

Tecnificación Mecatrónica

Vargas Soto José Emilio, Obil Martínez Octavio
y Orozco Ramírez Jorge Enrique



Tecnificación Mecatrónica

1ª Edición

Vargas Soto José Emilio
Obil Martínez Octavio
Orozco Ramírez Jorge Enrique

“Tecnificación Mecatrónica”

Vargas Soto José Emilio

Obil Martínez Octavio

Orozco Ramírez Jorge Enrique

© Asociación Mexicana de Mecatrónica A.C.
Prolongación Corregidora Norte No. 168, interior E
Col. El Cerrito, C.P. 76160, Querétaro, Qro.
México.

Internet: <http://www.mecamex.net>

1ª Edición, 09 de Octubre del 2019

ISBN: 978-607-9394-20-2

Derechos reservados.

Esta obra es propiedad intelectual de sus autores y los derechos de publicación han sido legalmente transferidos a la editorial. Las opiniones y la información que se muestran en los capítulos del libro son exclusivas de los autores de cada capítulo y no representan la postura de la Asociación Mexicana Mecatrónica A.C. Prohibida su reproducción total o parcial por cualquier medio sin la autorización por escrito del propietario de los derechos del copyright.

Impreso en México – Printed in Mexico



Índice

Capítulo 1	
Determinación de Daño en Elementos Lineales con Software Basado en el MEF	1
Candia García F., Flores Ascencio D., Castillo Flores M., Rodríguez Mora J.I., Heredia Jiménez A.H. y García Sánchez E.R.	
Capítulo 2	18
Segmentación de Imágenes Aplicado al Proceso de Secado de Granos de Café Pergamino	
Palma Cruz Lorena y Nieves Vázquez José Ángel.	
Capítulo 3	32
Instrumentación Virtual con ISE-LabVIEW©	
Castañeda Espinoza Angélica y Cortés Ramírez Ulises.	
Capítulo 4	46
Vivero con Sistema de Riego Inteligente Monitoreado con Sensores	
Itzel Pahua Angel, Aguilar Chamu Hassel Yabin e Ileri Valeria Villaseñor García.	
Capítulo 5	60
Sobre la Regularización Dual en el Balance de Postura de un Robot Humanoide Empleando Control de Cuerpo	
Rojas Rodríguez José Cruz, Aguilar Bustos Ana Yaveni y Bugarin Carlos Eusebio.	
Capítulo 6	75
Impresora 3D con Extrusora de Plástico Triturado	
Reyes Baltazar Edgar Alfredo, Aceves Arias Gerardo, Aguilar Molina Yehoshua y Domínguez García Rodolfo Omar.	
Capítulo 7	91
Optimización en el Diseño de Brazos Robóticos: Algoritmos de Enjambre	
Méndez Guzmán Hugo Antonio, Ledesma Silva Gabriel Daniel y Zavala Rojas Pablo Alejandro	



Índice

Capítulo 8	112
Diseño de una Máquina que Cocina Huevos González Valdivia Antonio de Jesús, Alatorre González Cristian Daniel y Aguilar Molina Yehoshua.	
Capítulo 9	125
Inclusión de la Dinámica de los Actuadores y Manejadores de Potencia en el Modelo de un Robot Móvil Omnidireccional Correa Fernández Erick Aarón, Serrano Pérez Omar, Rodríguez Molina Alejandro, Corona Ramírez Leonel Germán y Villarreal Cervantes Miguel Gabriel ¹ .	
Capítulo 10	139
Redes Neuronales Físicas Artificiales Covix Gracia Evelin, Nieves Vázquez José Ángel, Cervantes Morales Emmanuel de Jesús y Zubiaga Garate Augusto.	
Capítulo 11	153
Máquina Selladora de Bolsas Fuentes Flores Lorena	
Capítulo 12	161
Robot Neumático de Seis Grados de Libertad Monter Hernández Adolfo, Alegría Anaya Enrique, Ramos Arreguin Juan Manuel, Gorrostieta Hurtado Efrén y Aceves Fernández Marco Antonio.	
Capítulo 13	169
Control PID-Wavelet para Sistemas Lineales Invariantes en el Tiempo García Blancas Jesús y Flores Moreno Julian	
Capítulo 14	181
Sistema de Control de Acceso Inteligente a través de Cerraduras Electrónicas Figueroa Soto Israel, Gutiérrez Enríquez David Ramón, Domínguez García Rodolfo Omar y Aguilar Molina Yehoshua.	



Índice

Capítulo 15	200
Máquina Formadora de Hamburguesas a base de Carne Molida Mascot Ramírez Dulce Janet, Roque González Héctor Manuel y Aguilar Molina Yehoshua.	
Capítulo 16	213
Preparador Automático de Bebidas Ayón Rubio Cristóbal, Santana Garrafa Ulises, Domínguez García Rodolfo Omar y Aguilar Molina Yehoshua.	
Capítulo 17	223
Máquina para la Producción de Pan Caliente Gamiño Coronado Jesús Antonio, Rosas Garibay Francisco, Aguilar Molina Yehoshua y Domínguez Aguilar Rodolfo Omar.	
Capítulo 18	239
Separador Inteligente de Residuos Cervantes Morales Emmanuel de Jesús, Nieves Vázquez José Ángel, Covix Gracia Evelin, Camacho Torres José Luis y Martínez Domínguez Ángel Guadalupe.	
Capítulo 19	251
Mecanismo para rehabilitación (MEC-RE) García González Gladis, Chi Rodríguez Yesenia del Carmen y Hernández Gutiérrez Ramón.	
Capítulo 20	266
Diseño y Simulación de un Prototipo de Alarmas con Ubicación en Tiempo Real para Objetos Móviles Utilizando un Microcontrolador PIC 18f4550 Ramiro Ramiro José, Tacuepian Damian Oscar, Hernández García Edgar y Méndez Alonso José Miguel.	
Capítulo 21	280
Implementación de Sistema Mecatrónico para Simulación de Movimientos de la ATM Cortés Ramírez Ulises, Castañeda Espinoza Angélica, Ascanio Balderas Martín y Juárez Rosete Manuel E.	
Capítulo 22	294
Implementación de un Algoritmo de Control de Trayectoria en un Sistema de Arquitectura Abierta Pájaro Adrián J. Domingo, Reyes Báez Rodolfo, Cid Monjaraz Jaime J. y Reyes Cortés Fernando.	



Índice

Capítulo 23 Análisis de la Presencia de Actuadores en Sistemas de Amortiguamiento Activo con Controladores en Tiempo-Finito Rosales Martínez Antonio, Martínez Martínez Jesús, García Trinidad Enrique y Juárez Carrilo Cesar.	310
Capítulo 24 Prototipo de Embotelladora para Agua en PET de 600 ml para Pequeños Productores Palma Cruz Victor, Valencia Benitez Roberto, Amador Chagala Héctor M. y Palma Cruz Lorena.	323
Capítulo 25 Diseño de un Cicloconvertidor para un Sistema Mecatrónico de Rehabilitación Motriz de Rodilla Navarrete Hernández Gabriel Antonio, Galván Chávez Jorge Moisés y Grande Olvera Daniel.	338
Capítulo 26 Automatización de un Invernadero para el Cultivo de Hongos Avila Mata Estela Nataly, Rentería Ramírez Luis Felipe, Aguilar Molina Yehoshua y Brizuela Mendoza Jorge Aurelio.	350
Capítulo 27 Automatización del Lago Artificial e Implementación de Fuentes Danzantes Aceves Becerra Octavio Martin, Morales Medina José Luís, Tostado Estrada Álvaro Emanuel, Aguilar Molina Yehoshua y Domínguez García Rodolfo Omar.	364
Capítulo 28 Desarrollo de un Simulador Robótico Utilizando el Método Evolutivo de las Entregas Incrementales Jiménez López Eusebio, Zarza Herreros Melquisedec, Rivera Nieblas Jorge Oswaldo, Urbalejo Contreras Arturo, Amavizca Valdez Laura Olivia y Vázquez Brindis Juan Carlos.	372
Capítulo 29 Diseño e Implementación de un Tablero Educatrónico para Apoyo de la Enseñanza de los Microcontroladores Montiel Jiménez Elmar, Guzmán Rodríguez Alejandro, Córdova Morales Guillermo y Bonilla Reyes Susana.	385



Determinación de Daño en Elementos Lineales con Software Basado en el MEF

¹Candia García F., ²Flores Ascencio D., ³Castillo Flores M., ⁴Rodríguez Mora J.I.,
⁵Heredia Jiménez A.H., ⁶García Sánchez E.R.

^{1, 2, 3, 4}Facultad de Ingeniería, Benemérita Universidad Autónoma de Puebla.

⁵Escuela de Ingeniería Electrónica, Universidad Popular Autónoma de Estado de Puebla.

⁶Doctorado en Ingeniería Mecatrónica, Universidad Popular Autónoma de Estado de Puebla.
filinc@hotmail.com

Resumen

Este trabajo evalúa la capacidad del software NX Siemens, para determinar daño en elementos lineales con pérdida del continuo, se aborda desde la teoría de las discontinuidades débiles y las discontinuidades fuertes, ambas teorías rigen el comportamiento del material lineal cuando existen discontinuidades como grietas. Se evalúan elementos lineales con discontinuidades mediante la simulación por medio del método de los elementos finitos. Dentro de las aportaciones de este trabajo fue el comprobar si el software NX, tiene la capacidad de aproximar los resultados de sus simulaciones a los reportados por Robles en sus trabajos con elementos lineales con pérdida del continuo, realizando una comparativa de la aproximación de los resultados. Los datos obtenidos y el comportamiento de la simulación concluyen que el software NX Siemens cuenta con un algoritmo de cálculo robusto con la capacidad de aproximar resultados confiables en la evaluación de elementos lineales con pérdida del continuo.

Palabras clave: Diseño estructural, Elementos lineales, Continuo, Discontinuidad débil, Discontinuidad fuerte, Determinación de daño.

1. Introducción

Los antecedentes generales de este trabajo se enmarcan en los objetivos e hipótesis de investigación, bajo el supuesto "Si se valida el software NX Siemens para determinar daño en elementos lineales con pérdida del continuo y se utiliza la teoría de la discontinuidad fuerte del continuo como criterio de falla. Entonces es posible simular estructuras complejas en 3D optimizando el tiempo de análisis para la obtención de resultados confiables de su comportamiento ante discontinuidades fuertes" [1].

Como marco referencial se aborda la teoría de las discontinuidades débiles y fuertes y la mecánica de la fractura para establecer las condiciones de frontera ante la pérdida del continuo. El marco conceptual aborda la teoría del método de los elementos finitos, que configura el algoritmo de cálculo del software de simulación NX Siemens. Se prioriza la simulación virtual como medio de experimentación-exploratoria. Se organiza la simulación mediante un diagrama de flujo y se proponen modelos en los cuales se establecen condiciones de frontera y de falla (tipo de carga, restricción de los apoyos y profundidad de grieta), posteriormente se lleva a cabo la simulación en el software NX Siemens, con los datos obtenidos se comparan los resultados identificados en la literatura y estudios anteriores verificando si existe similitud para emitir las conclusiones. La técnica propuesta por Robles (2005) reporta que es una importante herramienta, para ser empleada al estudiar patologías en estructuras de acceso limitado, tal que permite determinar la existencia de daño, localizarlo y cuantificarlo. Permite localizar un seguimiento de la evolución del daño que tiene una estructura, tanto



sea metálica o de hormigón armado, con el transcurrir del tiempo, para establecer en la capacidad portante residual del elemento afectado [2].

En el desarrollo de la simulación primero se verificó mediante el análisis lineal (método exacto) los resultados de la simulación de una viga sin discontinuidades. Una vez configurados los modelos virtuales a través de los estudios de Robles (2005) se realizó la simulación y se compararon los resultados. Contando con los resultados de la simulación y mediante las gráficas de los resultados, se desarrolló la discusión de la aproximación de los valores y del comportamiento de los elementos lineales con discontinuidades. Se concluyó que ha sido favorable la evaluación y determinación de que NX Siemens es un software robusto con capacidad para realizar simulaciones a elementos lineales con discontinuidades y se favorece su uso y aplicación para estimar la detección de daño y falla en elementos estructurales complejos.

1.1. Objetivos

La finalidad del desarrollo de esta investigación está orientada a validar el uso del software NX Siemens [3] para la simulación y análisis de elementos lineales con discontinuidades fuertes. Ello requirió determinar los modelos en dos y tres dimensiones, para simulación y análisis de los trabajos realizados por la metodología de Robles, elaborar los archivos .prt y .fem de los modelos seleccionados, así como la aplicación de condiciones frontera al archivo .sim de los distintos modelos.

Cumpliendo con los requerimientos anteriores se realizó la simulación de los modelos en 2D y 3D generados y la comparación de los resultados de los ensayos obtenidos de la simulación contra la metodología de Robles. A fin de cumplir este propósito, se ha desarrollado modelos en 3D basados en la literatura y estudios anteriores que permitieron obtener gráficas y resultados numéricos para comparar el comportamiento teórico y el comportamiento de los modelos de simulación digital en la detección de daño.

2. Metodología

El desarrollo de este proyecto se lleva en función de las capacidades de simulación del software NX Siemens, para la detección de daño en elementos lineales con grietas (algoritmo de solución robusto, mallado adaptativo, etc.) [3]. Partiendo de este preámbulo se compararon los resultados de la técnica propuesta por Robles. Como parte del procedimiento se propusieron modelos en los que se establecen condiciones de frontera y falla, basados en el tipo de carga, restricción de los apoyos y profundidad de grieta, posteriormente se llevó a cabo la simulación en el software NX Siemens, con los datos obtenidos y se observaron los resultados numéricos y gráficos verificando si existe similitud con los estudios anteriores, de ser así se concluye el proceso con conclusiones favorables, en caso contrario se repite el procedimiento (figura 1), retornando a la formulación de nuevos modelos.

2.1. Análisis comparativo

Una fisura en un elemento estructural que produce cambios locales en la flexibilidad de esta, que se traducen en variaciones de su comportamiento estático y dinámico. La respuesta de estructuras [4] se puede clasificar según el tipo de carga a la cual estén sometidas o por el tipo de respuesta que presenten. Las cargas pueden ser estáticas o dinámicas; las cargas dinámicas dependen del tiempo, de la posición y de su magnitud. La respuesta de una estructura, a su vez, puede ser estática o dinámica, si es dinámica actuarán en la estructura fuerzas de inercia. Una estructura se puede dañar por diversas causas, sobrecargas accidentales que excedan aquellas para las cuales fue diseñada, por el cambio o deterioro de sus propiedades físicas o mecánicas en el tiempo, influencia del medio ambiente, etc. Por lo anterior se estima que la detección de daño se puede efectuar mediante métodos basados en ensayos destructivos y no destructivos.

En este trabajo se presenta una alternativa no destructiva para la detección de daño, basada en la respuesta estática de estructuras de acero lineales, tales como vigas, así como también pórticos de concreto armado, empleando un algoritmo de detección de daño, el cual incluye cuatro etapas:



detección, localización, cuantificación y predicción de la vida y/o resistencia residual de la estructura, para el caso de estudio solo se emplearán las dos primeras etapas: detección y localización.

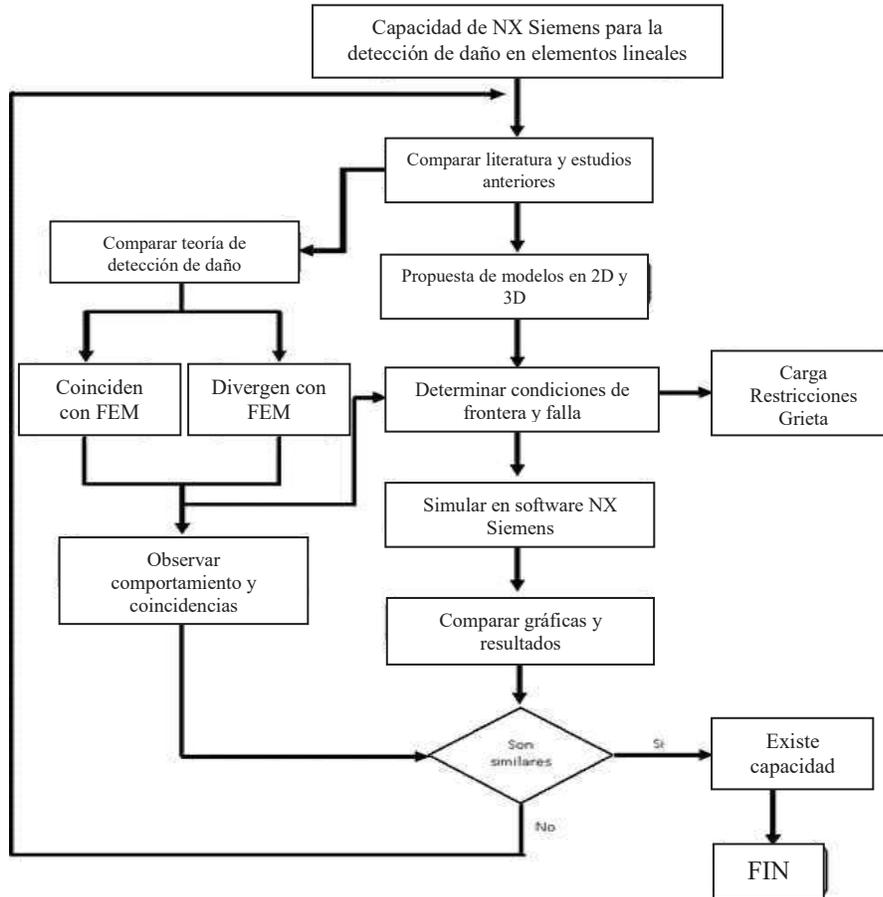


Figura 1. Diagrama de flujo para realizar simulaciones, elaboración propia.

Con base a la técnica de Robles para la detección de daño basada en la respuesta estática se propone el modelo de una viga simplemente apoyada de 1.00m de largo, 0.05m de alto y 0.05m de espesor, como se muestra en la figura 2, así mismo, se sobrepone una carga vertical $F = 200 \text{ N}$, dicho modelo es analizado considerando una formulación de estados planos de tensiones en 2D con elementos rectangulares de 4 nodos. El cambio de rigidez, debido al daño fue modelado mediante la incorporación de una fisura discreta. Las propiedades del material correspondientes al acero son: $\rho = 7855 \text{ Kg/m}^3$, $E = 1.9 \times 10^{11} \text{ Kg/m}^2$ y $\mu = 0.29$ [2].

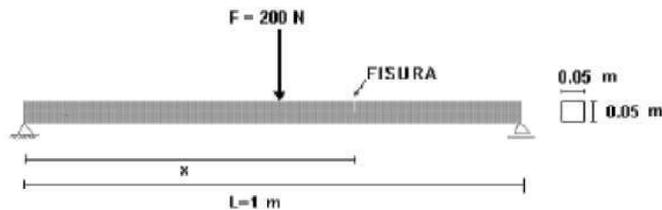


Figura 2- Modelo de viga simplemente apoyada, tomada de [2].

Para el modelo de la viga en voladizo, como el mostrado en la figura 3, las dimensiones son iguales a las de la viga simplemente apoyada, la carga puntual sobrepuesta $F = 200 \text{ N}$, de igual manera se consideró la formulación de estados planos de tensiones en 2D con elementos



rectangulares de nodos. El cambio de rigidez, debido al daño fue modelado mediante la incorporación de una fisura discreta. Las propiedades del material correspondientes al acero son: $\rho = 7855 \text{ Kg/m}^3$, $E = 1.9 \times 10^{11} \text{ Kg/m}^2$ y $\mu = 0.29$ [2].

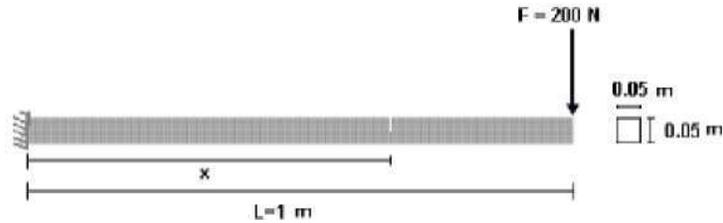


Figura 3- Modelo de viga en voladizo, tomada de [2].

Por último, se propuso un modelo basado en un pórtico biarticulado con una longitud de 4.00m de largo, 3.00m de alto mostrado en la figura 4, con una sección variable, para este caso se sobreponen dos cargas puntuales $F=300\text{N}$ cuya distancia entre cada una de estas es de 1.00m, se consideró la formulación de estados planos de tensiones en 2D con elementos rectangulares de 4 nodos. El cambio de rigidez, debido al daño fue modelado mediante la incorporación de una fisura discreta. Las propiedades del material correspondientes al concreto armado son: $\rho = 2405 \text{ Kg/m}^3$, $E = 2.06 \times 10^9$ y $\mu = 0.15$ [2]. La técnica de Robles establece que una vez cuantificado el daño se establece su magnitud y para ello se analizan modelos con distintas ubicaciones y severidades de daño, obteniéndose gráficas en tres dimensiones de los parámetros: ubicación, severidad y un tercero vinculado con el comportamiento mecánico de la estructura dañada. Luego se ingresa al gráfico realizado en 3D (figura 5), con la ubicación del daño x/L , y con el desplazamiento medido u otro parámetro mecánico, obteniéndose el valor de a/h . Para este estudio se utilizaron como parámetros a los desplazamientos, desplazamientos volumétricos y deformaciones volumétricas, con la finalidad de observar cual era más sensible a la severidad de daño.

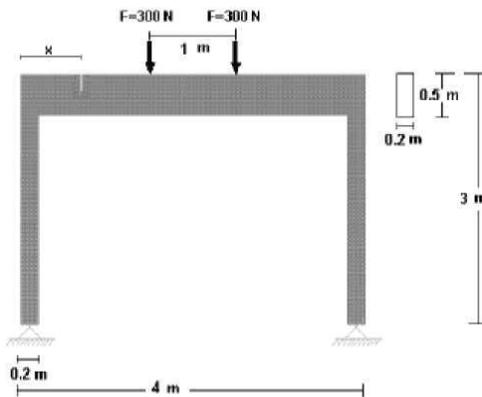


Figura 4- Modelo de pórtico biarticulado, tomada de [2].

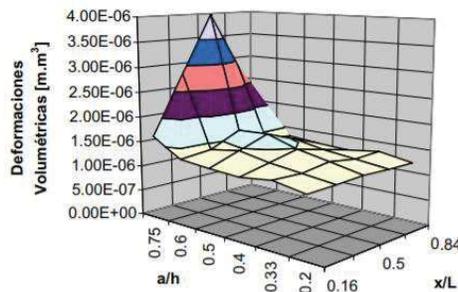


Figura 5- Deformaciones volumétricas en función de la ubicación y severidad del daño, tomada de [2].



En la tabla 1, se muestran los valores obtenidos para la viga simplemente apoyada, concluyendo que los desplazamientos lineales y los desplazamientos volumétricos presentan una variación porcentual. Resulta conveniente observar los desplazamientos volumétricos debido a que se emplea mayor cantidad de información, por lo que un error en la mediación podría amortiguarse.

Tabla 1- Desplazamiento del punto medio, desplazamiento volumétrico y deformación volumétrica de una viga simplemente apoyada en función de la ubicación x/L y severidades del daño a/h, tomada de [2].

a/h	Desplazamiento Punto Medio (m)			Desplazamiento Volumétrico (m.m ³)			Deformación Volumétrica (m.m ³)		
	x/L			x/L			x/L		
0.16	0.16	0.33	0.50	0.16	0.33	0.50	0.16	0.33	0.50
0.75	5.26E ⁻³	9.39E ⁻³	1.59E ⁻²	1.66E ⁻⁶	2.84E ⁻⁶	4.00E ⁻⁶	7.74E ⁻⁷	8.60E ⁻⁷	9.80E ⁻⁷
0.60	4.39E ⁻³	5.62E ⁻³	7.55E ⁻³	1.31E ⁻⁶	1.49E ⁻⁶	1.69E ⁻⁶	7.31E ⁻⁷	7.80E ⁻⁷	8.49E ⁻⁷
0.50	4.25E ⁻³	4.98E ⁻³	6.14E ⁻³	1.26E ⁻⁶	1.47E ⁻⁶	1.68E ⁻⁶	7.19E ⁻⁷	7.56E ⁻⁷	8.10E ⁻⁷
0.40	4.16E ⁻³	4.60E ⁻³	5.31E ⁻³	1.22E ⁻⁶	1.35E ⁻⁶	1.49E ⁻⁶	7.09E ⁻⁷	7.38E ⁻⁷	7.78E ⁻⁷
0.33	4.11E ⁻³	4.36E ⁻³	4.78E ⁻³	1.30E ⁻⁶	1.28E ⁻⁶	1.36E ⁻⁶	7.02E ⁻⁷	7.22E ⁻⁷	7.55E ⁻⁷
0.20	4.05E ⁻³	4.11E ⁻³	4.21E ⁻³	1.19E ⁻⁶	1.20E ⁻⁶	1.23E ⁻⁶	6.91E ⁻⁷	4.00E ⁻⁷	7.17E ⁻⁷
Variación % Min y Máx	30	128	277	40	136	225	12	23	36

En la tabla 2, se muestran los valores obtenidos para la viga en voladizo, concluyendo que los desplazamientos lineales y los desplazamientos volumétricos presentan una variación porcentual significativa.

Tabla 2- Desplazamiento del punto medio, desplazamiento volumétrico y deformación volumétrica de una viga en voladizo en función de la ubicación x/L y severidad del daño a/h, tomada de [2].

a/h	Desplazamiento Punto Extremo (m)			Desplazamiento Volumétrico (m.m ³)			Deformación Volumétrica (m.m ³)		
	x/L			x/L			x/L		
0.16	0.16	0.33	0.50	0.16	0.33	0.50	0.16	0.33	0.50
0.75	2.00E ⁻⁴	1.48E ⁻³	1.11E ⁻³	4.63 E ⁻⁵	2.98 E ⁻⁵	2.07 E ⁻⁵	1.19E ⁻⁴	1.07E ⁻⁴	1.03E ⁻⁴
0.60	1.04E ⁻³	8.92 E ⁻⁴	7.82 E ⁻⁴	2.34 E ⁻³	1.85 E ⁻³	1.59 E ⁻³	1.04E ⁻⁴	9.66E ⁻⁵	9.44E ⁻⁵
0.50	8.82 E ⁻⁴	7.91 E ⁻⁴	7.26 E ⁻⁴	1.95 E ⁻³	1.66 E ⁻³	1.51 E ⁻³	9.77E ⁻⁵	9.46E ⁻⁵	9.15E ⁻⁵
0.40	7.86 E ⁻⁴	7.32 E ⁻⁴	6.91 E ⁻⁴	1.72 E ⁻³	1.55 E ⁻³	1.46 E ⁻³	9.44E ⁻⁵	9.07E ⁻⁵	8.94E ⁻⁵
0.33	7.25 E ⁻⁴	6.93 E ⁻⁴	6.70 E ⁻⁴	1.57 E ⁻³	1.47 E ⁻³	1.43 E ⁻³	9.14E ⁻⁵	8.85E ⁻⁵	8.77E ⁻⁵
0.20	6.67E ⁻⁴	6.51E ⁻⁴	6.47 E ⁻⁴	1.42E ⁻³	1.41E ⁻³	1.40E ⁻³	8.68E ⁻⁵	8.51E ⁻⁵	8.49E ⁻⁵
Variación % Min y Máx	67	56	42	69	53	37	37	26	22

La tabla 3, muestra los valores obtenidos para el pórtico biarticulado. En este caso se observa que los desplazamientos volumétricos presentan una variación porcentual mayor que los otros dos parámetros, independientemente de la ubicación del daño.

Tabla 3- Desplazamiento del punto medio, desplazamiento volumétrico y deformación volumétrica del dintel de un pórtico biarticulado en función de la ubicación x/L y severidad del daño a/h, tomada de [2].

a/h	Desplazamiento Punto Medio (m)			Desplazamiento Volumétrico (m.m ³)			Deformación Volumétrica (m.m ³)		
	x/L			x/L			x/L		
0.16	0.16	0.33	0.50	0.16	0.33	0.50	0.16	0.33	0.50
0.60	1.54E ⁻⁴	2.17 E ⁻⁴	2.86 E ⁻⁴	3.10E ⁻⁴	6.16E ⁻⁴	7.29E ⁻⁴	2.92E-5	3.24E-5	3.34E-5
0.50	1.47 E ⁻⁴	1.83 E ⁻⁴	2.23 E ⁻⁴	3.08E ⁻⁴	4.59E ⁻⁴	5.25E ⁻⁴	2.82E-5	3.05E-5	3.11E-5
0.40	1.43 E ⁻⁴	1.62 E ⁻⁴	1.85 E ⁻⁴	2.74E ⁻⁴	3.66E ⁻⁴	4.05E ⁻⁴	2.73E-5	2.87E-5	2.92E-5
0.33	1.40 E ⁻⁴	1.50 E ⁻⁴	1.62 E ⁻⁴	2.49E ⁻⁴	3.03E ⁻⁴	3.27E ⁻⁴	2.67E-05	2.75E-5	2.77E-5
0.20	1.39 E ⁻⁴	1.43 E ⁻⁴	1.47 E ⁻⁴	2.31E ⁻⁴	2.62E ⁻⁴	2.75E ⁻⁴	2.62E-5	2.65E-5	2.67E-5
Variación % Min y Máx	10	34	48	26	58	62	10	18	20

La revisión de las tablas 1, 2, y 3, se observa que existe un comportamiento importante en el desplazamiento y deformación volumétrica, por lo tanto, para este trabajo la posibilidad de emplear el análisis en 3D del software NX Siemens, se apertura como la alternativa de simulación que proporcionaría mayor información. Por ello se realizó una comparativa de resultados entre un análisis 2D y 3D con elementos lineales sin daño.

3. Desarrollo de la simulación

En el estudio estático lineal de los elementos lineales sin discontinuidad, entre los métodos más utilizados actualmente para determinar la deflexión máxima se encuentra la ec. 1 de Euler también llamada ecuación de la curva elástica de una viga es una expresión matemática que se expresa como



$v = f(x)$. Esta expresión representa una ecuación diferencial no lineal de segundo orden. Su solución se denomina elástica, y da la forma exacta de la curva elástica, suponiendo que las deflexiones de la viga se producen solo por la flexión [5].

$$\frac{M}{EI} = \frac{\frac{d^2v}{dx^2}}{\left[1 + \left(\frac{dv}{dx}\right)^2\right]^{\frac{3}{2}}} \quad (1)$$

Este método tiene el inconveniente de no proporcionar información sobre la apertura de fisura y rotación a lo largo de las secciones, siendo este último aspecto un parámetro cada vez más necesario en el diseño estructural, sobre todo en los marcos biarticulados, los cuales mostraron un comportamiento de mucho interés (grandes deformaciones y rotaciones) al incrementar el tamaño a/h de la grieta (figura 6).

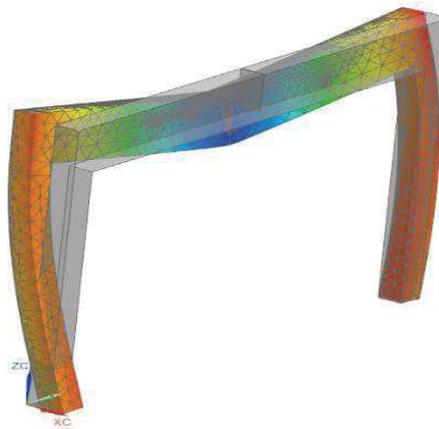


Figura 6- Comportamiento de marco biarticulado con gran tamaño de a/h .

3.1. Modelo de Viga Simplemente Apoyada

Se consideró una sección cuadrangular de acero sin ninguna fisura, las diferentes variables geométricas que intervienen en el problema se muestran en la figura 7. La viga tiene dimensiones de 50 mm, en el eje "X"; 1000 mm, en el eje "Y" y 50 mm, en el eje "Z".

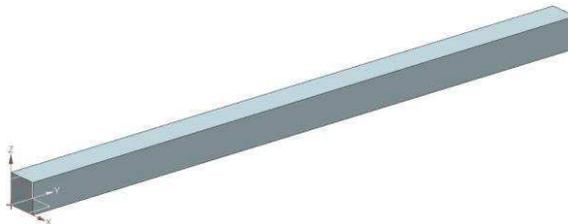


Figura 7- Sección cuadrangular de acero sin fisura.

Para determinar por simulación en NX Siemens, cómo se comporta el elemento se requiere el uso de la solución 101, tipo de solución estático lineal con restricciones globales [3] [4]. Para empezar con la simulación del modelo se tiene que hacer una discretización del modelo 3D como se muestra en la figura 8, empleando todas las constantes que se presentan en el método exacto de esta forma obtienen los resultados con los cuales se podrá hacer la comparación respectiva.



Para aprovechar la capacidad adaptativa del Software NX Siemens, se emplearon para el modelo 3D elementos finitos tetraédricos de 10 nodos y para el modelo 2D de elementos finitos elementos cuadrangulares de 8 nodos [3] [4].

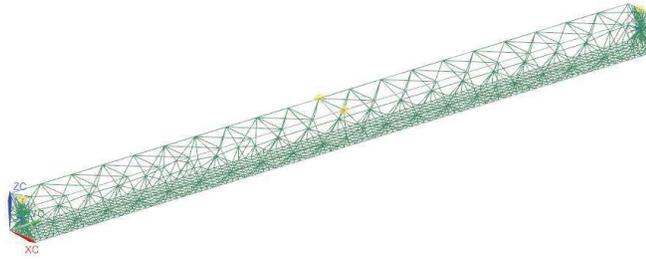


Figura 8- Modelo .fem de una viga simplemente apoyada.

En las figuras 9 y 10, se muestran los resultados de la simulación en términos del desplazamiento, tanto en simulación 2D como 3D, resultados que fueron coincidentes con el modelo analítico obtenidos por medio de la ec. 2, que es la solución exacta para la viga simplemente apoyada [5] y se comparan en la tabla 4, mostrando una alta aproximación.

$$\delta_{\max} = \frac{PL^3}{48EI} = 0.0429 \text{ mm} \quad (2)$$

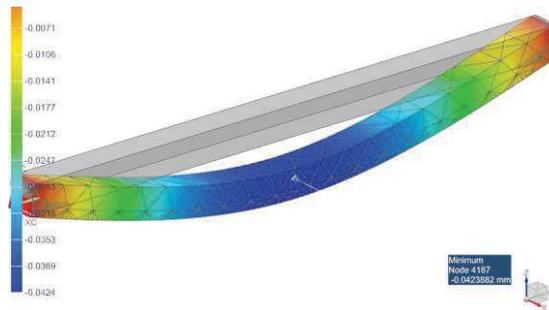


Figura 9- Desplazamientos obtenidos viga simplemente apoyada, por simulación con elementos 3D.

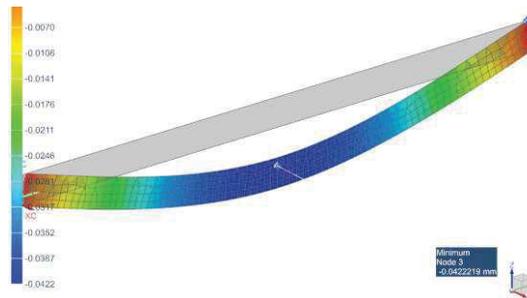


Figura 10- Desplazamientos obtenidos viga simplemente apoyada, por simulación con elementos 2D.

3.2. Modelo de viga en voladizo

Para realizar la simulación de este modelo se tienen que seguir el mismo procedimiento que para la simulación anterior con la diferencia que en esta ocasión solo existirá una sola restricción por empotramiento en la viga, que restringe los seis grados de libertad en un extremo de la viga (figura 11).

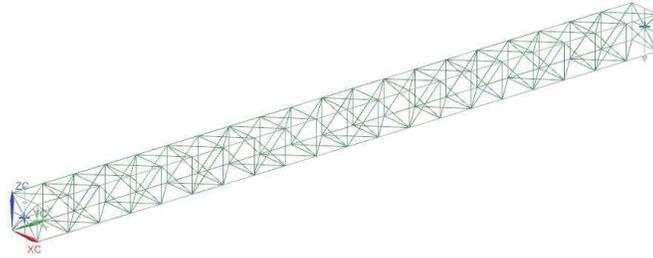


Figura 11- Modelo .fem de una viga en voladizo.

Al ejecutar la solución 101, solución estático lineal, se muestran los desplazamientos de la simulación en 3D (figura 12) y 2D (figura 13) del modelo. La solución analítica se obtuvo por medio de la ec. 3, la cual es la solución exacta para la viga en voladizo [5] y se comparan en la tabla 4, mostrando una alta aproximación.

$$\delta_{\max} = \frac{PL^3}{3EI} = 0.673 \text{ mm} \quad (3)$$

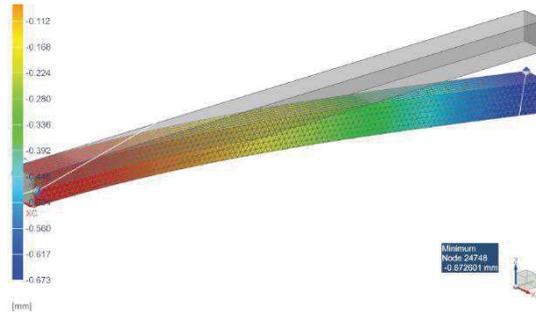


Figura 12- Desplazamientos obtenidos viga en en voladizo, por simulación con elementos 3D.

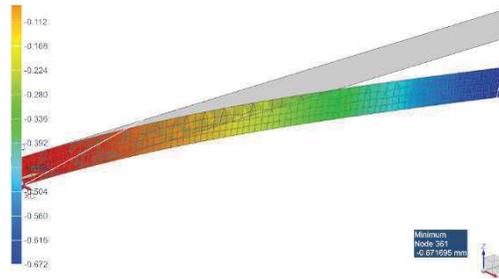


Figura 13- Desplazamientos obtenidos viga en voladizo, por simulación con elementos 2D.

3.3. Concepto de la discontinuidad

Para determinar que un elemento lineal tiene una discontinuidad [1] se considera que existe un sólido tridimensional, con un dominio $\Omega \in \mathbb{R}^3$ y frontera Γ , el cual se somete a carga hasta la aparición de una discontinuidad en el campo de desplazamientos $[u]$ a lo largo de una banda Ω^h de espesor k o superficie S (figura 10b), donde las deformaciones inelásticas y otros procesos no lineales se concentran. En consecuencia, el dominio se divide en $\Omega = \Omega^- + \Omega^h + \Omega^+$, y la frontera $\Gamma = \Gamma^- + \Gamma^+$. Las condiciones de frontera son: las tracciones prescritas t^* en $\Gamma_\sigma = \Gamma_\sigma^- + \Gamma_\sigma^+$ y los desplazamientos prescritos u^* en $\Gamma_u = \Gamma_u^- + \Gamma_u^+$ de tal forma que $\Gamma_\sigma \cup \Gamma_u = \Gamma$ y $\Gamma_\sigma \cap \Gamma_u = \emptyset$. La discontinuidad inicia cuando se satisface un criterio de falla, que en su caso más simple la falla se presente cuando se



excede el límite de comportamiento elástico en un modelo de plasticidad o daño, o mediante un análisis más elaborado a nivel material, un análisis de bifurcación (pérdida de elipticidad de las ecuaciones que gobiernan el problema) cuando se satisface la ec. 4:

$$\det (n * C^d * n) = 0 \quad (4)$$

Donde n es el vector normal a la discontinuidad y C^d es el tensor constitutivo tangente. El proceso de falla descrito anteriormente se puede idealizar mediante el modelo de Discontinuidades Interiores en sus tres aproximaciones: Discontinuidad Débil (DDE), Discontinuidad Fuerte (DFU) y Discontinuidad Discreta (DDI). Las primeras dos aproximaciones consideran que el dominio Ω permanece continuo después de que el material pierde estabilidad; el comportamiento del material se basa en una relación constitutiva continua esfuerzo-deformación (figura 14).

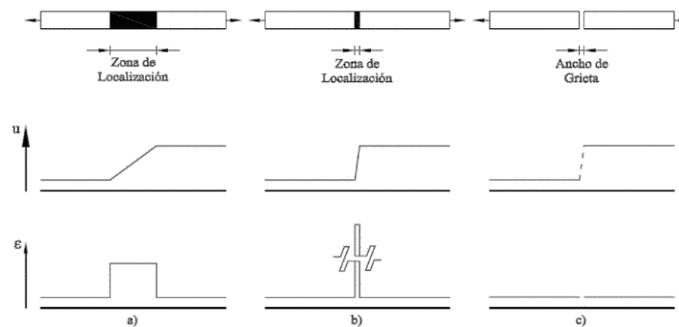


Figura 14- Cinemática de desplazamientos y deformaciones de las aproximaciones de Discontinuidad a) Débil, b) Fuerte y c) Discreta, tomada de [1].

Para la determinación de los desplazamientos es importante resaltar, que el comportamiento entre un elemento continuo y uno que llega a presentar una discontinuidad llega a ser significativo, mientras uno se apeg a la ley de Hooke el otro tiende a presentar un comportamiento no lineal, es en este preámbulo donde la mecánica de la fractura busca determinar la distribución de esfuerzos y deformaciones en la punta de la grieta, por lo que para el caso de estudio y con apego a la teoría de Inglis se determina la amplitud de la grieta [7] en base a las ecuaciones sugeridas por su teoría a partir de la figura 15.

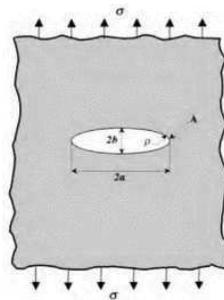


Figura 15- Agujero elíptico de una placa plana.

Inglis asumió que el agujero no está influenciado por las condiciones de contorno de la placa; la anchura de la placa $\gg 2a$ y la altura $\gg 2b$. A partir de este enunciado la tensión en el extremo del eje mayor (punto A de la figura 15) está dada por la ec. 5:

$$\sigma_A = \sigma \left(1 + \frac{2a}{b} \right) \quad (5)$$



La relación σ_A/σ se define como el factor de concentración de tensiones, K_t . Cuando $a = b$, el agujero es circular y en este caso $K_t = 3$, cifra que aparece en gran número de manuales. Cuando el eje mayor a , aumenta respecto a b , el agujero elíptico comienza a tener la apariencia de una grieta aguda. Para este caso, Inglis encontró una nueva ec. 6 en función del radio de curvatura ρ .

$$\sigma_A = \sigma \left(1 + 2\sqrt{\frac{a}{\rho}}\right) \quad (6)$$

Donde:

$$\rho = \frac{b^2}{a} \quad (7)$$

Cuando $a \gg b$, la ecuación (6) pasa a ser:

$$\sigma_A = 2\sigma\sqrt{\frac{a}{\rho}} \quad (8)$$

Inglis demostró que esta ec. 8, proporciona una buena aproximación para la concentración de tensiones en una placa con una entalla que no es elíptica excepto en la punta [7].

3.4. Simulación con discontinuidad

Para este modelo de simulación en 3D, se proponen distancias x/L a 0.5% de la longitud de la viga con la finalidad obtener resultados homogéneos en todos los modelos simulados (figura 16). Las propiedades mecánicas del material asignadas se pueden ver en la figura 21).

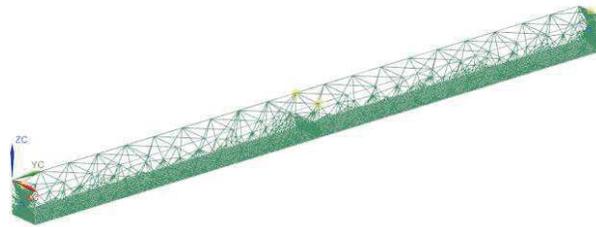


Figura 16- Modelo .fem viga simplemente apoyada en 3D con discontinuidad.

Al ejecutar la simulación con un desplazamiento de grieta $x/L = 0.5$ y $a/h = 0.4$ para la viga simplemente apoyada (figura 17) y $x/L = 0.5$ y $a/h = 0.5$ para la viga en voladizo (figura 18). Se obtienen los desplazamientos de las vigas, los resultados se concentran en las tablas 6 y 7 respectivamente, los cuales observan una alta aproximación con los obtenidos por Robles al ser comparados contra los valores de las gráficas 23 y 25. Las figuras 24 y 26 muestran un comportamiento elástico dentro de la zona lineal tal como lo refiere Robles antes de la identificación de la falla.

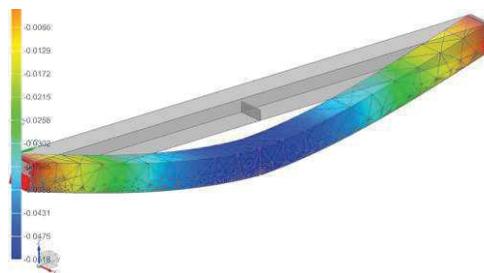


Figura 17- Desplazamientos del modelo de viga simplemente apoyada con fisura $x/L = 0.5$ y $a/h = 0.4$.

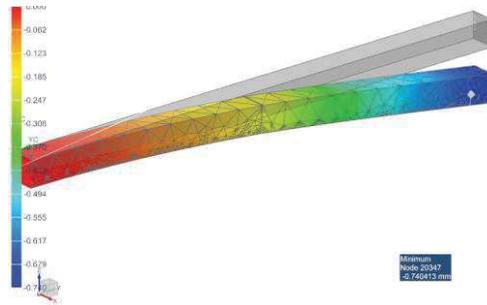


Figura 18- Desplazamientos del modelo de viga en voladizo con fisura $x/L = 0.5$ y $a/h = 0.5$.

3.5. Simulación de marco para $x/L = 0.5$ y $a/h = 0.5$

La simulación obtenida de emplear la solución estático lineal (figura 19 y 20) presenta los desplazamientos que existen en el marco cuando se le asignan las propiedades y constantes de acuerdo al reporte de Robles (figura 22). Los resultados se concentran en la tabla 8.

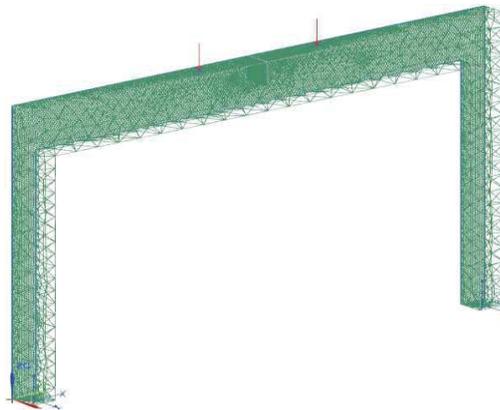


Figura 19- Modelo de elemento finito del marco en 3D.

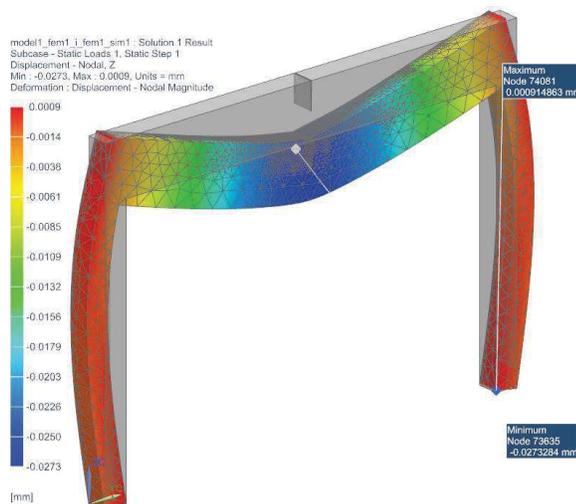


Figura 20- Desplazamientos del modelo de marco simplemente apoyado con fisura $x/L = 0.5$ y $a/h = 0.5$.

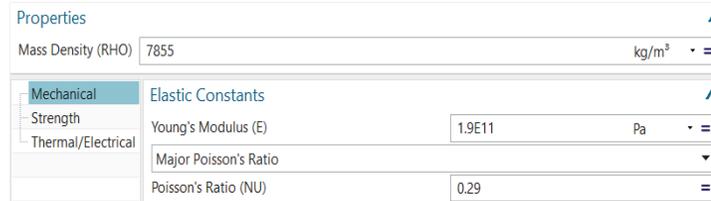


Figura 21- Datos de los materiales para la simulación de vigas.



Figura 22- Datos de los materiales para la simulación de marcos.

4. Resultados y discusión

Los resultados obtenidos para una viga simplemente apoyada con discontinuidad por la técnica de Robles, se puede observar en la figura 18, encontrando que el máximo desplazamiento ocurre en el centro de la viga siendo este de 0.0470mm y para NX Siemens (figura 24) es de 0.0518mm, como se indica en la tabla 6.

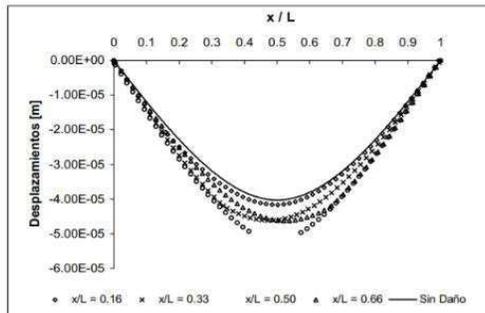


Figura 23- Desplazamientos, para viga simplemente apoyada con x/L en distintas ubicaciones y $a/h=0.4$, tomada de [2].

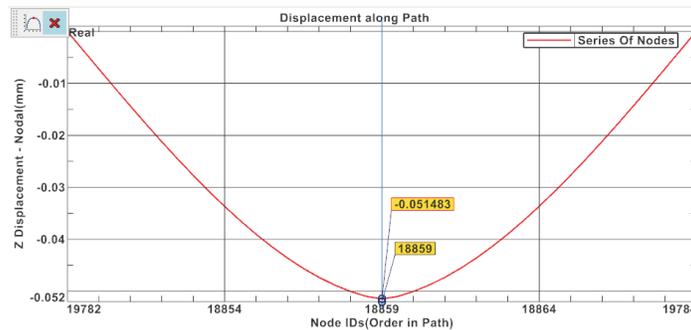


Figura 24- Resultados con NX Siemens en 3D, viga simplemente apoyada $x/L=0.5$, $a/h=0.4$.

Para el caso de una viga en cantiléver con discontinuidad en la figura 20, se muestran los resultados obtenidos por Robles de una viga en cantiléver con discontinuidad para una relación $a/h=0.5$ como se observa, el máximo desplazamiento se alcanza en el extremo del voladizo con un valor de 0.730mm, en la figura 21 se muestran los resultados obtenidos con NX Siemens en 3D con



un valor de 0.741mm, los cuales no difieren significativamente entre uno y otro método como se observa en la tabla 7.

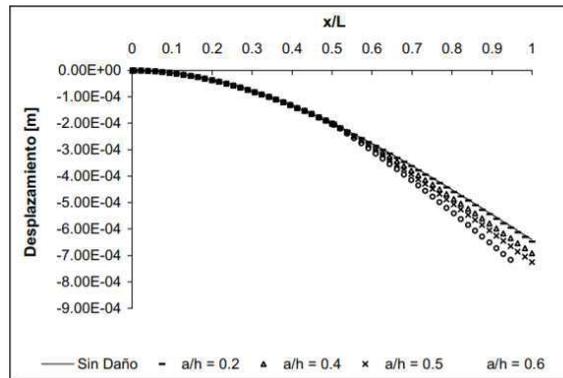


Figura 25- Resultados de Robles (2005), para viga en cantiléver $x/L=0.5$, $a/h=0.5$, tomada de [2].

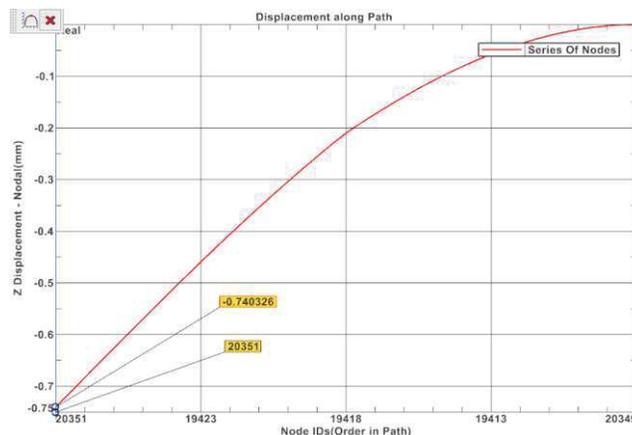


Figura 26- Resultados obtenidos con NX Siemens en 3D, viga en voladizo, $x/L=0.5$, $a/h=0.5$.

Para el caso un marco hiperestático con discontinuidad, los resultados obtenidos por Robles se muestran en la figura 27, como se observa en dicha figura se determinan los desplazamientos para cada segmento del marco, el máximo desplazamiento ocurre en el centro del claro con un valor de 0.260, mientras que en la figura 28 se muestran los resultados obtenidos con NX Siemens en 3D con un valor de 0.0277, como se observan en la tabla 8.

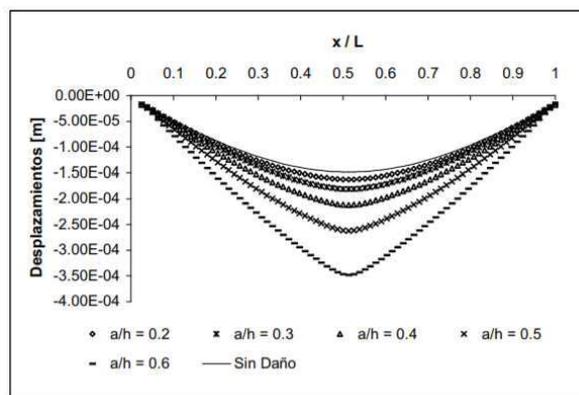


Figura 27- Resultados de Robles (2005), para marco biarticulado, $x/L=0.5$, $a/h=0.5$, tomada de [2].

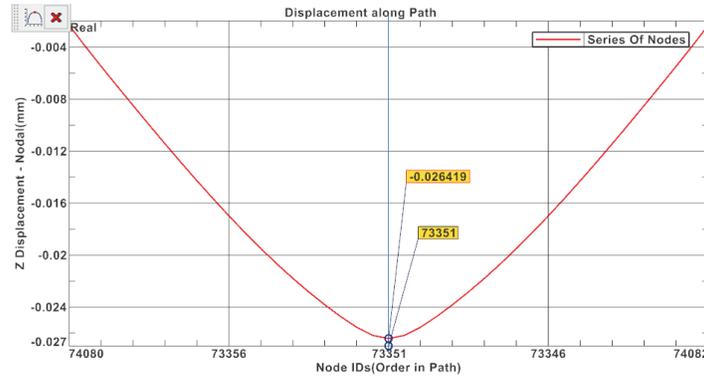


Figura 28- Resultados marco articulado con NX Siemens en 3D, $x/L=0.5$, $a/h=0.5$.

4.1. Resumen de resultados

En la tabla 4, se muestran los valores correspondientes a los parámetros de desplazamiento para los tres tipos de métodos, como se puede observar la variación porcentual entre el método analítico y la simulación en 2D es de un 1.6%, mientras que la diferencia entre el método analítico y la simulación en 3D es apenas de un 1.16% por lo que dichos parámetros son aceptables y viables para modelar y simular elementos lineales con discontinuidades por medio de modelos 3D en NX Siemens. Asimismo, coincide con las recomendaciones de Robles en cuanto a considerar los resultados volumétricos de los desplazamientos y deformaciones para cuantificar el daño (tablas 1, 2 y 3).

Tabla 4- Viga simplemente apoyada sin discontinuidades.

Viga simplemente apoyada		
Desplazamiento (mm)		
<i>Método analítico</i>	<i>Simulación 2D</i>	<i>Simulación 3D</i>
-0.0429	-0.0422	-0.0424

Para el caso de la viga en voladizo (tabla 5) sin discontinuidades la variación porcentual entre el método analítico y la simulación en 2D varía en un 0.14%, mientras que la diferencia entre el método analítico y la simulación 3D es del 0%.

Tabla 5- Viga en voladizo sin discontinuidades.

Viga en voladizo		
Desplazamiento (mm)		
<i>Método analítico</i>	<i>Simulación 2D</i>	<i>Simulación 3D</i>
-0.673	-0.672	-0.673

La variación de los desplazamientos de una viga simplemente apoyada con daño (tabla 6), correspondiente al punto x/L (donde x es la coordenada de cada punto y L es la longitud total de la viga), en función de la ubicación relativa de cada punto, para distintas relaciones de a/h (donde a es la profundidad y h es la altura total de la viga). Considerando que el valor de referencia es el de Robles la variación porcentual es del 10.2%.

Tabla 6- Viga simplemente apoyada con discontinuidades en comparativa.

Relaciones	Desplazamiento (mm)	
Datos gráfica	Robles	Simulación 3D
Sin daño	-0.040	-0.0424
$x/L=0.5$ y $a/h=0.4$	-0.0470	-0.0518



La variación de los desplazamientos de una viga en voladizo con daño (tabla 7), correspondiente al punto $x/L=5$ (donde x es la coordenada de cada punto y L es la longitud total de la viga), en función de la ubicación relativa de cada punto, para distintas relaciones de a/h (donde a es la profundidad y h es la altura total de la viga). Considerando que el valor de referencia es el de Robles la variación porcentual es del 1.5%.

Tabla 7- Viga en voladizo con discontinuidades en comparativa.

Desplazamiento (mm)		
Relaciones	Robles	Simulación 3D
Sin daño	-0.673	-0.673
$x/L= 0.5$ y $a/h= 0.2$	-0.650	-0.680
$x/L= 0.5$ y $a/h= 0.4$	-0.690	-0.710
$x/L= 0.5$ y $a/h= 0.5$	-0.730	-0.741
$x/L= 0.5$ y $a/h= 0.6$	Falla	-0.787
$x/L= 0.5$ y $a/h= 0.8$	Sin datos	-0.989

La variación de los desplazamientos de un marco biarticulado con daño (tabla 8), correspondiente al punto $x/L=5$ (donde x es la coordenada de cada punto y L es la longitud total de la viga), en función de la ubicación relativa de cada punto, para distintas relaciones de a/h (donde a es la profundidad y h es la altura total de la viga). Considerando que el valor de referencia es el de Robles la variación porcentual es del 6.5%.

Tabla 8- Marco con discontinuidades en comparativa.

Desplazamiento (mm)		
Relaciones	Robles	Simulación 3D
Sin daño	-0.150	-0.145
$x/L= 0.5$ y $a/h= 0.2$	-0.170	-0.162
$x/L= 0.5$ y $a/h= 0.4$	-0.210	-0.216
$x/L= 0.5$ y $a/h= 0.5$	-0.260	-0.277
$x/L= 0.5$ y $a/h= 0.6$	-0.340	-0.372
$x/L= 0.5$ y $a/h= 0.8$	Sin datos	-0.959

De forma general, los resultados obtenidos de la elaboración del presente trabajo se sintetizan en la tabla 9.

Tabla 9- Comparación entre técnicas de simulación.

Robles	NX
Usa elementos cuadriláteros 2D	Usa diferentes tipos de elementos finitos 2D y 3D
No minimiza el error debido al tamaño de paso	Minimiza el error debido al tamaño de paso
Dependiendo del tamaño de paso, es el tiempo requerido para la resolución.	No presenta ningún inconveniente el tamaño de paso para la resolución.
El MEF no se adapta a la forma de la grieta	El mallado de NX se adapta en ambos modelos
Empleo de análisis lineal	Empleo de análisis lineal y no lineal
No emplea ecuaciones de procesamiento en paralelo	Emplea ecuaciones de procesamiento en paralelo
Carece de asociatividad en los modelos simulados por lo que no se puede llevar una rápida modificación de la estructura	Proporciona asociatividad de modelos simulados con modelos CAD, permitiendo una rápida modificación de la estructura
No maneja solucionadores adicionales	Maneja varios tipos de solucionadores
Presenta un esquema de difícil manejo, por lo que se requieren avanzados conocimientos de programación	Se debe contar con un mínimo dominio del programa el cual es intuitivo



Con base a las observaciones realizadas al comportamiento de las gráficas para elementos con daño y sin daño, se puede notar que tanto el método exacto como la técnica de NX Siemens coinciden significativamente más entre ellos. Sin embargo, el resultado de la técnica de Robles se aleja considerablemente de dichos métodos, ello puede ser debido a diversos aspectos como el empleo de elementos cuadriláteros que no logran adaptarse en su totalidad a la fisura, caso contrario de la metodología de NX Siemens que emplea elementos tetraédricos que se adoptan mejor al tamaño de grieta que minimiza el error debido al tamaño de paso y a los diferentes tamaños de los elementos finitos que emplea.

Asimismo, el manejo de una gama extensa de solucionadores en NX Siemens facilita elegir la interfaz y comandos que mejor nos convenga, por lo que facilita la resolución y simulación de los problemas en 2D y 3D. Como hallazgo relevante las figuras 28, 29 y 30, muestran una considerable pérdida de la curva elástica cuando la relación falla a/h se incrementa por arriba del 0.6 de la longitud de la viga, indicando una pérdida de rigidez, coincidiendo con los hallazgos de la técnica de Robles.

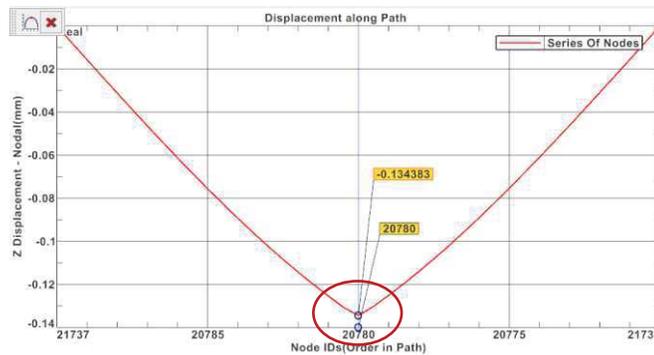


Figura 29- Viga simplemente apoyada $x/L= 0.5$, $a/h=0.8$.

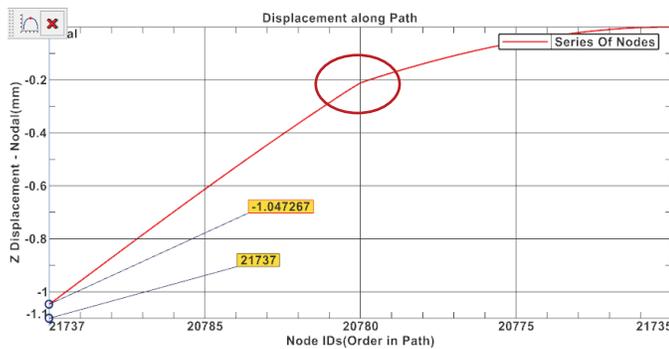


Figura 30- Viga en voladizo $x/L= 0.5$, $a/h=0.8$.

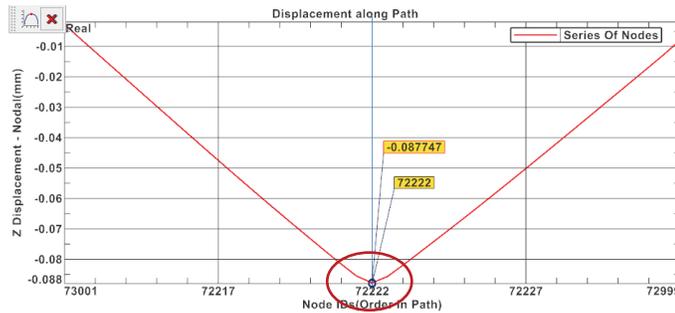


Figura 31- Marco biarticulado $x/L= 0.5$, $a/h=0.8$.



5. Conclusión

En el presente trabajo se buscó una alternativa para poder solucionar problemas tridimensionales de geometrías complejas, para la detección de daño, basada en la respuesta estática de estructuras de acero lineales, tales como vigas y pórticos de concreto armado, empleando la adaptación de un algoritmo de detección de daño el cual incluye dos etapas: detección, localización. Partiendo de la detección, los desplazamientos y deformaciones de una estructura se establecieron en función de su geometría con y sin pérdida del continuo, propiedades del material y de las cargas actuantes. Por lo tanto, se ha concluido que el empleo del software NX Siemens es una herramienta de gran importancia para ser empleada al estudiar patologías en estructuras de acceso limitado, tal que permite determinar la existencia de daño y localizarlo.

Referencias

- [1] Juárez, L. G. (2006). Modelado Numérico de Problemas de Fractura en Solidos Mediante Discontinuidades Interiores. D.F.: UNAM.
- [2] Robles, S. I., & Ortega, N. F. (2005). Detección de daños en elementos estructurales lineales. Mecánica Computacional Vol. XXIV, 19.
- [3] Goncharov, P. A. (2014). Engineering Analysis with NX. Lulu Publishing Services.
- [4] Boroschek, R. (2009). Dinámica de estructuras. Recuperado el 19 de mayo de 2018.
- [5] Hibbeler, R. C. (2011). Mecánica de Materiales. México: Pearson Education.
- [6] Gokhale, N. D. (2008). Practical Finite Element Analysis. Maharashtra, India: Finite to Infinite.
- [7] Arana, J. & Gonzalez, J. L. (s.f.). Mecánica de Fractura. Universidad del País Vasco Euskal Herriko Unibertsitateko Argitalpen Zerbitzua.



Segmentación de Imágenes Aplicado al Proceso de Secado de Granos de Café Pergamino

Palma Cruz Lorena y Nieves Vázquez José Ángel

Instituto Tecnológico Superior de San Andrés Tuxtla.
lorenapalma@itssat.edu.mx, josenieves@itssat.edu.mx

Resumen

El presente artículo muestra una de las áreas de interés de la Visión Artificial para el proceso de secado de granos de café pergamino, mediante la aplicación del método de segmentación utilizando el software Python en conjunto con la biblioteca Open CV, ambos de acceso libre, por lo que son considerados como una opción viable para el desarrollo del presente trabajo. La técnica propuesta permite la caracterización de los granos de café para que a futuro se elabore un código que determine en tiempo real su nivel de secado que indique si se debe retirar la carga de café del horno. El método de segmentación propuesto ha resultado eficiente, demostrando que se puede realizar una comparación entre diversas muestras de granos de café con aproximadamente 30 horas de secado en un horno mecánico, detectando con facilidad las semejanzas entre las muestras obtenidas. Por otra parte, resulta una técnica innovadora puesto que no se encontraron trabajos relacionados al proceso de secado utilizando la técnica propuesta.

Palabras clave: Segmentación, Adquisición de Imágenes, Visión Artificial, Café Pergamino.

1. Introducción

La visión artificial es una disciplina que engloba todos los procesos y elementos que proporcionan ojos a una máquina y se podría decir que: la visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo [1].

Por otra parte, el software que posee las herramientas necesarias para el análisis de imágenes en el área de Visión Artificial, entre los cuales destacan MathCad, Matlab y LabView; la principal desventaja para su utilización es el alto costo de la licencia, además en el caso de LabView tiene un gran número de módulos para diversas aplicaciones (entre ellas Vision Development), que se adquieren de forma independiente. Por las razones anteriores se opta por la utilización del software Python en conjunto con OpenCV (Open source Computer Vision Library).

Python es un lenguaje de programación que se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Python permite dividir el programa en módulos reutilizables desde otros programas, además viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a utilizar Python). También hay módulos incluidos que proporcionan E/S (entrada/salida) de ficheros, llamadas al sistema, Sockets y hasta interfaces a GUI (graphical user interface) como TK, GTK, QT entre otros. Estos últimos son bibliotecas de componentes gráficos.



OpenCV es un grupo de funciones en lenguaje C y algunas clases en lenguaje C++, desarrolladas para implementar algunos de los más populares algoritmos de procesamiento de imágenes y visión artificial con la característica adicional de ser de tipo **open source**, lo que indica que además de ser gratis se puede modificar [2]. La segmentación es una etapa determinante en un sistema de visión artificial. El objetivo de esta etapa es aislar los objetos de interés, para luego realizar el análisis de sus características. Los algoritmos desarrollados están basados en las propiedades locales de color [3]. En el presente documento se aplica la técnica de segmentación para luego encontrar el histograma del objeto bajo estudio. Con la información del histograma podemos determinar la Intensidad de Iluminación vs Cantidad de Píxeles que contiene en el objeto analizado, para determinar los cambios de color en el área de interés, es decir en donde el grano presenta un secado idóneo. Las técnicas de visión artificial se han aplicado en diversos problemas relacionados con la cosecha y procesamiento de café, a continuación se mencionan algunos de ellos. Sandoval-Prieto [4] en Medellín, Colombia, desarrollaron un sistema de visión artificial para la clasificación de frutos de café en once categorías dependiendo de su estado de madurez. Para la descripción de la forma, el color y la textura de cada fruto se extrajeron 208 características, que se redujeron a 9 con base en los resultados de dos métodos de selección de características, uno univariado y otro multivariado. La caracterización de café cereza se evaluó en dos técnicas de clasificación: Bayesiano y Redes Neuronales.

Por otra parte, Guzmán P., y colaboradores [5] en Colombia midieron la severidad de la mancha de hierro del cafeto, que es una enfermedad causada por *Cercospora coffeicola* Berk y Cook en hojas de café, utilizando Matlab 5.3, aplicaron técnicas de procesamiento de imágenes y reconocimiento por color que permitieron obtener con mayor precisión y rapidez el área foliar de plantas de café y el área afectada por la enfermedad o la producida por algún otro disturbio fitosanitario, obteniendo resultados confiables. Herrera y otros autores [6] proponen un sistema de visión artificial que detecta el fruto de café apto para la producción, desarrollando dos algoritmos, uno para la clasificación de los frutos en maduros y no maduros, otro para la detección de la plaga de la broca; extrayendo característica de color y un clasificador bayesiano. Es importante mencionar que actualmente muchos países utilizan la Visión Artificial como una alternativa para mejorar el control de calidad en productos agrícolas, lo cual les permite ser más competitivos en el mercado globalizado pues brinda productos de mayor calidad a los consumidores [7].

Para determinar si el grano de café ya puede ser considerado pergamino seco, en este documento presentamos una técnica novedosa para determinar de manera eficiente las características de los granos de café considerándolos con un nivel de secado óptimo y pudiéndose utilizar como un indicador para terminar el proceso de secado en el horno.

La estructura del artículo es el siguiente: En la sección dos se presenta el algoritmo que se va a utilizar en pseudocódigo, así como el diagrama de flujo de la metodología empleada para aplicar la técnica de segmentación. La sección tres presenta los resultados que se encontraron al emplear la herramienta de segmentación en el análisis de nivel de secado mediante Python-Open CV, así como los histogramas y la comparación de dichos histogramas obtenidos de las diferentes muestras. Finalmente, la sección cuatro presenta las conclusiones y comentarios del trabajo presentado.

2. Segmentación en el análisis de granos de café pergamino.

Como parte de la metodología empleada, se diseña el algoritmo para el análisis de los píxeles de las imágenes obtenidas mediante una cámara de 12 MP (Mega Píxeles). El método de segmentación se emplea para definir el área que indica un nivel de secado óptimo en el grano de café. Por otra parte, mediante la obtención del histograma de las imágenes adquiridas y la matriz de **Intensidad de Iluminación vs Cantidad de Píxeles** se logró caracterizar el grano de café utilizando la intensidad de su color la cual se utilizó para delimitar la cantidad de píxeles existentes en la imagen



que contenía el objeto de interés. Es importante mencionar que se utilizó como referencia las muestras con 30 horas de secado que es el tiempo aproximado en que el grano llega a buen término en su etapa de secado.

Se comparan seis muestras seleccionadas de diferentes procesos de secado, al realizar el análisis antes descrito se pudo obtener la caracterización del grano, que en un proyecto a futuro determinará si éste se debe retirar del horno utilizando muestras seleccionadas en tiempo real durante el proceso de secado.

Como primer punto, es de vital importancia presentar el diagrama de flujo que se muestra (Figura 1) dando a conocer la metodología empleada para la segmentación.

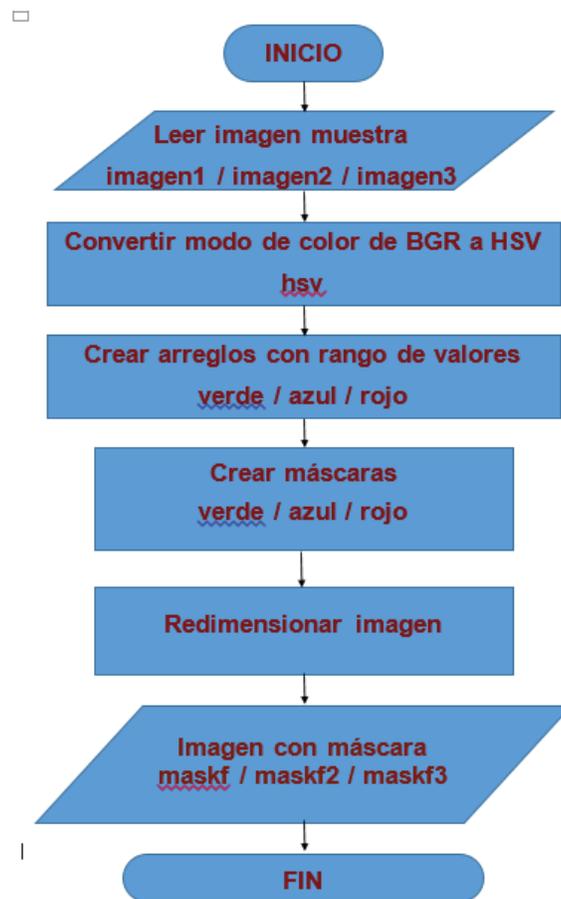


Figura 1. Diagrama de flujo.

A continuación, se muestra el pseudocódigo para aplicar la técnica de segmentación a una sola muestra.

```
Img1←Imagen1;  
hsv↔Img1
```

```
// Definir rangos de colores  
//verde  
Verde_b←array([h,s,v])  
verde_a ←array([h,s,v])
```



```
// azul
azul_a ←array([h,s,v])
azul_b ←array([h,s,v])

//rojo
rojo_a1 ←array([h,s,v])
rojo_b1←array([h,s,v])
rojo_a2 ←array([h,s,v])
rojo_b2←array([h,s,v])

//Creación de los filtros

filtro_verde←(verde_a, verde_b)
filtro_azul←(azul_a, azul_b)
filtro_rojo1←(rojo_a1, rojo_b1)
filtro_rojo2←(rojo_a2, rojo_b2)

mask←filtro_verde
mask2←filtro_azul
mask3←filtro_rojo1
mask4←filtro_rojo2
```

2.1 **Caracterización del entorno para la adquisición de las imágenes.**

Antes de analizar los granos de café es necesario que el entorno en que se realice la adquisición de imágenes cumpla con ciertas características; a continuación se mencionan las consideraciones y pruebas realizadas para que la imagen obtenida sea la óptima para su análisis evitando la reducción en el tamaño de las fotografías lo que daría como resultado una disminución en la información en la imagen. Al realizar estas consideraciones se logra conservar sus características reales lo que ayuda al realizar el procesamiento en la imagen.

2.1.1 **Tipo de iluminación.**

Se realizan pruebas para la obtención de imágenes con diferentes tipos de iluminación para establecer cuál es la más apropiada, que no afecte la calidad del resultado alcanzado.

En primer lugar se ha considerado un entorno con luz artificial (lámpara de luz blanca de la habitación) y una cámara con el flash desactivado (figura 2) y utilizando flash (figura 3).



Figura 2 Grano de café en un entorno con luz artificial y flash desactivado.



Figura 3 Grano de café en un entorno con luz artificial y flash activado.

Como se puede observar, al utilizar la fuente de luz artificial, la gama de colores se torna en verdosos cambiando considerablemente en comparación con la imagen real. Por esta razón, se descarta la posibilidad de tomar las muestras con la luminaria encendida en la habitación donde se realice dicha actividad.

Después se hizo la toma de fotografías con luz natural sin flash (figura 4) y con el flash activado (figura 5).



Figura 4 Granos de café con luz natural, con flash desactivado



Figura 5 Granos de café con luz natural, con flash activado.



Como segunda prueba se realizaron las fotografías con luz natural, de manera similar, se observa que al activar el flash la gama de colores en los granos de café se oscurecen, dando como resultando una imagen saturada de niveles de color cercanos a 0, haciendo que las características físicas del grano no son tan perceptibles como en la imagen en donde no se ha activado el flash. Por lo tanto, se ha determinado, para el trabajo aquí presentado, que la adquisición de imágenes se debe llevar a cabo con luz natural y con el flash desactivado.

2.1.2 Distancia de la lente al objeto de estudio.

Se toman fotografías a diversas distancias entre la lente y el grano de café, a 15cm. (figura 6) en donde las características del grano no se aprecian al máximo, a 12 cm. (figura 7) mejora la captación de las características aunque a 7.5 cm. (figura 8) se puede observar un grano más claro, donde se notan los colores así como los relieves que contiene el grano, por último se maneja una distancia de 3.5 cm. (figura 9) en donde se obtiene una imagen borrosa, de modo que no se aprecian las características requeridas., como se puede observar, la distancia más apta para la toma de muestras es de 7.5 cm; por lo tanto es la distancia a la que se analizaron todas las muestras de las siguientes secciones.

Como se aprecia, entre más corta es la distancia entre el grano y la lente de la cámara, se nota un grano de tamaño más grande además de que cada vez sus características físicas se observan más detalladamente; al llegar a la figura 9 el tamaño del grano es el máximo, pero por lo cercano de la lente se torna en una imagen borrosa.



Figura 6 Imagen captada a 15cm de distancia del grano a la lente de la cámara.



Figura 7 Imagen captada a 12cm de distancia del grano a la lente de la cámara.



Figura 8 Imagen captada a 7.5 cm de distancia del grano a la lente de la cámara.



Figura 9 Imagen captada a 3.5 cm de distancia del grano a la lente de la cámara.

2.1.3 Pruebas de iluminación sobre el objeto de estudio.

Se realizan pruebas de iluminación incorporando una fuente de luz blanca, haciendo variaciones con respecto a los ángulos de inclinación ubicándola en diferentes lugares para observar cuál es la más idónea. Se encontró que la luz blanca dio como resultado proporciones oscuras en el grano de café, en la mayor parte de los casos opacando sus características físicas, como se observa en las figuras de la 10 a la 13; además se generan sombras que afecta el análisis mediante el software, porque se reflejan en los cambios del histograma además de la imagen en forma de matriz donde se puede observar la cantidad de píxeles por intensidad de iluminación. Por otra parte, al comparar con la figura 14 en donde se muestra un grano con iluminación natural, se aprecia que las características físicas del café son mejores expuestas sin la fuente de luz artificial, puesto que no genera sombras sobre la base además de que las características físicas del grano son mejor apreciadas.



Figura 10 Fuente de luz a una inclinación de 30°



Figura 11 Fuente de luz a una inclinación de 45°



Figura 12 Fuente de luz a una inclinación de 70°



Figura 13 Imagen capturada con dos fuentes de iluminación laterales.



Figura 14 Imagen capturada con luz natural

2.3 Muestras utilizadas para la adquisición de imágenes.

Inicialmente, para el análisis de granos de café mediante el software libre Python 3.7 se recopilan tres muestras: como primera muestra se tiene el grano secado por 10 horas, la segunda muestra se obtiene con 20 horas dentro del horno; por último, la tercera muestra consiste en granos de café con 30 horas de secado. Lo anterior para determinar cuáles son las características de cada una de las muestras, que delimiten las semejanzas y diferencias entre las muestras por medio del histograma para analizar los diferentes valores de intensidad de iluminación, así como la cantidad de píxeles que contiene cada uno. Dicha comparación, se utilizará en un trabajo futuro para proponer un algoritmo que determine si el grano es apto o no para sacarlo del horno de secado y continuar con su proceso para el tostado.

Es importante mencionar que se consideran las imágenes que a continuación se muestran (Figuras 15, 16 y 17) que son de granos de café con 10, 20 y 30 horas de secado en el horno; capturadas con luz natural, con el flash desactivado; con una distancia entre el grano y la lente de la cámara de 7.5 cm por medio de una cámara fotográfica Fujifilm de 12 Megapíxeles. Además se han colocado las muestras sobre una base de madera pintada de color blanco para que la luminosidad sobre el grano sea uniforme en cada uno de los casos.



Figura 15 Granos de café con 10 horas de secado



Figura 16 Granos de café con 20 horas de secado



Figura 17 Granos de café con 30 horas de secado



2.4 Aplicación de la herramienta segmentación para análisis de nivel de secado.

Para la ejecución del análisis de la herramienta se utiliza una computadora en la cual se almacenan las imágenes adquiridas, además se requiere la instalación del software Python así como de las librerías OpenCV y Numpy para el manejo de los vectores de la intensidad de iluminación vs cantidad de píxeles del área de interés.

Es importante mencionar que para aplicar el método de segmentación se delimitó el rango de color, saturación y tonalidad para establecer arreglos que permiten filtrar las máscaras de color: verde, rojo y azul; lo que delimitará cuál es el filtro adecuado para el caso de estudio.

3 Resultados

Aplicando el método de segmentación por medio del algoritmo propuesto a una muestra de granos con 10 horas de secado se obtiene la figura 18, que como se puede observar al aplicar las máscaras verde (a la izquierda) y azul (a la derecha) no se hace presente ningún rastro de la presencia del objeto de interés, en contraste con la máscara roja (al centro) es la única que se refleja en la imagen resultante.



Figura 18 Segmentación con 10 horas de secado. De izquierda a derecha: máscaras verde, roja y azul.

Debido a los resultados encontrados, se concluye que las máscaras verde y azul no revelan alguna característica de las tonalidades que caracterizan el secado del grano en las imágenes de la muestra, por lo tanto se determina que para las pruebas sucesivas se aplicará únicamente la máscara roja.

En la figura 19 se muestran las imágenes obtenidas aplicando el algoritmo con únicamente la máscara roja, las muestras de granos de café con 10, 20 y 30 horas de secado, se aprecia visiblemente que la saturación de los tonos en rojo aumenta de forma considerable. Por lo tanto se determina que el algoritmo propuesto es aplicable al caso de estudio puesto que en los resultados obtenidos se observa cómo va aumentando el área existente con el filtro rojo al haber transcurrido mayor en el tiempo de secado.



Figura 19 De izquierda a derecha: segmentación de grano con 10, 20 y 30 horas de secado.

Hasta la presente sección el método de segmentación ha permitido visualizar su eficiencia al comprobar que efectivamente conforme se tienen muestras con mayor tiempo de secado, la cantidad de píxeles filtradas con la máscara roja es mayor.

A continuación se muestra en las figuras 20 y 21 las imágenes procesadas con el filtro rojo aplicando la técnica de segmentación a seis muestras de granos de café con 30 horas de secado.

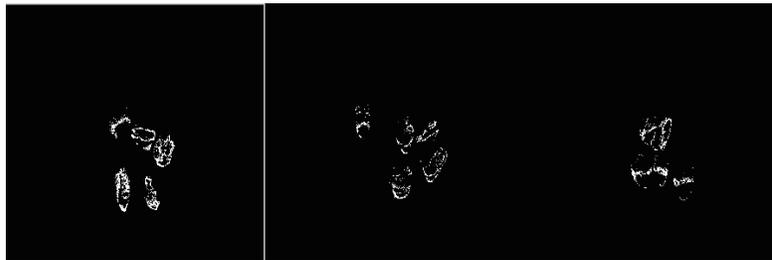


Figura 20 Imágenes procesadas de muestras de grano con 30 horas de secado.



Figura 21 Imágenes procesadas de muestras de grano con 30 horas de secado.

Al haber realizado los análisis anteriores, se requiere cuantificar los datos encontrados por medio de las imágenes obtenidas, por lo que se calcularon los histogramas de las 6 muestras de granos con 30 horas de secado (figura 22). Al interpretar es importante hacer mención que no se toma en cuenta el pico ubicado entre las intensidades de iluminación de 140 a 220 porque éste refleja la cantidad de píxeles del color de fondo de la imagen; por lo tanto se debe observar entre las intensidades 55 a 120 aproximadamente, en donde se indica que el nivel de cantidad de píxeles de todos los histogramas es semejante.

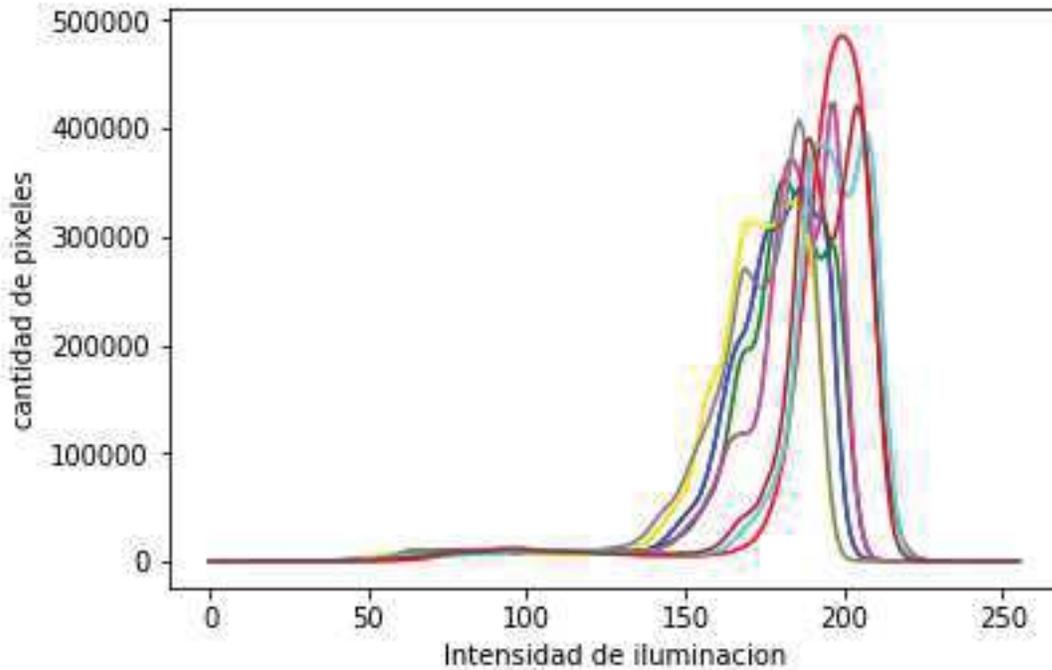


Figura 22 Histogramas obtenidos en ocho muestras con 30 horas de secado.

En la figura 23 se muestra el fragmento de una matriz de Intensidad de niveles de gris (iluminación) vs Cantidad de pixeles obtenida mediante Python, ésta contiene 256 elementos en total, para cuestiones de análisis sólo se extrajeron los rangos de intensidad de interés que se encuentran entre 55 y 120; éstos registros se tomaron para las seis muestras para compararlos de tal forma que se respalden numéricamente los indicadores del histograma, analizando los datos expuestos, la cantidad de pixeles oscila entre 7112 y 9718 en la intensidad de 55; por otra parte en la intensidad 120, la cantidad de pixeles oscila entre 6493 y 9199. Con base en las muestras utilizadas, se calcula un promedio de la cantidad de pixeles de 8130 en el rango de intensidades mencionado (55 a 120).

Matriz Intensidad de Iluminación & Cantidad de Pixeles

45 [5717.]	57 [7875.]	69 [6521.]	81 [6602.]	93 [9047.]	105 [8309.]	117 [6038.]
46 [5726.]	58 [7793.]	70 [6706.]	82 [6858.]	94 [9171.]	106 [8383.]	118 [5811.]
47 [5920.]	59 [7714.]	71 [6987.]	83 [7160.]	95 [9247.]	107 [7786.]	119 [5768.]
48 [6024.]	60 [7580.]	72 [7016.]	84 [7337.]	96 [9151.]	108 [7764.]	120 [5726.]
49 [5943.]	61 [7507.]	73 [7104.]	85 [7783.]	97 [8873.]	109 [7366.]	121 [5530.]
50 [6292.]	62 [7406.]	74 [6878.]	86 [7920.]	98 [8802.]	110 [7358.]	122 [5544.]
51 [6357.]	63 [7265.]	75 [6886.]	87 [8015.]	99 [8608.]	111 [7046.]	123 [5520.]
52 [6265.]	64 [7104.]	76 [6549.]	88 [8033.]	100 [8601.]	112 [6954.]	124 [5564.]
53 [6184.]	65 [7075.]	77 [6383.]	89 [8313.]	101 [8736.]	113 [6641.]	125 [5404.]
54 [6507.]	66 [6819.]	78 [6590.]	90 [8288.]	102 [8679.]	114 [6589.]	
55 [7112.]	67 [6593.]	79 [6486.]	91 [8500.]	103 [8707.]	115 [6275.]	
56 [7712.]	68 [6394.]	80 [6438.]	92 [8644.]	104 [8620.]	116 [6050.]	

Figura 23 Vector en forma de Matriz de Intensidad de Iluminación vs Cantidad de Pixeles de una muestra.



4 Conclusiones

El método de segmentación ha sido aplicado en diversos casos dentro del área agrícola, incluso en algunos trabajos relacionados con el café, aunque en ningún caso en el proceso de secado del grano; por lo tanto la técnica presentada es innovadora.

Es importante mencionar que los resultados obtenidos son eficientes y confiables; se ha demostrado que mediante la segmentación es posible caracterizar los granos de café para determinar su nivel de secado óptimo que puede ser utilizado como un indicador del término de su proceso en el horno mecánico.

Los resultados obtenidos en el presente análisis se aplicarán en trabajos futuros; a corto plazo para desarrollar un algoritmo que determine en tiempo real el momento exacto en que la carga de granos de café dentro del horno de secado mecánico haya finalizado, llevando a buen término dicho proceso.

Un proyecto para desarrollar a largo plazo es que mediante la técnica de segmentación se clasifiquen los granos de café por su grado de tostado, unificando la calidad del producto terminado para que cumpla con las especificaciones de calidad de cada tipo que existe en el mercado.

Referencias

- [1] V.S. Nalwa. A Guided Tour to Computer Vision. Addison-Wesley, 1993.
- [2] Martínez Mejía, David Alberto. Trabajo de Grado: Interceptor de trayectorias basado en visión artificial. 2005. Pontificia Universidad Javeriana. Facultad de Ingeniería.
- [3] Vives, L., Mejía, H. Vilcherrez, K., Vassallo, M. Visión Artificial: Aplicación de filtros y segmentación en imágenes de hojas de café – Rev. Ingeniería: Ciencia, Tecnología e Innovación VOL 1/Nº 2, ISSN: 2313-1926/Diciembre 2014.
- [4] Sandoval Niño, Zulma Liliana. Prieto Ortiz, Flavio Augusto. Caracterización de café cereza empleando técnicas de visión artificial. Revista Facultad Nacional de Agronomía Medellín. Vol.60 No. 2. p.4105-4127.2007.
- [5] Guzmán P., O.A.; Gómez G., E.O.; Rivillas O., C.A.; Oliveros T., C.E. Utilización del procesamiento de imágenes para determinar la severidad de La Mancha de Hierro del Cafeto, en hojas de café. Cenicafé (54(3):258-265.2003.
- [6] J.C. Herrera et al., "Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial," UISIngenierías, vol. 15, no. 1, pp. 7-14, May. 2016.
- [7] Luis ángel Silva, Sergio Lizcano Evaluación del estado de maduración de la piña en su variedad perolera mediante técnicas de visión artificial. ITECKNE Vol. 9 Número 1 • ISSN 1692 - 1798 • Julio 2012 • 31 - 41



Instrumentación Virtual con ISE-LabVIEW®

Castañeda Espinoza Angélica y Cortés Ramírez Ulises

Laboratorio de Mecatrónica, Universidad Tecnológica de Huejotzingo, Santa Ana Xalmimilulco,
Huejotzingo, Puebla, México
ulises@cortes.mitmx.net

Resumen

En este trabajo se da a conocer el lenguaje de programación de ISE-LabVIEW® para crear aplicaciones de instrumentación virtual, esta herramienta es un complemento para el entorno de programación de LabVIEW, dedicada a la adquisición y generación de señales a través de plataformas de bajo costo. El código fuente de ISE-LabVIEW® cuenta con el registro de propiedad intelectual ante INDAUTOR y actualmente está disponible en compilaciones para versiones 2016 a 2019 de LabVIEW en distribuciones de 32 y 64 bits, para sistemas operativos como: Windows, Linux y MAC. El lenguaje de programación de esta herramienta es análogo al de la línea DAQmx de National Instruments, el hardware utilizado son plataformas OpenSource en las que es necesario cargar el firmware de la librería. Con esta herramienta es posible realizar aplicaciones como: regulación de velocidad de motor de DC con inversión de giro, manipulación de un servomotor, lectura de sensores de humedad, temperatura, ultrasónico, acelerómetro, giroscopio y magneto brújula; en este trabajo se tiene el objetivo de dar a conocer la metodología para realizar tareas de adquisición y generación de señales analógicas y digitales, así como del tipo contador.

Palabras clave: Instrumentación virtual, Adquisición de datos, Generación de señales.

1. Introducción

El concepto de instrumentación virtual surge a partir de la implementación de la computadora en los sistemas de medición de los procesos industriales, conformada principalmente por sistemas de adquisición de datos y generación de señales, además de visualizar las variables de proceso, manipular y controlar los actuadores, con la instrumentación virtual se logra que los instrumentos de medición estén integrados en la computadora a través de una GUI (Interfaz Gráfica de Usuario, de sus siglas en inglés), y cuenten con un comportamiento, funcionalidad y apariencia similar al de los instrumentos físicos [1].

La instrumentación virtual consta de dos aspectos sumamente importantes, que son: el software y el hardware; el primero se puede clasificar en tres tipos: Software de interfaz (controladores), Software de aplicación (lenguaje de programación) y Software listo para ejecutar (aplicación realizada por el usuario) [2], mientras que el hardware puede clasificarse por el tipo de interfaz de comunicación, el tipo de tecnología con la que esta implementada o por el campo de aplicación al que esta designado [3].

National Instruments es líder a nivel mundial en cuanto a aplicaciones de instrumentación virtual, su software LabVIEW es una plataforma que integra todas las necesidades para este campo, que incluye un lenguaje de programación orientada a objetos [4] y conectividad con sistemas de adquisición de datos, en la parte de hardware cuenta con diferentes familias de tarjetas de adquisición de datos (DAQ), como:

- **DAQmx:** Esta línea cuenta con tarjetas de adquisición de datos con conectividad USB, PCI, PCIe, PXI, Wi-Fi o Ethernet [5], con tasas de muestreo de hasta 2MS/s,



- contadores de 32 bits [6], cuenta con drivers y librerías de LabVIEW para su configuración y programación [7].
- **CompactDAQ:** Las tarjetas de adquisición de datos de esta línea cuentan con conectividad USB y Ethernet, son sistemas modulares, sus drivers y programación son compatibles con la línea DAQmx, están integradas en chasis ranurados a los que se les puede adaptar diferentes módulos de entrada salida [8].
- **CompactRIO:** Esta línea cuenta con autonomía en la que es posible cargar la aplicación desarrollada en LabVIEW a la tarjeta de adquisición de datos [9], además de que posee compatibilidad con las versiones de LabVIEW Development System, LabVIEW Real Time y LabVIEW FPGA; dispone de conectividad Ethernet, USB, RS232/RS485 y VGA, con velocidades de transmisión de datos de hasta 1000Mb/s.

Además del hardware de National Instruments, también existen opciones de terceros, como:

- Arduino Compatible Compiler for LabVIEW [10]
- Digilent LINX MakerHub [11]
- LabVIEW Interface for Arduino [12]

Los cuales corresponde a hardware comercial designado para aplicaciones embebidas.

ISE-LabVIEW© surgió como parte del desarrollo del Robot ùBot-32b [13] en su fase de instrumentación de bajo nivel, como una necesidad emergente de contar con un monitoreo constante y en tiempo real de cada una de sus variables, las señales a observar estaban en función de la experimentación o pruebas realizadas; por esta razón se implementaron los subvi's o funciones con las que fuera posible indicar las variables y tipo de muestreo de una manera simple y rápida, pero a la vez fiable; con la finalidad de visualizar en una interfaz de usuario la respuesta del control de seguimiento de trayectorias y localización en el plano cartesiano del robot [14], en la Figura 1, se muestra la línea de tiempo de la evolución de la herramienta ISE-LabVIEW©.

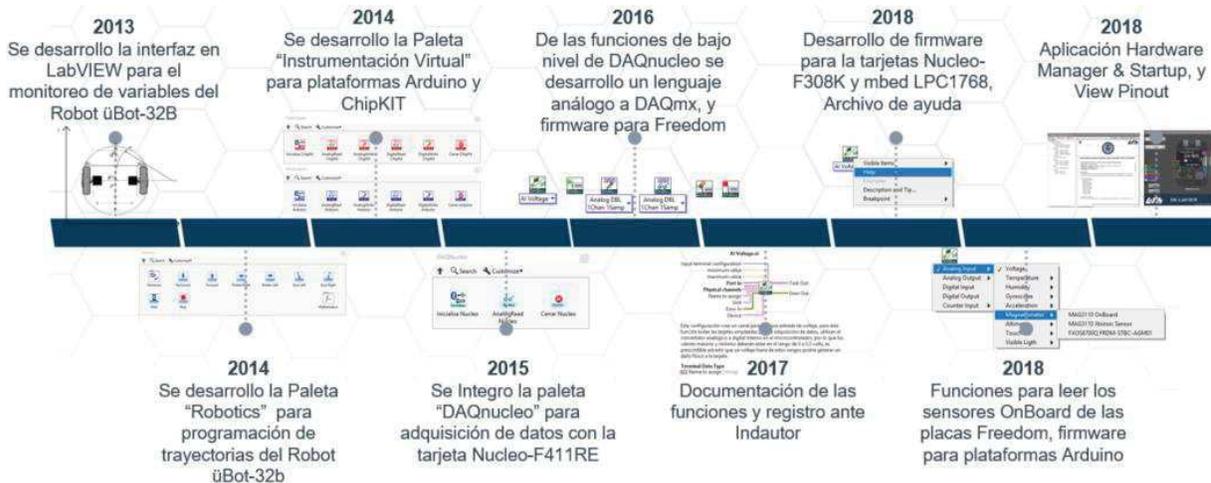


Figura 1. Evolución de ISE-LabVIEW©

La librería ISE-LabVIEW© (Interfaz de sistemas Embebidos con LabVIEW) es integrada a LabVIEW en la paleta de controles y funciones del panel frontal y diagrama a bloques respectivamente, su lenguaje de programación es análogo a la línea DAQmx; es posible utilizar familias de tarjetas de desarrollo como:

- NUCLEO del fabricante ST Microelectronics.
- Freedom de NXP (Freescale).
- mbed de NXP.
- Arduino



Esta aplicación ha sido desarrollada en el laboratorio de Mecatrónica de la Universidad Tecnológica de Huejutzingo y actualmente su código fuente cuenta con el registro de propiedad intelectual con el número: 03-2017-040510192900-01, realizado ante el Instituto Nacional de Derecho de Autor cabe mencionar que se cuenta con una metodología para realizar los procesos de registro de propiedad intelectual.

En este trabajo se muestra el lenguaje de programación para realizar aplicaciones de adquisición y generación de señales, utilizando hardware de bajo costo y con un concepto análogo a la línea DAQmx de National Instruments, esto con la intención de que los estudiantes de ingeniería cuenten con una herramienta alternativa y accesible en la que puedan aprender a realizar sistemas de instrumentación virtual, ya que, una de las problemáticas a resolver es que los estudiantes de la carrera de Mecatrónica de la Universidad Tecnológica de Huejutzingo, al realizar prácticas de adquisición de datos, no se limiten solo al uso del equipo de laboratorio, en los horarios disponibles o asignado, ya que, con esta herramienta contarán con más horas prácticas, lo que les permitirá obtener una mayor experiencia en aplicaciones de adquisición y generación de señales.

2. Interfaz de Sistemas embebidos con LabVIEW (ISE-LabVIEW®)

ISE-LabVIEW® es una librería para LabVIEW, la cual se agrega a las paletas de funciones y controles en el diagrama a bloques y panel frontal respectivamente a través de la plataforma VI Package Manager, fue implementada como primera instancia en sus bloques de bajo nivel en la plataforma Arduino, al realizar un lenguaje análogo a la serie DAQmx de National Instruments e implementar las funciones de adquisición de datos, se trabajó con la tarjeta Nucleo-F411RE del fabricante STMicroelectronics, y posteriormente se han ido añadiendo placas de la familia Freedom, mbed y Arduino. La finalidad de que la herramienta ISE-LabVIEW® posea un lenguaje de programación análogo a la línea DAQmx, es ser una alternativa ante aplicaciones de instrumentación virtual, que brinde a los estudiantes de ingeniería una herramienta en la que puedan desarrollar sus proyectos, y al mismo tiempo aprender el lenguaje de programación de las tarjetas DAQmx y CompactDAQ, con plataformas comerciales y de bajo costo.

Las funciones y controles de ISE-LabVIEW® se agregaron a la paleta de Funciones y Controles de LabVIEW; en una subpaleta nombrada como: "ISE-LabVIEW", se eligió este nombre debido a que será la subpaleta principal que alojará los productos desarrollados en un futuro, hasta el momento esta paleta contiene a DAQise.

Con estas funciones, es posible encender y apagar leds, controlar motores de CD, relevadores, servomotores, o leer sensores de temperatura, luz, humedad, encóder, ultrasónicos, acelerómetros, giroscopios, magneto brújulas, y dispositivos que cuenten con salida o entrada, del tipo analógica o digital.

2.1 Instalación de la paleta ISE-LabVIEW®

La última compilación de ISE-LabVIEW® (al momento de redactar este documento) es la versión 2019.0.0.34, para esta versión se han generado los archivos firmware para cada una de las plataformas incluidas. Es de suma importancia advertir que, si se llegará a cargar un firmware que no corresponda a la tarjeta en específico a emplear, podría generar daños físicos al microcontrolador, esto como consecuencia de que cada tarjeta difiere en la distribución de pines en los conectores, aunque también existe la posibilidad que el programa no se ejecute en el microcontrolador. Para instalar ISE-LabVIEW® es necesario ejecutar el archivo mostrado en la Figura 2, a través de la aplicación VI Package Manager.



Figura 2. Archivo de instalación de ISE-LabVIEW©.

Para que la comunicación entre LabVIEW y cualquiera de las tarjetas de desarrollo integradas en la aplicación ISE-LabVIEW© sea posible, es necesario instalar los controladores VISA (de sus siglas “Virtual Instrument Software Architecture”) de National Instruments, ya que su conectividad es a través del puerto USB con el protocolo COM.

2.2 Administrador de Hardware y Puesta en Marcha de ISE-LabVIEW©

La herramienta ISE-LabVIEW© cuenta con un gestor de tarjetas llamado “*Hardware Manager & Startup*” mostrado en la Figura 3, esta plataforma tiene la finalidad de administrar y designar un nombre único a cada uno de los dispositivos, además de que facilita la generación y carga del firmware, así como la instalación de drivers.

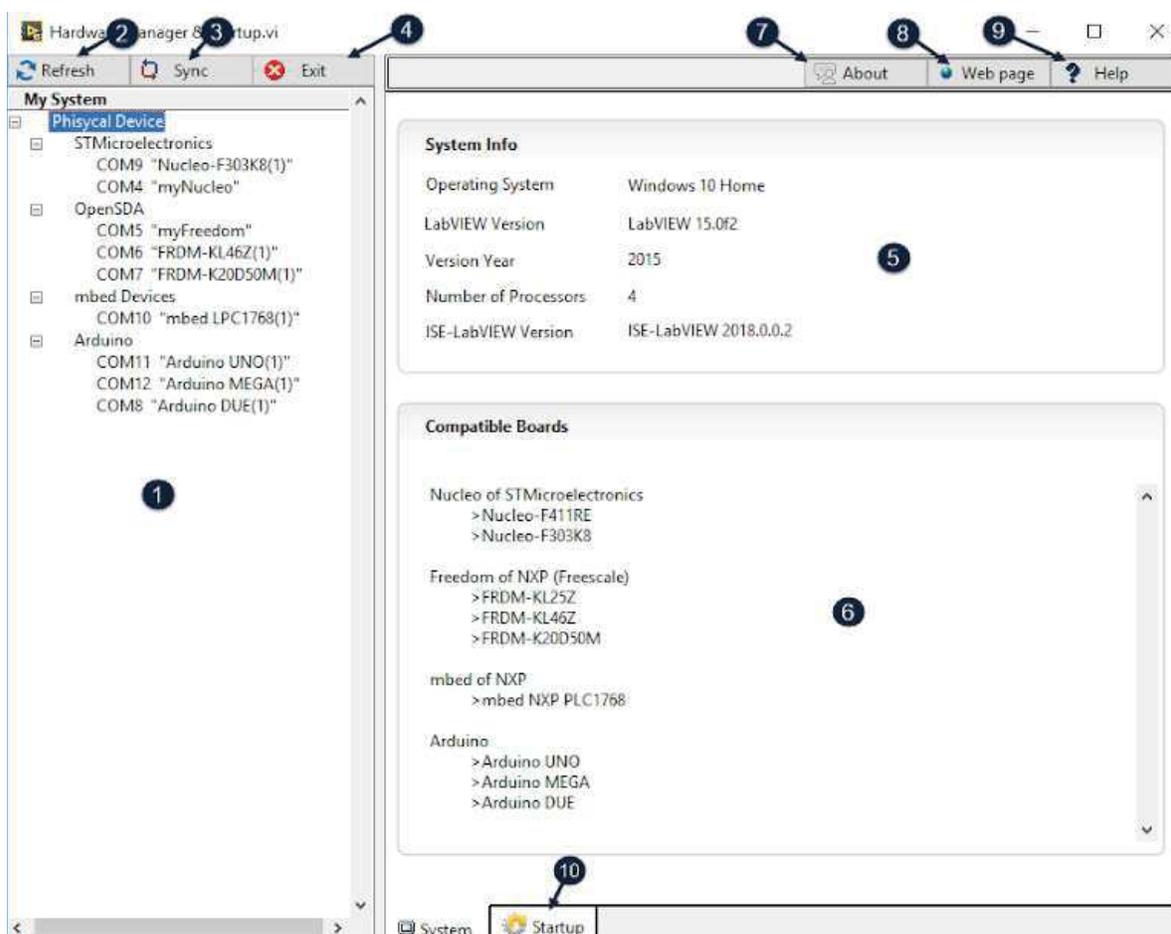


Figura 3. Hardware Manager & Startup de ISE-LabVIEW©.

Las partes que lo conforman son:

- 1) Lista de dispositivos sincronizados con el PC.
- 2) Actualiza la lista de dispositivos.
- 3) Sincroniza los dispositivos conectados.
- 4) Finaliza y cierra la aplicación.
- 5) Información del sistema operativo y versión de LabVIEW instalada.



- 6) Tarjetas compatibles con ISE-LabVIEW©.
- 7) Información de ISE-LabVIEW©.
- 8) Página Web.
- 9) Ayuda de ISE-LabVIEW©.
- 10) Pestaña Startup.

Cada una de las tarjetas compatibles con ISE-LabVIEW©, requieren de un firmware para establecer la comunicación con las funciones DAQise sea posible, este firmware es generado y cargado a través del “Hardware Manager & Startup” en la pestaña Startup como se observa en la Figura 4.

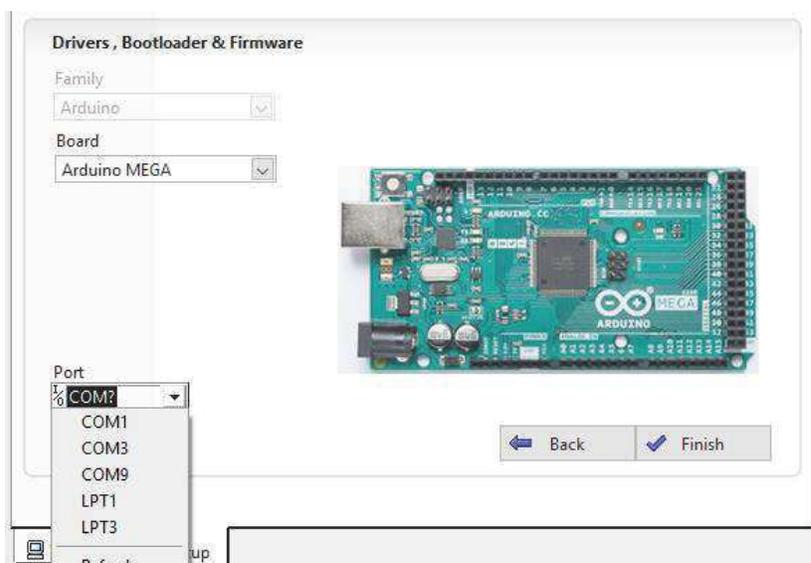


Figura 4. Carga de Firmware a tarjeta Arduino MEGA.

2.3 Plataformas Incluidas en ISE-LabVIEW©

De las plataformas que actualmente cuentan con firmware para la paleta ISE-LabVIEW©, se tienen:

- Nucleo-F411RE
- Nucleo-F303K8
- FRDM-KL25Z
- FRDM-KL46Z
- FRDM-K20D50M
- mbed NXP PLC1768
- Arduino UNO
- Arduino MEGA
- Arduino DUE

Estas tarjetas de desarrollo están implementadas en un hardware compatible con la plataforma Arduino Uno, y tienen embebido un microcontrolador de 32 bits, que en algunos casos alcanzan velocidades de 10KS/s, cada una de las tarjetas tienen entradas y salidas digitales, entradas contador, entradas analógicas y salidas analógica del tipo PWM, además de que algunas plataformas cuentan con salidas analógicas de voltaje.

Las tarjetas de desarrollo de NXP y STMicroelectronics incluyen una interfaz OpenSDA, por lo que, una vez que se conecte al PC a través de este bus será detectada como un dispositivo de almacenamiento masivo correspondiente a la memoria de programa, esto hace que la programación del microcontrolador pueda realizarse con solo copiar y pegar el archivo. S19 para el caso del



bootloader y .bin para el firmware, una vez generados en el *Hardware Manager & Startup*. Previo a ello es necesario, para cada una de las tarjetas instalar sus controladores, para que sea detectado por el PC como un puerto de comunicaciones COM.

2.4 Paleta de controles de ISE-LabVIEW©

La subpaleta de controles ISE-LabVIEW© mostrada en la Figura 5, está ubicada en la paleta de controles del panel frontal, estos controles están implementados en un control del tipo “Combo Box”, y son:

- **Board.ctl:** con este control se indica el tipo de tarjeta que se está utilizando.
- **Physical Channel AI.ctl:** Este control indica el canal analógico a leer.
- **Physical Channel AO.ctl:** Con este control se indica un canal de salida analógica, el cual puede ser PWM o de voltaje según el tipo de tarjeta a utilizar.
- **Physical Channel ctr.ctl:** Con este control se indica un contador/temporizador.
- **Physical Channel DIO.ctl:** Este control indica un canal digital, que puede ser de entrada o salida, un bit o puerto, según la configuración utilizada.

Es importante denotar que cada uno de los controles está desarrollado para una función en específico, por lo que, por ejemplo, usar un Physical Channel AI.ctl para una salida digital no sería adecuada y generaría un error al momento de ejecutar la aplicación en LabVIEW, sin embargo, este uso indebido del control se marcaría con un punto de coerción.

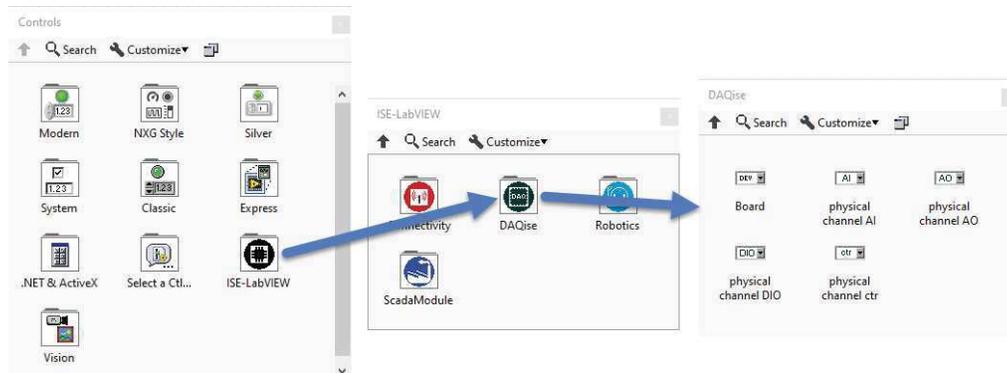


Figura 5. Controles de la paleta ISE-LabVIEW©.

2.5 Paleta de Funciones de ISE-LabVIEW©

La paleta ISE-LabVIEW© mostradas en la Figura 6 está conformada por seis funciones con las que es posible adquirir y generar señales analógicas y digitales, además de entradas tipo contador; de las funciones implementadas se tiene: DAQise Create Channels, DAQise Read, DAQise Write, DAQise Start, DAQise Stop, y DAQise Clear, que posteriormente se describirán más a detalle.

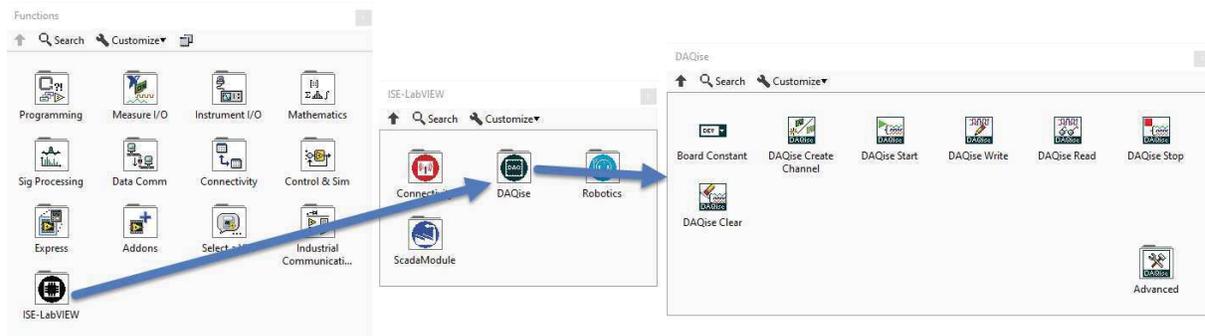


Figura 6. Paleta de funciones de ISE-LabVIEW©.



3. Desarrollo de ISE-LabVIEW©

Como ya se mencionó previamente, ISE-LabVIEW© se desarrolló con un lenguaje análogo a la línea DAQmx de National Instruments, esto con la finalidad de que los estudiantes que realicen aplicaciones con ISE-LabVIEW© les sea de facilidad migrar al entorno DAQmx, por otro lado, para quien tengan ya conocimiento en el uso de hardware de National Instruments, no tengan ninguna dificultad al migrar a esta herramienta.

3.1 Funciones ISE-LabVIEW© vs Funciones DAQmx

Las funciones DAQmx corresponden a la librería Measurement I/O de LabVIEW, y con ellas es posible realizar aplicaciones de adquisición y generación de señales con hardware como CompactDAQ y DAQ de National Instruments, ISE-LabVIEW© está basada en el lenguaje de programación de esta librería. De las funciones principales se tienen:

- **Función para crear un Canal:** Esta función permite crear un canal que puede ser de entrada o salida, ya sea analógico, digital o del tipo contador, en la Figura 7, se muestran la función para crear un canal tanto para la librería DAQmx como para ISE-LabVIEW©.

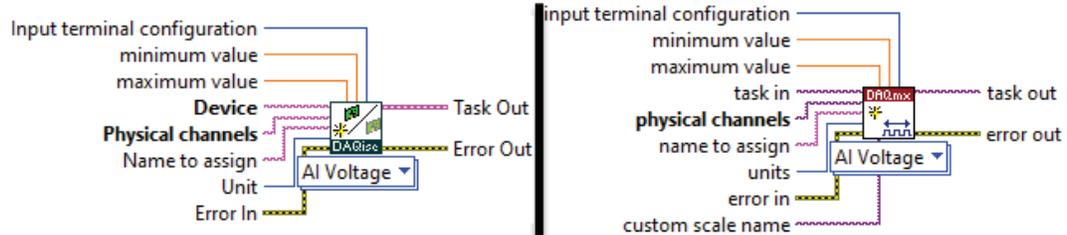


Figura 7. Funciones para crear canal (Izquierda: ISE-LabVIEW©, Derecha: DAQmx).

Esta función está compuesta por un selector donde se indica que tipo de canal se va a crear, para el caso de ISE-LabVIEW© para indicar el canal y el dispositivo, se emplean las terminales Physical channels y Device, en las que se indica el pin y nombre de la tarjeta, a diferencia de la línea DAQmx que solo emplea la terminal physical channels, en ella se indica canal y dispositivo.

- **Función para leer un canal:** Esta función mostrada en la Figura 8, permite la lectura de un canal, se encuentra integrada en un vi polimórfico para seleccionar el tipo de lectura a realizar, si será de una entrada del tipo digital, analógica o contador, y las opciones del número de canales y lecturas a realizar, así como el tipo de dato de los valores leídos.

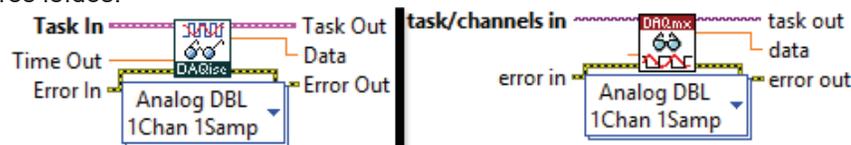


Figura 8. Función para leer un canal (Izquierda: ISE-LabVIEW©, Derecha: DAQmx).

En ambas funciones el canal a leer debe de coincidir con el canal creado, es decir si se creó un canal analógico, debe leerse un canal analógico, para el caso de una lectura de múltiples canales, el rango de canales a leer también debe estar indicado en la función donde se crea el canal.

- **Función para escribir un canal:** Esta función mostrada en la Figura 9, permite la escritura de uno o múltiples canales, que puede ser analógico, digital o contador, tanto la función para ISE-LabVIEW© como para DAQmx, están integradas en vi polimórfico, con lo que se puede seleccionar el tipo de escritura a realizar, así como el tipo de dato. En ambas el tipo de canal a leer debe coincidir con el tipo de canal creado, para el caso



de una escritura a múltiples canales el rango de estos también debe estar indicado en la creación del canal.



Figura 9. Función para leer un canal (Izquierda: ISE-LabVIEW©, Derecha: DAQmx).

- **Función de Inicialización:** Esta función permite inicializar un canal, en la Figura 10, se muestra para ISE-LabVIEW© y DAQmx, a nivel usuario tiene la misma función de inicializar el canal creado, pero en su bajo nivel difieren en que la función DAQmx Start Task está enfocada a inicializar una tarea reservando los recursos del canal creado para una tarea en específico, mientras que la función DAQise Start se enfoca solo en inicializar el canal y verificar que el puerto de comunicaciones esté disponible.



Figura 10. Función para inicializar un canal (Izquierda ISE-LabVIEW©, Derecha DAQmx).

- **Función para detener el canal creado:** Esta función permite detener un canal, en la Figura 11 se muestra para ISE-LabVIEW© y DAQmx, para el caso de las funciones DAQmx cuando la opción auto start se habilita esta función detiene el task creado, para el caso de ISE-LabVIEW© indica a la tarjeta que la comunicación se ha finalizado.

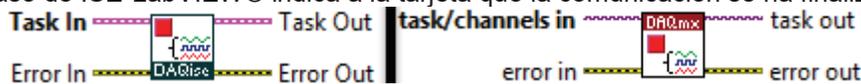


Figura 11. Función para detener un canal (Izquierda ISE-LabVIEW©, Derecha DAQmx).

- **Función para limpiar el canal:** En la Figura 12, se muestra para ISE-LabVIEW© y DAQmx, para el caso de la librería DAQmx esta función libera el recurso reservado para el canal que se creó y para el caso de ISE-LabVIEW© resetea en microcontrolador de la tarjeta que se está usando para limpiar la configuración creada..

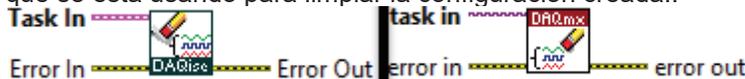


Figura 12. Función para limpiar un canal (Izquierda ISE-LabVIEW©, Derecha DAQmx).

3.2 Polimorfismo en ISE-LabVIEW©

Las funciones DAQise Create Channels, DAQise Write y DAQise Read, están implementadas en un vi polimórfico el cual permite la selección de diferentes opciones, para las tres funciones mencionadas cada opción corresponde a un subvi perteneciente a la librería ISE-LabVIEW©, la implementación es análoga a la línea DAQmx, para el caso de la creación del canal se selecciona el tipo de canal y posteriormente el tipo de señal de medida o escritura como se muestra en la Figura 13.

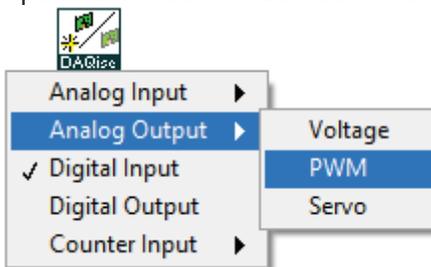


Figura 13. Polimorfismo de la función DAQise Ceate Channels.



Par la función de lectura se sigue la metodología de la línea DAQmx, primero se elige el tipo de canal a leer, si será un solo canal o múltiples canales, precedido de ello se indica si se realizará una sola muestra o múltiples muestras y finalmente se selecciona el tipo de dato en que se obtendrá el valor leído, como se observa en la Figura 14. Y para el caso de la función de escritura se sigue un procedimiento análogo como se muestra en la Figura 15.

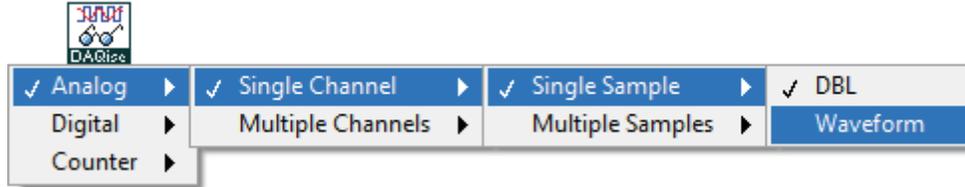


Figura 14. Polimorfismo de la función DAQise Read.

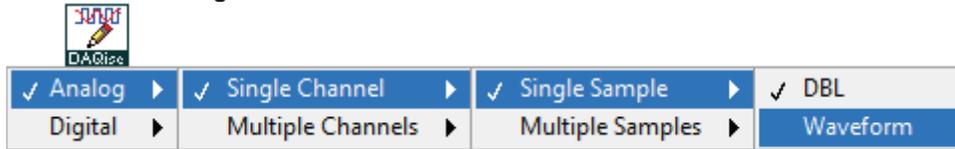


Figura 15. Polimorfismo en la función DAQise Write.

3.3 Errores en ISE-LabVIEW©

Nótese que en la Figura 7, donde se muestra la función DAQise Create Channels, algunas de sus entradas se muestran en letras negritas, esto indica que esas terminales de entrada son requerida y que necesariamente deben estar conectadas, cuando no son conectadas generara que el botón Run de la barra de herramientas de LabVIEW aparezca rota y no permitirá ejecutar el VI, en ISE-LabVIEW© cuando ocurre un error LabVIEW desplegara una ventana emergente como la que se muestra en la Figura 16, en la que se identificara el error con un código numérico, y se indicara una posible causa para ayudar al usuario a corregir el error.

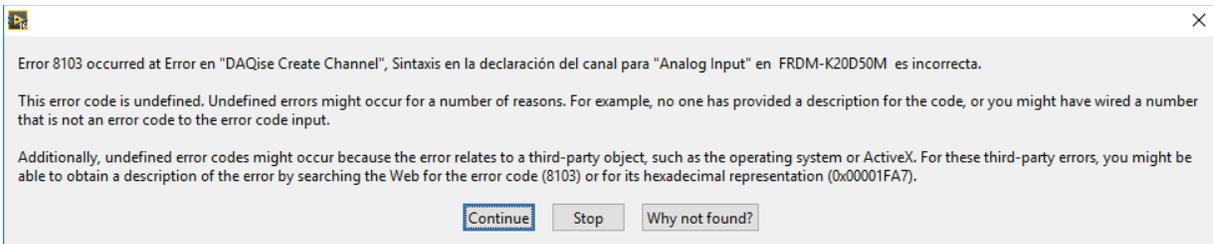


Figura 16. Ventana emergente de error ocurrido en ISE-LABVIEW©.

4. Instrumentación Virtual con ISE-LabVIEW©

La instrumentación virtual se compone de tres elementos esenciales definidos como:

- **Sensores y Actuadores:** Son elementos que se encargan de convertir las señales físicas en señales eléctricas mientras que los actuadores son los que transforman una señal eléctrica en la activación de un proceso.
- **Adquisición o Generación de Señales:** Esta se implementa utilizando una tarjeta de adquisición de datos DAQ (de sus siglas en ingles de Data Aquisition), la cual tiene la función de adquirir y generar señales que son proporcionadas por los sensores para su visualización, y manipulación de los actuadores.
- **Computadora:** Es un elemento físico que a través de un software especializado muestra una interfaz gráfica de usuario GUI (de sus siglas en inglés, Graph Interface User) las variables de adquiridas por el sistema de adquisición de datos, y además contiene controladores y protocolos de comunicación para la conectividad entre la computadora con la tarjeta de adquisición de datos, generando un instrumento virtual.



4.1 Adquisición y generación de señales con ISE-LabVIEW®

Para el desarrollo de sistemas de adquisición de datos y generación de señales con ISE-LabVIEW® se siguió el estándar en el entorno de LabVIEW, el cual consta de tres etapas mostradas en la Figura 17, donde se tiene:

- **Configuración e Inicialización:** en este bloque se comenzará con la creación del canal, donde se indicará si se trabajará con un canal analógico o digital, tipo de entrada o salida; además de indicar el pin que se estará utilizando, las unidades del valor recibido o escrito, los límites máximo y mínimo; seguido de esta configuración se colocará la función para inicializar y limpiar algún dato existente en el bus de comunicaciones de la tarjeta.
- **Lee/Escribe:** en este bloque se colocará una función de escritura o lectura, dependiendo si se realizará una tarea de adquisición o generación de señales.
- **Finalización y Cierre:** con esta función se limpia y cierra el puerto de comunicación al que está conectada la tarjeta que se esté empleando, además de que se resetea su configuración interna.

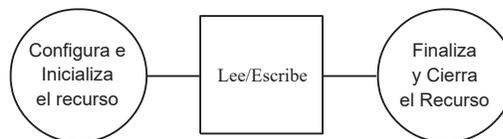


Figura 17. Diagrama a bloques para la Adquisición de Datos.

Para su implementación óptima el bloque lectura/escritura deberá quedar incluido en un ciclo repetitivo (while loop), ya que esta etapa estará realizando la adquisición o generación de la señal, mientras que los bloques de inicialización y cierre solo será necesario que se ejecute una vez, al inicio y al final de la ejecución respectivamente. Con las funciones integradas en la paleta ISE-LabVIEW®, se han implementado ejemplos de aplicación, en los que se realizan lecturas analógicas y digitales, generación de señales analógicas de voltaje y digitales; estos ejemplos, son visualizados en el administrador de tarjetas (*Hardware Manager & Startup*), una vez que el dispositivo este completamente sincronizado.

4.2 Lectura analógica de un potenciómetro

Para la adquisición de datos analógicos, se utilizará un potenciómetro como divisor de voltaje, para la programación en LabVIEW se inicia con la función DAQise Create Channel, con la opción de entrada analógica de voltaje denominada como AI Voltage. Para esta configuración en el "input terminal configuration" se debe elegir el valor de 3.3, ya que el voltaje de referencia del convertidor analógico a digital es de 3.3volts, por lo tanto, el rango del voltaje de la entrada analógica es de 0 a 3.3 Volts, por lo que en los terminales minimum value y maximum value deberán estar dentro del rango de 0 a 3.3V, es de suma importancia considerar que un valor de voltaje mayor a 3.3V o inferior a 0V podría generar un daño físico en el pin. En la terminal Physical channels se deberá indicar como una entrada del tipo AI precedida del número del canal a utilizar y las unidades serán de voltaje; después de configurar el canal se implementará la función DAQise Start. Para la acción de lectura en la función DAQise Read se deberá elegir la opción Analog>>Single Channel>>Single Sample>>DBL.

En la Figura 18, se muestra el diagrama a bloques, nótese que la función DAQise Read está dentro de un ciclo While loop, y para detener la ejecución del ciclo es necesario pulsar el botón de Stop en el panel frontal, si llegará a ocurrir un error, la terminal condicional del ciclo esta concatenada a través de una función OR al status del error, lo que detendría el ciclo cuando ocurra un evento de error.

En la Figura 19, se muestra el diagrama de conexión para realizar una lectura de voltaje a través del pin AI0 en la tarjeta FRDM-KL25Z.

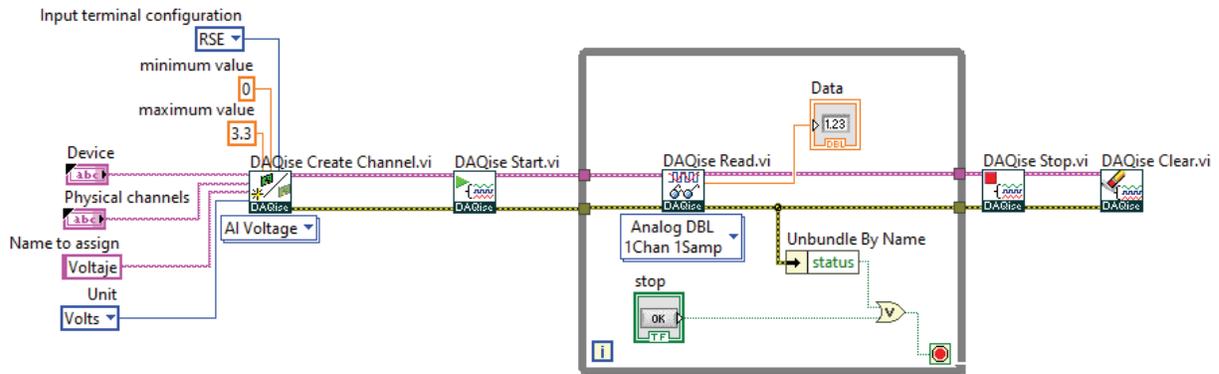


Figura 18. Diagrama a bloque para la lectura de voltaje.

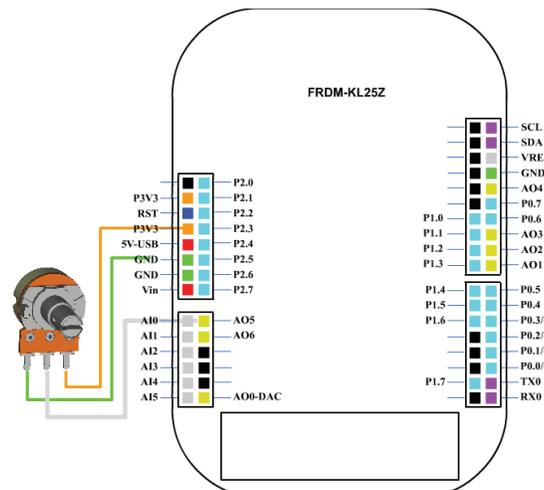


Figura 19. Diagrama de conexión para lectura de voltaje.

4.3 Lectura de un encóder en cuadratura

En este ejemplo se realiza la lectura de un encóder en cuadratura a través de la tarjeta FRDM-K20D50M conectado al contador 0, esta opción está disponible en la función DAQmx Create Channel como CI Angular Encóder. Para esta configuración debido a que los encóder en cuadratura disponen de dos Canales (CHA y CHB), se han hecho pares de dos canales en los contadores que son agrupados de la siguiente forma: “ctr0-ctr1”, “ctr2-ctr3”, “ctr4-ctr5”, “ctr(n)-ctr(n+1)”; aunque para el caso en particular de la tarjeta a emplear solo dispone de un par.

En la configuración se deberá indicar los pulsos por revolución del encóder, las unidades como grados o radianes, el ángulo inicial, y el tipo de codificación, definida como:

- **X1:** El contador del encóder solo realiza cuentas de incremento o decremento si llega un flanco de subida del canal CHA.
- **X2:** El conteo se realiza cuando llega un flanco de subida y un flanco de bajada del canal CHA.
- **X4:** El conteo se realiza con los flancos de subida y los flancos de bajada del canal CHA y CHB.

En la Figura 20, se tiene el diagrama a bloques para leer un encóder en cuadratura, en el selector de la función DAQmx Read se eligió la opción de contador, un solo canal, una sola muestra y un tipo de dato DBL.

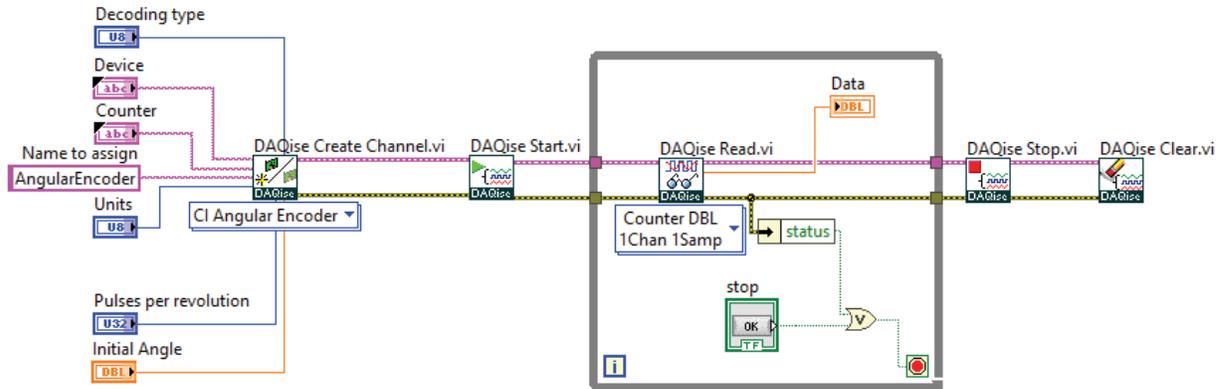


Figura 20. Ejemplo lectura de encóder en cuadratura.

En la Figura 21, se puede observar la conexión del encóder a la tarjeta FRDM-K20D50M, para una lectura en cuadratura, donde el canal CHA del encóder está conectado al pin ctr0 de la tarjeta Nucleo y el canal CHB está en ctr1.

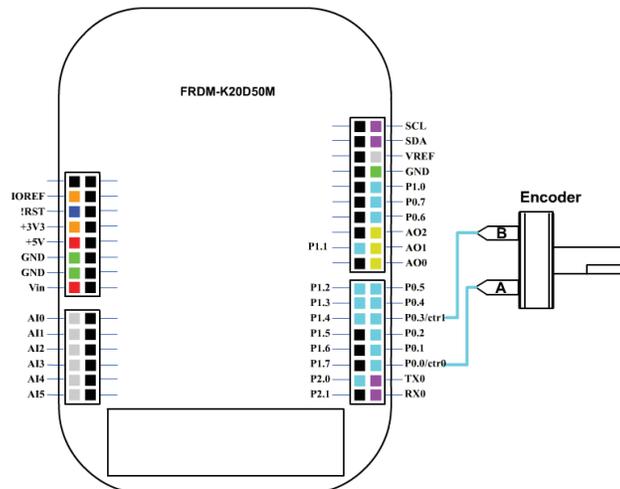


Figura 21. Conexión de encóder en tarjeta Nucleo-F411RE.

Básicamente las salidas de los encóder son trenes de pulsos que constan de 90 pulsos por vuelta, esto quiere decir que si se lee el encóder con una codificación X1 por cada vuelta se tendrá un conteo de 90, si realizamos la codificación X2 se obtendrá un conteo de 180, esto será como consecuencia de un conteo de un solo canal en donde se observarán los flancos de subida y de bajada, y para el caso de la codificación X4 se tendrá un conteo de 360, esto debido a que el conteo se realizará en los flancos de subida y bajada de cada canal, la codificación para saber si el encóder está girando en sentido horario o anti-horario ya está implementada en las funciones ISE-LabVIEW©.

Para conocer los pulsos por vuelta del encóder que se vaya a utilizar es necesario consultar la hoja de datos del fabricante y algo que se debe considerar es la forma de proporcionar los pulsos por canal o cuadratura, es decir para el caso del encóder del ejemplo se tienen 90 pulsos por vuelta de cada canal, y 360 pulsos por vuelta en cuadratura.

4.4 Generación de una señal de Voltaje

Para este ejemplo se generará una señal senoidal para la salida analógica, la función DAQmx Create Channel se configurará como: Analog Output>>Voltage, mientras que la función DAQmx Write se seleccionará la opción Analog>>Single Channel>>Single Sample>>DBL, donde su salida será una forma de onda. En la Figura 22, se muestra el código para generar una forma de onda en un canal de salida analógica, la señal se está creando con la función Generar Forma de Onda, pero también



puede hacerse con una función Basic Function Generator, algo de suma importancia es que la salida analógica de la tarjeta está definida en el rango de 0 a 3.3Volts, por lo que un valor negativo o mayor a 3.3 causara un error en la escritura del canal analógico.

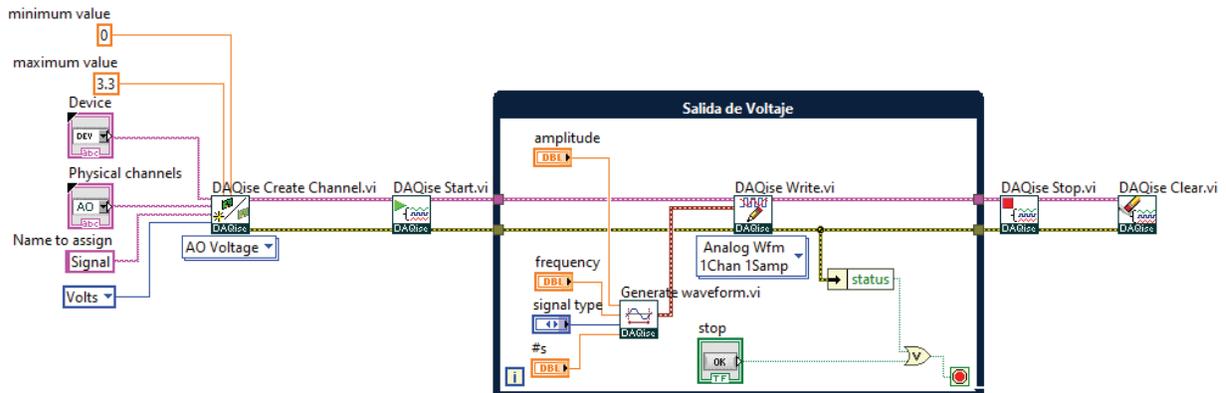


Figura 22. Diagrama a bloques para generar una forma de onda.

En el gráfico de la Figura 23, se tiene la señal de salida del canal analógico de voltaje, la cual corresponde a una forma de onda del tipo senoidal, con voltaje pico-pico de 2.740Volts y una frecuencia de 9.377 Hertz. En el vi de ejemplo se tiene el control signal type, donde se puede elegir una señal diente de sierra, cuadrada o triangular.

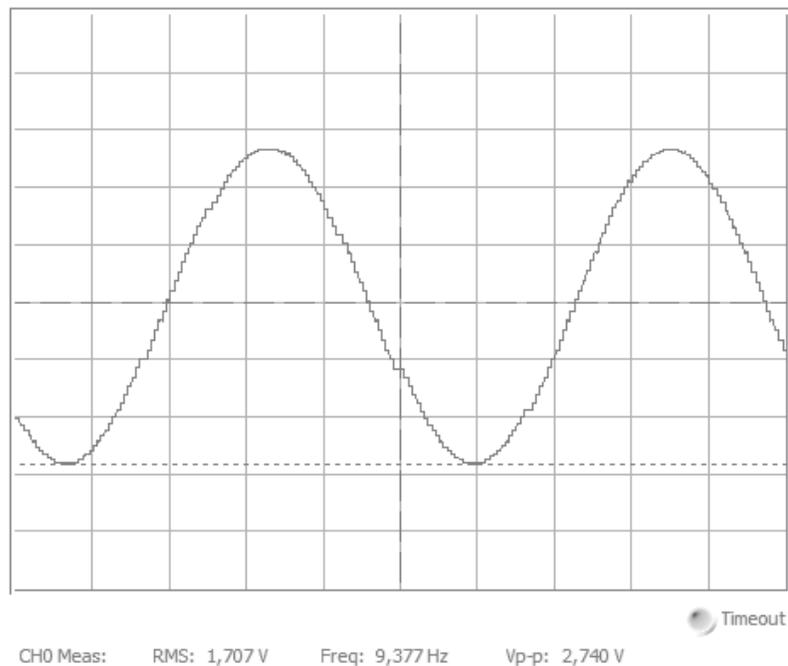


Figura 23. Señal generada en el canal analógico.

5. Conclusiones

Las funciones DAQmx además de generar o adquirir señales analógicas, también brindan la posibilidad de adquirir y generar señales digitales, o a través de sus entradas Contador, es posible leer sensores ultrasónicos o encóder; otra gran característica que tiene ISE-LabVIEW®, es que su programación y configuración es muy similar a la línea DAQmx; debido a ello es ideal como una herramienta didáctica en temas de Instrumentación Virtual, además de que en el desarrollo de la



plataforma se consideraron arquitecturas de 32bits, sus capacidades brindan la posibilidad de resolver aplicaciones reales en el ámbito de prototipado. Actualmente se está trabajando con extensión del lenguaje, e incluir tarjetas como Arduino DUE, LPC1768 de NXP, plataformas ChipKit y el desarrollo de hardware de adquisición de datos propia, la cual cuente con los recursos y acondicionamiento necesario para la adquisición y generación de señales.

Actualmente la última versión se está adaptando para su revisión por National Instruments para su integración en los repositorios de la aplicación VI Package Manager, para que el acceso a ISE-LabVIEW® sea de mayor facilidad, al igual se está revisando su manual de usuario para ponerlo a disposición de los usuarios en medios electrónicos.

Referencia

- [1] E. Valdés Zaldívar and V. Escartín Fernández, "Instrumentación Virtual: Conceptos y Aplicaciones," *Ingeniería Electrónica, Automática y comunicaciones*, vol. XXI, no. 3, pp. 11-15, 2000.
- [2] E. Valdés Zaldívar and V. Escartín Fernández, "Instrumentación Virtual: Elementos de Software," *Ingeniería Electrónica, Automática y Comunicaciones*, vol. XXI, no. 3, pp. 21-25, 2000.
- [3] E. Valdés Zaldívar and V. Escartín Fernández, "Instrumentación Virtual: Elementos de Hardware," *Ingeniería Electrónica, Automática y Comunicaciones*, vol. XXI, no. 3, pp. 16-20, 2000.
- [4] J. del Río Fernández, S. Shariat-Panahi, D. Sarriá Gandul and A. Manuel Lázaro, *LabVIEW, Programación para Sistemas de Instrumentación*, Madrid, España: Alfaomega, 2013.
- [5] R. Nelson, "From USB to PXI, System Aid Capture," *EE: Evaluation Engineering*, vol. 51, no. 8, pp. 12-17, 2012.
- [6] National Instruments Corp., "Data Acquisition, Extending family," *Environmental Engineering*, vol. 18, no. 2, p. 37, 2005.
- [7] J. Essick, *Hands-On Introduction to LabVIEW for Scientists and Engineers*, New York: Oxford, 2016.
- [8] OLYMPUS Corp., "Product Showcase," *Insight: Non-Destructive Testing & Condition Monitoring*, vol. 48, no. 7, pp. 444-447, 2006.
- [9] W. Webb, "National Instruments Extends Embedded Controller," *EDN*, vol. 52, no. 10, p. 19, 2007.
- [10] National Instruments Corporation, "National Instruments," 2017. [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/213121>. [Accessed 27 Mayo 2017].
- [11] National Instruments, "LabVIEW MakerHub," [Online]. Available: <https://www.labviewmakerhub.com/>. [Accessed 2017 Agosto 03].
- [12] National Instruments, "LabVIEW Interface for Arduino Documents," [Online]. Available: <http://forums.ni.com/t5/LabVIEW-Interface-for-Arduino/LabVIEW-Interface-for-Arduino-Setup-Procedure/ta-p/3521346>. [Accessed 2017 Agosto 03].
- [13] U. Cortés Ramírez, A. Castañeda Espinoza, A. Benítez Ruiz and S. A. Arroyo Díaz, "Control de Movimiento de un Robot Móvil Tipo Diferencial, Robot ùBot-32b," *Congreso Nacional de Control Automático, AMCA 2015*, Vols. Octubre 14-16, 2015, pp. 314-319, 2015.
- [14] U. Cortés Ramírez, *Implementación de un Control de Movimiento en Robot Móvil tipo Diferencial*, Puebla: Universidad Politécnica de Puebla, 2014.
- [15] J. Díaz Rodríguez, F. J. Meca Meca and J. A. Jiménez Calvo, *Sistemas de Adquisición de Datos*, España: Universidad de Alcalá, 1995.
- [16] F. Pérez Roque, E. Valdés Zaldívar and O. Arias de Fuentes, "Sistema de Adquisición de Datos con Comunicación Inalambrica," *Ingeniería Electrónica, Automática y Comunicación Inalambrica*, vol. XXXIV, no. 3, pp. 63 - 73, 2013.
- [17] J. C. Herrera Benítez, "Comunicación USB entre Aplicaciones Desarrolladas en LabVIEW y Microcontroladores de Silicon Labs," *Ingeniería Electrónica, Automática y Comunicaciones*, vol. XXXIV, no. 2, pp. 1-8, 2013.



Vivero con Sistema de Riego Inteligente, Monitoreado con Sensores

Itzel Pahua Angel, Aguilar Chamu Hassel Yabin y Ileri Valeria Villaseñor García

Universidad Michoacana De San Nicolás De Hidalgo

Resumen

Los cambios climáticos han causado la pérdida de más de 100,000 hectáreas de cultivo conllevando a una caída económica, por ello se presenta la elaboración de un vivero con el fin de aumentar la calidad y producción de los cultivos, en el que se crea un ambiente artificial totalmente aislado, donde se pretende controlar las magnitudes físicas como la temperatura, la humedad del suelo, la humedad, temperatura del aire y la luminosidad para desarrollar de una mejor manera el cultivo y protegerlo de los bruscos cambios climáticos que se presentan en la actualidad. El proyecto se elaboró principalmente aplicando sistemas de control para poder manipular las variables físicas, de tal manera que se establezcan los parámetros necesarios de cualquier tipo de cultivo, por lo que el sistema por si solo se estabilizara a las características necesarias. Debido a la aplicación de sistemas de control, se establece el momento exacto en que el cultivo debe ser regado por lo cual el sistema de riego es automatizado, de esta manera el ambiente artificial se encarga del total desarrollo del cultivo sin la necesidad de la intervención personal.

Palabras clave: Cambio climático y vivero inteligente.

1. Introduccion.

Un vivero es un conjunto de instalaciones agronómicas en el cual se cultivan todo tipo de plantas hasta que alcanzan el estado adecuado para su distribución y venta [1]. Los viveros tienen como requerimientos específicos para el crecimiento y desarrollo del cultivo por lo tanto es necesario el análisis de su ubicación ya que este debe de contar con los siguientes aspectos que son [1]:

- Umbráculo
- Invernadero con cobertura de plástico
- Invernadero de cristal (cada vez más en desuso)
- Embalse
- Terreno de cultivo al aire libre

Estas características deben ser adecuadas para el tipo de cultivo que se desea cosechar ya que dependiendo de ello se verá si el terreno es el adecuado o no, además se debe de tomar en cuenta la temporada en que se puede dar el cultivo. Todo ello conlleva a que la selección del cultivo se vaya reduciendo a solo unos pocos que puedan sobrevivir con las condiciones y características donde se construya el vivero [1].



Figura 1: vivero convencional

En los últimos años el cambio climático ha afectado los cultivos provocando una problemática en la obtención de una de las necesidades básicas del ser humano por lo cual afecta tanto a la población en general como a la economía del país [2].

En México según el Instituto Nacional de Estadísticas y Geografía (INEGI) se perdió en el último año 2018 un total de 102,000 hectáreas de cultivo a lo largo de la república por el cambio climático, causando una afección en la demanda de ciertos cultivos y una decaída económica por las pérdidas de estos [3].

La principal problemática que está causando el deterioro de los cultivos es el cambio climático ya que este afecta directamente los cultivos y no se tiene una manera de protección para ellos [3].

En otro enfoque existe otra problemática y es que el 80% de la superficie cultivable del país es considerada "temporal", es decir, se siembra sólo en alguna época del año, apuntan datos oficiales. Son tierras que no reciben riego artificial, pues las cosechas dependen de las lluvias y el sol que reciben durante el periodo de cultivo [3].

La mayor parte de esta superficie pertenece a campesinos de escasos recursos, con poca producción por hectárea. Éstos han sido los más afectados con las variaciones del clima, señala el Ministerio de Agricultura [2].

Según un informe presentado el Foro Panamericano de la Ingeniería al Mejoramiento del Medio Ambiente, en los últimos años el país registra variaciones significativas en su clima. Los fenómenos perjudican sobre todo a comunidades indígenas, donde se concentra la mayoría de la población marginada del país, según el Instituto Nacional de Estadísticas [3].

La creación de ambientes artificiales es algo que el hombre ha aplicado en las últimas décadas principalmente para controlar el ambiente de manera deseada, por ello la propuesta presenta la implementación de un sistema artificial dentro de un vivero teniendo el control de las características esenciales del cultivo, esto equivale a que los cultivos estarán protegidos y su crecimiento y desarrollo serán controlados sin temor a que las afecciones climáticas puedan dañar la cosecha brindando protección y un desarrollo seguro del cultivo.

2. Estructura y funcionamiento del sistema.

Los ambientes artificiales son creados por el ser humano, ya que estos son adaptados a la conveniencia y necesidades que se desean dando la principal característica que es la adaptación o enfoque al que se le quiera dar al ambiente. La creación de un ambiente artificial nos asegura un control de las características ambientales necesarias para el desarrollo y cuidado del cultivo por lo que el proyecto se enfoca en la adaptación de un espacio donde se controle la temperatura, la humedad



del aire, la humedad del suelo, el riego y la luminosidad, de tal forma se establecerán las magnitudes necesaria para cualquier tipo de cultivo sin necesidad de buscar una zona con las características necesarias o la espera de temporadas agilizando la cosecha.

Como tal un sistema artificial debe constar principalmente de un control con el cual determine si las magnitudes físicas del ambiente son las ideales en base a las características ambientales necesarias para el crecimiento y desarrollo de la planta, dentro de este control se tiene lo que son sensores y actuadores para estabilizar el sistema, indicadores y en algunos casos alarmas, además se debe contemplar que el ambiente artificial debe de estar aislado del medio ambiente de tal manera que se un sistema independiente del otro.



Figura 2: diagrama del sistema o ambiente artificial.

Para la implementación del proyecto se creó un sistema en cual este aislado del medio ambiente principalmente, pero teniendo las características de un vivero convencional utilizando un umbráculo similar y una malla contra sombra.

2.1 MICROCONTROLADOR

Para la elaboración del proyecto se utilizó el microcontrolador arduino por su compatibilidad con los sensores y su entorno de programación además sus características nos permiten utilizar varios sensores al mismo tiempo.

Arduino UNO es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector jack de alimentación, terminales para conexión ICSP y un botón de reseteo.



Figura 3: microcontrolador arduino



2.2 REGISTRO DE TEMPERATURA Y HUMEDAD

Para el registro de temperatura y humedad se utilizó el sensor DHT11, es un sensor digital de Temperatura y Humedad. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de los datos.

Este sensor trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad (RH) de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos.

Estas características nos favorecen debido a que se pueden determinar dos magnitudes físicas a la misma vez utilizando solo un pin del arduino sin exceder la frecuencia de muestreo.

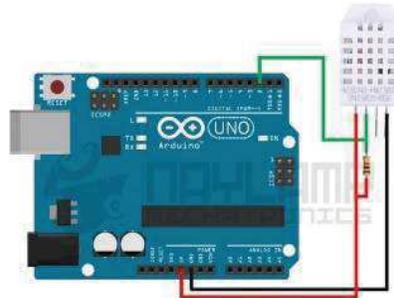


Figura 4: conexión del sensor DHT11 con arduino

El sensor de humedad de suelo (higrómetro) también conocido como modulo FC-28 es un sensor empleado mayormente en sistemas de riego automático, su funcionamiento es simple permite medir la conductividad del suelo, si el suelo está muy húmedo mayor será la conductividad y si el suelo está muy seco la conductividad será mucho menor. El modulo FC-28 que es la sonda que se utiliza para medir la humedad viene acompañado de otra board que es la que se encarga de entregar la medición del sensor. El sensor en conjunto con la board encargada de las mediciones entregara resultados análogos entre los valores 0 (suelo muy húmedo) y 1023 (suelo demasiado seco).

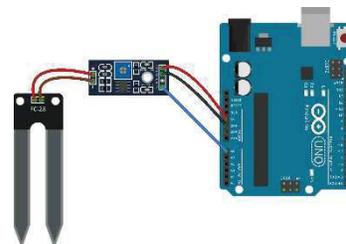


Figura 5: conexión del sensor de humedad del suelo con arduino

Con la medición de este sensor se ajustará un control de humedad para mantener el valor en las características necesarias de la planta, permitiendo un rápido crecimiento y desarrollo

2.3 NIVEL DE AGUA Y SISTEMA DE REUTILIZACION

El sensor de ultrasonidos HC-SR04, se compone por un emisor y un receptor de ultrasonidos que trabajan a una frecuencia de 40KHz.

Con este sensor se puede determinar una cierta distancia en base a la medición de una onda ultrasónica y el revote de la misma.

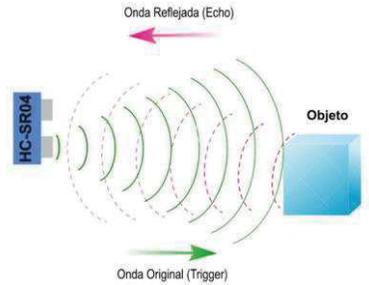


Figura 6: principio de operación del sensor ultrasónico

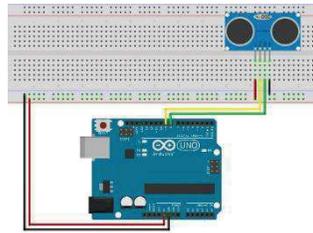


Figura 7: conexión del sensor ultrasónico con arduino

En base esto nos permite medir el nivel de agua en contenedor principal y si esta llegara a descender emitir una alarma indicando la falta de agua a su vez el sistema cuenta con un sistema de recuperación de agua esto en base que a al momento de regar gran cantidad de agua es desperdiciada y el mismo cultivo no alcanza a captar toda el agua en el mismo instante por ello en la parte baja del vivero tendrá conductos que llevaran esta agua a un contenedor donde se le brindara una determinada oxigenación cada cierto tiempo para evitar el estancamiento y la generación de bacterias, esta reserva de agua se puede volver a reutilizar para el riego dando una forma sustentable en el emplea miento del sistema.

2.4 BOMBA DE AGUA

La bomba sirve para trasladar agua a superficies que se encuentran elevadas. En este caso utilizamos dos bombas una para aspersión de agua y la otra para la reutilización de agua las que se utilizaron en fueron: bomba de agua mini 6-12V 6w y Mini Bomba de Agua Sumergible 6V estas fueron elegidas por sus siguientes características:

- ❖ Bomba de agua mini 6-12V 6w
- ❖ Voltaje de funcionamiento: 6-12V DC
- ❖ corriente de funcionamiento: 0.5 - 0.7A
- ❖ Caudal máximo: 1-3L*Min
- ❖ cabeza máxima: 3 m
- ❖ Rango de aspiración máxima: 2 m
- ❖ Temperatura: 80°C Máximo
- ❖ Vida útil: 2500H
- ❖ Diámetro de entrada de la tubería: 6mm
- ❖ Mini Bomba de Agua sumergible 6 V
- ❖ Voltaje de operación: 2.5-6 V
- ❖ Elevación máxima: 40-110 cm
- ❖ Flujo: 80-120l/h (2 litros por minuto)
- ❖ Tamaño de orificio de salida: 7.5mm
- ❖ Tamaño de orificio de entrada: 5mm
- ❖ Diámetro: aprox. 24mm
- ❖ Longitud: Aprox. 45mm
- ❖ Altura: Aprox. 30mm



- ❖ Material: Plástico
- ❖ Levante: ≈40cm-110cm



Figura 8: Bomba de agua

2.5 GENERACION DE CALOR.

Esto se genera a través del uso de resistencias térmicas, las cuales dependiendo de la corriente inducida generara cierto calor, de tal manera que serán nuestra fuente de calor para mantener una temperatura estable.



Figura 9: resistencia calorífica

En este caso se utiliza un control de potencia en base a la señal que proporcione el microcontrolador será la generación de calor para el sistema.

2.6 LUMMINOSIDAD.

La luz es uno de los factores más importantes que regulan el crecimiento y desarrollo de las plantas. Es precisamente este efecto regulador de la LUV en general el que puede tener buenos beneficios en cultivos controlados. Los estudios aplicados que se han realizado hasta el momento, aseguran que de este modo se pueden conseguir efectos como la regulación del crecimiento y desarrollo de los cultivos y mejoras en el color.

Para ello se implementó un control de luminosidad de manera analógica para establecer el nivel de luz en el vivero de esta manera la fotosíntesis no se ve interrumpida y se regulariza la cantidad de luz que necesita el cultivo esto debido a que ciertos cultivos requieren más de sombra que de luz o viceversa.



Figura 10: aplicación de luz ultra violeta en plantas

Los potenciómetros limitan el paso de la corriente eléctrica (Intensidad) provocando una caída de tensión en ellos al igual que en una resistencia, pero en este caso el valor de la corriente y la tensión en el potenciómetro las podemos variar solo con cambiar el valor de su resistencia.

De esta manera se puede controlar el paso de corriente para bajar o elevar la luminosidad del vivero, teniendo un control análogo pero a la vez este valor análogo es leído por el microcontrolador para determinar la cantidad de luz en base a la cantidad de corriente que se deja pasar a la fuente de luminosidad.

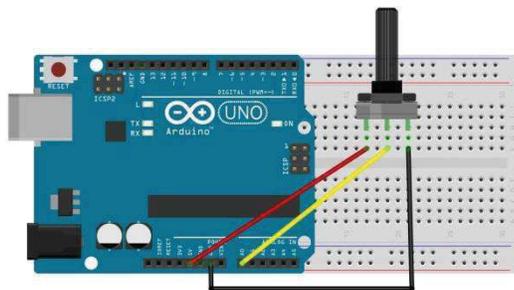


Figura 11: lectura analógica de un potenciómetro con arduino

El riego automático es un sistema para proveer con agua a las plantas del jardín de manera tecnificada, por medio de sistema de aspersión y/o goteo normalmente. Se puede entregar el agua en la ubicación, cantidad, frecuencia y horario que se desee.

2.7 VENTILACION

La ventilación es parte importante del sistema ya que en base a su funcionamiento se mantendrá una oxigenación en el ambiente artificial y fungirá como el actuador para bajar la temperatura. En muchos casos los ventiladores son empleados debido a que su objetivo es limpiar y regularizar la entrada del aire.



Figura 11: Ventilador.

2.8 AISLAMIENTO DEL MEDIO

Para el aislamiento entre el medio externo y el medio interno que en este caso es el ambiente artificial se utilizaron los mismos materiales para los viveros convencionales pero de manera que el sistema fuese sellado para mantener un aislamiento y no afecten los factores externos al cultivo.

Para la cuestión del techo se utilizó igualmente una malla-sombra que es una herramienta esencial para el crecimiento de las plantas en cultivos agrícolas, ya que los rayos UV del sol pueden causar daños a nuestro cultivo como deshidratación, la insolación y las quemaduras del sol es por esto que se requiere que los cultivos estén protegidos y la mejor forma para realizar esto es con una malla que les brinde cierta cantidad de sombra pero sin bloquear los efectos benéficos que el sol le da a las plantas. Es por eso que utilizamos la malla-sombra Monofilamento 70% Sombra, esta le proporciona sombra a la cosecha y beneficia su crecimiento, también funciona como una barrera contra los cambios de clima, ataques de depredadores y para evitar la propagación de plagas.

2.9 CONTROL

El control es la parte esencial del proyecto ya que con él se crea el ambiente artificial en base de las características necesarias para la planta para ello se realizaron 5 tipos de control los cuales fueron:

- Control de temperatura
- Control de humedad del aire
- Control de humedad del suelo
- Control de luminosidad
- Control de nivel de agua

Hay diferentes tipos de control por lo que en nuestro caso utilizamos varios de tal manera que se tenga un ambiente totalmente controlado y más que nada manipulable para poder cosechar varios tipos de cultivos a lo largo de todo el año.

2.10 CONTROL DE TEMPERATURA.

En este caso, fue el control más completo y difícil de elaborar ya que para su funcionamiento se implementó un control proporcional derivativo e integrador (PID) de manera analógica y con lectura digital para la facilidad de solo establecer la temperatura deseada y el sistema por sí solo empiece a estabilizarse con el tiempo.

Para ello se realizó un circuito donde utilizamos amplificadores operacionales para determinar las ganancias k_p , k_i y k_d el cual fue el siguiente:



Figura 12: control de temperatura

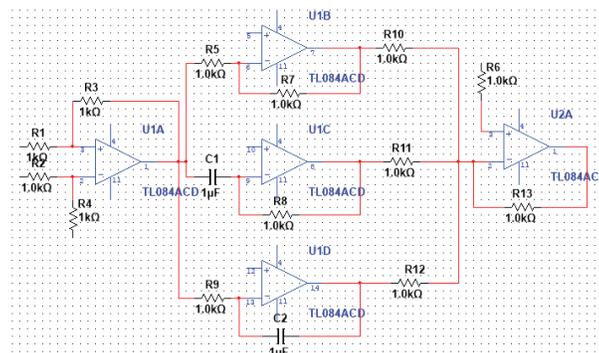


Figura 13: control PID

Para la aplicación de este control se establecieron unas ganancias para que la respuesta no tardará mucho en estabilizarse, para la facilitación de la implementación se realizó una conversión acerca del sensor para que en este caso y como debe ser el sensor sea la aparte de retroalimentación y de esta manera eliminar el error, para ello el set-point lo pone el arduino para solo establecer la temperatura deseada, de allí si la respuesta del control PID nos indica una caída de temperatura se encenderán las resistencias y los ventiladores para mantener la temperatura deseada. En el caso de que la temperatura se empiece a elevar solo se activaran los ventiladores para estabilizar la temperatura, la importancia de este control es que se realizara de manera automática sin la necesidad de una intervención directa.

2.11 CONTROL DE HUMEDAD DEL AIRE

Para este caso solo se implementó un control PD de manera digital donde el arduino que es nuestro microcontrolador realiza una conversión de la lectura del sensor donde si la humedad empezara a bajar se activaría la esparció de gotas de agua a través del aire de manera controlada, en dado caso de que la humedad empezara a subir con las mismas resistencias caloríficas se trataría de bajar la humedad. De esta parte sale una problemática en la acción de dos sistemas de control el de temperatura y el de humedad del aire ya que para eliminar la humedad excesiva de aire se tiene que incrementar la temperatura pero el control de temperatura no lo permitiría ya que es un control más completo por ello solo se tomaría un máximo de 10 minutos para bajar la humedad, en esos 10 minutos el sistema de control de temperatura será apagado momentáneamente para estabilizar la humedad después de esos 10 minutos se volverá a activar el control de temperatura para ajustarla a la deseada de tal manera que el sistema se regulara en un máximo de 15 a 20 minutos determinando la humedad necesaria en conjunto con su temperatura.

Para este control no es necesario la implementación de un sistema analógico ya que el arduino lo podrá resolver de manera digital solamente maneja la activación de las resistencias y las bombas para la esparció de micro gotas de agua.



Figura 14: control de humedad del suelo

2.12 CONTROL DE HUMEDAD DEL SUELO

Para este control se utilizó igualmente un control PD igualmente de manera digital, también hay de recalcar que con este control se implementa lo que es el riego automatizado ya que en base a las características de la planta si la humedad del suelo de esta baja, nos indica que necesita el consumo de agua, por lo que se mantendrá siempre hidratada de tal manera que nunca se podrá ver afectado el cultivo por sequía, a la vez nosotros determinamos el rango de humedad en porcentaje para la facilitación de su uso, de igual manera el sistema tratara de mantener la humedad indicada y necesaria del cultivo.

En este control interviene lo que es sensor de humedad del suelo FC-28 quien funge como la retroalimentación de este control y como actuador tenemos la activación de aspersores de goteo para no dañar el cultivo con la presión del agua. En este caso es una aspersión diferente a la que se menciona en el control de humedad del aire ya que en ese caso se usa una aspersión de micro goteos y en el caso de la humedad del suelo son gotas más grandes para poder brindar la hidratación necesaria al cultivo.



Figura 15: control de humedad del suelo

2.13 CONTROL DE LUMINOCIDAD.

Este control es más fácil del sistema ya que es un control proporcional solamente donde mediante un potenciómetro se establece la luminosidad necesaria del cultivo, para ello solamente se lee la lectura analógica del potenciómetro a través del microcontrolador para establecer el rango de luminosidad en porcentaje.



Figura 16: Control de luminosidad

2.14 CONTROL DE NIVEL DE AGUA

En este caso se implementó un control ON-OFF de manera digital, esto debido a que se reutilizara el agua excedente en el riego por lo que si en el contenedor principal que es la fuente de agua del control de riego, llegara a bajar el nivel de agua a un estado menor al 25% de su capacidad total del contenedor se encenderá el bombeo del agua reutilizable al contenedor principal de esta manera el proceso se hace sustentable, por ello no se ocupa un control más potente ya que solo se ocupa el bombeo de agua.

Para este sistema de control se implementa lo que es el sensor ultrasónico que nos permitirá ver el volumen de agua en el contenedor principal, en dado caso de que el contenedor principal llegara a descender y no hubiera agua en el segundo contenedor (agua reutilizable) se encenderá un indicador que avisara de la falta de agua.

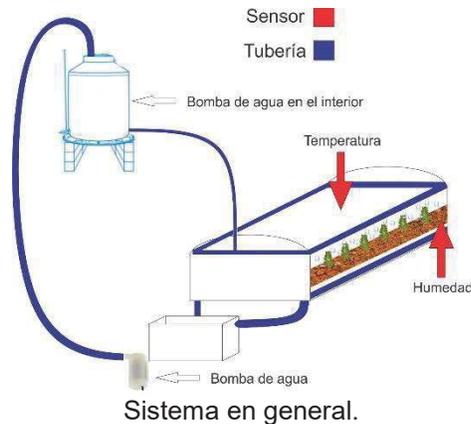
Para evitar el estancamiento del agua a reutilizar se agregó un bombeo de aire en este tanque esto para evitar a la creación de bacterias que puedan afectar al cultivo al momento del riego.



Figura 17: control de humedad del suelo

2.15 FUNCIONAMIENTO EN GENERAL.

En general el conjunto de controles nos brinda lo que es un ambiente artificial que establecerá las magnitudes físicas ambientales necesaria para el cultivo deseado, permitiendo un aseguramiento en el crecimiento y desarrollo del cultivo mejorando en tiempos y logrando una mejor calidad del cultivo ya que se permitirá desarrollar al 100% en sus condiciones ideales.



3. RESULTADOS

A lo largo del desarrollo de este proyecto se analizó la técnica y la forma correcta de trabajar con cierto tipo de plantío basado en la implementación de un sistema automatizado, donde se tomó como prueba inicial la cultivación de la hierbabuena para ello se tuvo que analizar las características de este cultivo, la revisión de esta planta nos permitió establecer bases sólidas, para poder realizar el experimento correcto y la implementación de nuestro proyecto para su cultivo.



Las características de esta planta nos indican que requiere de un riego abundante, por lo que se debe de estar atento que no le falta nada de humedad, esto nos favorece para comprobar que los sensores de humedad del suelo están funcionando correctamente al detectar con más frecuencia la falta de humedad para la planta. Sin embargo, se debe de evitar encharcar la planta, porque las raíces pudieran morirse, de aquí parte la aplicación del sistema de control de humedad para evitar el daño del cultivo.

La hierbabuena se caracteriza por ser una planta perenne. Para ello se requiere una temperatura entre los 15°C y 30°C, tomando en este caso una media de entre los 23°C debido a que en temporadas heladas este cultivo puede llegar a perecer. Para mantener el sistema a esta temperatura nosotros pusimos el set-point a 23°C por lo cual el control de temperatura se estabilizará a la deseada y necesario de la planta de esta manera aseguramos que la planta no se verá afectada por la temperatura externa del medio ambiente, aun a pesar de que la temperatura externa sea menor a la deseada.

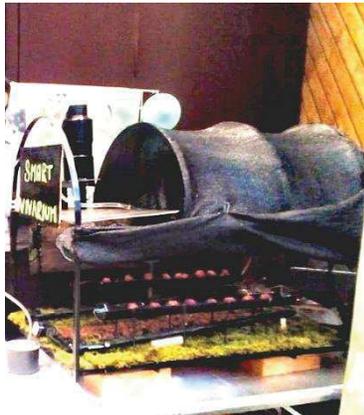
La hierbabuena se cultiva en clima templado. Específicamente en lugares con sombras, aunque también puede llegar a soportar los rayos de sol, por lo que nosotros utilizamos una malla de sombra especial para evitar el paso de rayos solares en cantidades grandes, ante ello se aplica el control de luminosidad ya que la malla sombra no permite pasar todo el brillo del sol, por ello la luminosidad lo establecimos en un 75% lo que genera que la fotosíntesis no sea interrumpida aun en la noche.



Como resultado, en la etapa final de la prueba se notó un mejor crecimiento del cultivo mejorando en los tiempos y más que nada protegiendo el cultivo de los cambios bruscos del ambiente, por lo tanto, los beneficios del sistema que se ha implementado se puede observar en lo siguiente:

- Tiempo
- Crecimiento
- Fructificación
- Cuidado
- Automatización

A lo largo de la prueba se observó el funcionamiento del control de temperatura y humedad ya que estos cumplían con su funcionamiento, logrando la protección del cultivo.



4. CONCLUSION

Se logró el objetivo deseado, de crear un ambiente artificial, de acuerdo a los parámetros necesarios del cultivo, como prueba se pudo observar el crecimiento de la hierbabuena, respetando cada una de sus características, para ello el sistema funciona de manera simple donde solamente se ingresa los datos necesarios de la planta, para que el ambiente artificial se adapte a este, mediante el control de automatización, donde por si solo estará desarrollando el cultivo adecuadamente.

A pesar de que fue una prueba pequeña se sabe que al poder controlar las magnitudes ambientales con las características necesarias para un cultivo cualquiera, este crecerá de la manera más adecuada mejorando la calidad del cultivo, además se estará protegiendo de los cambios bruscos del medio ambiente que puede llegar a estropearlo evitando la perdida innecesaria de cosecha.

Como tal el objetivo era crear un sistema donde se pudieran proteger los cultivos, pero además de eso brindarles la mejor calidad para su desarrollo. Lo cual se demostró con el experimento, por lo tanto, se pudo lograr una mejora en los cultivos y en las cosechas al implementar este sistema.

5. ANEXOS

```
void setup () {  
  Serial.begin(9600);  
  
}  
  
void loop()  
{  
  int valPot=analogRead(A0);  
  valPot=map(valPot,0,1023,0,255);
```



```
analogWrite(9,valPot);

int valorHumedad1 = map(analogRead (A1), 0, 1023, 100, 0);
Serial.print("humedad1: ");
Serial.print(valorHumedad1);
Serial.print("%");
Serial.println();

int valorHumedad2 = map(analogRead (A2), 0, 1023, 100, 0);
Serial.print("humedad2: ");
Serial.print(valorHumedad2);
Serial.print("%");
Serial.println();

int valorHumedad3 = map(analogRead (A3), 0, 1023, 100, 0);
Serial.print("humedad3: ");
Serial.print(valorHumedad3);
Serial.print("%");
Serial.println();

int valorHumedad4 = map(analogRead (A4), 0, 1023, 100, 0);
Serial.print("humedad4: ");
Serial.print(valorHumedad3);
Serial.print("%");
Serial.println();

float tempC = analogRead(A5);

// Calculamos la temperatura con la fórmula
tempC = (tempC /1023)*5000;
float t= tempC/10;

// Envía el dato al puerto serial
Serial.print(t);
// Salto de línea
Serial.println();

if(valorHumedad1 < 64 or valorHumedad2 < 64 or valorHumedad3 < 64 or valorHumedad4 <
64){
digitalWrite(10,1);
delay(5000);
}
}
```

6. REFERENCIAS BIBLIOGRAFICAS

- [1] Juan Reyes Quiñones Santo Domingo, República Dominicana. (2015). *Manual y organización de viveros. República Dominicana*: CEDAF.
- [2] Leonardo Manrique Ducuara, Otoniel López Valencia, Mario Oswaldo Triana. (1998). *instalación de un vivero. República de Colombia*: CEUDES.
- [3] INEGI. (----). *Datos agrícolas, ganaderos y pescas. ----*, de Instituto Nacional de Estadísticas Geográficas Sitio web: <https://www.inegi.org.mx/>



Sobre la Regularización Dual en el Balance de Postura de un Robot Humanoide Empleando Control de Cuerpo Completo

Rojas Rodríguez José Cruz, Aguilar Bustos Ana Yaveni y Bugarin Carlos Eusebio.

Tecnológico Nacional de México/I. T. Ensenada.
josec_rojas19@hotmail.com

Resumen

En el presente trabajo se introduce un algoritmo para el balance de postura de un robot humanoide con control de cuerpo completo regulando el momento centroidal (lineal y angular) y empleando regularización dual (este último método para la solución de un programa cuadrático con la técnica del punto interior para incrementar la eficiencia y robustez ante configuraciones singulares e inconsistencias en la ejecución de múltiples tareas). Para verificar la teoría propuesta se consideran 5 escenarios de simulación en donde se aplican perturbaciones en movimiento. Finalmente, de acuerdo con las gráficas de seguimiento del centro de masas, se concluye que el desempeño del algoritmo propuesto es satisfactorio.

Palabras clave: Robot humanoide, programación cuadrática, momento centroidal.

1. Introducción

El balance de postura es una de las tareas fundamentales para un robot con patas, y además precursor a su caminata. En la literatura existen diversas técnicas para el balance con control de cuerpo completo; por ejemplo, en [1] se introduce el balance de un robot cuadrúpedo empleando control PD para regular la posición y orientación de la base utilizando la dinámica centroidal y se formula un programa cuadrático para el computo de las fuerzas de contacto y estas son mapeadas a las patas del robot mediante Jacobiano transpuesto, mientras que en [2] se agrega un término adicional en la función objetivo penalizando la desviación con respecto a la solución anterior para evitar discontinuidades en los pares articulares durante la locomoción. En [3] se introduce control multi-objetivo mediante programación cuadrática empleando la dinámica de cuerpo completo y las fuerzas de contacto se restringen a permanecer dentro de un cono de fricción. Experimentos demuestran que los humanos minimizan el momento angular durante la caminata [4], esto ha motivado el desarrollo de controladores que regulan el momento centroidal. En [5] y [6] se introduce el balance de un robot humanoide mediante la regulación del momento lineal y angular empleando la matriz del momento centroidal [7] y el problema se resuelve mediante cuadrados mínimos, mientras que en [8] se resuelve un solo programa cuadrático.

Una de las dificultades en el control de cuerpo completo es el manejo de múltiples tareas que pueden entrar en conflicto, en [9] se introduce el control de tareas múltiples para sistemas de base flotante. En [10] el balance de cuerpo completo se formula mediante una jerarquía de tareas y se resuelve mediante programación cuadrática jerárquica (PCJ) en donde las tareas de menor prioridad están restringidas a no modificar las tareas de mayor prioridad, mientras que en [11] se emplea programación cónica jerárquica. En [12] se reduce el costo computacional de la (PCJ) proyectando las tareas de menor prioridad en el espacio nulo de las tareas de mayor prioridad mediante descomposición QR reduciendo así la dimensión de los programas cuadráticos en la jerarquía de tareas, mientras que en [13] se emplea descomposición del valor singular.



Para incrementar la robustez ante errores de modelado en [14] la dinámica inversa se complementa con la cinemática inversa y ambos problemas se resuelven mediante programación cuadrática. Sin embargo, en ninguno de los trabajos previamente mencionados que emplean programación cuadrática pueden manejar tareas singulares eficientemente ya que la técnica de solución empleada es el método del conjunto activo el cual presenta varias limitaciones como por ejemplo sensibilidad al condicionamiento del problema y un costo computacional que incrementa exponencialmente con el número de desigualdades. Por lo tanto, el objetivo del presente artículo es manejar configuraciones singulares de manera eficiente y robusta en la ejecución de múltiples tareas durante el balance de postura con control de cuerpo completo de un robot humanoide con 18 articulaciones (ver figura 1) mediante programación cuadrática empleando el método del punto interior con regularización dual.

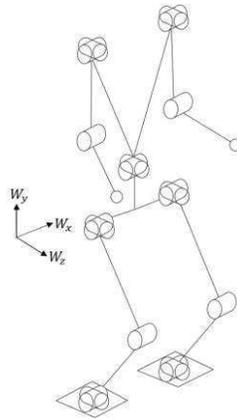


Figura 1. Estructura cinemática del robot bípedo bajo estudio.

2. Optimización convexa

2.1 Programación cuadrática

El problema del balance de postura con control de cuerpo completo se formula como un programa cuadrático, esto es:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{s. t.} \quad & Ax = b \\ & Gx \leq d \end{aligned} \quad (1)$$

donde $Q \in \mathbb{R}^{n \times n}$ es la matriz Hessiana, $c \in \mathbb{R}^n$ es el vector de constantes del término lineal, $A \in \mathbb{R}^{l \times n}$ es la matriz de las l restricciones de igualdad, $b \in \mathbb{R}^l$ es el vector de constantes de las l restricciones de igualdad, $G \in \mathbb{R}^{m \times n}$ es la matriz de las m restricciones de desigualdad y $d \in \mathbb{R}^m$ es el vector de constantes de las m restricciones de desigualdad. En general, se pueden considerar 3 métodos de solución para (1) estos son el método del conjunto activo, el método del punto interior y el método de multiplicadores. El método del conjunto activo es eficiente cuando se inicializa con una buena estimación del conjunto activo óptimo requiriendo en este caso pocas iteraciones para converger, sin embargo, este método es sensible al condicionamiento del problema y su costo computacional incrementa de manera considerable con el número de desigualdades y cuando el conjunto activo inicial difiere de manera significativa del conjunto activo óptimo [15]. El método de multiplicadores es robusto y se puede inicializar empleando una solución previa, sin embargo, es un algoritmo ineficiente dado que es un método de primer orden requiriendo un número considerable de iteraciones para converger, aunque es posible acelerar la convergencia utilizando sobre relajamiento y pre condicionamiento como la equilibración de matrices [16].



En términos de robustez y eficiencia un optimizador basado en el método del punto interior es superior, además el número de iteraciones para converger es aproximadamente constante, sin embargo, es difícil inicializar este método con una solución previa ya que tiende a volverse inestable, esta deficiencia del método del punto interior no presenta un problema en la práctica ya que es extremadamente eficiente incluso con una iniciación por ceros [17], por ende este es el método que se utiliza para el balance de cuerpo completo. En el método del punto interior las desigualdades se convierten a igualdades introduciendo una nueva variable $s \in \mathbb{R}^m$ restringida a ser positiva [18] y empleando regularización dual en (1) se tiene que:

$$\begin{aligned} \min_{x,s} \quad & \frac{1}{2} x^T Q x + c^T x + \frac{\delta}{2} (x^T x + y^T y + z^T z) \\ \text{s. t.} \quad & Ax - b = 0 \\ & Gx + s - d = 0 \\ & s \geq 0 \end{aligned} \quad (2)$$

donde $\delta \in \mathbb{R}$ es una constante positiva que regulariza las variables duales, $y \in \mathbb{R}^l$ son los multiplicadores de Lagrange de las restricciones de igualdad y $z \in \mathbb{R}^m$ son los multiplicadores de Lagrange de las restricciones de desigualdad. Empleando las condiciones de optimalidad de Karush Kuhn Tucker en (2) se obtiene:

$$(Q + \delta I)x + c + A^T y + G^T z = 0 \quad (3a)$$

$$SZe - \mu e = 0 \quad (3b)$$

$$Ax - b + \delta y = 0 \quad (3c)$$

$$Gx + s - d + \delta z = 0 \quad (3d)$$

$$(s, z) \geq 0 \quad (3e)$$

donde $S = \text{diag}(s) \in \mathbb{R}^{m \times m}$ es una matriz con s en la diagonal, $Z = \text{diag}(z) \in \mathbb{R}^{m \times m}$ es una matriz con z en la diagonal, $I \in \mathbb{R}^{n \times n}$ es la matriz identidad, $e \in \mathbb{R}^m$ es un vector de unos, $\mu = s^T z / m$ es la brecha complementaria y $\sigma \in \mathbb{R}$ es una constante positiva.

Reescribiendo (3) como una función no lineal se tiene:

$$F(x, s, y, z) = \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe - \sigma \mu e \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix} \quad (4)$$

Empleando el método de Newton en (4) se obtiene:

$$\begin{bmatrix} Q + \delta I & 0 & A^T & G^T \\ 0 & Z & 0 & S \\ A & 0 & \delta I & 0 \\ G & I & 0 & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe - \sigma \mu e \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix} \quad (5)$$

El emplear regularización dual permite introducir 2 matrices de la forma δI en la diagonal inferior de (5) permitiendo resolver el sistema mediante cualquier técnica de factorización ya que de otra forma el sistema (5) sería indefinido. El método del punto interior consiste en la solución de múltiples sistemas de la forma (5) disminuyendo en cada iteración el valor de la brecha complementaria μ mediante la constante σ hasta el valor de cero. Una vez resuelto (5) se realiza una búsqueda de línea de forma tal que las condiciones $(s, z) \geq 0$ no se violen. El nuevo punto de búsqueda se obtiene como:

$$(x_{k+1}, s_{k+1}) = (x_k, s_k) + \alpha_{\tau}^{pri} (\Delta x, \Delta s) \quad (6a)$$

$$(y_{k+1}, z_{k+1}) = (y_k, z_k) + \alpha_{\tau}^{dual} (\Delta y, \Delta z) \quad (6b)$$



donde la longitud de los pasos $\alpha_{\tau}^{pri} \in \mathbb{R}$ y $\alpha_{\tau}^{dual} \in \mathbb{R}$ se obtienen de la siguiente forma:

$$\alpha_{\tau}^{pri} = \max\{\alpha \in (0, 1]: s + \alpha\Delta s \geq (1 - \tau)s\} \quad (7a)$$

$$\alpha_{\tau}^{dual} = \max\{\alpha \in (0, 1]: z + \alpha\Delta z \geq (1 - \tau)z\} \quad (7b)$$

donde $\tau \in (0, 1)$ es una constante positiva que define la cercanía de las variables s y z de sus valores mínimos de 0, un valor típico de τ es 0.995. En el presente trabajo se emplea la estrategia de [19] $\tau_k = \max(0.995, 1 - \mu_k)$ para controlar el valor de τ . La estrategia de resolver (5) en secuencia no necesariamente resulta en un algoritmo eficiente, las implementaciones modernas del método del punto interior suelen emplear una estrategia alternativa basada en el algoritmo de Mehrotra el cual consiste en calcular un paso predictor de (5) con $\sigma = 0$, esto es:

$$\begin{bmatrix} Q + \delta I & 0 & A^T & G^T \\ 0 & Z & 0 & S \\ A & 0 & \delta I & 0 \\ G & I & 0 & \delta I \end{bmatrix} \begin{bmatrix} \Delta x^{aff} \\ \Delta s^{aff} \\ \Delta y^{aff} \\ \Delta z^{aff} \end{bmatrix} = - \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix} \quad (8)$$

Posteriormente se calcula un paso corrector mediante el siguiente sistema:

$$\begin{bmatrix} Q + \delta I & 0 & A^T & G^T \\ 0 & Z & 0 & S \\ A & 0 & \delta I & 0 \\ G & I & 0 & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe - \sigma\mu_k e + \Delta S^{aff} \Delta Z^{aff} e \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix} \quad (9)$$

donde $\Delta S^{aff} = \text{diag}(\Delta s^{aff}) \in \mathbb{R}^{m \times m}$ es una matriz con Δs^{aff} en la diagonal y $\Delta Z^{aff} = \text{diag}(\Delta z^{aff}) \in \mathbb{R}^{m \times m}$ es una matriz con Δz^{aff} en la diagonal. En la figura 2 se muestra un pseudo código del algoritmo de Mehrotra para la solución de programas cuadráticos.

```

k = 0
while E(xk, sk, yk, zk) ≤ εTOL
    (x, s, y, z) = (xk, sk, yk, zk)
     $\begin{bmatrix} Q + \delta I & 0 & A^T & G^T \\ 0 & Z & 0 & S \\ A & 0 & \delta I & 0 \\ G & I & 0 & \delta I \end{bmatrix} \begin{bmatrix} \Delta x^{aff} \\ \Delta s^{aff} \\ \Delta y^{aff} \\ \Delta z^{aff} \end{bmatrix} = - \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix}$ 

    μk = sTz/m
    αaff = max{α ∈ (0, 1]: (s, z) + α(Δsaff, Δzaff) ≥ 0}
    μaff = (s + αaffΔsaff)T(z + αaffΔzaff)/m
    σ = (μaff/μk)3

     $\begin{bmatrix} Q + \delta I & 0 & A^T & G^T \\ 0 & Z & 0 & S \\ A & 0 & \delta I & 0 \\ G & I & 0 & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} (Q + \delta I)x + c + A^T y + G^T z \\ SZe - \sigma\mu_k e + \Delta S^{aff} \Delta Z^{aff} e \\ Ax - b + \delta y \\ Gx + s - d + \delta z \end{bmatrix}$ 

    τk = max(0.995, 1 - μk)
    ατkpri = max{α ∈ (0, 1]: s + αΔs ≥ (1 - τk)s}
    ατkdual = max{α ∈ (0, 1]: z + αΔz ≥ (1 - τk)z}
    α̂ = min(ατkpri, ατkdual)
    (xk+1, sk+1, yk+1, zk+1) = (xk, sk, yk, zk) + α̂(Δx, Δs, Δy, Δz)
    k = k + 1
end
    
```



Figura 2. Algoritmo de Mehrotra para programación cuadrática.

El algoritmo de Mehrotra selecciona la constante σ de manera dinámica empleando la heurística propuesta por [20] con $\sigma = (\mu_{aff}/\mu_k)^3$ en donde $\mu_{aff} \in \mathbb{R}$ se obtiene como:

$$\mu_{aff} = (s + \hat{\alpha}_{aff}\Delta s^{aff})^T (z + \hat{\alpha}_{aff}\Delta z^{aff})/m \quad (10)$$

donde $\hat{\alpha}_{aff} \in \mathbb{R}$ es una constante que se obtiene realizando una búsqueda de línea, esto es:

$$\hat{\alpha}_{aff} = \max\{\alpha \in (0, 1]: (s, z) + \alpha(\Delta s^{aff}, \Delta z^{aff}) \geq 0\} \quad (11)$$

Por último, en el algoritmo de la figura 2 la función de error se define como:

$$E(x, s, y, z) = \max\{\|Q + \delta I + c + A^T y + G^T z\|, \|S z\|, \|A x - b + \delta y\|, \|G x + s - d + \delta z\|\} \quad (12)$$

Tradicionalmente los sistemas de ecuaciones (8) y (9) se resuelven mediante factorización simétrica indefinida con librerías como MA57 [21], algunos optimizadores que emplean factorización simétrica indefinida son OOQP [22], LOQO [23], KNITRO [24] y IPOPT [25], sin embargo, en nuestra formulación al emplear regulación dual es posible resolver los sistemas (8) y (9) mediante descomposición LU disminuyendo así el costo computacional del algoritmo.

3. Balance de postura con control de cuerpo completo

3.1 Tareas múltiples

Cuando un robot es redundante la ejecución de tareas secundarias son posibles realizarlas en el espacio nulo de las tareas de mayor prioridad. En la formulación del problema de balance se consideran tareas de dinámica y cinemática. El balance se efectúa empleando las ecuaciones del cuerpo completo, esto es:

$$M(q)\ddot{q} + h_b(q, \dot{q}) = S^T \tau + J^T(q)\lambda \quad (13)$$

donde $M(q) \in \mathbb{R}^{(6+n_j) \times (6+n_j)}$ es la matriz de inercia de base flotante, $h_b(q, \dot{q}) \in \mathbb{R}^{6+n_j}$ son las fuerzas de Coriolis, centrífugas y gravedad, $S = [I_{n_j \times n_j} \quad 0_{n_j \times 6}]$ es la matriz de selección de pares actuados, $\tau \in \mathbb{R}^{n_j}$ es el vector de pares articulares, $J(q) \in \mathbb{R}^{3n_c \times (6+n_j)}$ es el Jacobiano aumentado de los n_c puntos de contacto, $\lambda \in \mathbb{R}^{3n_c}$ es el vector de las n_c fuerzas de contacto y n_j es el número de grados de libertad del sistema sin tener en cuenta la posición y orientación de la base. Las fuerzas de contacto se modelan con la fricción estática de Coulomb, por lo tanto la fuerza de contacto λ_j está restringida por:

$$\|\lambda_j - (n_j^T \lambda_j) n_j\| \leq \mu n_j^T \lambda_j \quad (14)$$

donde $n_j \in \mathbb{R}^3$ es el vector normal a la superficie de contacto y $\mu \in \mathbb{R}$ es el coeficiente de fricción. Sin embargo (14) es una restricción cónica lo que convierte el problema de balance en una optimización cónica de segundo orden, por lo tanto, el cono de fricción de Coulomb se aproxima mediante una pirámide compuesta de 4 vectores base $w_{ij} \in \mathbb{R}^3$ que se definen como:

$$w_{ij} = n_j + \mu d_{ij} \quad (15)$$

donde $d_{ij} \in \mathbb{R}^3$ es un vector tangente a la superficie de contacto. Por lo tanto, la pirámide de fricción es:

$$\forall j \quad \lambda_j = W_j \beta_j \quad (16a)$$



$$\forall j \quad \beta_j \geq 0 \quad (16b)$$

donde $W_j = [w_{1j} \ w_{2j} \ w_{3j} \ w_{4j}] \in \mathbb{R}^{3 \times 4}$ define una base vectorial de la pirámide de fricción y $\beta_j \in [\beta_{1j} \ \beta_{2j} \ \beta_{3j} \ \beta_{4j}]^T \in \mathbb{R}^4$ es un vector de constantes positivas. El uso de (16) en lugar de (14) permite abordar el problema de balance mediante programación cuadrática. El restringir a las fuerzas de contacto dentro del cono de fricción por (16) y obedecer las ecuaciones de movimiento (13) generaliza el criterio de balance del ZMP a superficies irregulares [3] y esta es la tarea de mayor prioridad en la formulación del problema de balance. La ecuación de movimiento es lineal en las variables \ddot{q} , τ y λ , por lo tanto (13) y (16a) se pueden expresar en una matriz aumentada, esto es:

$$\begin{bmatrix} M & -S^T & -J^T & 0 \\ 0 & 0 & I & -W \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} = - \begin{bmatrix} h_b \\ 0 \end{bmatrix} \quad (17)$$

donde $W = \begin{bmatrix} W_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_8 \end{bmatrix} \in \mathbb{R}^{3n_c \times 4n_c}$ y $I \in \mathbb{R}^{3n_c \times 3n_c}$ es la matriz identidad. Ahora si se define el vector de optimización $x = [\ddot{q}^T \ \tau^T \ \lambda^T \ \beta^T]^T$ entonces (17) se puede reescribir como:

$$A_1 x = b_1 \quad (18)$$

donde $A_1 = \begin{bmatrix} M & -S^T & -J^T & 0 \\ 0 & 0 & E & -W \end{bmatrix}$ y $b_1 = - \begin{bmatrix} h_b \\ 0 \end{bmatrix}$.

Durante el balance se especifica una tarea que restringe el movimiento de los puntos contacto con respecto a la superficie de contacto, estas restricciones se representan mediante la siguiente función:

$$c_j(q) = p_j^d \quad (19)$$

donde $c_j(q)$ es la función cinemática directa del punto de contacto j y $p_j^d \in \mathbb{R}^3$ es la posición deseada del punto de contacto j . Se deriva (19) dos veces con respecto al tiempo para expresar la restricción cinemática al nivel de la aceleración:

$$J_j(q)\ddot{q} + \dot{J}_j(q)\dot{q} = 0 \quad (20)$$

donde $J_j(q)$ es el Jacobiano de la restricción cinemática j . La restricción cinemática (20) se reescribe en términos del vector de optimización x , por lo tanto se tiene:

$$[J_j \ 0 \ 0 \ 0] \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} = -J_j \dot{q} \quad (21)$$

Ahora cuando se tienen n puntos de contacto (21) se generaliza a:

$$\begin{bmatrix} J_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ J_n & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} = - \begin{bmatrix} J_0 \\ \vdots \\ J_n \end{bmatrix} \dot{q} \quad (22a)$$

$$A_2 x = b_2 \quad (22b)$$



$$\text{donde } A_2 = \begin{bmatrix} J_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ J_n & 0 & 0 & 0 \end{bmatrix} \text{ y } b_2 = - \begin{bmatrix} j_0 \\ \vdots \\ j_n \end{bmatrix} \dot{q}.$$

Para incrementar la robustez del balance se considera la regulación del momento lineal y angular. El momento centroidal está relacionado con las velocidades articulares por [7]:

$$h_G = A_G(q)\dot{q} \quad (23)$$

donde $h_G \in \mathbb{R}^6$ es el momento centroidal y $A_G(q) \in \mathbb{R}^{6 \times (6+n_j)}$ es la matriz del momento centroidal. Tomando la derivada temporal de (12) se tiene:

$$\dot{h} = A_G(q)\ddot{q} + \dot{A}_G(q)\dot{q} \quad (24)$$

Reescribiendo (24) en términos de x se obtiene:

$$[A_G \ 0 \ 0 \ 0] \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} = \dot{h}_{des} - \dot{A}_G(q)\dot{q} \quad (25a)$$

$$A_3 x = b_3 \quad (25b)$$

donde $\dot{h}_{des} \in \mathbb{R}^6$ es una referencia deseada, $A_3 = [A_G \ 0 \ 0 \ 0]$ y $b_3 = \dot{h}_{des} - \dot{A}_G(q)\dot{q}$. A continuación, se emplea la siguiente referencia deseada:

$$\dot{h}_{des} = \begin{bmatrix} -k_G h \\ m k_p (r_d - r) - k_v l \end{bmatrix} \quad (26)$$

donde $l \in \mathbb{R}^3$ es el momento lineal, $r_d \in \mathbb{R}^3$ es la posición deseada del centro de masas, $k_G \in \mathbb{R}^{3 \times 3}$ es una matriz de ganancias para la regulación del momento angular, $k_p \in \mathbb{R}^{3 \times 3}$ y $k_v \in \mathbb{R}^{3 \times 3}$ son matrices definidas positivas de ganancias proporcional y derivativa, respectivamente.

Por último, se especifica una postura de referencia deseada, esto es:

$$\ddot{q}_{des} = k_p(q_d - q) - k_v \dot{q} \quad (27)$$

donde $q_d \in \mathbb{R}^{6+n_j}$ es la referencia deseada. De (27) se define la siguiente tarea:

$$\ddot{q} - \ddot{q}_{des} = 0 \quad (28a)$$

$$[I \ 0 \ 0 \ 0] \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} = \ddot{q}_{des} \quad (28b)$$

$$A_4 x = b_4 \quad (28c)$$

donde $A_4 = [I \ 0 \ 0 \ 0]$ y $b_4 = \ddot{q}_{des}$.

3.2 Control de cuerpo completo

El problema de balance se formula como la solución a una jerarquía de 4 tareas, sin embargo, esto requiere la solución de múltiples programas cuadráticos, por lo tanto, en el presente trabajo se le asigna un peso $w_j \in \mathbb{R}$ a cada tarea de forma tal que sólo es necesario la solución de un único programa cuadrático reduciendo así el costo computacional del control de cuerpo completo. La jerárquica de tareas se obtiene de (18), (22b), (25b) y (28c), ahora se crea un sistema aumentado concatenando las 4 tareas, esto es:



$$\begin{bmatrix} w_1 A_1 \\ w_2 A_2 \\ w_3 A_3 \\ w_4 A_4 \end{bmatrix} x = \begin{bmatrix} w_1 b_1 \\ w_2 b_2 \\ w_3 b_3 \\ w_4 b_4 \end{bmatrix} \quad (29a)$$

$$Ax = b \quad (29b)$$

donde $A = [w_1 A_1^T \ w_2 A_2^T \ w_3 A_3^T \ w_4 A_4^T]^T$ y $b = [w_1 b_1^T \ w_2 b_2^T \ w_3 b_3^T \ w_4 b_4^T]^T$. Por último, las desigualdades (16b) se expresan en términos del vector de optimización x y se multiplican por la ganancia w_1 , esto es:

$$w_1 \begin{bmatrix} 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \tau \\ \lambda \\ \beta \end{bmatrix} \leq 0 \quad (30a)$$

$$w_1 Gx \leq w_1 d \quad (30b)$$

donde $I \in \mathbb{R}^{4n_c \times 4n_c}$ es la matriz identidad, $G = [0 \ 0 \ 0 \ -I]$ y $d = 0$.

Por lo tanto, de (29b) y (30b) el problema de optimización a resolver en cada ciclo de control para el balance de cuerpo completo es:

$$\begin{aligned} \min_{x, u_1, u_2, v_1, v_2} \quad & \frac{1}{2} x^T Q x + c^T x + \gamma e^T (u_1 + u_2 + v_1 + v_2) \\ \text{s. t.} \quad & Ax - b + u_1 = v_1 \\ & Gx - d + u_2 \leq v_2 \\ & u_1, u_2, v_1, v_2 \geq 0 \end{aligned} \quad (31)$$

donde $u_1 \in \mathbb{R}^l, u_2 \in \mathbb{R}^m, v_1 \in \mathbb{R}^l, v_2 \in \mathbb{R}^m$ son variables adicionales, $\gamma \in \mathbb{R}$ es una constante positiva que penaliza la violación de las restricciones. El optimizador del punto interior de la figura 2 agrega regulación a las variables duales y de optimización del programa cuadrático (31) y esta es la razón por la cual estos términos adicionales no aparecen en la función objetivo de (31). La función objetivo cuadrática que se emplea en (31) consiste en minimizar los pares articulares, esto es:

$$\frac{1}{2} x^T Q x + c^T x = \frac{1}{2} \tau^T \tau \quad (32)$$

donde $c = 0$ es un vector de ceros y $Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

Para incrementar la eficiencia durante la solución de (31) todas las tareas A_j y la matriz Hessiana Q se representan mediante matrices dispersas. La formulación del programa cuadrático (31) está inspirado en el manejo de problemas cuadráticos inconsistentes en la programación cuadrática secuencial y se denomina programa elástico o modo elástico [19]. La inclusión de variables adicionales en (31) permite lidiar con posibles tareas singulares y el uso de regularización dual incrementa la robustez y eficiencia del optimizador.

4. Resultados

Se consideran 5 escenarios de simulación en donde las ecuaciones de movimiento se integran empleando el método de Euler simpléctico durante 8000 pasos de integración ($\Delta t = 1 \times 10^{-3}$). Para realizar las simulaciones se desarrolla un simulador en Matlab empleando el algoritmo del cuerpo articulado para sistemas de base flotante [26] en el cálculo de las aceleraciones articulares dados los



pares articulares y fuerzas externas, también se implementa el algoritmo descrito en [27] para el cómputo de las fuerzas de contacto.

De (13) la matriz de inercia $M(q)$ se calcula mediante el algoritmo del cuerpo compuesto, el vector $h_b(q, \dot{q})$ se obtiene mediante el algoritmo recursivo de Newton Euler. De (24) la matriz del momento angular centroidal $A_G(q)$ se obtiene mediante el algoritmo descrito en [7] y para el cálculo del término $\dot{A}_G(q)\dot{q}$ se desarrolla un algoritmo $O(N)$ empleando álgebra de Lie.

El número de puntos de contacto que se consideran en los 5 escenarios es $n_c = 8$ (4 para cada pie) y el número de grados de libertad actuados en el robot es $n_j = 18$. Las propiedades del modelo del robot bajo estudio se inspiran en [28] para las piernas mientras que para la parte superior del cuerpo se adaptan de forma tal que la masa total del sistema sea de 80kg, los datos empleados se muestran en la tabla 1.

Tabla 1. Propiedades del modelo del robot bípedo.

Extremidad	Masa	Longitud
Torso	30.125kg	0.6m
Pelvis	13.375kg	0.2m
Muslo	8.5kg	0.5m
Pierna	3.5kg	0.5m
Pie	1.25kg	0.1m
Brazo derecho	2.5kg	0.4m
Antebrazo derecho	2.5kg	0.4m
Brazo izquierdo	2.5kg	0.4m
Antebrazo izquierdo	2.5kg	0.4m

Primero se considera un escenario 0 para visualizar el comportamiento del sistema sin ningún tipo de perturbación. El sistema comienza (al igual que en todos los escenarios) con una postura inicial con los brazos flexionados similar a la postura que se muestra en la figura 1 y su postura deseada en este escenario consiste en terminar con los brazos estirados levantados, los resultados del escenario 0 se presentan en la figura 3. Observe que la variación en “x” es insignificante y que levanta los brazos al solicitarle una posición 4 cm por arriba del centro de masas inicial. Hay errores aceptables en “y” y en “z” pues el robot se mantiene de pie.

El escenario 1 consiste en mantener el balance aplicando una perturbación sobre el torso de $f^E = [200 \ 0 \ 100]^T N$ a partir del primer segundo durante 80 pasos de integración, los resultados del escenario 1 se muestran en la figura 4. Note que el robot reacciona manteniéndose siempre de pie.

El escenario 2 consiste en el balance del robot puesto sobre una plataforma que alterna su movimiento con una velocidad de 0.4 m/s en el eje x , los resultados del escenario 2 se muestran en la figura 5. Nuevamente, a pesar de que el piso se mueve, el robot mantiene su balance erguido.

El escenario 3 al igual que el 2 consiste en mantener el balance en una plataforma que alterna su movimiento con una velocidad de 0.4 m/s , sin embargo, en este caso es sobre el eje z , también se aplica una perturbación sobre el torso de $f^E = [200 \ 0 \ 0]^T N$ a partir del primer segundo durante 80 pasos de integración, los resultados del escenario 3 se muestran en la figura 6.

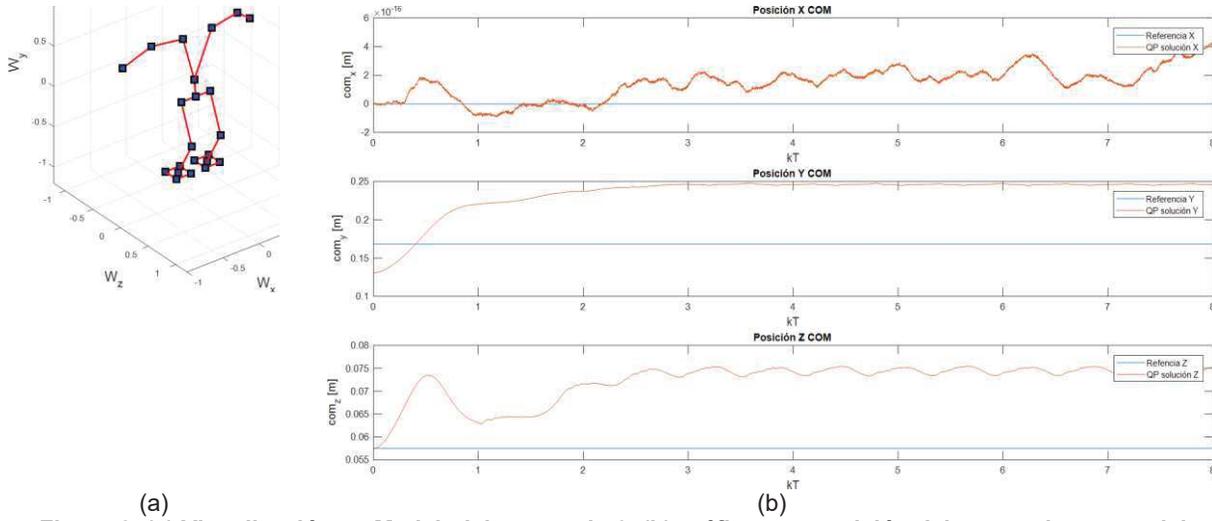


Figura 3. (a) Visualización en Matlab del escenario 0, (b) gráficas en posición del centro de masas del escenario 0.

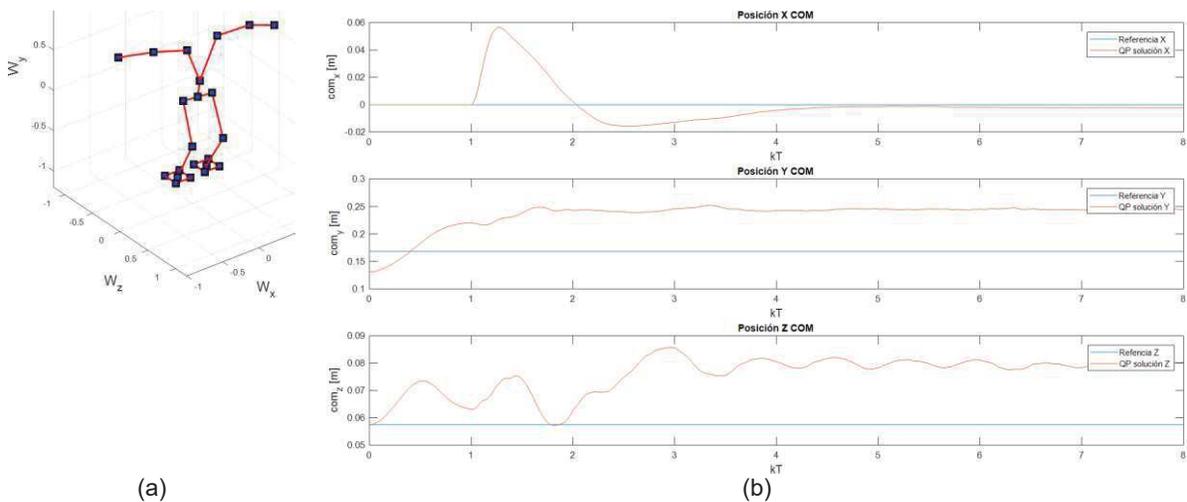


Figura 4. (a) Visualización en Matlab del escenario 1, (b) gráficas en posición del centro de masas del escenario 1.

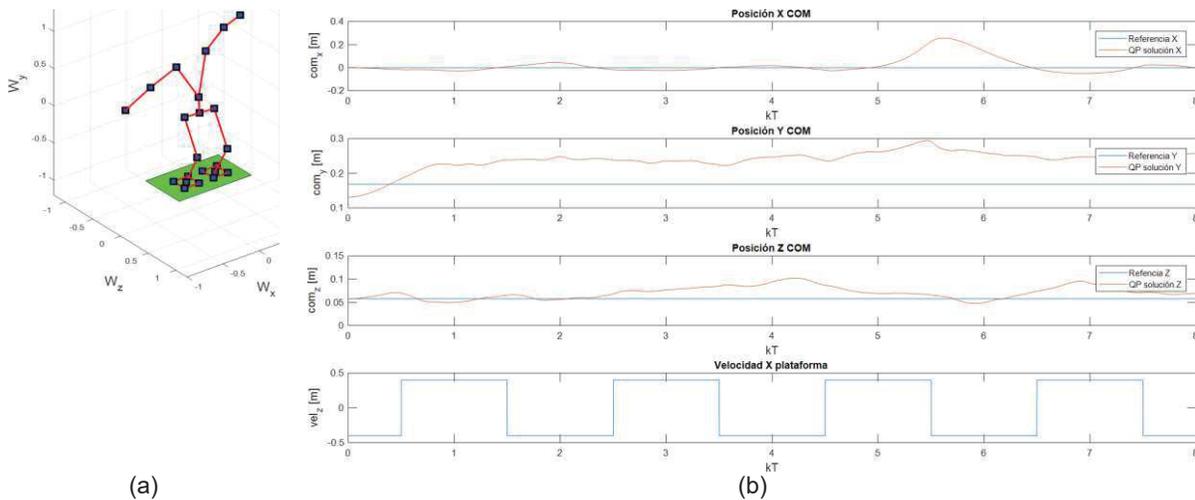


Figura 5. (a) Visualización en Matlab del escenario 2, (b) gráficas en posición del centro de masas del escenario 2.

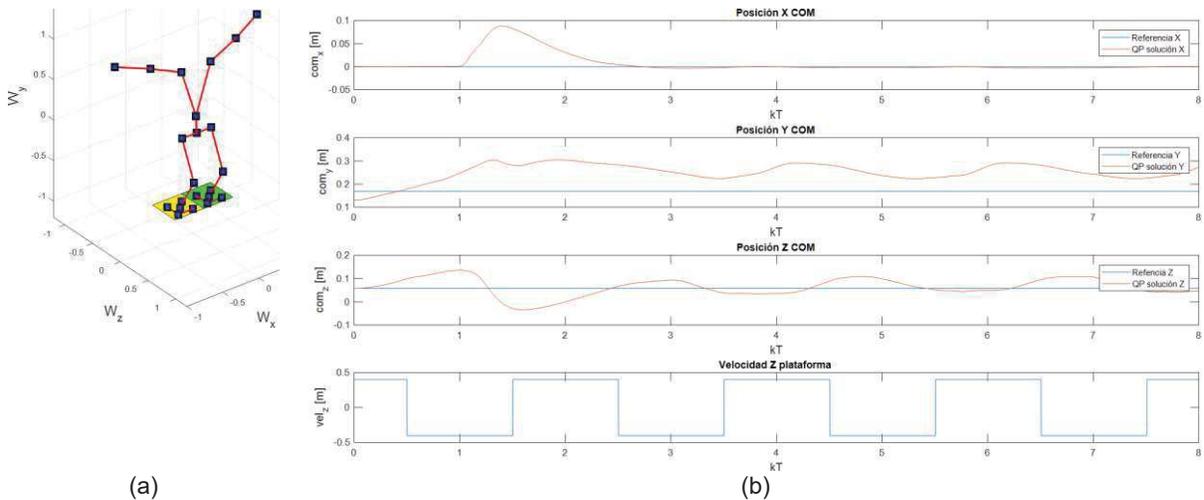


Figura 6. (a) Visualización en Matlab del escenario 3, (b) gráficas en posición del centro de masas del escenario 3.

Por último, en el cuarto escenario se considera el balance con el robot puesto sobre dos plataformas dispuestas con inclinación de 20° y -20° grados que se desplazan en direcciones opuestas sobre el eje z de manera alternante con una velocidad de 0.4 m/s , también se aplica una perturbación sobre el torso de $f^E = [400 \ 0 \ 0]^T \text{ N}$ a partir del primer segundo durante 80 pasos de integración, los resultados del escenario 4 se ilustran en la figura 7.

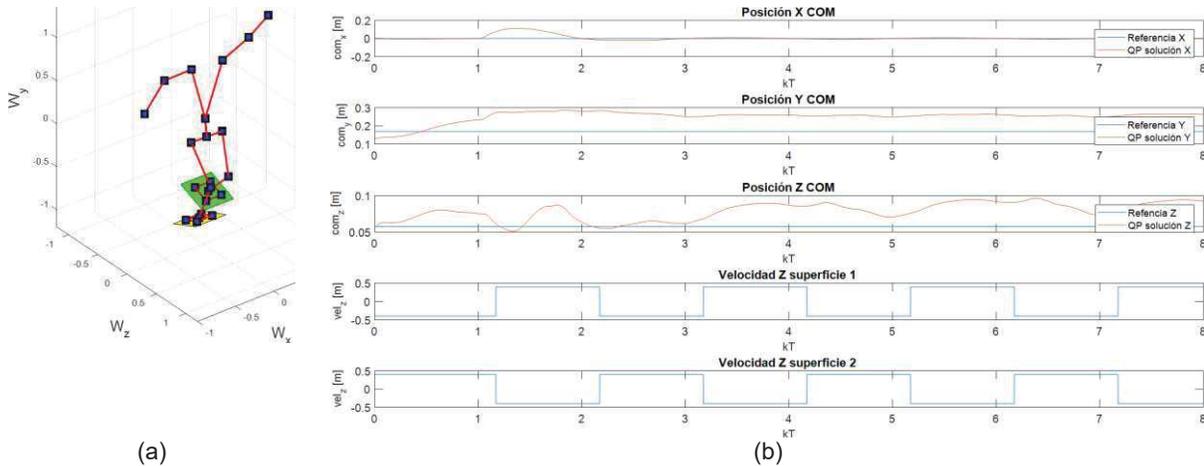


Figura 7. (a) Visualización en Matlab del escenario 4, (b) gráficas en posición del centro de masas del escenario 4.

En los 5 escenarios se observa que la posición del centro de masas se mantiene en valores bastante aceptables mostrando así que el sistema es capaz de mantener el balance ante la fuerza externa en el torso y el movimiento de la plataforma.

A continuación en la figuras 8, 9, 10, 11 y 12 se presentan los errores con respecto a la referencia deseada del centro de masas, momento lineal y momento angular de los 5 escenarios.

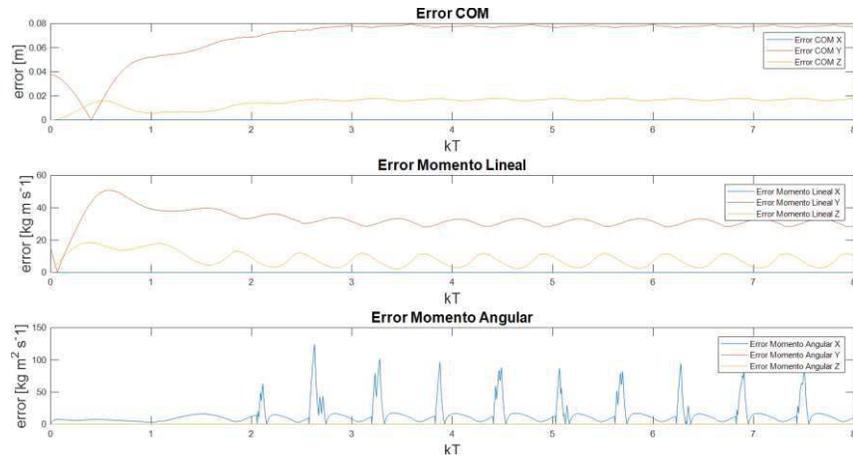


Figura 8. Errores del COM, momento lineal y angular en el escenario 0.

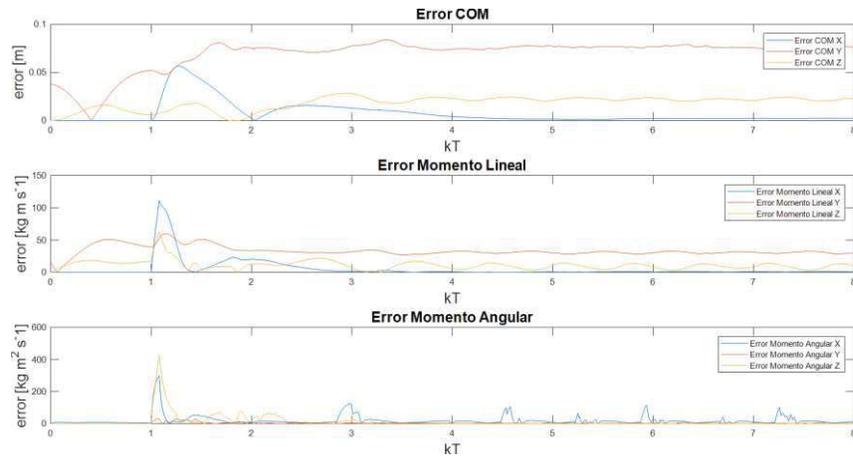


Figura 9. Errores del COM, momento lineal y angular en el escenario 1.

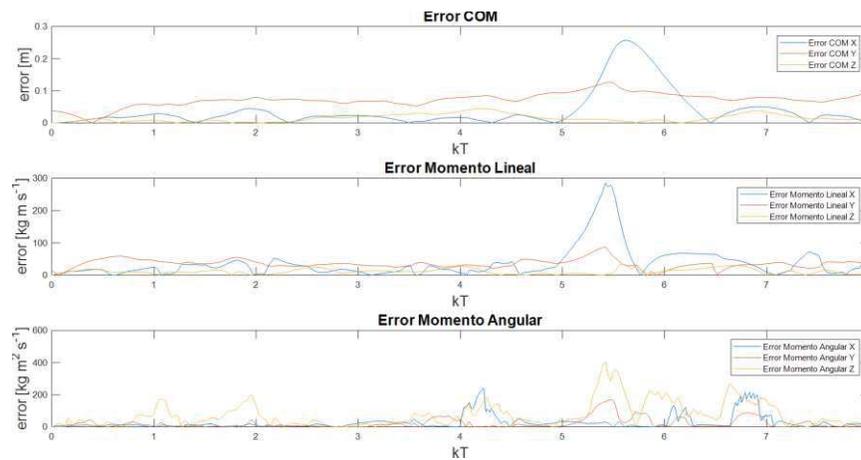


Figura 10. Errores del COM, momento lineal y angular en el escenario 2.

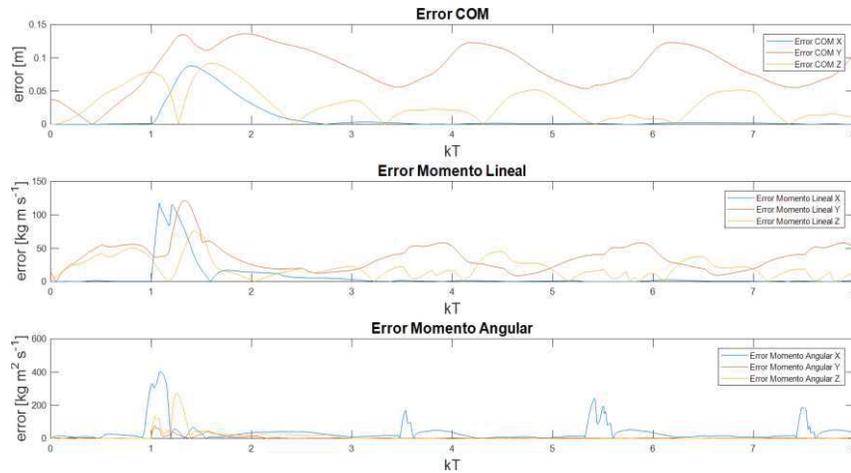


Figura 11. Errores del COM, momento lineal y angular en el escenario 3.

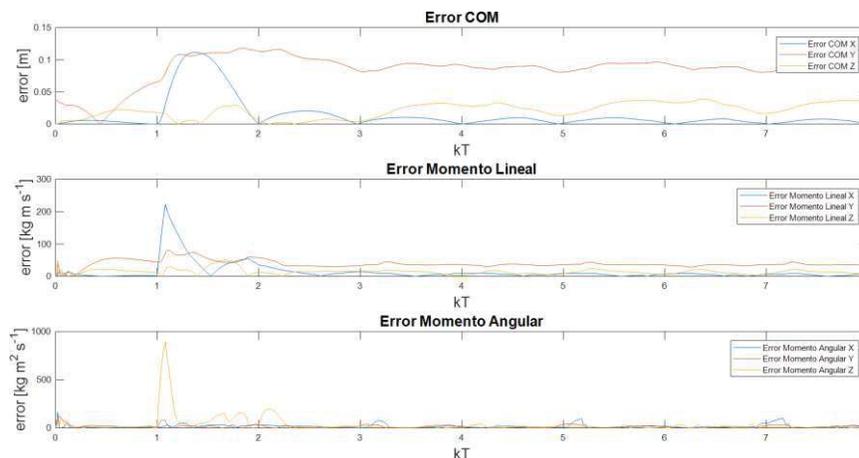


Figura 12. Errores del COM, momento lineal y angular en el escenario 4.

En los 5 escenarios se observa que la magnitud del error del centro de masas, momento lineal y angular se mantiene con algunas fluctuaciones aceptables. El instante en donde se aprecia el mayor error en los escenarios es donde se aplica la perturbación en el torso (a partir del primer segundo). El uso de las ganancias para especificar prioridades en las tareas incrementa la robustez del control del cuerpo completo ya que sin éstas el sistema tiende a converger a configuraciones que se desvían de manera considerable de la postura deseada, además no fue difícil especificar los valores de las ganancias en los 5 escenarios considerados en este trabajo, los valores que se emplearon fueron potencias de 10. Sin embargo, diferentes ganancias en las tareas producen diferentes comportamientos del sistema y el grado de cumplimiento de las tareas dependen de éstas. También existe el riesgo de inestabilidad numérica si la diferencia entre las ganancias es muy grande, en los experimentos numéricos se observa inestabilidad con una diferencia mayor a 1000.

5. Conclusiones

En el presente artículo se ha introducido un algoritmo para el balance de postura de un robot humanoide con control de cuerpo completo capaz de lidiar con múltiples tareas y configuraciones singulares de manera eficiente y robusta. El número de iteraciones que le toma al algoritmo en



converger durante cada ciclo de control en los 5 escenarios considerados es de aproximadamente 10. Los escenarios simulados en este artículo demuestran el potencial de este algoritmo para el balance en superficies irregulares, esto permite visualizar su aplicación en la caminata de robots con patas en terrenos irregulares. Un trabajo futuro a considerar consiste en la aplicación del optimizador del punto interior en el balance mediante programación cuadrática jerárquica y planeación de cuerpo completo con programación cuadrática secuencial para la síntesis de movimientos dinámicos como caminar y correr.

Referencias

- [1] Gehring, C., Bellicoso, C. D., Coros, S., Bloesch, M., Fankhauser, P., Hutter, M., & Siegwart, R. "Dynamic trotting on slopes for quadrupedal robots". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, (pp. 5129-5135), Hamburg, Germany, 2015.
- [2] Bledt, G., Powell, M. J., Katz, B., Di Carlo, J., Wensing, P. M., & Kim, S. "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, (pp. 2245-2252), Madrid, Spain, 2018.
- [3] Abe, Y., Da Silva, M., & Popović, J. "Multiobjective control with frictional contacts". In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, (pp. 249-258), San Diego, California, USA, 2007.
- [4] Popovic, M., Hofmann, A., & Herr, H. "Angular momentum regulation during human walking: biomechanics and control". IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, IEEE, (Vol. 3, pp. 2405-2411), New Orleans, LA, USA, 2004.
- [5] Lee, S. H., & Goswami, A. "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground". *Autonomous Robots*, (Vol. 33(4), pp. 399-414), 2012.
- [6] Lee, S. H., & Goswami, A. "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground". 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, (pp. 3157-3162), Taipei, Taiwan, 2010.
- [7] Orin, D. E., Goswami, A., & Lee, S. H. "Centroidal dynamics of a humanoid robot". *Autonomous Robots*, (Vol. 35(2-3), pp. 161-176), 2013.
- [8] Koolen, T., Bertrand, S., Thomas, G., De Boer, T., Wu, T., Smith, J., & Pratt, J. "Design of a momentum-based control framework and application to the humanoid robot atlas". *International Journal of Humanoid Robotics*, (Vol. 13(01), 1650007), 2016.
- [9] Sentis, L., & Khatib, O. "Control of free-floating humanoid robots through task prioritization". Proceedings of the 2005 IEEE International Conference on Robotics and Automation, IEEE, (pp. 1718-1723), Barcelona, Spain, 2005.
- [10] Herzog, A., Righetti, L., Grimminger, F., Pastor, P., & Schaal, S. "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics". 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, (pp. 981-988), Chicago, IL, USA, 2014.
- [11] Wensing, P. M. "Optimization and control of dynamic humanoid running and jumping" (Doctoral dissertation, The Ohio State University), 2014.
- [12] Escande, A., Mansard, N., & Wieber, P. B. "Hierarchical quadratic programming: Fast online humanoid-robot motion generation". *The International Journal of Robotics Research*, (Vol. 33(7), 1006-1028), 2014.
- [13] De Lasa, M., Mordatch, I., & Hertzmann, A. "Feature-based locomotion controllers". *ACM Transactions on Graphics (TOG)*, ACM, (Vol. 29(4), p. 131), 2010.
- [14] Feng, S., Whitman, E., Xinjilefu, X., & Atkeson, C. G. "Optimization based full body control for the atlas robot". 2014 IEEE-RAS International Conference on Humanoid Robots, IEEE, (pp. 120-127), Madrid, Spain, 2014.
- [15] Maes, C. M. "A regularized active-set method for sparse convex quadratic programming" (Doctoral dissertation, Stanford University), 2011.
- [16] Stellato, B. "Mixed-integer optimal control of fast dynamical systems" (Doctoral dissertation, University of Oxford), 2017.
- [17] Wang, Y., & Boyd, S. "Fast model predictive control using online optimization". *IEEE Transactions on control systems technology*, (Vol. 18(2), pp. 267-278), 2009.



- [18] Nocedal, J., & Wright, S. “*Numerical optimization*”. Springer Science & Business Media, 2006.
- [19] Andrei, N. “*Continuous nonlinear optimization for engineering applications in GAMS technology*”. Springer International Publishing, 2017.
- [20] Mehrotra, S. “*On the implementation of a primal-dual interior point method*”. *SIAM Journal on optimization*, (Vol. 2(4), pp. 575-601), 1992.
- [21] Duff, I. S. “*MA57---a code for the solution of sparse symmetric definite and indefinite systems*”. *ACM Transactions on Mathematical Software (TOMS)*, (Vol. 30(2), pp. 118-144), 2004.
- [22] Gertz, E. M., & Wright, S. “*OOQP user guide (No. ANL/MCS-TM-252)*”. Argonne National Lab., IL (US), 2002.
- [23] Vanderbei, R. J. “*LOQO: An interior point code for quadratic programming*”. *Optimization methods and software*, (Vol. 11(1-4), pp. 451-484), 1999.
- [24] Waltz, R. A., Morales, J. L., Nocedal, J., & Orban, D. “*An interior algorithm for nonlinear optimization that combines line search and trust region steps*”. *Mathematical programming*, (Vol. 107(3), pp. 391-408), 2006.
- [25] Wächter, A. “*An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*” (Doctoral dissertation, Carnegie Mellon University), 2002.
- [26] Featherstone, R. “*Rigid body dynamics algorithms*”. Springer, 2014.
- [27] Kokkevis, E., & Metaxas, D. “*Efficient dynamic constraints for animating articulated figures*”. *Multibody system dynamics*, (Vol. 2(2), pp. 89-114), 1998.
- [28] Geyer, H., & Herr, H. (2010). A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, (Vol. 18(3), pp. 263-273), 2010.



Impresora 3D con Extrusora de Plástico Triturado

Reyes Baltazar Edgar Alfredo, Aceves Arias Gerardo, Aguilar Molina Yehoshua y Domínguez García Rodolfo Omar

Universidad de Guadalajara
Centro Universitario de los Valles
alfre.98reyes@gmail.com

Resumen

El presente documento describe el proyecto de una impresora que realiza piezas en tres dimensiones (3D), con la inclusión de un extrusor que se alimenta de plástico reciclado, triturado previamente, extruyendo directamente el material fundido en la plataforma de la impresora para dar forma a la pieza a elaborar, la impresora es del tipo cartesiana, contando con 3 grados de movimiento (X, Y, Z) con dimensiones de 1 m. de ancho por 1 m. de largo, y 40 cm de alto, para el control de la máquina se utiliza el software Marlin Firmware de código abierto de arduino, que nos permite realizar el manejo de la máquina y el diseño del producto que queremos elaborar, enfocándonos en la utilización de plástico para su reciclaje, creando piezas que pueden satisfacer las necesidades dentro del hogar, de espacios recreativos, entre otras aplicaciones, generando un círculo del uso del plástico dentro de pequeñas comunidades o personas.

Palabras clave: Impresora 3D, plástico, reciclaje, comunidad, reducción.

1. Introducción

El desarrollo de productos ha cambiado drásticamente con la mejora de los procesos industriales, la evolución de las tecnologías ha permitido un gran avance en la mejora de los procesos de producción, siendo enfocadas a las demandas de un mercado cada vez más estricto en cuestión de calidad, demandas que han cambiado con el pasar de los años, puesto que en cada generación hay ideales distintos. Actualmente el mercado al que se dirigen las empresas en general, es un mercado que ha optado por los productos que denotan cuidado hacia la ecología, empresas que estén comprometidas con la protección del medio ambiente, desde el inicio de los procesos de producción, hasta el producto final que ofrecen, deben ser procesos que no tengan repercusiones ecológicas negativas, y debe ser un producto que no se mantenga como un daño perpetuo para la naturaleza.

La revolución industrial es un punto de referencia para señalar la evolución del modo de operación dentro de los procesos de producción, siendo procesos que a largo plazo han sido muy dañinos para el medio ambiente, se distinguían por ser procesos en los que las máquinas eran robustas y requerían de energía obtenida de la combustión de carbón, esto ha ido cambiando con la implementación de la energía eléctrica a las máquinas, siendo reducidas en su tamaño y en el manejo de las mismas con los avances en la tecnología digital.

Con el avance tecnológico han surgido nuevas maneras de producción, en las que resalta la compactación de los procesos hacia una sola máquina, dando otro sentido al concepto de manufactura, en las que, para la creación de un producto, se hace más eficiente tener el control de todo el proceso desde un solo lugar, yendo desde el diseño del producto, hasta la opinión del



consumidor final. Una de las máquinas implementadas son las máquinas de control numérico (CNC), y las impresoras 3D.

Las impresoras 3D son máquinas que se han desarrollado para la creación de piezas sólidas tridimensionales casi de cualquier forma a partir de un material en forma de filamento, elaborandolas mediante líneas vectorizadas las cuales van creando capa a capa el cuerpo de la figura determinada. Esta tecnología se produjo a mediados de los 80's cuando se desarrollaban sistemas computacionales y sistemas de control y ha ganado atención en estos últimos años ya que ha probado ser compatible con la manufactura a nivel industrial, incluso después del prototipado [1].

La propuesta de este proyecto es con la intención de utilizar la tecnología de la impresión 3D, para el reciclado del plástico desechado, integrando en la impresora un extrusor de plástico que utiliza un tornillo sin fin, una vez triturado, se vierte en un contenedor que va alimentando continuamente el extrusor, el cual funde y distribuye el material directamente en la plataforma donde se da forma a la pieza que se está fabricando, de esta manera se reduce el proceso de obtener el filamento desde un extrusor individual, y volverlo a fundir en el extrusor de la impresora. Al utilizar plástico triturado se evita que se siga acumulando en gran cantidad, pudiendo darle otro uso al material, su propiedad moldeable nos permite volver a darle forma para otro fin, y con la tecnología de impresión 3D, tenemos los medios para desarrollar el reciclaje de desechos plásticos en un entorno doméstico, o dentro de edificios de trabajo, como por ejemplo en una escuela, reduciendo el uso excesivo de plásticos, y alargando la vida útil del mismo material.

El deterioro ambiental es un tema de gran relevancia para quienes desarrollamos la tecnología en el área industrial, un punto que debemos tomar en cuenta en la implementación de todos los proyectos, con las herramientas con las que contamos hoy en día podemos llevar a cabo nuestras ideas de la mano con el cuidado ambiental, llevando así al desarrollo industrial de una manera sustentable.

2. Fundamentos

La impresión 3D es una tecnología muy novedosa que ha tenido un gran e importante impacto en el mundo. Se espera que esta revolucionaria tecnología cambie los modelos de negocio, los lugares de producción, las cadenas de suministros y alteren el orden económico global [2].

Esta tecnología representa un proceso para la creación de objetos mediante capas de cierto material a partir de un diseño digital, trabajando en conjunto equipo de software, hardware y el material que se esté utilizando para la creación de la pieza, también se puede definir como manufactura por adición, esto por la acción de la máquina en añadir líneas vectorizadas para así dar forma al objeto [3]. Esto resulta especialmente útil en lo que se conoce como "prototipado rápido".

Para la elaboración de una impresora 3D se precisan de una serie de conceptos y componentes, los cuales serán descritos a continuación.

2.1 Impresoras 3D

Existen distintas formas de fabricación digital mediante impresoras 3D, como lo es la estereolitografía, que utiliza rayos UV para endurecer resinas, sinterizado selectivo por láser, este método es mediante la pulverización de materiales, otra forma es mediante la fabricación con filamento fundido, que utiliza como material, plásticos calentados y extruidos [3], siendo este último con el trabajado en este proyecto.



Dentro del tipo de impresora que trabaja con filamento fundido se encuentran distintas estructuras, una de ellas es la cartesiana, es la más común en el mercado, siendo la estructura más sencilla, utiliza articulaciones prismáticas, que están orientadas de manera perpendicular entre ellas [4], permitiendo movimientos líneas en 3 dimensiones, X, Y y Z, teniendo ubicado el cabezal sobre el eje X y Y. La otra estructura son las polares, de esta forma se trabaja con un ángulo y una longitud, siendo la cama de la máquina la que gira, con el cabezal teniendo movimientos lineales, subiendo, bajando, alejándose y acercarse al eje, siendo similar al funcionamiento de una grúa industrial [5*]. Finalmente la otra estructura de una impresora 3D, son las delta, debiéndose el nombre a la forma en que se localizan 3 motores con una configuración triangular, para posicionar el cabezal sobre la cama de impresión circular, trabaja con coordenadas cartesianas moviendo los 3 motores hacia arriba y abajo el cabezal sobre todo el espacio superior a la cama [5].

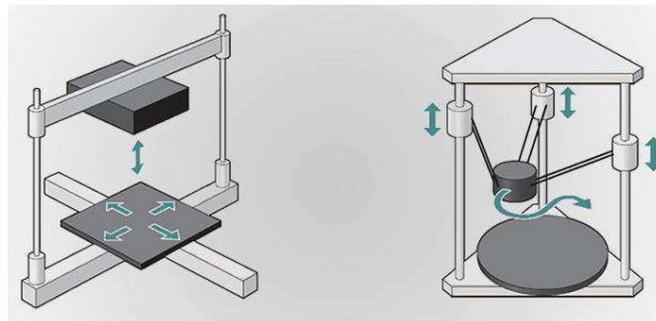


Figura 1. Estilo cartesiana y delta

2.2 Extrusor

El funcionamiento de los cabezales utilizados en las impresoras de filamento fundido, es calentar el material hasta ser fundido, a su vez estar inyectando continuamente el material extruido en forma de filamento, la estructura que tienen es simple, se utiliza un motor con un engranaje dentado en el eje, para hacer la función de un piñón cremallera, realizando un movimiento circular, convirtiéndolo en movimiento lineal, moviendo así el filamento hacia la parte caliente que lo fundirá. Otra forma de extrusión es mediante un extrusor de plástico, utilizada para la creación de filamento comúnmente de material triturado, su funcionamiento yace en la acción del giro de un tornillo de arquímedes denominado husillo, la estructura de este tipo de extrusor es de un tolva para captar el material triturado, consta de 3 partes dentro del husillo, que son la parte de adquisición del material dentro del tornillo, siendo las 3 partes con espacios más angostos para la compresión del material y fuerza en la eyección, así como sacar el aire dentro del material y así obtener resultados más uniformes en el filamento final, la cual se calienta alrededor de 100 °C, temperatura a la cual el material se funde.



Figura 2. Extrusor de impresora 3D



2.3 **Motor a pasos**

Para la parte del movimiento mecánico de los ejes de la máquina existen varias opciones, dentro de las más comunes es la utilización de motores a pasos, o bien, servomotores. Además, ofrecen una mejor relación costo-beneficio. Los motores van conectados directamente con la transmisión que hará posible que el movimiento rotatorio del motor se convierta en movimientos lineales haciendo posible que la máquina pueda trabajar bajo vectores.



Figura 3. Motor a pasos

2.4 **Vías de movimiento**

En la parte de transmisiones se cuentan distintas opciones para transmisión de movimiento a partir de los motores, las cuales son por bandas y poleas, tornillo sin fin y tuercas o piñón-cremallera. Para seleccionar el tipo de transmisión es importante considerar las velocidades que se requieren para trabajar así como la fuerza que los motores nos pueden entregar, siendo más práctico para la impresora, utilizar el funcionamiento de piñón cremallera, por la velocidad adecuada.

2.5 **Microcontrolador**

En la parte lógica, las impresoras 3D incorporan un sistema embebido que soporta corrientes relativamente altas haciendo uso de drivers que facilitan además la programación, incluyen también monitoreo de sensores de temperatura y finales de carrera, estos últimos ayudan en la parte de seguridad a delimitar los movimientos de la máquina y que está no ocasione daños eléctricos ni mecánicos.

2.6 **Sensor final de carrera**

Dentro de la automatización industrial existe una gran diversidad de dispositivos electrónicos que facilitan los procesos de producción, entre los que se encuentran los sensores finales de carrera. Este tipo de sensor detecta la posición de un elemento mediante accionamiento mecánico, los cuales necesitan estar en contacto con el objeto para detectar la llegada de un elemento móvil a una determinada posición. Su salida es binaria y la única información que nos da es si el objeto está en una posición determinada o no.

2.7 **Ramps 1.4**

Cuando se realiza un diseño en 3D, y se materializa, se precisa de una herramienta que permita controlar todos los componentes que en nuestra impresora se encuentran, así como de alimentarlos sin quemarse. Para ello se encuentra la tarjeta *Ramps*, la cual es una shield compatible con Arduino Mega que traduce las órdenes digitales de nuestro ordenador en órdenes por pasos, esto se hace a través de los drivers para los motores paso a paso.

2.8 **Termistores**



Son elementos de detección de temperatura compuestos por un material semiconductor sinterizado que presentan un gran cambio en la resistencia en proporción a un cambio pequeño en la temperatura. Son muy útiles en el control de temperatura en sistemas autónomos.

2.9 Resistencia cerámica

Estas son componentes electrónicos que proveen de cantidades determinadas de resistencia a una corriente eléctrica en circuitos electrónicos. Las resistencias cerámicas tienen su función en la base del extrusor, siendo los componentes que calientan el material a imprimir hasta un estado en que pueda ser moldeado.

2.10 Marlin firmware

Marlin es el software utilizado en el microcontrolador para que este pueda interpretar las instrucciones que la computadora le da y convertirlas en coordenadas que puedan mover los actuadores a la posición necesaria en tiempo, además controla la temperatura del extrusor, la cama caliente, los sensores de fin de carrera.

Este software resulta bastante conveniente ya que incorpora muchos sistemas de seguridad, que pueden evitar situaciones peligrosas. Por ejemplo, si uno de los sensores se daña, Marlin lo detecta y detiene todo en plan emergencia [11]. Además, al ser un open source, está conformado por una gran comunidad que ayudan a mejorar el mismo.

2.11 Prototipado rápido

Este se define como la manera más rápida de obtener una réplica exacta tridimensional de un diseño generado mediante aplicación CAD en 3D. Estos modelos físicos pueden ser únicamente estéticos y útiles para el estudio del mercado potencial al que van dirigidos, o pueden cumplir con algunas o buena parte de los requerimientos mecánicos que tendría la pieza definitiva, ofreciendo en este caso la posibilidad de realizar pruebas funcionales y de homologación antes de que existan los moldes preliminares.

3. Diseño

Para el diseño simulado de la impresora, se utilizó el software de solidworks, en el que se muestra pieza por pieza ensamblada, basada en una estructura tridimensional, de un prisma rectangular de un metro por un metro, por 40 cm de alto, con los perfiles de aluminio como vías de movimiento para los carros que permiten el movimiento al eje X, de los cuales, dos perfiles de aluminio en paralelo, sostienen el carro que da movimiento al eje Y, de sí mismo da soporte a las vías de movimiento del eje Y, ascendente y descendente.

3.1 Piezas

A continuación se muestran las piezas utilizadas con sus respectivas cantidades por unidad, y las medidas correspondientes a cada una de ellas.



Fig. 4. 6 x Perfil de aluminio



Fig. 5. 4 x Tubular cuadrado



Fig. 6. 18 x Tornillo

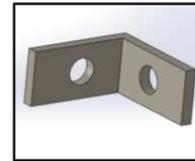


Fig. 7. 16 x escuadra



20*20*1000 mm

20*20*250 mm

3/8 "

1 1/2 "

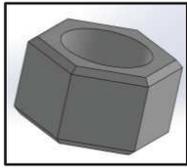


Fig. 8. 54 x Tuerca

3/8 "

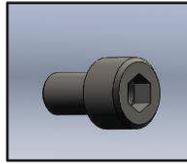


Fig. 9. 32 x Tornillo milimetrico flexible

M5*8

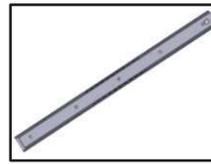


Fig. 10. 2 x Guia telescopica

30cm



Fig. 11. 1 x Eje

5mm a 8mm

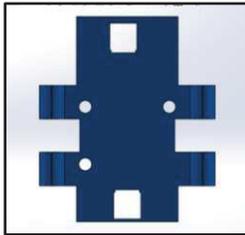


Fig. 12. 2 x Pieza impresa

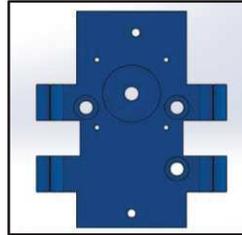


Fig. 13. 2 x Pieza impresa

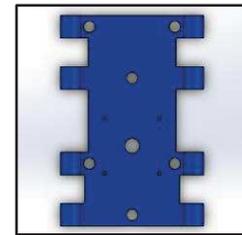


Fig. 14. 1 x Pieza impresa

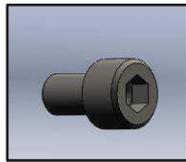


Fig. 15. 20 x Tornillo milimetrico

M4*12

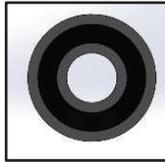


Fig. 16. 22 x Balero campana

6000 2rs

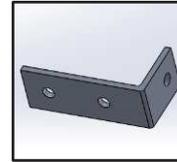


Fig. 17. 4 x Escuadra

1 1/2 " x 2 1/2 "

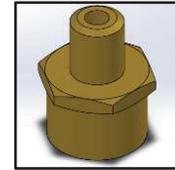


Fig. 18. 1 x Niple

3/8 " x 1/2 "



Fig. 19. 2 x Step motor Nema 17

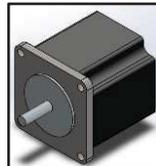


Fig. 20. 3 x Step motor Nema 23

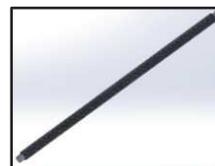


Fig. 21. 1 x Varilla roscada

5/8 " Ø x 15.748"

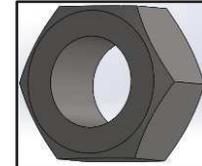


Fig. 22. 2 x Tuerca

5/8 "



Fig. 23. 20 x Tornillo 1/4 " 1/2 "

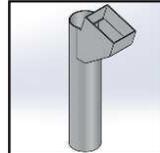


Fig. 24. 1 x Tubo con tolva

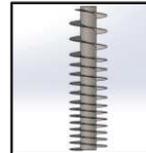


Fig. 25. 1 x Husillo



Fig. 26. 1 X Tuerca 1

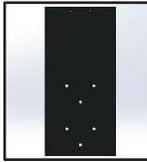


Fig. 27. 1 x Hoja de madera
Madera
40 x 18 cm

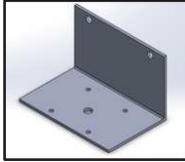


Fig. 28. 2 x Lámina metálica
18 x 8 cm



Fig. 29. 1 x Madera
18 x 20 cm

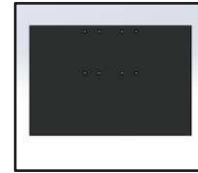


Fig. 30. 1 x

3.2 Montaje de piezas

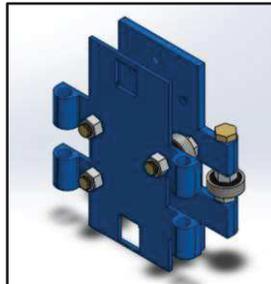


Fig. 31. Carro para el eje X
Vista isométrica



Figura 32. Diseño simulado del eje Y.

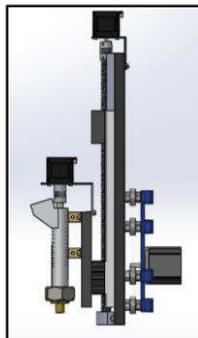


Figura 33. Diseño simulado del eje Z con extrusor.

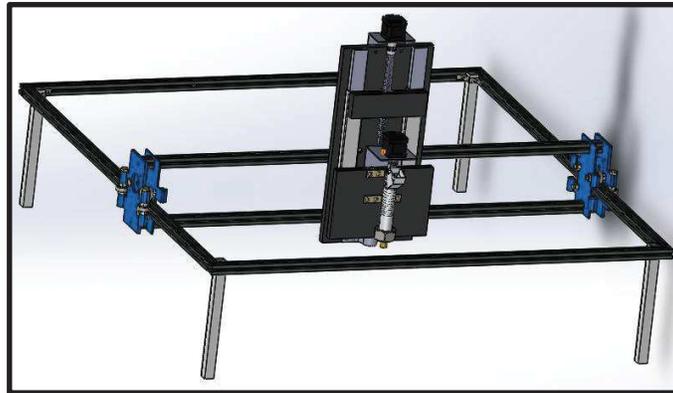


Figura 34. Prototipo de la impresora con extrusor de plástico

4. Resultados

Estructura de la impresora

Para la definición de un diseño adecuado se realizaron muchas pruebas y varios prototipos de lo que pudiera ser la impresora, algunas ideas tuvieron que ser descartadas por otras mejores. Es el caso del primer prototipo el cual consistía en unir los 4 perfiles de aluminio (que conforman el marco de la impresora) y los soportes (que dan la altura de la impresora) con coples de PLA impresos con impresora 3D; de esta manera sería más fácil unir las piezas. Sin embargo, dichos coples no suministraban la fuerza ni la estabilidad necesaria para soportar el peso de la impresora ni la misma en funcionamiento antes de que esta se rompiera. Por ello se optó por cambiar los coples de aluminio. Como solución, se decidió utilizar los soportes de la misma impresora; los cuales son tubos cuadrados (con dimensiones 2.54 x 2.54 x 25 cm). En la parte superior de los mismos se atornillaron 2 escuadras, en los que a su vez, a cada uno, se atornilla uno de los perfiles de aluminio; de modo que, por esquina, cada tubo sostiene 2 perfiles de aluminio. De esta forma pudimos unir la estructura básica de la impresora, incluyendo la base de soporte de toda la máquina.



Figura 35. Soporte de la impresora.

Para el segundo prototipo se utilizó varillas roscadas (Espárragos) como transmisiones de los motores hacia los ejes X,Y,Z que dieran lo largo de los perfiles de aluminio (100 x 100 x 25 cm, respectivamente) así que conseguimos varilla roscada de 11.113 mm (7/16 ") de diámetro por 100 cm de largo, las cuales mandamos al taller de torno para que les redujeron las orillas a un diámetro de 10 mm para que pudieron entrar en nuestros coples flexibles que usamos para conectar hacia los motores junto con esta idea se diseñaron unos carros para los ejes X, Y, los cuales fueran



fáciles de acoplar a estas varillas roscadas y que pudieran convertir el movimiento circular de la varilla a un movimiento lineal con ayuda de las tuercas.

4.1 Carros

El diseño de los carros se realizó acorde a las medidas del motor y de los perfiles de aluminio, se obtuvieron con ayuda de una impresora 3D integrando los rodamientos correspondientes, se comenzó con las pruebas con la máquina ya armada con la base hecha por los perfiles de aluminio, pero la varilla necesitaba bastantes vueltas para poder avanzar tan sólo algunos milímetros en la máquina, siendo que los motores son de baja velocidad tardaba alrededor de 4 minutos en recorrer de lado a lado el riel lo que ocasiona que la máquina sea muy lenta aunque eso sí la precisión obtenida fue muy alta. Para este proyecto no es una buena opción, debido a que se requiere que la máquina vaya a velocidades razonables y que la parte de precisión no afecte tanto porque aún así cambiando el tipo de transmisión resulta ser buena.

En ese momento se opta por cambiar la forma de transmitir el movimiento, cambiando las varillas por poleas y bandas, descartando los modelos anteriores y se hacen nuevos diseños donde el motor iría sobre el mismo eje que estaría desplazándose y éste al final no tendría un cople flexible sino que sería reemplazado por una polea dentada, una vez que se tenían los carros diseñados e impresos se prosigue a reemplazarlos por los que ya se tenían. Cuando se realizarán las pruebas ya con estos nuevos cambios se hacen de nuevo los ajustes del software ya que las configuraciones cambian en cuestión a los milímetros por segundo qué puede avanzar la máquina con el giro del motor, se constató que justamente es lo que se requiere para la impresora debido a que trabaja velocidades buenas y una precisión satisfactoria además de que está tenía la suficiente fuerza (torque o par) para cargar con el peso de los diferentes ejes.



Figura 36. Transmisión por banda y polea.

El modelo que determinado para los carros, iría inicialmente solamente uno en el eje X, de esta manera se estaría usando un motor por cada uno de los ejes, reduciendo el costo de los materiales usados y espacio para el microcontrolador, pero al momento de experimentar el movimiento de este eje alimentando los motores, no realizaba el movimiento esperado de los dos perfiles de aluminio, los cuales tienen que hacer un movimiento perpendicular al perfil por el cual se traslada el motor, tanto del lado del motor, como del lado contrario al que se encuentra este, lo que pasaba era que primeramente se trasladaba el lado donde estaba el motor, haciendo que los dos perfiles se se ladeara, arrastrando el lado contrario del motor, esto no era favorable, ya que no permitiría movimientos controlados de toda la impresora, de tal manera que se colocó dos motores para el eje X, los cuales realizaron movimientos simétricos, el accionamiento de los motores a pasos es mediante la activación de bobinas para que giren en un sentido, para que el movimiento de estos dos motores colocados en una posición contraria giraran al mismo lado, se invirtió la señal de las bobinas en uno de los motores, de esta manera se trasladan hacia la misma dirección ambos motores desde un solo comando para su control, descartando otras opciones como la de



colocar un eje de transmisión desde un motor a otro, resultando más difícil de controlar por las dimensiones de los materiales planeados a utilizar, quedando en el producto final, el diseño con los dos motores.



Figura 37. Carro eje X

Para el eje Y, se colocó un solo carro, el cual sería de la misma forma que los carros anteriores, pero en esta ocasión, su función es deslizarse a lo largo de dos perfiles de aluminio paralelos, por lo que se cambió el diseño, modificando solamente lo alto para poder sujetar los baleros de ambos perfiles, el problema en esta parte, fue que debía estar sujeto a este carro, la base para el eje Z, habiendo inconsistencias en que el mismo diseño serviría para ello, pero al colocarlo, solamente se tuvo que perforar la tabla base, para que las tuercas de ajuste entren en ese espacio, así dejando espacio libre para eje Z.

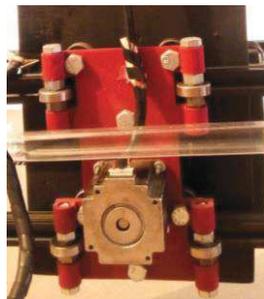


Figura 38. Carro Y

4.2 Base eje Z

La base para el eje Z, se realizó con una madera de 40 cm x 20 cm x 1 cm, colocando sobre ella 1 par de rieles deslizantes de acero inoxidable para cajones de 16 pulgadas y 3 secciones cada uno, los cuales tienen la función de transmitir un movimiento más libre en cuestión a la fricción que se produce.

A partir de la base mencionada se montan varios dispositivos de los cuales tienen una acción fundamental para el funcionamiento del eje Z así como del área del extrusor. Tales dispositivos se describen a continuación:

1. Lamina de acero inoxidable de 12 cm x 8 cm x 2 mm, la función de esta lámina es ser la base que sostiene al motor que mueve todo el eje Z. Los inconvenientes que se presentaron fueron, las perforaciones para las sujeciones hacia la base principal y el motor ya que esta lamina de acero inoxidable es bastante rígida, también para el dobles de 90° fue un factor importante pero estas consignas se solucionaron con el uso de brocas de metal de diferentes tamaños, pinzas y taladro.



2. Madera 5 cm x 18 cm x 1cm, esta tabla tiene la función de transmitir una conexión entre los rieles de deslizamiento y el espárrago que mueve la base del extrusor por medio de dos tuercas que embonan en el espárrago. Antes de ensamblar dicha pieza se cortó con el router la parte central superior y central inferior de la tabla con dimensiones de la tuerca anteriormente mencionada, para la sujeción de dicha tabla con los rieles se utilizaron tornillos en ambos rieles.

3. Tabla de madera de 18 cm x 15 cm x 1 cm, esta tabla tiene como función ser la base del extrusor de la impresora, la conexión entre los rieles de deslizamiento se hizo de la misma forma que la tabla de conexión del eje Z (con tornillos en cada riel pero en la parte inferior de estos).

4. Espárrago $\frac{1}{2}$ " de 30 cm con extremos lisos hechos con torno, la función de esta pieza es ser el mecanismo de transmisión de movimiento del eje Z por medio del giro del motor.

5. Balero y Lámina doblada con silueta de $\frac{1}{2}$ balero, la función de estos dispositivos es dar libre movimiento de Z, la lámina se fija a la base principal junto con el balero los cuales no permiten el movimiento ascendente o descendente del espárrago sino que permiten el giro del espárrago en el mismo punto, de tal forma que a través del mecanismo de transmisión de movimiento (Tabla 18 cm x 5 cm x 1 cm) se mueven los rieles y estos a su vez mueven la base del extrusor.

6. Sensor de final de carrera, como su nombre lo dice, la función de este es dar un punto final al proceso o paro de la etapa, en este caso se opta por poner este sensor en la parte superior izquierda del riel derecho con el fin de parar el movimiento del motor cuando el mecanismo de transmisión de movimiento lo toque, también tiene el uso de punto de referencia.



Figura 39. Base del eje Z

4.3 Sujeción de bandas

El movimiento circular de los motores se transmite a una banda, transformándola en forma lineal, un problema que surgió fue la de colocar las bandas de manera firme, para que toda la fuerza de los motores no se perdiera, obteniendo así más precisión, la forma en que deben ir sujetas es de tal manera que no abarque el espacio libre para los ejes de la impresora, por lo que deben estar lo más pequeña posible pero a la vez muy fuerte para mantener la banda tensa, primeramente se implementaron piezas impresas en otra máquina de impresión 3D, pero resultaron demasiado grandes, por lo que se sustituyó por placas de acrílico perforadas, de tal manera que los extremos de las bandas se mantuvieran sujetas con solo tensarlas una vez, estas placas fueron sujetas con tornillos para tener mayor fuerza, abarcando un espacio delimitado por los perfiles de aluminio y el tubo cuadrado como soporte de cada esquina.

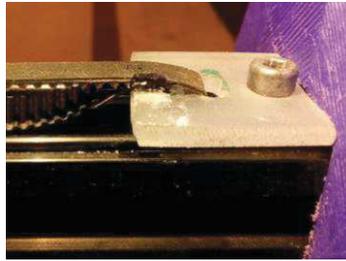


Figura 40. Ajuste de banda

Un inconveniente que surgió fue al momento de sujetar las bandas en el eje Y, ya que el carro realiza un movimiento lineal a través de dos perfiles paralelos, siendo en esta parte donde no hay algun lugar de donde atornillarlas de la manera que con los del eje X, para ello, se implementó las mismas placas de acrilico, pero ahora más pequeñas, con la misma finalidad de que no abarque mucho espacio para dejar más libertad de movimiento al motor, se posicionaron hasta los extremos de los perfiles, con las placas perforadas para sujetar la banda, y para atornillarla, pero esta ocasión con una tuerca colocada en la ranura que tienen los perfiles de aluminio, para así realizar presión de la parte inferior de la placa, fue más complicado colocarlo en esta parte porque se requiere mantener tensa la banda y así permitir al carro realizar movimientos libres.

4.4 *Finales de carrera*

Los movimientos lineales de cada eje tienen como delimitación los finales de carrera, los cuales tuvieron un funcionamiento óptimo, el problema que surgió fue en la fijación de ellos, porque en ocasiones que no estaban muy fijos, se quitaban por la fuerza que tenían los carros, por lo que se hicieron varias pruebas, inicialmente colocándolos sobre las placas de acrílico de los que se sujetaban las bandas, pero no funcionó por la lejanía al lugar donde debería terminar el movimiento de los carros y por la forma en cómo se sujetaron, utilizando pegamento de fijación rápida, por lo que en cada eje se colocaron de distinta manera acorde al material en donde debería ir colocado, en el eje X se posicionaron dos finales de carrera, esto por los dos motores utilizados en este eje, los cuales funcionan a la vez como localizaciones en las cuales el eje toma como referencia para realizar las funciones requeridas, en este lugar se fijaron mediante tornillos, para el eje Y, fue solamente un final de carrera, de igual manera para que en ese lado, se localiza la posición inicial de la cual partirán todos los movimientos de la máquina, por último para el eje Z, siendo su movimiento vertical, el final de carrera se localiza en la parte superior, en ese lugar será la posición de donde partirá.

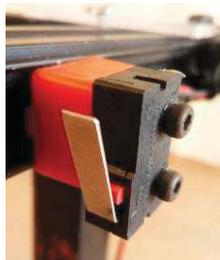


Figura 41. Sensor final de carrera

4.5 *Cama*

La superficie destinada para ser la base sobre la que se imprima las piezas, es una cama de madera de un metro por un metro, requiere libre movimiento hacia arriba y abajo, ya que se debe



calibrar para que quede a una misma distancia hacia el extrusor en todos los puntos en donde se mueva, esto con el fin de que no haya roces de la punta del extrusor con la cama y a la vez, no tenga problemas con realizar la pieza, para ello se modificaron las esquinas de la tabla para que se acoplara a los soportes de cada vértice de la impresora. Una vez posicionada la cama, se buscó la manera en como hacer para que toda la cama realizará movimientos verticales en los 4 puntos, para así calibrar la altura óptima, una de las opciones fue en realizar perforaciones en forma hexagonal en las bases de madera inferiores, para embonar una tuerca, y esta sea la guía para hacer subir y bajar un tornillo de $\frac{3}{8}$ " , al cual en el cabezal hexagonal, estaría fija la tabla, esto en los 4 puntos especificados, otra opción con el mismo funcionamiento, fue con coples para los tornillos en lugar de las tuercas, siendo más fácil la perforación de la base inferior, y con mejor manejo de los tornillos al ser de menor espesor, siendo estos con cabezal hexagonal cóncavo, utilizando de esta manera un desarmador para así calibrar, en lugar de una llave que se requería en caso de colocar los tornillos de $\frac{3}{8}$ ". Para mantener las esquinas de la cama fijas al cabezal de los tornillos, se usaron resortes de compresión, para que su fuerza se direcciona hacia arriba, y el cabezal del tornillo, hacia abajo, de esta forma el tornillo bajara, permitiendo a la cama, deslizarse manteniéndose siempre fija a la altura del tornillo, y con una rondana en cada lado de la cama para un apoyo de sujeción más uniforme.



Figura 42. Mecanismo de ajuste de altura

4.6 *Extrusor*

La impresora 3D de este proyecto tiene como visión o punto adicional la implementación de extrusión a base de pet reciclado. Para llevar a cabo esto se realizó un prototipo de extrusor adaptado a las condiciones, necesidades y disponibilidad de materiales.

El primer prototipo se realizó de la siguiente manera, en ella se pueden apreciar diferentes materiales entre los cuales destaca un tubo de acero de $1\frac{1}{2}$ " de diametro x 12 cm de largo, tuerca de acero de 1", niple campana de $\frac{3}{8}$ " x $\frac{1}{2}$ " (la cual funciona como la boquilla del extrusor, lámina con corte en forma de tolva. husillo, tornillos, escuadras.

Una vez terminado el diseño físico se toma la primer prueba de experimental, para esta primer prueba hubo inconsistencias. En primer momento la variable de temperatura es muy importante a tomar en cuenta ya que las resistencias colocadas en la superficie del extrusor, las cuales no proporcionaba la temperatura suficiente para la fundición del plástico, por tal motivo el motor a pasos nema 17 no proporcionaban el torque suficiente para girar el uso husillo el cual es el encargado de la conducción de la viruta de PET.

En cuestión al problema más importante que se presentó en la primer prueba se tomó la decisión de perforar la tuerca que sujeta y conecta el tubo con el Niple (Boquilla del extrusor) y



meter las resistencias dentro de esta para obtener una conducción más uniforme y a la vez elevar el nivel de temperatura para poder fundir la viruta de PET.

Al llevar a cabo la segunda prueba experimental, la opción planteada soluciona el problema presentado en la primer prueba, pero se presenta otro obstáculo que irrumpía el objetivo, para este problema la viruta de PET era demasiado grande para llevarla de la tolva hacia la boquilla por medio del husillo. La solución para esta problemática lo encontramos en la trituración más fina de la viruta.

Para la prueba experimental número 3 se llevó a cabo la solución encontrada, en este aspecto tuvimos una respuesta positiva, aunque nuevamente se presentó una problemática más. En esta ocasión el flujo de PET fundido no era suficiente para poder extruir y por lo tanto el motor no podía mover el mecanismo del extrusor de manera eficiente, por lo que ahora el problema estaba en la transmisión de fuerza para el mecanismo del extrusor, visto a este problema se buscó otra opción de transmisión de fuerza de tal modo que tuviera el torque necesario que satisfaga la extrusión. Para solucionar este último problema presentado se optó por cambiar de motor (de un motor a pasos a un motorreductor).

Se llevó a cabo la siguiente prueba experimental y en esta se implementó la opción de solución del motorreductor, para este caso se volvió a solucionar esa área aunque aún persistía el problema del flujo de plástico, nos dimos cuenta que la temperatura aún no era suficiente para fundir la cantidad de plástico entrante aun con 2 resistencias conectadas, la opción de usar pet reciclado era un opción aún más compleja de lo que parece, ya que al poner más resistencias se elevaría el consumo de la máquina (3 A por cada resistencia más a parte el consumo de los demás componentes).



Figura 43. Prototipo de extrusor.

4.7 **Conexión y software**

En la tarjeta diseñada especialmente para impresoras 3D vienen integrados los pines necesarios y distribuidos de forma que solo hace falta cargar el software a la placa y conectar cada parte del hardware en la parte que corresponde. En esta placa encontramos pines para motores paso a paso, finales de carrera, resistencia de extrusor (hotten), cama, termopares, driver para motores paso a paso y algunos otros pines adicionales para mejorar cuestiones mecánicas de la impresora.

El software utilizado es Marlin Firmware v.1.1.9, Marlin es un firmware de código abierto para la familia RepRap de replicadores de prototipos rápidos, popularmente conocido como "impresoras 3D". Se derivó de Sprinter y grbl. Este software se comenzó a popularizar por su flexibilidad para trabajar con microcontroladores Atmel AVR, como los que se incorporan el los arduinos, inclusive se puede contar con la opción de compilar dicho código desde la IDE de arduino. Marlin Firmware se ejecuta en la placa principal de la impresora 3D, gestionando todas las actividades en tiempo real de la máquina. Coordina los hotten, los motores paso a paso, los sensores, la pantalla LCD, los botones y todo lo relacionado con el proceso de impresión 3D.



El lenguaje de control para Marlin es un derivado del G-código . Los comandos de código G le dicen a la máquina que haga cosas simples como “configurar el calentador 1 a 180 °” o “moverse a XY a velocidad F.” Para imprimir un modelo con Marlin, se debe convertir a código G usando un programa que pueda vectorizar un diseño y posteriormente convertir esos vectores en comandos que pueda leer la máquina.

A medida que Marlin recibe comandos de movimiento, los agrega a una cola de movimiento que se ejecutará en el orden recibido. La "interrupción por pasos" procesa la cola, convirtiendo los movimientos lineales en impulsos electrónicos de tiempo preciso en los motores paso a paso. Incluso a velocidades modestas, Marlin necesita generar miles de pulsos paso a paso cada segundo.

Los calentadores y los sensores se administran en una segunda interrupción que se ejecuta a una velocidad mucho más lenta, mientras que el bucle principal maneja el procesamiento de comandos, la actualización de la pantalla y los eventos del controlador. Por razones de seguridad, Marlin se reiniciará realmente si la CPU se sobrecarga demasiado para leer los sensores.

Este software incorpora también sistemas de control retroalimentados para estar monitoreando distintas variables y notificar al usuario en caso de existir alguna anomalía, incluso puede suspender el proceso que esté realizando la máquina para protegerla mecánicamente, o bien, electrónicamente.

5. Conclusiones

El trabajo de este proyecto ha sido ventajoso, para verificar el funcionamiento del software de la impresora, junto con la mecánica de la misma, se han hecho pruebas con un extrusor de plástico de impresora 3D en nuestro prototipo, obteniendo buenos resultados al realizar algunas pruebas. Continuamos en desarrollo con el primer prototipo para continuar con las modificaciones y mejoras para obtener el objetivo, de unificar el proceso de extrusión de plástico triturado en una impresora 3D para la agilización del desarrollo de productos en entornos domésticos.



Figura 44. Prueba de la impresora

Como trabajo a futuro de este proyecto, pretende ser parte de una campaña ecológica para el Centro universitario de los valles, la cual consiste en, que los alumnos aprendan a identificar la clasificación correcta de la basura, posteriormente, se usará solamente el PET (envases plásticos), este será procesado mecánicamente para ser triturado y quedando en pequeñas virutas. Una vez realizado el proceso anterior, esta viruta es capaz de servir como materia prima para la impresora 3D, con lo que se plantea imprimir estructuras para crear un tipo de pequeños domos, estos son utilizados para dar sombra en los pasillos de la escuela, beneficiando a la misma escuela y a los alumnos, de esta manera se consigue la colaboración de ambas partes.



Nuestro proyecto también está pensado para poder trabajar con empresas e instituciones en las cuales el uso y desecho de plástico sea en grandes proporciones, con ello, se crea la oportunidad de poder reciclar este material de manera ventajosa tanto para la empresa como para el medio ambiente. Por parte de la empresa es reutilizado por medio de un proceso de reciclado mecánico del que se obtiene la viruta de plástico, y esta puede ser transformada en un producto que sea de utilidad para su infraestructura, puede bien ser transformada en bancas comunitarias, estructuras para toldos, mesas, sillas, botes para basura, entre otros productos.

Por ser una máquina de fácil replicación puede tomar una escalabilidad mucho más tentativa, pues bien se puede plantear como un proyecto gubernamental para las grandes ciudades, el cual consistiría en hacer conciencia en la sociedad para que separe la basura correctamente.

Referencias

- [1] Gebler M., Schoot Uiterkamp A., Visser C. , "A *global sustainability perspective on 3D printing technologies*", Energy Policy, vol. 74, pp. 158-167, 2014. Available: 10.1016/j.enpol.2014.08.033
- [2] J. M. M., "*Proceso de la impresión 3D como aporte al diseño gráfico*", Proyecto de grado, Universidad Rafael Landívar, Guatemala de La Asunción, 2015.
- [3] "*¿Qué es la impresión 3D?*", AUTODESK, San Rafael, California, Tech.
- [4] "*Robótica industrial, estructura mecánica del robot*", class notes for Sistemas Robóticas, Departamento de Ciencias Exactas e Ingeniería, Universidad de Guadalajara, Primavera 2019.
- [5] Lucía, C., "*Tipos de impresoras 3D FDM: Delta, Cartesiana, Polar y Brazo robótico*", 1st ed., 2017.
[Online]. Available: <https://www.3dnatives.com/es/tipos-impresoras-3d-fdm-190620172/>



Optimización en el Diseño de Brazos Robóticos: Algoritmos de Enjambre

Méndez Guzmán Hugo Antonio*, Ledesma Silva Gabriel Daniel
y Zavala Rojas Pablo Alejandro

Universidad Politécnica del Bicentenario
Carretera Silao-Romita km. 2, San Juan de los Durán, C.P. 36283. Tel. (472) 72 38 700
hmendezg@upbcentenario.edu.mx

Resumen

Se presenta la metodología para la optimización de las dimensiones de los eslabones que conforman un brazo robótico de 4GDL a partir de dimensiones iniciales y una trayectoria inicial LIN dadas, con el propósito de determinar las dimensiones óptimas que produzcan la menor cantidad de movimiento angular y torque dinámico para un trabajo asignado. Se presenta una comparativa en el desempeño de los algoritmos PSO, ABC, BAT y PIO para la optimización de este procedimiento, así como una comparativa del movimiento angular y torque dinámico entre las dimensiones iniciales y optimizadas.

Palabras clave: Algoritmos de Enjambre, Optimización, PSO, BAT, ABC, PIO, Dinámica de Robots.

1. Introducción

El diseño es el primer paso para lograr el máximo desempeño de cualquier producto o servicio que se desee implementar. Dependiendo del objeto que se trate debemos considerar ciertas propiedades y parámetros como la forma, el tamaño, el peso, su consumo de energía o el costo de fabricación.

Para lograr que un diseño obtenga los mejores resultados es necesaria la optimización de cada una de sus características, de acuerdo a las tareas y trabajos a lo que se le asignará. En el caso de la robótica puede ser para trabajos de mantenimiento, trabajos de manufactura como pintar, soldar, ensamblar o mover objetos de un lado a otro.

Cuando se trata de implementar un robot a un proceso, se suele tomar en cuenta el alcance y el peso del robot, pero se suele ignorar la trayectoria o el espacio que necesita para trabajar. Esto puede llevar a un exceso de trabajo y de consumo de energía por parte de los motores y demás elementos, que a largo plazo se convierten en altos costos o baja producción. Debido a lo anterior, en este proyecto se decidió analizar algoritmos evolutivos como una forma eficaz de optimizar el diseño de un robot, específicamente la longitud de sus eslabones. Con esto se busca minimizar el torque generado por las articulaciones, así como su movimiento angular y, por lo tanto, el consumo de energía y el desgaste que sufrirá el robot a lo largo de una trayectoria predeterminada por el trabajo que va a realizar.

2. Metodología

Partiendo de las dimensiones iniciales para cada eslabón, se asignan posiciones iniciales y finales de una trayectoria lineal. Para cada iteración del algoritmo se calculan 20 puntos angulares



intermedios y equidistantes de la trayectoria LIN por las que pasa el robot, estas posiciones son calculadas usando cinemática inversa. Una vez obtenidas se mide el incremento angular total que tendrá que efectuar el Robot de una posición intermedia otra y a la vez se calcula el torque dinámico efectuado a lo largo de la trayectoria. Este proceso se ve representado en la figura 1. El proceso de optimización se realiza por 1000 iteraciones por los algoritmos PSO, ABC, BAT y PIO para su posterior análisis.

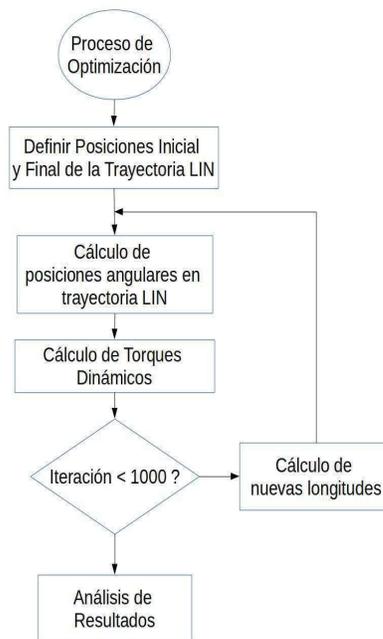


Figura 1. Metodología de Optimización.

3. Cinemática de Robots

La cinemática es el estudio del movimiento del robot respecto a un sistema de referencia, sin considerar las fuerzas que lo ocasionan [1]. Para nuestro trabajo es necesario definir tanto la cinemática directa como inversa, ya que son indispensables para la representación de movimiento de la trayectoria lineal y el cálculo de los incrementos angulares sufridos en una trayectoria lineal respectivamente.

3.1 Cinemática Directa

En la cinemática directa conocemos la posición de cada una de las articulaciones ($q_1; q_2; q_3; \dots; q_n$), es decir, el valor angular de la articulación de revoluta. El objetivo es utilizar estos parámetros para encontrar la posición ($x; y; z$) y orientación ($\alpha; \beta; \gamma$) del efector final del brazo [2]. Esto se logra a través del uso de matrices de transformación homogénea que permiten referenciar cada eslabón del robot con el sistema de referencia principal ubicado en la base del robot, a partir de ahí se coloca un sistema de coordenadas en la base de cada eslabón. Una vez establecidos todos los ejes de cada sistema, estos quedan definidos respecto al sistema anterior por medio de cuatro variables, conocidas como los parámetros Denavit-Hartenberg [2]. Los parámetros D-H para nuestro caso de estudio están definidos en la tabla 1.



Tabla 1. Parámetros D-H de Robot Antropomórfico de 4GDL.

i	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	$\pi / 2$
2	θ_2	0	l_2	0
3	θ_3	0	l_3	0
4	θ_4	0	l_4	0

Esto se logra substituyendo los parámetros Denavit-Hartenberg en una matriz general de transformación homogénea (ecuación 1) para cada eslabón.

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Con estas matrices podemos transformar un punto referenciado a cualquiera de los sistemas multiplicando directamente las matrices de los sistemas anteriores (ecuación 2), esto forma una nueva matriz referenciada al sistema de origen.

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \quad (2)$$

Para el caso de estudio, la posición del efector final está dada por:

$$\begin{aligned} x &= [l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3) + l_4 \cos(\theta_2 + \theta_3 + \theta_4)] \cos(\theta_1) \\ y &= [l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3) + l_4 \cos(\theta_2 + \theta_3 + \theta_4)] \sin(\theta_1) \\ z &= l_1 + l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3) + l_4 \sin(\theta_2 + \theta_3 + \theta_4) \end{aligned} \quad (3)$$

3.2 Cinemática Inversa

En la cinemática inversa consiste en la obtención de los valores angulares de cada articulación a partir de la posición y orientación del robot [1]. Usando el método algebraico los valores angulares de las primeras articulaciones está dado por:

$$\theta_1 = \arctan \left(\frac{y}{x} \right) \quad (4)$$



$$c_3 = \frac{W_x^2 + W_z^2 - l_2^2 - l_3^2}{2l_2l_3} \quad (5)$$

$$s_3 = \pm\sqrt{1 - c_3^2} \quad (6)$$

$$\theta_3 = \arctan\left(\frac{s_3}{c_3}\right) \quad (7)$$

$$s_2 = \frac{W_x(l_3c_3) - W_z(l_2 + l_3c_3)}{l_2^2 + l_3^2 + 2l_2l_3c_3} \quad (8)$$

$$c_2 = \frac{-W_z(l_3s_3) - W_x(l_2 + l_3c_3)}{l_2^2 + l_3^2 + 2l_2l_3c_3} \quad (9)$$

$$\theta_2 = \arctan\left(\frac{s_2}{c_2}\right) \quad (10)$$

donde W_x y W_z están dados por:

$$W_x = x\cos(\theta_1) + y\sin(\theta_1) - l_4\cos(B) \quad (11)$$

$$W_z = z - l_1 - l_4\sin(B) \quad (12)$$

finalmente, la última articulación se puede calcular por:

$$\theta_4 = B - \theta_3 - \theta_2 \quad (13)$$

donde B es el ángulo de orientación del efector final con respecto a las articulación de hombro, brazo y muñeca del Robot.

4. Dinámica de Robots

La dinámica se encarga de estudiar la relación del movimiento de un objeto y las fuerzas que lo provocan [1]. Aquí se abordan las ecuaciones que le dan movimiento al robot según la metodología propuesta, así como los pares de torsión generados en cada articulación debido a este movimiento y generado por los actuadores. Al igual que la cinemática existen en la dinámica dos problemas básicos. En el primero buscamos el movimiento que tendrá el mecanismo al aplicarle pares de torsión dados en un vector τ . En el segundo calculamos este mismo vector τ en base a los vectores de posición, velocidad y aceleración (θ , θ' y θ'') del mecanismo.

En el caso de este proyecto nos enfocaremos en encontrar la dinámica inversa, donde buscamos los pares de torsión necesarios en cada grado de libertad para generar la rotación deseada



en cada eslabón y que son uno de los criterios principales de la función objetivo que definirá la opción más óptima para las dimensiones del robot.

4.1 Ecuación de Movimiento Newton-Euler

La ecuación de movimiento Newton-Euler es un modelo recursivo en el cual obtenemos la posición, velocidad y aceleración del eslabón i a partir del eslabón $i - 1$. Desde el eslabón 1 hasta el eslabón n también se pueden calcular las fuerzas y pares de torsión del eslabón i referenciado a la base del robot a partir de los que hay en el eslabón $i + 1$.

De esta forma se establece este modelo como:

$$\tau = D \ddot{q} + C + G \quad (14)$$

D es la matriz de inercia generalizada, C son las fuerzas de Coriolis y centrípetas, G es el vector de la fuerza de gravedad que influye en el eslabón. Este modelo involucra varios conceptos dimensionales y posicionales, así como las fuerzas y los pares de torsión.

4.2 Tensor de Inercia

Para un objeto que existe en tres dimensiones existen infinitud de posibles ejes de rotación, por ello se necesita una forma general de representar la distribución de la masa de todo el cuerpo respecto a cada uno de los ejes del sistema coordenado donde se encuentra. Esto se conoce como tensor de inercia, que puede definirse como la generalización del momento escalar de inercia [2]. La matriz de inercia se puede definir como:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (15)$$

con

$$I_{xx} = \iiint_V \rho (y^2 + z^2) dV \quad (16)$$

$$I_{yy} = \iiint_V \rho (x^2 + z^2) dV \quad (17)$$

$$I_{zz} = \iiint_V \rho (x^2 + y^2) dV \quad (18)$$

$$I_{xy} = I_{yx} = \iiint_V \rho (xy) dV \quad (19)$$

$$I_{xz} = I_{zx} = \iiint_V \rho (xz) dV \quad (20)$$



$$I_{yz} = I_{zy} = \iiint_V \rho(yz) dV \quad (21)$$

Considerando cada eslabón del robot como cilindros sólidos (figura 2) de radio r y altura h , su tensor de inercia se puede calcular cambiando el sistema a coordenadas cilíndricas.

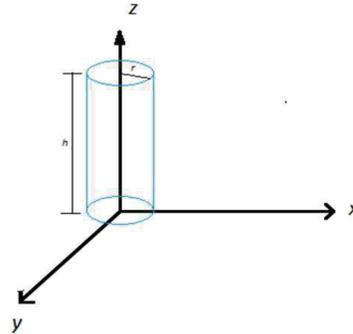


Figura 2. Cilindro en coordenadas cartesianas.

Donde los componentes de la matriz de inercia se definen como:

$$I_{xx} = \int_0^h \int_0^{2\pi} \int_0^r \rho(r^2 \sin^2 \theta + z^2) |J| dr d\theta dz \quad (22)$$

$$I_{yy} = \int_0^h \int_0^{2\pi} \int_0^r \rho(r^2 \cos^2 \theta + z^2) |J| dr d\theta dz \quad (23)$$

$$I_{zz} = \int_0^h \int_0^{2\pi} \int_0^r \rho r^2 |J| dr d\theta dz \quad (24)$$

$$I_{xy} = I_{yx} = \int_0^h \int_0^{2\pi} \int_0^r \rho(r^2 \sin \theta \cos \theta) |J| dr d\theta dz \quad (25)$$

$$I_{xz} = I_{zx} = \int_0^h \int_0^{2\pi} \int_0^r \rho(r \cos \theta) z |J| dr d\theta dz \quad (26)$$

$$I_{yz} = I_{zy} = \int_0^h \int_0^{2\pi} \int_0^r \rho(r \sin \theta) z |J| dr d\theta dz \quad (27)$$

Donde r es el determinante de la matriz Jacobiana del cilindro (28). La diferencia con las coordenadas cartesianas es que los términos x , y y z son sustituidos por las expresiones de (29)



$$J = \begin{bmatrix} \frac{dx}{dr} & \frac{dx}{d\theta} & \frac{dx}{dz} \\ \frac{dy}{dr} & \frac{dy}{d\theta} & \frac{dy}{dz} \\ \frac{dz}{dr} & \frac{dz}{d\theta} & \frac{dz}{dz} \end{bmatrix} \quad (28)$$

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \\ z &= z \end{aligned} \quad (29)$$

y ρ es la densidad de volumen:

$$\rho = \frac{m}{\pi r^2 h} \quad (30)$$

Sustituyendo cada uno de los elementos el tensor de inercia queda definido como:

$$I = \begin{bmatrix} \frac{mr^2}{4} + \frac{mh^2}{3} & 0 & 0 \\ 0 & \frac{mr^2}{4} + \frac{mh^2}{3} & 0 \\ 0 & 0 & \frac{mr^2}{2} \end{bmatrix} \quad (31)$$

4.3 Matriz de Inercia Generalizada

D_i es una matriz cuadrada de nxn conocida como matriz de inercia debido a que en ella se incluyen términos como el tensor de inercia I [3].

$$D_i = m_i J_{li}^T J_{li} + J_{wi}^T (R^T I_i R) J_{wi} \quad (32)$$

Dónde m_i es la masa del eslabón, J_{li} y J_{wi} son los jacobianos lineales y rotacionales respectivamente, I_i es el tensor de inercia y R es la matriz de rotación necesaria para referenciar el tensor de inercia en el sistema de coordenadas base del robot.

Esta ecuación se trata de la suma de los dos factores que intervienen en el movimiento, la fuerza y el momento causantes de la traslación y la rotación, dando como resultado la matriz de inercia generalizada (32) del modelo dinámico y expresada en cada una de sus componentes, a través de (33).



$$D = \sum_{i=1}^n D_i = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix} \quad (33)$$

4.4 Fuerzas Centrifugas y de Coriolis

Dentro del modelo dinámico (14) se encuentran expresadas la matriz C , que incluye las fuerzas centrífugas y de Coriolis. Este elemento C se obtiene a partir de la matriz de inercia generalizada D (33) mediante los símbolos de Christoffel [3]:

$$C = \sum_{k=1}^n c_{ijk} \dot{q}_k \quad (34)$$

siendo:

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial d_{ij}}{\partial q_k} + \frac{\partial d_{ik}}{\partial q_j} - \frac{\partial d_{jk}}{\partial q_i} \right) \quad (35)$$

El número de símbolos de Christoffel que debemos calcular está determinado por el número de articulaciones del robot. Para la morfología del robot que estamos analizando, un robot con cuatro grados de libertad, se calcularon 64 símbolos de Christoffel dados por la multiplicación $n \times n \times n$.

A partir de los símbolos de Christoffel la matriz de fuerzas centrifugas y de coriolis queda definida como la suma de estos por las velocidades angulares (36) y representado por un vector columna (37).

$$C_{ijk} = \sum_{i=1}^n \sum_{j=1}^n c_{ijk} \theta_i \theta_j \quad (36)$$

$$C = \begin{bmatrix} C_{ij1} \\ C_{ij2} \\ C_{ij3} \\ C_{ij4} \end{bmatrix} \quad (37)$$

4.5 Energía de Potencial

La energía de potencial es la energía contenida en un cuerpo determinada por su altura respecto del suelo [3]. Esta energía se expresa de la siguiente forma:



$$P = mgh \quad (38)$$

Siendo m la masa del cuerpo, g la aceleración de la gravedad igual a -9.81 m / seg^2 y h la altura del objeto. En un sistema robótico la energía potencial se encuentra concentrada en el centro de masa de sus eslabones, así que la altura es la proyección del eslabón sobre el eje vertical z del sistema base.

Esto aplicado al robot de cuatro grados de libertad que hemos analizado, quedaría expresado de la siguiente forma para cada eslabón:

$$P_1 = m_1 g l_{c1} \quad (39)$$

$$P_2 = m_2 g (l_1 + l_{c2} \sin \theta_2) \quad (40)$$

$$P_3 = m_3 g (l_1 + l_2 \sin \theta_2 + l_{c3} \sin(\theta_2 + \theta_3)) \quad (41)$$

$$P_4 = m_4 g (l_1 + l_2 \sin \theta_2 + l_3 \sin(\theta_2 + \theta_3) + l_{c4} \sin(\theta_2 + \theta_3 + \theta_4)) \quad (42)$$

Así es que la energía potencial total del robot es la suma de estas energías:

$$P = \sum_{i=1}^n P_i \quad (43)$$

A partir de estos términos, el vector de fuerza gravitacional G está conformado por:

$$G = \begin{bmatrix} \frac{\partial P}{\partial \theta_1} \\ \frac{\partial P}{\partial \theta_2} \\ \frac{\partial P}{\partial \theta_3} \\ \frac{\partial P}{\partial \theta_4} \end{bmatrix} \quad (44)$$

4.6 Modelo dinámico completo

Finalmente se puede armar por completo el modelo dinámico (14) conformado por las tres fuerzas que generan un par de torsión en las articulaciones el robot [1]. La matriz de inercia generaliza (33) que multiplicada por el vector de aceleración angular representa el torque provocado por la rotación de cada eslabón.

La matriz de fuerzas centrífugas y de Coriolis (37) abarca las fuerzas que el movimiento de un eslabón transmite a todos los demás. Por último el vector de fuerzas de gravedad (44), la cual expresa el torque causado por la gravedad en cada grado de libertad.



$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix} + \begin{bmatrix} c_{ij1} \\ c_{ij2} \\ c_{ij3} \\ c_{ij4} \end{bmatrix} + \begin{bmatrix} \frac{\partial P}{\partial \theta_1} \\ \frac{\partial P}{\partial \theta_2} \\ \frac{\partial P}{\partial \theta_3} \\ \frac{\partial P}{\partial \theta_4} \end{bmatrix} \quad (45)$$

5. Algoritmos de Enjambre

Los algoritmos de enjambre son algoritmos que incorporan elementos de adaptación y aprendizaje incorporando la toma de decisiones en forma social. El objetivo de estos se enfoca en encontrar el óptimo de un problema de optimización por medio de la comunicación entre sus agentes [4], entre algunos de ellos se encuentran el Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Pigeon-Inspired Optimization (PIO) y Bat Algorithm Inspired (BAT).

5.1 Particle Swarm Optimization.

Propuesto por James Kennedy y Russell Eberhart en 1995 [5]. Inspirado por el comportamiento y movimiento de algunos animales en la naturaleza tales como aves, insectos y peces.

El algoritmo inicia con un enjambre de partículas distribuidas de forma aleatoria en el área de búsqueda. Durante un predeterminado número de iteraciones N , estas partículas se mueven mejorando su posición y velocidad de manera autónoma y dinámica. Esto se lo hacen basándose en la experiencia propia y de todo el conjunto, cada partícula cuenta con una memoria donde guardan su mejor posición histórica $pBest$ y la mejor posición global del sistema $gBest$ de acuerdo al criterio establecido en la función objetivo f , que es el punto al que el enjambre debe llegar, o acercarse lo más posible, para lograr el objetivo. En términos generales los parámetros que necesita el algoritmo son:

- A Tamaño del enjambre.
- $pBest_i$ Mejor posición de la partícula x_i .
- v_i Velocidad de la partícula x_i .
- f Función objetivo.

La influencia que el $pBest$ y el $gBest$ tienen determinan hacia donde se moverá cada agente del enjambre. Al actualizar la dirección y velocidad todo el grupo se va acercando al objetivo (figura 3). La partícula que contenga el $gBest$ al final de todas las iteraciones N será tomada como la mejor solución jamás encontrada.

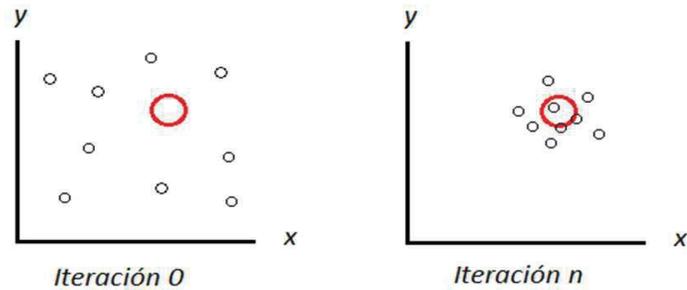


Figura 3. Comportamiento del enjambre de la iteración 0 a la iteración N.

Las reglas básicas para actualizar cada partícula son las siguientes:

$$v_i(k) = w(k)v_i(k-1) + c_1\phi(p_{Best} - x_i(k-1)) + c_2\phi(g_{Best} - x_i(k-1)) \quad (46)$$

$$x_i(k) = x_i(k-1) + v_i(k) \quad (47)$$

dónde:

- x_i Posición de la partícula.
- v_i Velocidad de la partícula x_i .
- c_1 Peso de la información local.
- c_2 Peso de la información global.
- ϕ Valor aleatorio entre 0 y 1.
- $pBest_i$ Mejor posición de la partícula x_i .
- $gBest$ Mejor posición global.
- w Factor de convergencia.

Factor de convergencia

Con el objetivo de acelerar el proceso del PSO se le agrega un factor χ [6]:

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad \text{con } \phi_1 + \phi_2 \geq 4 \quad (48)$$

dónde κ es una constante $0 \leq \kappa \leq 1$ que regula a c_1 , c_2 y $w(k)$ tal que:

$$\begin{aligned} w(k) &= \chi \\ c_1 &= \chi\phi_1 \\ c_2 &= \chi\phi_2 \end{aligned} \quad (49)$$

5.2 Artificial Bee Colony.



Como su nombre lo indica, está basado en los enjambre de abejas y el proceso que utilizan para encontrar y recolectar el néctar que necesitan. Como primer paso envían a las abejas exploradoras a buscar fuentes probables de néctar. Cuando encuentran una vuelven al panal para informar a sus compañeras mediante un baile donde les comunican la distancia, la dirección y la calidad de la fuente, a mayor calidad mayor será el número de abejas que acudirán al llamado [5].

Todo el proceso puede ser expresado en un pseudocódigo:

```

Inicio
while condición no completada do
    Fase abejas obreras
    Fase abejas espectadoras
    Fase abejas exploradoras
    Memorizar mejor solución
end while
    
```

Inicio. Al iniciar el algoritmo se crean fuentes de néctar de forma aleatoria $x_i, i=1,2,\dots,n$, las abejas exploradoras acuden a esas fuente a una velocidad:

$$v_i = x_i + \varphi (x_i - x_j) \tag{50}$$

Donde φ es un valor aleatorio $-1 \leq \varphi \leq 1$ y x_j es una fuente aleatoria diferente de x_i . Con la velocidad puede actualizarse la posición de la abeja para buscar el néctar:

$$x_i = x_i + v_i \tag{51}$$

Fase de abejas obreras. Estas abejas buscan en estas fuentes para evaluarlas de acuerdo a su calidad o fitness F_i . Este valor se calcula de acuerdo a la función objetivo $f(x_i)$:

$$F_i = \begin{cases} \frac{1}{1+f(x_i)} & \text{if } f(x_i) \geq 0 \\ 1+f(x_i) & \text{if } f(x_i) < 0 \end{cases} \tag{52}$$

Fase de abejas espectadoras. Después de recibir la información de las abejas obreras, las espectadoras recopila esos datos y los usan para calcular la probabilidad que tiene cada fuente de ser visitada de acuerdo a su *fitness*. Osea, la probabilidad $p(i)$:

$$p(i) = \frac{F_i}{\sum_{i=1}^n F_i} \tag{53}$$

Fase de abejas exploradoras. Estas abejas inician nuevamente el proceso buscando nuevas fuentes de néctar de acuerdo a las ecuaciones 50 y 51. Si después de un número predeterminado de iteraciones la fuente no ha sido visitada entonces se abandona y la abeja busca una nueva.

Mejor solución. Al final de cada ciclo de compara cada fuente en base a la función objetivo y se guarda aquella que tiene la mejor solución global.



5.3 Pigeon-Inspired Optimization.

Las palomas son aves muy comunes en gran parte del mundo, sobre todo en zonas urbanas. Durante la Segunda Guerra Mundial fueron de gran ayuda para los ejércitos por su capacidad de navegar en base al Sol, al campo magnético de la Tierra y otros puntos de referencia para encontrar su posición y su camino a casa.

En 2004, Tim Guilford desarrolló un modelo matemático para predecir en que momento las palomas cambiarán de un técnica a otra [7], El los definió como operadores:

Factor compás. Las palomas pueden usar el Sol como compás para dirigir su trayectoria, además sienten el magnetismo de la Tierra con lo cuál pueden trazar el mapa en su cabeza.

Puntos de referencia. Usan puntos de referencia ara ubicarse respecto a su destino, si no reconocen ningún punto entonces siguen a la paloma que sí lo hace.

Al iniciar el algoritmo se crean una serie de palomas virtuales con posición y velocidad aleatorios. Las posiciones y velocidades siguientes de cada paloma con respecto al modelo de Guilford [7] están expresadas como:

$$v_i = v_i e^{-Rt} + r(x_{best} - x_i) \quad (54)$$

$$x_i = x_i + v_i \quad (55)$$

dónde:

R es el factor compás $0 \leq R \leq 1$.

t es la iteración en la que se encuentra el algoritmo.

r número aleatorio $0 \leq r \leq 1$.

x_{best} es la paloma que representa la mejor solución global del problema en ese momento.

x_i es la paloma evaluada en ese instante.

Una vez que cada paloma actualizó su velocidad y posición se procede a seguir los puntos de referencia, que en el caso del algoritmo se trata de reducir el grupo a la mitad. Esto consiste en ordenar todas las palomas de acuerdo a la función objetivo, se queda entonces la mitad que tenga mejor costo.

$$N_p = \frac{N_p}{2} \quad (56)$$

Ahora se determina el centro del grupo x_c mediante la siguiente ecuación:

$$x_c = \frac{\sum x_i \text{fitness}(x_i)}{N_p \sum \text{fitness}(x_i)} \quad (57)$$

Con el centro definido se actualiza una vez más la posición de cada paloma:



$$x_i = x_i + r(x_c - x_i) \quad (58)$$

El término *fitness* como se explico anteriormente representa la calidad de la partícula, o en este caso, de la posición de la paloma. Si se trata de conseguir el óptimo mínimo:

$$fitness(x_i) = \frac{1}{f(x_i)} \quad (59)$$

En el caso de buscar el máximo óptimo, se toma simplemente como el costo de la función objetivo:

$$fitness(x_i) = f(x_i) \quad (60)$$

5.4 Bat Algorithm Inspired.

Los murciélagos utilizan la eco-localización para moverse evadiendo obstáculos y encontrar la mejor ruta hacia su presa, además de ubicarse perfectamente en la oscuridad. Estos murciélagos emiten un sonido de alta frecuencia, ellos escuchan el eco que se produce al chocar las ondas de sonido con los objetos que se encuentran alrededor.

En el algoritmo cada uno de los murciélagos empieza con una posición x_i y velocidad v_i aleatorios. El conjunto posee una frecuencia F_{min} , una amplitud de onda λ y un volumen A_0 . Tanto la amplitud como el volumen pueden ir variando de acuerdo a la proximidad que tiene el murciélago respecto su objetivo [8].

Todos los murciélagos ajustan su posición mediante la siguientes ecuaciones:

$$F_i = F_{min} + (F_{max} - F_{min})\beta \quad (61)$$

$$v_i = v_i + (x_i - x_{best})F_i \quad (62)$$

$$x_i = x_i + v_i \quad (63)$$

donde:

- β es un valor aleatorio $0 \leq \beta \leq 1$.
- x_{best} es el mejor valor global.
- F_{min} y F_{max} representan la frecuencia mínima y máxima.

Después de actualizarse, se crea una nueva solución a partir de la mejor solución global x_{best} en esa iteración:

$$x_{new} = x_{old} + \epsilon A^{(t)} \quad (64)$$

Siendo $A^{(t)}$ el promedio de los volúmenes de cada murciélago en esa iteración y ϵ un número aleatorio $0 \leq \epsilon \leq 1$. Esta ecuación permite al algoritmo intensificar la búsqueda local cuando



el murciélago se está acercando a su objetivo, justamente en las iteraciones finales. Es por eso que el radio de búsqueda r_i debe aumentar conforma avanza el algoritmo al mismo tiempo que ϵ y el volumen deben disminuir [8].

$$r_i(t+1) = r_i(0)(1 - e^{-\alpha t}) \quad (65)$$

Dónde $r_i(0)$ es un radio de pulso constante mayor a 0. Por otro lado el volumen disminuye usando la ecuación:

$$A_i^t = \alpha A_i^t \quad (66)$$

dónde α es una constante $0 \leq \alpha \leq 1$.

El algoritmo queda expresado en el siguiente pseudo-código donde al principio se crean los murciélagos con velocidad y posición aleatorias y se define la mejor solución global:

```

while condición no completada do
  for cada murciélago do
    Actualizar velocidad y posición}
    if rand >  $r_i$  then
      Buscar solución local  $x_{new}$  a partir de solución global
    end if
    If rand <  $A_i$  and  $f(x_i) < f(\bar{x})$  then
      Actualizar murciélago  $x_i$  a la posición  $x_{new}$ 
      Incrementar el radio de emisión  $r_i$ 
      Disminuir el volumen  $A_i$ 
    end if
    Definir la mejor solución global
  end for
end while
    
```

6. Proceso de Optimización

El proceso de optimización consiste en encontrar las longitudes óptimas para las cuatro articulaciones del Robot, tal que a partir de dos puntos dados $\{x_i, y_i, z_i\}$ y $\{x_f, y_f, z_f\}$, sea capaz de realizar una trayectoria lineal produciendo el mínimo desplazamiento angular para sus articulaciones y el menor torque en su trayectoria. Este proceso se puede describir a través del siguiente pseudocódigo:

```

Especificar dimensiones iniciales del Robot a Optimizar
Definir punto inicial y final de la trayectoria LIN
    
```



```

while it < itmax do
  for cada posición intermedia  $j$  do
    Definir siguiente punto  $\{x^{j+1}, y^{j+1}, z^{j+1}\}$  de la trayectoria
    Calcular la siguiente posición angular  $\{\theta_1^{j+1}, \theta_2^{j+1}, \theta_3^{j+1}, \theta_4^{j+1}\}$ 
    Calcular Torques Dinámico  $\tau_1, \tau_2, \tau_3, \tau_4$ 
    Calcular Incrementos Angulares  $|\theta_i^{j+1} - \theta_i^j|$ 
  end for
  Calcular el promedio del Torque Dinámico y Movimiento Angular en trayectoria LIN
  Estimar la función costo
  Búsqueda de óptimo por algoritmos de enjambre
end while
    
```

6.1 Costo asociado al movimiento angular en trayectoria LIN.

El torque dinámico de un Robot se ve afectado por las velocidades y aceleraciones efectuadas en una trayectoria, por lo que si se reduce la cantidad de movimiento ejercida en una trayectoria se contribuye a la disminución de esfuerzos dinámicos. Por tanto la medición del incremento angular promedio en una trayectoria LIN dada puede definirse como:

$$C_{\theta} = \frac{1}{4} \sum_{i=1}^4 \sum_{j=1}^{20} |\theta_i^{j+1} - \theta_i^j| \quad (67)$$

6.2 Costo asociado al Torque generado en trayectoria LIN.

Para estimar la cantidad de Torque C_{τ} ejercida en una trayectoria LIN se considero obtener el Torque dinámico para cada articulación en cada una de 20 posiciones intermedias para finalmente promediarlo.

$$C_{\tau} = \frac{1}{4} \sum_{i=1}^4 \sum_{j=1}^{20} \tau_i^j \quad (68)$$

6.3 Función Objetivo.

La función objetivo tiene el fin de evaluar la contribución de cada coste asociado, que para este caso nos interesa descubrir las mínimas dimensiones de las articulaciones del Robot que puedan efectuar una trayectoria LIN dada, produciendo la mínima cantidad de movimiento angular y torques dinámicos, es decir:

$$\operatorname{argmin}_l C(l, x_i, x_f) = C_{\theta} (1 + C_{\tau}) \quad (69)$$



7. Resultados

El proceso de optimización se ejecutó 100 veces para cada algoritmo, 20 agentes y con 1000 iteraciones, mostrando la evolución de cada algoritmo tal como lo muestra la figura 4, en esta se aprecia que a pesar de que el algoritmo PIO presentó una etapa de estancamiento llegó a un mejor costo que los demás. También se puede apreciar de como el PSO llega en muy pocas iteraciones a un óptimo muy bueno. En la tabla 2 se muestra el mejor costo encontrado, costo promedio y su desviación estándar tras 100 ejecuciones para cada algoritmo.

Tabla 2. Comparativa de Algoritmos de Enjambre.

Algoritmo	Mejor Costo	Costo Promedio	Desviación Estándar
PSO	1.8853	2.2201	0.1619
ABC	1.8820	1.9788	0.0576
BAT	2.0005	2.2784	0.1232
PIO	1.8760	1.8999	0.0114

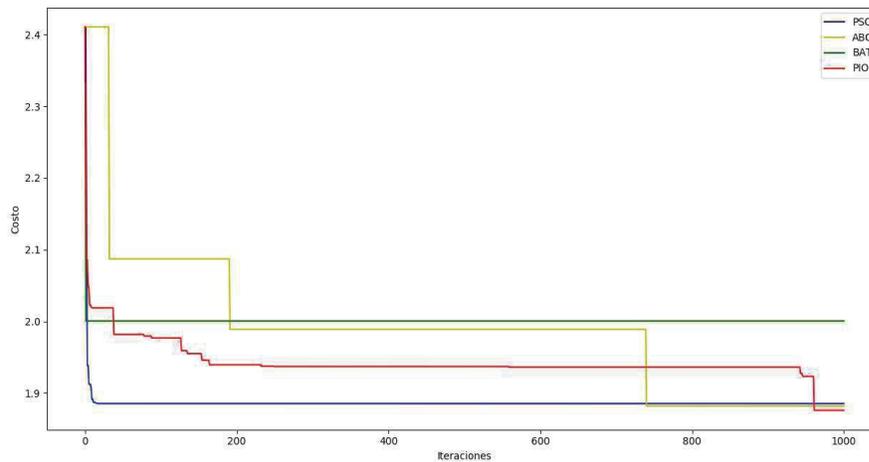


Figura 4. Evolución del Costo tras 1000 iteraciones.

Las dimensiones iniciales con las que se partieron para el diseño fueron $l_1 = 0.1 m$, $l_2 = 0.15 m$, $l_3 = 0.15 m$, $l_4 = 0.08 m$ (figura 5a) donde el algoritmo PIO con un costo promedio de 1.876 logró optimizarlas para ejecutar la misma trayectoria inicial hasta $l_1 = 0.0081 m$, $l_2 = 0.2251 m$, $l_3 = 0.2239 m$, $l_4 = 0.0011 m$ (figura 5b). Como se puede apreciar esta reducción implica que el mismo trabajo efectuado con un Robot de 4GDL pudo haber sido efectuado con uno de 3GDL con la misma capacidad de generación de trayectoria LIN.

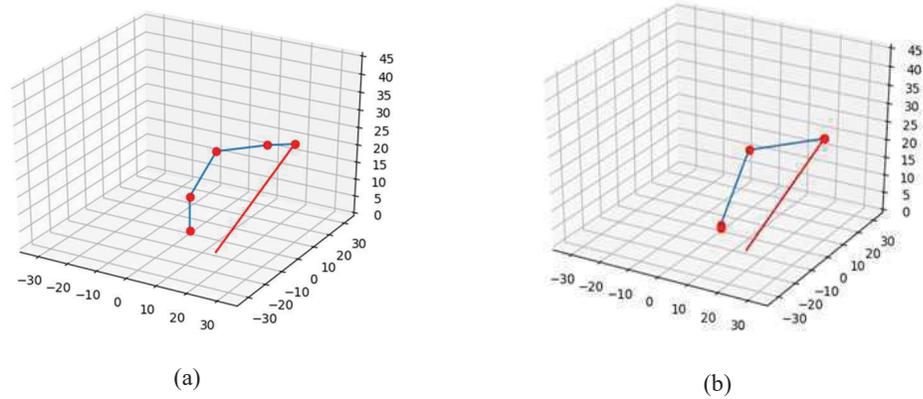
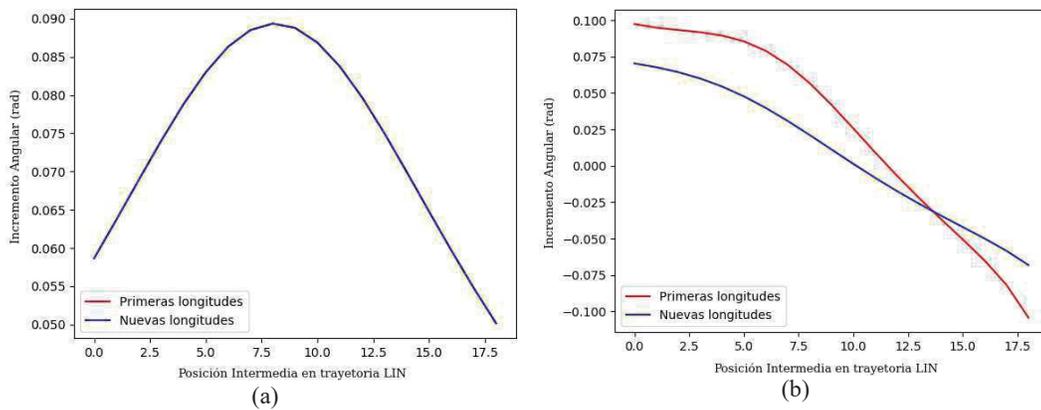


Figura 5. Optimización de dimensiones de Brazo Robótico, (a) Dimensiones iniciales, (b) Dimensiones Óptimas.

En la figura 6 se muestra una comparativa de los incrementos angulares tras pasar por los 20 puntos intermedios que conforman una trayectoria LIN, notando una reducción usando las dimensiones optimizadas. Las figuras 6a, 6b, 6c y 6d muestran la disminución del incremento angular para las articulaciones θ_1 , θ_2 , θ_3 y θ_4 respectivamente.



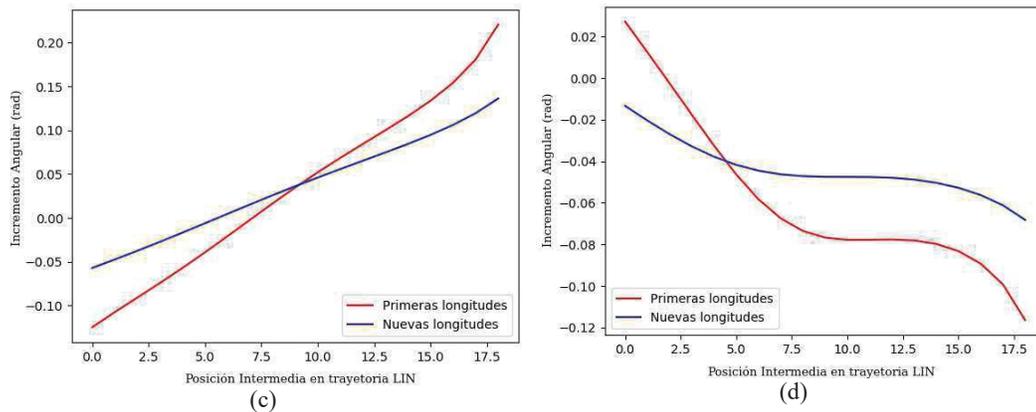
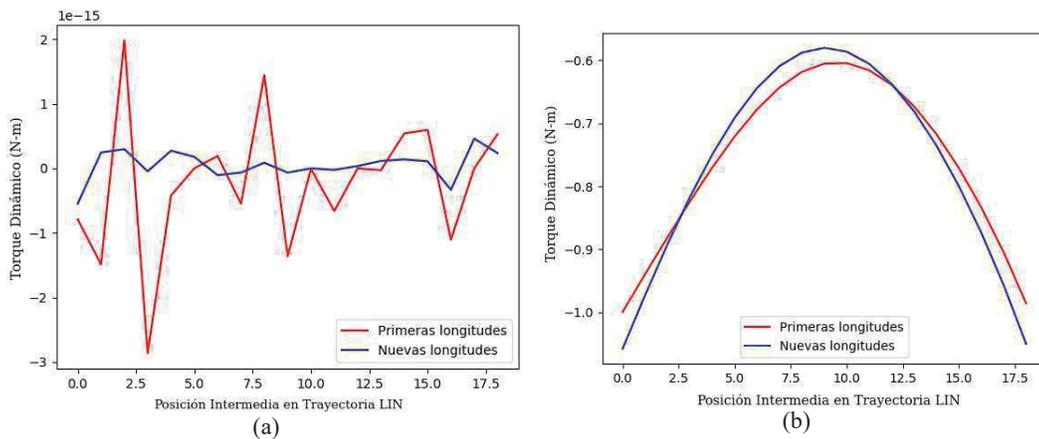


Figura 6. Incremento Angular para ejercer una trayectoria LIN, (a) Respecto a θ_1 , (b) Respecto a θ_2 , (c) Respecto a θ_3 , (d) Respecto a θ_4 .

En la figura 7 se muestra también la disminución del Torque asociado al movimiento del Robot en la trayectoria LIN en cada las articulaciones. Mientras que para τ_1 hubo una reducción considerable (figura 7a), para τ_2 y τ_3 se presentó una reducción gradual (figuras 7b y 7c). También se puede apreciar que al disminuir el tamaño de la articulación 4 considerablemente τ_4 se ve reducido radicalmente (figura 7d).



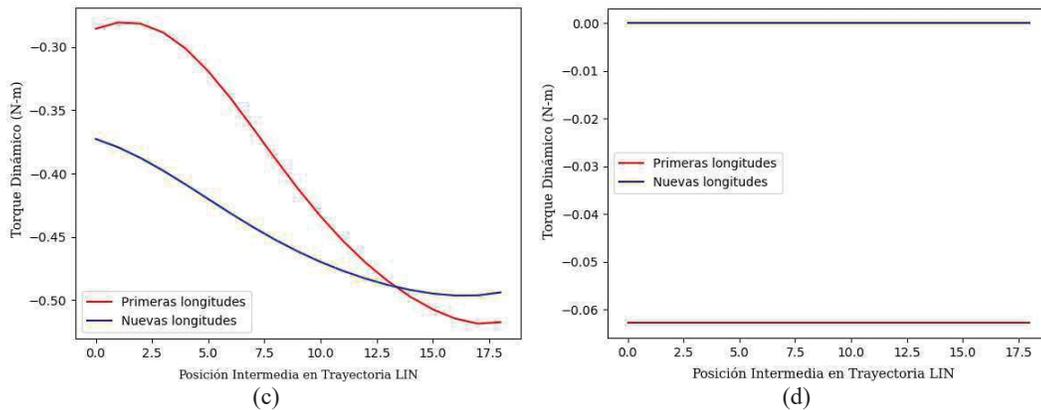


Figura 7. Torque dinámico al ejercer una trayectoria LIN, (a) Respecto a θ_1 , (b) Respecto a θ_2 , (c) Respecto a θ_3 , (d) Respecto a θ_4 .

8. Conclusiones

Los algoritmos de enjambres de partículas representan una valiosa herramienta para la optimización, debido a su flexibilidad han podido adaptarse a cualquier ambiente en tiempo real, lo que ha permitido a ingenieros y diseñadores alrededor del mundo mejorar sus creaciones según las necesidades del entorno en que se desarrollarán.

Aplicar la optimización al mundo del diseño logró abrir una nueva área de oportunidad donde ahora no solo importa la apariencia estética de un objeto o proceso, sino que cada una de sus propiedades se encuentre con las mejores dimensiones, cantidades y ubicaciones posibles, logrando así un mayor aprovechamiento de los recursos disponibles.

A lo largo de este trabajo se presentó la metodología seguida para optimizar las longitudes de los eslabones de un robot, logrando de este modo que el torque de cada articulación disminuyera en función de la trayectoria que el efecto final debe seguir en su rutina de trabajo.

El método conocido como Pigeon-Inspired Algorithm también demostró poseer características satisfactorias en cuanto a las pruebas a las que fue sometido, teniendo un promedio de costo suficientemente bajo además de una desviación estándar pequeña, lo que aumenta las posibilidades de recrear los resultados, esto le da al algoritmo repetibilidad y puede aumentar la confianza que se le tiene al momento de optimizar otras propiedades, no solo dentro de la robótica sino en cualquier campo de trabajo.

Referencias

- [1] Barrientos A. "Fundamentos de Robótica", Mc-Graw Hill, 1997.
- [2] Kumar S. "Introducción a la Robótica", Mc-Graw Hill, México, 2010.
- [3] Pérez M., Cuevas E. y Zaldívar D. "Fundamentos de Robótica y Mecatrónica con Matlab y Simulink", Alphaomega, 2014.
- [4] Rojas A. "Topologías de Optimización por Enjambre de Partículas basadas en Información Mutua", Centro de Investigación en Matemáticas A.C., 2015.
- [5] Hu Y. "Swarm Intelligence", 2017.
<http://docplayer.net/44658517-Swarm-intelligence-yichen-hu.html>



- [6] Méndez H., Flores V., Solano M. "*Descomposición de Trayectoria Óptima para Robots Mviles Generada por PSO y Descriptores de Forma*", Ingeniería y automatización, Capítulo 3, Asociación Mexicana de Mecatrónica A.C., México, 2018.
- [7] Duan, Haibin y Qiao Peixin. "*Pigeon-Inspired Optimization: a new swarm intelligence optimizer for air robot path planning*". Beijing University of Aeronautics and Astronautics, (2014).
- [8] Olivares R. "*Enumeración adaptativa y rankeadores de estrategias basados en algoritmo BAT*", Pontificia Universidad Católica de Valparaiso, 2015.



Diseño de una Máquina que Cocina Huevos

González Valdivia Antonio de Jesús, Alatorre González Cristian Daniel y Aguilar Molina Yehoshua.

Universidad de Guadalajara
Centro Universitario de los Valles
Correo: antonio.glez.1000@gmail.com

Resumen

El presente documento describe a un proyecto con una alternativa radical e innovadora en donde podemos hacer que todas las personas que viven día a día trabajando por las mañanas hasta altas horas de la noche puedan tener su desayuno listo solo para comer, y esto con solo programar su máquina que realizara la acción de cocinar su huevo a su gusto y en la hora y tiempo específico, describe una máquina que pueda cocinar huevos de forma automatizada y sin necesidad de la intervención humana, así como una alternativa de cocina moderna y accesible para todas las personas y en grandes y pequeñas comunidades, así como de describe el proceso por el cual fue inspirado y creado el proyecto así como los materiales utilizados.

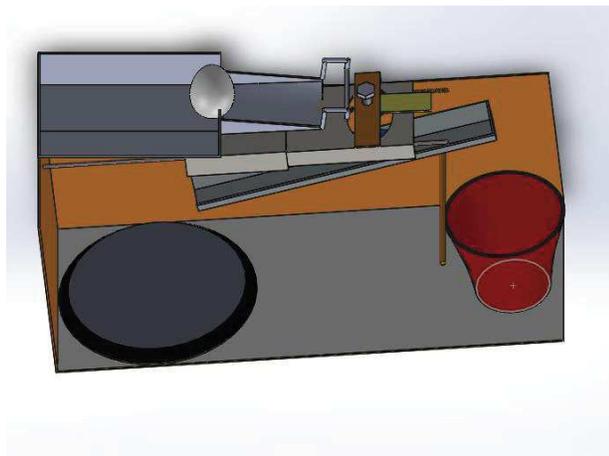


Figura 1. Principio

Palabras claves: Huevo, mecánico, temporizado, desarrollado, vanguardista, inteligente, barato e interactivo.



1. Introducción

Se desarrolla una maquina en donde tendremos como materia prima el huevo de gallina y que posteriormente el huevo pasara a ser cocinado por medio de la inducción o gas (que son los métodos por los cuales se calentara para poder cocinar) esto dependiendo del gusto del cliente que adquiera la máquina, en fin el huevo se cocinara en dos etapas la de cocción que es cuando se quiebra y se coloca en el lugar necesario para poder comenzar a cocinarlo y la de proceso de presentación es aquí donde se coloca en el plato con los ingredientes que son al gusto previamente programados.

Todas las personas cuando se levantan antes e ir a trabajar siempre se tienen que levantar un poco de tiempo más temprano para cocinar su alimento que son los más importantes del día que es el desayuno, ahora con esta gran innovación la persona que adquiera este producto solo bastara con programar a una hora exacta del día, para que el huevo sea cocinado y solo se preocupe por comerlo y no por cocinarlo.

“Todos necesitamos descansar solo tenemos que escoger que aparato nos puede ayudar a tener mejor desempeño con la menor cantidad de tiempo perdido en cosas que pueden obtenerse de otra manera”.

2. Diseño de una máquina que cocine huevos

Se realizará el prototipo de una máquina que prepare huevos con un sistema de control para las variables de temperatura y humedad. Con esto obtendremos las condiciones favorables para el huevo al cocinar y así conseguir el mayor nivel de producción en la máquina.

Se propone crear un sistema que tendrá una serie de sensores y actuadores para que pueda hacerse efectivo la manipulación del medio donde se estará trabajando cuando se cocine el huevo y finalmente su consumo.

Los sensores se encargaran de recoger información de control para que no se queme el huevo: temperatura y de humedad. Con estos parámetros tendremos que controlarlos para que el huevo termine con el término deseado (bien cocido, término medio o tres cuartos).

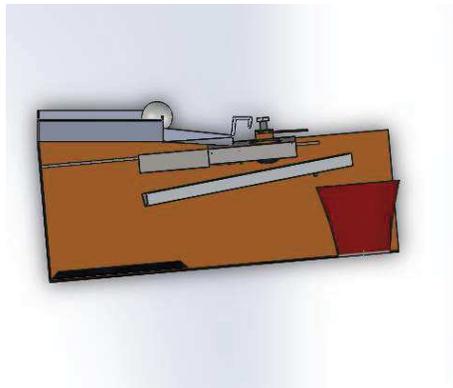


Figura 2. Diseño de maquina vista lateral



Una serie de actuadores que serán los encargados de cocinar y mejorar las condiciones de la maquina cuando trabaje.

2.1 *Funcionamiento*

Tenemos una canasta o bandeja que es la que se muestra en la figura 2 en la parte superior es allí donde el huevo se colocara y posterior mente se dirigirá (por gravedad el huevo se moverá) llegando a una par de engranes colocados horizontalmente que serán los encargado de quebrar por mitad el huevo dejando salir todo el “liquido” del mismo y este líquido se dirigirá al platillo (por gravedad el líquido se moverá) colocado en la parte inferior de la maquina es aquí donde se cocinara de acuerdo a las especificaciones previamente programadas y con un motor de corriente directa el platillo de metal se estará moviendo para así poder obtener una mejor cocción del huevo.

Cabe mencionar que la cascara de los huevos cuando se rompen para que salga el líquido se colocaran en un vaso (figura 2 esta de color rojo) es donde se depositaran los desperdicios y cuando proceso este por terminar un timbre estará avisando el fin del proceso para que el usuario pueda pasar a recoger su desayuno.

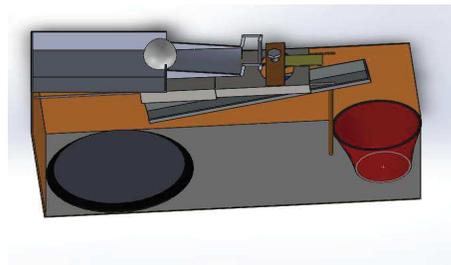


Figura 3. Vista lateral superior

2.2 *Arduino UNO*

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos.



Figura 4. Arduino UNO



2.3 *Motor de corriente directa*

El motor de corriente continua (motor DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio. En algunas modificaciones, ejercen tracción sobre un riel. Estos motores se conocen como motores lineales.

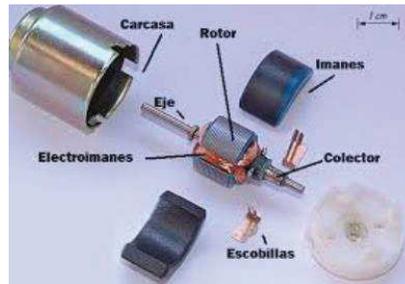


Figura 5. Partes de motor de DC

2.4 *Engranajes impresos en 3D*

Están serán las piezas encargadas de quebrar el huevo para que el líquido salga así mismo serán los encargados de que colocar el casaron en el vaso de desechos.

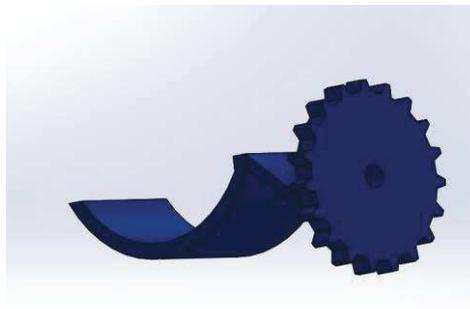


Figura 6. Diseño de engranes de presión

2.5 *Sensor de temperatura*

Es un sensor analógico que está calibrado directamente en grados Celsius, a diferencia de otros dispositivos electrónicos como los termistores en donde la temperatura se determina por la variación de su resistencia eléctrica, el dispositivo LM35 proporciona una salida de voltaje proporcional a la temperatura.

Cuenta con una precisión garantizada de 0.5°C a 25°C, su rango de medición abarca desde -55°C hasta 150°C. El dispositivo LM35 tiene un comportamiento lineal con la temperatura y su razón de incremento es de 10mV por cada grado centígrado.

La baja impedancia de salida que presenta, así como su salida lineal y su precisa calibración hacen posible que este circuito integrado sea fácil de utilizar en nuestro circuito de control. Una ventaja que debemos considerar es que el consumo de corriente es muy bajo (aproximadamente 60uA) y por ende produce un efecto de calentamiento muy reducido.

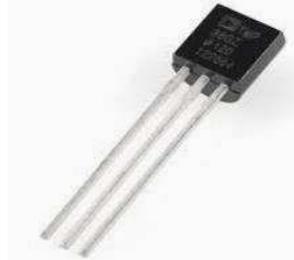


Figura 7. Sensor de temperatura

2.6 Sensor de humedad

El sensor DHT11 que nos permite medir la temperatura y humedad con Arduino. Una de las ventajas que nos ofrece el DHT11, además de medir la temperatura y la humedad, es que es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviarnos la información y por lo tanto, estaremos más protegidos frente al ruido.

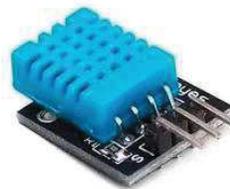


Figura 8. Sensor de humedad

2.7 Display de comandos

Los displays LCD son elementos muy útiles en la mayoría de proyectos para poder visualizar información sin tener conectado a nuestro Arduino un ordenador, es por ello que vamos a hacer un pequeño repaso a los diferentes tipos de displays LCD que podemos encontrar y que son 100% compatibles con Arduino. El conocer diferentes displays y sus características nos será muy útil a la hora de decidir que display usaremos en nuestros proyectos.





Figura 9. Pantalla LCD

2.8 Metal de estructura exterior

El inoxidable austenítico más popular es el Tipo 304, que contiene básicamente 18% de cromo y 8% de níquel, con un tenor de carbono limitado a un máximo de 0,08%. Tiene gran aplicación en las industrias químicas, farmacéuticas, de alcohol, aeronáutica, naval, uso en arquitectura, alimenticia, y de transporte. Es también utilizado en cubiertos, vajillas, piletas, revestimientos de ascensores y en un sin número de aplicaciones.

En determinados medios, especialmente en aquellos que contienen iones cloruro, el inoxidable 304 muestra propensión a una forma de corrosión llamada corrosión por picado.

Es un tipo de corrosión extraordinariamente localizada, en la cual en determinados puntos de la superficie del material, el medio agresivo consigue quebrar la película pasiva para después progresar en profundidad. El crecimiento de los picados se da en un proceso auto catalítico y aunque la pérdida de masa pueda ser a veces insignificante, esta forma de corrosión es muy insidiosa, ya que muchas veces un picado es suficiente para dejar un equipo fuera de servicio.

La corrosión por rendijas, puede ser considerada como una corrosión por picado artificial. El aspecto es frecuentemente semejante al de la corrosión por picado y el proceso de crecimiento es también auto catalítico. Pero, la existencia de una rendija es necesaria para la ocurrencia del fenómeno, lo que no sucede en la corrosión por picado.

Los mismos medios capaces de provocar la corrosión por picado, promueven la corrosión por rendijas en los aceros inoxidables.



Figura 10. Acero Tipo 304

Cuando hablamos del material que se va a utilizar también tenemos que tener en cuenta que tipo de soldadura vamos a utilizar en este caso el que será utilizado es la soldadura TIG.

El proceso de soldadura por arco bajo la protección de gas con electrodo (no consumible), también llamado TIG (Tungsten Inert Gas), usa un arco eléctrico como fuente de energía que se establece entre el electrodo no consumible y la pieza a soldar con la envoltura protectora del gas



inerte. Cuando se usa material de relleno, éste se proporciona mediante el uso de varillas, de la misma forma que en la soldadura de oxiacetileno.

Durante el proceso, la antorcha TIG debe estar conectada al polo negativo (-) y la pinza de masa al polo positivo (+).

El proceso TIG se puede utilizar para la soldadura de todos los materiales, incluidos el aluminio el magnesio y los materiales sensibles a la oxidación como el titanio.

Se trata de un proceso mayoritariamente manual con tasa de deposición menor comparado con otros procesos por lo que es utilizado para aplicaciones donde se precisa un acabado visualmente perfecto.

La posibilidad de soldar un material u otro, dependerá de las características del equipo de que dispongamos. El proceso TIG puede utilizarse tanto con corriente continua (DC) como con corriente alterna (AC). La elección de la clase de corriente y polaridad se hará en función del material a soldar. Para ello tenemos la soldadura con corriente continua y la soldadura con corriente alterna, estos dos tipos se estarán describiendo a continuación:

TIG corriente continua (DC): Suministran corriente continua (DC) cualquiera de los equipos INVERTER de SOLTER por lo que se puede soldar perfectamente con cualquiera de ellos mediante el procedimiento TIG. Estos equipos se utilizan mayoritariamente para el soldeo de los materiales ACERO y ACERO INOXIDABLE. Utilizando equipos INVERTER convencionales el cebado del arco será mediante contacto con la pieza a soldar y la apertura del gas será manual. Existen también equipos SOLTER de TIG profesionales en los que el cebado del arco se realiza sin contacto y mediante alta frecuencia (HF) y el gas se abre automáticamente mediante electro-válvula

TIG corriente alterna (AC): Suministran corriente alterna (AC) para soldadura TIG equipos SOLTER INVERTER específicos ya que además es preciso que estos mantengan la alta frecuencia (HF) en continuo funcionamiento para evitar la extinción de arco durante el proceso.

Estos equipos suministran normalmente los dos tipos de corriente; AC y DC; por lo que es posible soldar los materiales ALUMINIO, TITANIO... con AC, además de los anteriormente descritos en DC.

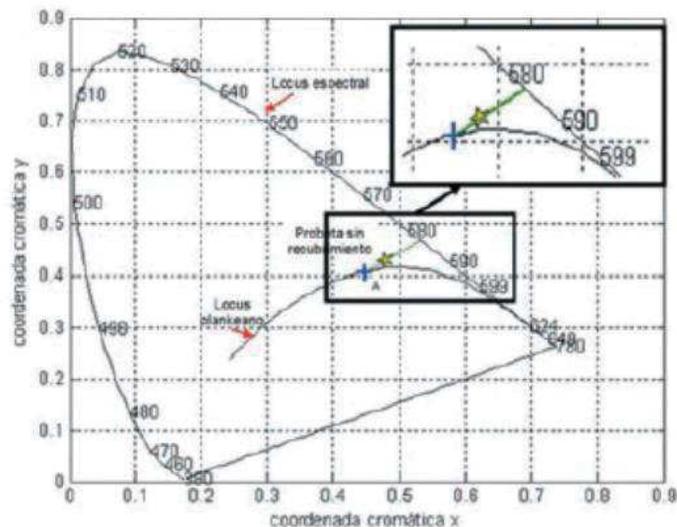




Figura 11. Diagrama de cromática para acero Tipo 304

2.9 Propiedades del filamento de impresora

En el ámbito de la impresión 3D para principiantes, el ácido poliláctico (PLA) es el más importante. Aunque a menudo se compara con el filamento ABS (el segundo filamento 3D más usado), el filamento PLA es con diferencia el filamento 3D más popular, y con razón.

Para empezar, es fácil de extruir. Se puede extruir a temperaturas más bajas que el filamento ABS y no se deforma con la misma facilidad. En otras palabras, no necesitas una cama de impresión (aunque ayuda a obtener mejores impresiones 3D). Otro aspecto positivo: el PLA no emite mal olor durante la impresión 3D. En general se considera un filamento 3D sin olor, pero muchos afirman que tiene un olor dulce.

Como termoplástico biodegradable, el PLA es más ecológico que la mayoría de los filamentos 3D. Deriva de recursos renovables como el almidón de maíz o la caña de azúcar.

Al igual que sucede con el filamento ABS, el PLA es un material de consumo básico utilizado para muchos filamentos exóticos o lúdicos, como los filamentos conductores de la electricidad, los que brillan en la oscuridad o los que incorporan madera o metal.

Para obtener más información acerca de las diferencias entre los filamentos PLA y ABS consulta el siguiente artículo PLA vs ABS: comparación de filamentos para impresión 3D.



Figura 12. Filamento de impresora

→ Dureza: Alta | Flexibilidad: Baja | Durabilidad: Media: esto con la finalidad de cuando sea utilizado para imprimir las piezas y estas sean impresas tengan una dureza alta y resistente a algunas fuerzas exteriores que se le pueda aplicar a la pieza.

→ Dificultad de uso: Baja: Con la finalidad de que la impresora como el usuario puedan utilizar el filamento sin problema, con lo que concierne al usuario es cuando se vaya a instalar.

→ Temperatura de impresión 180 °C – 230 °C : son las temperaturas que maneja la impresora para poder tener el filamento en estado líquido.



- Temperatura de la cama de impresión: 20 °C – 60 °C (aunque no necesaria)
- Contracción/deformación: Mínima
- Soluble: No
- Seguridad alimentaria: Consulta las especificaciones del fabricante

3 Formas de cocinar el huevo

Hoy en día existen muchas formas de cocinar el huevo pero tenemos que tener en cuenta el grande proceso que el preparar un huevo puede conllevar y la gran dificultad tanto manual como la mecánica pero la diferencia de la mecánica es que solamente programaremos como queremos el huevo y después procederemos a comerlo, en los siguientes dos apartaos tenemos los pasos para cocinarlo de forma manual y después una muy breve explicación de cómo cocinara la maquina el huevo y como llevara el control de temperatura.

3.1 Formas de cocinar el huevo (forma manual)

Huevos fritos:

Para la mayoría de personas, el huevo frito perfecto tiene la yema líquida y la clara cuajada. Hay dos formas de conseguirlo: dejar la yema hacia arriba mientras se fríe e ir bañándole con el aceite o dándole la vuelta con cuidado cuando esté medio hecha.

Este segundo método es menos popular porque es fácil que la yema se rompa al dar la vuelta al huevo porque pierde el tono amarillo fuerte que le caracteriza.

Huevos revueltos:

Consiste en cocinarlos a fuego lento y removeros con paciencia todo el tiempo. Nunca intentar hacerlos con prisa pues quedarán duros y grumosos. Para dos personas utiliza 4 huevos, 2 cucharadas de nata y salpimienta al gusto. Bate los huevos junto a la nata y la salpimienta durante un minuto.

Cubre la base de la sartén con aceite de oliva y cuando esté caliente, incorpora la mezcla. Remueve constantemente a fuego lento de 5 a 8 minutos. Retira de la sartén y sigue removiendo un par de minutos.

3.2 Formas de cocinar el huevo (maquina)

Huevos fritos:

En este apartado se colocara el huevo sobre el platillo metálico para poder evitar que se revuelva el huevo y con el motor cambiando de sentido para que la cocción sea uniforme. En esta parte los engranes son los encargados de quebrar el huevo para que este caiga de manera uniforme en el platillo después de esto el control de temperatura y humedad será el encargado de precisar el término del huevo.



Huevos revueltos:

En este apartado se colocara el huevo sobre el platillo metálico antes de esto la yema como la clara del huevo rodara por el camino que tendrá como final el platillo metálico y con el motor cambiando de sentido para que la cocción sea uniforme.

En esta parte los engranes son los encargados de quebrar el huevo igual que en el huevo frito.

4 Resultados

Un sistema (maquina) en el que se puede cocinar cualquier tipo de huevo (principalmente el de gallina) con cualquier terminación y en cualquier tiempo, claro con un margen de producción y con cierta calidad dependiendo la urgencia del mismo.

Tenemos una máquina que tiene como propósito el desarrollar una ambiente amigable con el usuario cuando este quiera hacer su desayuno, hablando más claro el huevo y que el usuario no le tome mucho tiempo en usar la máquina, sino tan solo un par de minutos para especificar las necesidades o requerimientos.

Es decir, se persigue la posibilidad de un mercado con alcances en todos los niveles, ya que todas las personas que trabajan siempre desean comer aunque sea un huevo a la semana y esta máquina ayuda a que no pierdan mucho tiempo cocinando ni preocupándose por cocinar el mismo. La máquina pueda hacer el trabajo por nosotros sin necesidad de estar pendiente del proceso de producción del huevo.

Resultado final es una maquina encargado de cocinar huevos con diferentes términos y condiciones que el usuario previamente ha establecido.



Figura 13. Toma de medidas en acero 304



Figura 14. Soldadura realizada

4.1 Tablas.

Tabla 1. Programa de avance de proyecto por semanas.

Etapas/Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Diseño CAD															
Selección de materiales															
Selección de actuadores y sensores															
Fabricar estructura del prototipo															
Fabricar componentes															
Ensamblaje del prototipo															
Diseño de electrónica/eléctrica															
Cableado de dispositivos															
Construcción de PCB															



Diseño de sistema de control															
Diseño de interfaz hombre-máquina															
Pruebas y ajustes															

5 Conclusiones

La máquina está en proceso de fabricación, en pocas palabras tenemos el diseño en 3D pero aún está en proceso de fabricación todas sus piezas, ya que lo que se pretende es hacerlo con materiales altamente resistentes esto para poder evitar que se pueda dañar en corto plazo, ya que es muy importante que sea duradero y también muy amigable con el medio ambiente por tanto el material que se utilizara va a ser plástico para algunas piezas que se imprimirán en 3D y otras de metal que son las partes del diseño exterior o del casaron de la máquina para que no tenga algún problema con algún liquido o golpe superficial.

De esta manera no afecte su funcionamiento ya que para esto es muy importante que cuide la estructura de forma integral.

Se pretende una máquina que pudo usarse por todas las edades, claro que sean mayores a 6 años para que así se pueda evitar algún accidente o incidente y no poner en riesgo la seguridad física del usuario, por tanto esta máquina es y será una herramienta muy importante en la cocina matutina.

Se espera que el producto final termine de la siguiente manera:

Como huevo estrellado se pretende que esté listo después de que sea cocinado.



Figura 15. Huevo estrellado esperado



Como huevo revuelto se pretende que esté listo después de que sea cocinado.



Figura 16. Huevo revuelto esperado.

Referencias

- [1]"Breve historia del huevo / Conoce lo que comes / Enciclopedia de los Alimentos / Huevo", *Alimentacion.es*, 2019. [Online]. Recuperado de: http://www.alimentacion.es/es/conoce_lo_que_comes/bloc/huevo/breve-historia-del-huevo/.
- [2]"Arduino - Home", *Arduino.cc*, 2019. [Online]. Recuperado de: <https://www.arduino.cc/>.
- [3]"¿Qué es LabVIEW? - National Instruments", *Ni.com*, 2019. [Online]. Recuperado de: http://www.ni.com/es-mx/shop/labview.html?cid=Paid_Search-135945-Rest_of_Latam-Google_ESW1_LabVIEW_Broad&s_kwcid=AL!6304!3!243941345035!b!!s!!%2Blabview&gclid=CjwKCAjwnrjrBRAMEiwAXsCc4xb_YhgawglPYUJ5bfcx95uw2kh8LuywQ_hA4wgsygh8DBfYCHVV_RoCOQAQAvD_BwE.
- [4]"Soldadura TIG", *SOLTER Soldadura*, 2019. [Online]. Recuperado de: <https://www.solter.com/es/procesos-soldadura/tig>.
- [5]R. Diosdado, "Zona Maker - Tipos de LCD para Arduino", *Zonamaker.com*, 2019. [Online]. Recuperado de: <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/tipos-de-lcd-para-arduino>.
- [6]"Los 25 mejores filamentos para impresoras 3D - Guía 2019 | All3DP", *All3DP*, 2019. [Online]. Recuperado de: <https://all3dp.com/es/1/filamento-3d-filamento-impresora-3d/>.



Inclusión de la Dinámica de los Actuadores y Manejadores de Potencia en el Modelo de un Robot Móvil Omnidireccional

Correa Fernández Erick Aarón ¹, Serrano Pérez Omar ^{1*}, Rodríguez Molina Alejandro ¹,
Corona Ramírez Leonel Germán ² y Villarreal Cervantes Miguel Gabriel ¹.

¹ Departamento de Posgrado, sección de Mecatrónica, Laboratorio de Diseño Mecatrónico Óptimo, CIDETEC, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, 07700, CDMX, México.

² Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional, Av. Instituto Politécnico Nacional No. 2580, Col Barrio la Laguna Ticomán, Gustavo A. Madero, 07340, CDMX, México
e-mail: *omarserrano95@gmail.com

Resumen

En este trabajo se presenta el análisis cinemático y dinámico de un robot móvil omnidireccional (RMO), en el cual, se emplea la formulación de Euler-Lagrange para obtener la dinámica de la parte mecánica y se realiza un acoplamiento dinámico para incluir la dinámica de los motores y la dinámica de los manejadores de potencia del prototipo real de laboratorio. Para estimar el comportamiento de la dinámica de los manejadores de corriente se consideró que su comportamiento se rige a través de un controlador Proporcional-Derivativo (PI) y la dinámica de los actuadores corresponde a la del motor de corriente directa. El modelo dinámico propuesto se valida a través de resultados en simulación y experimentación, mostrando un comportamiento similar bajo ciertas condiciones de operación.

Palabras clave: Robot móvil omnidireccional, modelo cinemático, modelo dinámico, dinámica del actuador, dinámica de manejadores de corriente.

1. Introducción

En la actualidad el uso de robots móviles terrestres es esencial para alcanzar la movilidad sobre superficies irregulares y/o de difícil acceso para el ser humano, donde su gran ventaja radica en conseguir movilidad en entornos con limitaciones de espacio o en presencia de obstáculos peligrosos para realizar diversas tareas cotidianas [1]. Por esta razón, se han desarrollado varios trabajos de investigación que muestran una gran variedad de mecanismos para conseguir el desplazamiento en superficies irregulares y diversas propuestas de configuraciones de locomoción, con lo cual se mejora la maniobrabilidad y la precisión de dirección en los robots móviles terrestres [2], [3].

Una de las propuestas que logra un buen desempeño en el contexto de los robots móviles, es la plataforma triangular equipada con tres ruedas omnidireccionales, éstas permiten el control independiente del movimiento del robot por las direcciones de rotación longitudinal de las llantas [4], dotando al robot la capacidad de cambiar simultáneamente su posición lineal y angular continuamente. Esta ventaja se debe a que la rueda se acopla directamente con el actuador montado en la plataforma, logrando así reducir el espacio y el tiempo requerido para el movimiento. Sin embargo, sus grandes inconvenientes son la baja capacidad de carga y la sensibilidad a la vibración



vertical que se produce en las llantas cuando los rodillos tocan el suelo. Este tipo de robots se denomina en la literatura como robot móvil omnidireccional [5], [6].

Por otra parte, para obtener un buen desempeño con el RMO al realizar tareas de seguimiento y regulación para distintas aplicaciones cotidianas, es importante garantizar una buena eficiencia en sus movimientos a través de un adecuado diseño de su sistema de control. Existen diversos trabajos sobre el uso de los robots móviles en tareas de reconocimiento espacial o terrestre [7], en la cosecha de vegetales [8], en su uso para sillas de ruedas eléctricas [9], en tareas cooperativas [10], tareas de detección y localización de minas [11], entre otros usos. De manera general, dichos trabajos mencionan que para mejorar el rendimiento de los sistemas de control en los robots móviles, es necesario diseñar controladores que puedan garantizar estabilidad. Para lograr lo anterior, diversos controladores se diseñan considerando únicamente la cinemática del robot y en otros casos el modelo dinámico [12].

Para ejecutar tareas que requieren altas velocidades o transporte de cargas, es fundamental considerar la cinemática del robot, además de su dinámica [13], debido a que en el modelo dinámico se estudian las fuerzas y velocidades ejercidas por las ruedas y el efecto que estas causan en el movimiento del robot. Diversos autores destacan la complejidad que se tiene al considerar todos los parámetros o efectos que se producen en el móvil, como por ejemplo la interacción con la superficie en donde opera el robot, fricciones, deslizamientos, etc. [14], [15], [16], mismos que dificultan la controlabilidad del sistema, aun cuando el robot móvil navega con una velocidad reducida y muchos de los comportamientos no lineales resultan despreciables. Por otra parte, la identificación y la estimación de parámetros de sistemas lineales y no lineales son de gran interés para la comunidad científica. El objetivo principal de la tarea de identificación del sistema es la estimación de un modelo equivalente en simulación que imite el comportamiento real del sistema [17]. Este proceso es denominado en la literatura como Model-in-the-Loop [18], [19]. Por consiguiente, surge así la importancia de obtener un modelo cinemático/dinámico aproximado y a su vez, procurar una alta precisión en la estimación de los parámetros del modelo que se apegue al funcionamiento y desempeño real del robot [20].

En el estado del arte se pueden observar propuestas para la obtención de un modelo cinemático y dinámico del RMO. En [21] utilizan una modificación de la convención Denavit-Hartenberg, originalmente utilizada para describir el comportamiento de manipuladores robóticos, con el propósito de obtener un modelo cinemático del robot móvil, para lo cual es necesario obtener una matriz Jacobiana de las ruedas. En [22] proponen la obtención de un modelo cinemático del RMO a través del mapeo de las velocidades relativas del robot. Asimismo, en [23] se propone un modelado dinámico del RMO a través de la formulación Euler-Lagrange, considerando el comportamiento de la plataforma móvil y de las ruedas unidas a los motores. En [24] se reporta el modelo dinámico de un RMO que considera la dinámica de actuadores, teniendo como entrada de control el voltaje de armadura de los motores. Por tal motivo, en este trabajo se describe la obtención del modelo cinemático y dinámico del RMO considerando las dinámicas de los actuadores (motor) y manejadores (servo-amplificadores) de un prototipo real diseñado en el Laboratorio de Diseño Mecatrónico Óptimo en el CIDETEC-IPN en CDMX, México. El desempeño del modelo dinámico se valida a través de experimentación a través de dos escenarios de prueba.

Lo que acontece del artículo se organiza de la siguiente manera: En la sección 2 se describe el modelo cinemático del RMO. En la sección 3 se describe el modelo dinámico que rige el comportamiento del RMO con la inclusión de la dinámica de los actuadores y manejadores. Los resultados en simulación y experimentación se discuten en la Sección 4 y finalmente se proporcionan las conclusiones y trabajo a futuro en la Sección 5.

2. Modelo cinemático del robot móvil omnidireccional

En la figura 1 se muestra el diagrama esquemático del robot móvil, donde la masa y la inercia del robot móvil están representadas por m e I_z ; r y J son el radio y la inercia de las ruedas; L es la distancia entre el centro de masa del robot móvil y las ruedas; θ_i y $\dot{\theta}_i$ son la posición y la velocidad angular de la i -ésima rueda.

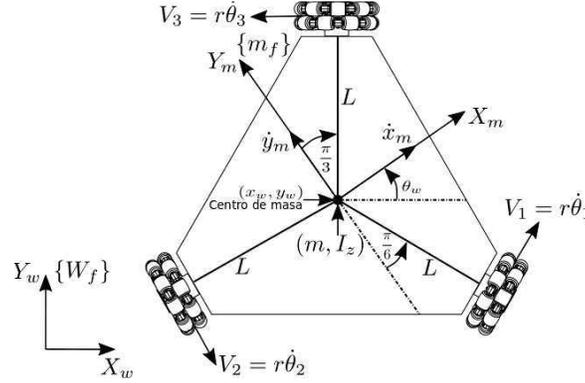


Figura 1. Diagrama esquemático del robot móvil omnidireccional.

Para determinar las posiciones y orientaciones del robot, se asume que el sistema de coordenada absoluto es $X_w - Y_w$, cuyo origen está en O_w y es fijo en el plano. El sistema de coordenadas de movimiento del robot móvil es $X_m - Y_m$, cuyo origen está en O_m , y está fijo al centro de gravedad. Además, se considera como θ_w al ángulo entre el sistema de coordenadas de movimiento y el sistema de coordenadas absoluto, por lo que el mapeo de las velocidades relativas del móvil $(\dot{x}_m, \dot{y}_m, \dot{\theta}_m)$ a las velocidades absolutas en el sistema coordenado $X_w - Y_w$ está dado por (1).

$$\begin{aligned}\dot{x}_w &= \dot{x}_m \cos(\theta_w) - \dot{y}_m \sin(\theta_w) \\ \dot{y}_w &= \dot{x}_m \sin(\theta_w) + \dot{y}_m \cos(\theta_w) \\ \dot{\theta}_w &= \dot{\theta}_m\end{aligned}\quad (1)$$

Considerando $\dot{\eta}_m = [\dot{x}_m, \dot{y}_m, \dot{\phi}_m]^T$ como el vector de velocidad lineal y angular del móvil con respecto al sistema de coordenadas de movimiento $X_m - Y_m$ y $\dot{\eta}_w = [\dot{x}_w, \dot{y}_w, \dot{\phi}_w]^T$ como el vector de velocidad lineal y angular del móvil con respecto al sistema de coordenadas absoluto (inercial) $X_w - Y_w$, la ecuación se puede representar como $\dot{\eta}_w = {}^w_m R \dot{\eta}_m \in R^{3 \times 1}$, donde ${}^w_m R$ es la matriz de rotación (2).

$${}^w_m R = \begin{bmatrix} \cos(\theta_w) & -\sin(\theta_w) & 0 \\ \sin(\theta_w) & \cos(\theta_w) & 0 \\ 0 & 0 & 1 \end{bmatrix}\quad (2)$$

El mapeo entre la velocidad lineal de las llantas y la velocidad angular y lineal en el sistema coordenado del robot móvil está dado por (3), donde $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T \in R^{3 \times 1}$ es el vector de velocidad angular de cada rueda, $\bar{R} = \text{diag}(r, r, r) \in R^{3 \times 3}$ es el radio de cada rueda y L es la distancia del centro geométrico del robot móvil al centro de la rueda.

$$\bar{R}\dot{\theta} = \bar{A}^T \dot{\eta}_m \quad (3)$$

donde:

$$\bar{A}^T = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & L \\ 0 & -1 & L \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & L \end{bmatrix}$$

3. Modelo dinámico del robot móvil omnidireccional

3.1 Dinámica de la plataforma móvil

Para obtener el modelo dinámico de la plataforma móvil se considera la formulación Euler-Lagrange, que involucra la energía potencial y cinética de todos los componentes del sistema. Se considera que el sistema de coordenadas O_m' está alineado al sistema O_m y tiene las coordenadas s_1 y s_2 con respecto al sistema O_m como se observa en la figura 2.

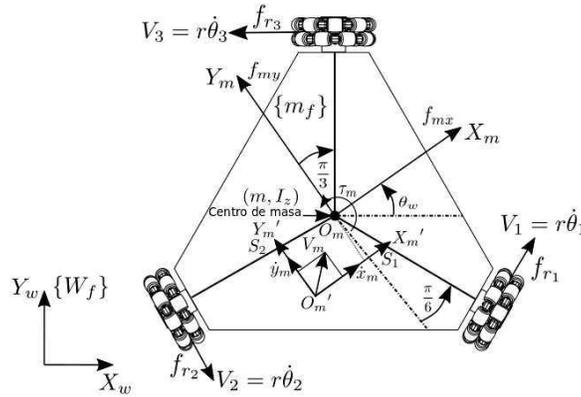


Figura 2. Diagrama esquemático del robot móvil.

La velocidad del centro de masa con respecto a O_m está dada por (4).

$$V_{C.M.} = [\dot{x}_m - \dot{\phi}_m s_2, \dot{y}_m + \dot{\phi}_m s_1, 0]^T \quad (4)$$

La energía cinética del robot móvil está dada por la energía debida a la rotación y traslación del robot móvil, por lo tanto, el Lagrangiano está dado por (5).

$$L = \frac{1}{2} m V_{C.M.}^T V_{C.M.} + \frac{1}{2} \dot{\phi}_m^2 I_z \quad (5)$$

$$L = \frac{1}{2} m (\dot{x}_m^2 - 2\dot{\phi}_m \dot{x}_m s_2 + \dot{\phi}_m^2 s_2^2 + \dot{y}_m^2 + 2\dot{\phi}_m \dot{y}_m s_1 + \dot{\phi}_m^2 s_1^2) + \frac{1}{2} \dot{\phi}_m^2 I_z$$

Considerando las coordenadas generalizadas $\dot{\eta}_m = [\dot{x}_m, \dot{y}_m, \dot{\phi}_m]^T$, las fuerzas generalizadas $F_m = [f_{mx}, f_{my}, \tau_m]^T$, y estableciendo $s_1 = s_2 = 0$ y $\dot{s}_1 = \dot{x}_m$, $\dot{s}_2 = \dot{y}_m$, las ecuaciones de movimiento de Lagrange se expresan en (6).



$$F_m = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\eta}_m} \right) - \frac{\partial L}{\partial \eta_m} \quad (6)$$

$$F_m = \begin{bmatrix} m\ddot{x}_m - m\dot{\phi}_m\dot{y}_m \\ m\ddot{y}_m + m\dot{\phi}_m\dot{x}_m \\ I_z\ddot{\phi}_m \end{bmatrix}$$

Por último, al realizar el análisis de fuerzas y momentos del robot móvil se obtiene que $F_m = \bar{A}F_r$, donde $F_r = [f_{r_1}, f_{r_2}, f_{r_3}]^T$ es la fuerza de rozamiento estático, por lo tanto, el modelo dinámico se puede expresar como en (7).

$$\tilde{M}\ddot{\eta}_m + \tilde{H}\dot{\eta}_m = \bar{A}F_r \quad (7)$$

donde:

$$\tilde{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad y \quad \tilde{H} = \begin{bmatrix} 0 & -m\dot{\phi}_m & 0 \\ m\dot{\phi}_m & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3.2 Dinámica de la rueda-motor

La rueda omnidireccional se define como una rueda estándar a la cual se dota de una corona de rodillo, cuyos ejes de giro resultan perpendiculares a la dirección normal de avance. En la figura 3 se muestra la representación esquemática de la rueda.

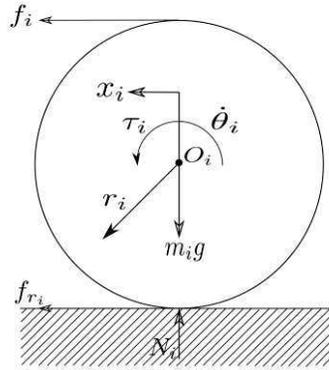


Figura 3. Dinámica de la rueda.

Al aplicar la segunda ley de Newton para el movimiento rotacional en la rueda de la figura 3 y teniendo en cuenta que $\tau_i = r_i f_i$, donde τ_i es el par ejercido por el motor de la i -ésima rueda, se obtiene (8).

$$\sum M_{O_i} = J\ddot{\theta}_i \quad (8)$$

$$\tau_i - r_i f_{r_i} = J\ddot{\theta}_i$$

Para el caso general de las tres ruedas del RMO se obtiene (9), donde $\tau = [\tau_1, \tau_2, \tau_3]^T \in \mathfrak{R}^{3 \times 1}$ es el par de entrada, $\bar{J} = \text{diag}(J, J, J) \in \mathfrak{R}^{3 \times 3}$ es el momento de inercia, y $\ddot{\theta} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3]^T \in \mathfrak{R}^{3 \times 1}$ es la aceleración angular de cada rueda.

$$\tau - \bar{R}F_r = \bar{J}\ddot{\theta} \quad (9)$$

3.3 Dinámica del motor

En la figura 4 se muestra el diagrama esquemático de la parte mecánica del motor, el cual muestra los parámetros físicos involucrados en el mismo para obtener una mejor aproximación a la realidad en la formulación del modelo dinámico de dicho sistema. En esta figura, V_i representa el voltaje de entrada de armadura, $b_o \in \mathfrak{R}$ es el coeficiente de fricción viscosa referido a la flecha, $J_o \in \mathfrak{R}$ es la inercia del motor referida a la flecha, q_{m_i} , \dot{q}_{m_i} , \ddot{q}_{m_i} son las posiciones, velocidades y aceleraciones angulares del rotor, respectivamente para la i -ésima rueda. τ_{m_i} es el par del motor, τ_i es el par de salida (carga), $\hat{r} \in \mathfrak{R}$ es la relación de reducción de engranes, θ_i , $\dot{\theta}_i$, $\ddot{\theta}_i$ son las posiciones, velocidades y aceleraciones angulares de salida, respectivamente para la i -ésima rueda, $k_m \in \mathfrak{R}$ es la constante de par, $R_a \in \mathfrak{R}$ es la resistencia de armadura y $k_e \in \mathfrak{R}$ es la constante de fuerza contraelectromotriz.

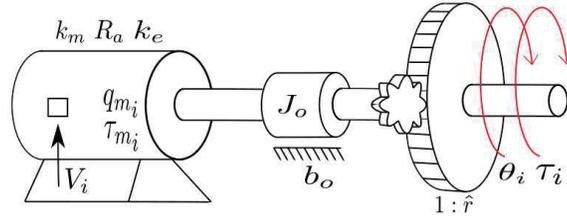


Figura 4. Diagrama esquemático de la mecánica del motor.

Aplicando la segunda ley de Newton, se puede calcular la dinámica de la parte mecánica del motor tal como se expresa en (10).

$$\sum M = \frac{d(J_o \dot{q}_{m_i})}{dt} \quad (10)$$

$$\tau_{m_i} - b_o \dot{q}_{m_i} - \frac{\tau_i}{\hat{r}} = J_o \ddot{q}_{m_i}$$

La ecuación para calcular el par del motor se define en (11), donde $i_{ai} \in \mathfrak{R}$ es la corriente de armadura.

$$\tau_{m_i} = k_m i_{ai} \quad (11)$$

Posteriormente se sustituye τ_{m_i} de (11) en (10), para que la dinámica de la parte mecánica del motor esté relacionada con la dinámica eléctrica del mismo. Para el caso general de las tres ruedas, se obtiene (12), donde $\bar{J}_o = \text{diag}(J_o, J_o, J_o) \in \mathfrak{R}^{3 \times 3}$, $\bar{b}_o = \text{diag}(b_o, b_o, b_o) \in \mathfrak{R}^{3 \times 3}$, $\hat{K}_m = \text{diag}(k_m, k_m, k_m) \in \mathfrak{R}^{3 \times 3}$, $\hat{R} = \text{diag}(\hat{r}, \hat{r}, \hat{r}) \in \mathfrak{R}^{3 \times 3}$, $\ddot{q}_m = [\ddot{q}_{m_1}, \ddot{q}_{m_2}, \ddot{q}_{m_3}]^T \in \mathfrak{R}^{3 \times 1}$, $\dot{q}_m = [\dot{q}_{m_1}, \dot{q}_{m_2}, \dot{q}_{m_3}]^T \in \mathfrak{R}^{3 \times 1}$ y $\hat{i}_a = [i_{a_1}, i_{a_2}, i_{a_3}]^T \in \mathfrak{R}^{3 \times 1}$.

$$\bar{J}_o \ddot{q}_m + \bar{b}_o \dot{q}_m + \hat{R}^{-1} \tau = \hat{K}_m \hat{i}_a \quad (12)$$

Después de obtener la dinámica de la parte mecánica del motor, se procede a calcular la dinámica de la parte eléctrica. Dicho proceso se generaliza para las tres ruedas con las que cuenta el RMO.

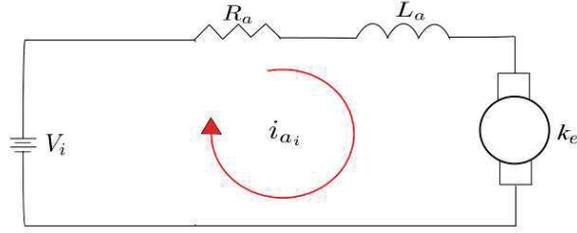


Figura 5. Representación eléctrica del motor.

Aplicando la ley de voltaje de Kirchhoff al circuito de la figura 5 se obtiene (13).

$$L_a \frac{di_{a_i}}{dt} + R_a i_{a_i} + k_e \dot{q}_{m_i} = V_i \quad (13)$$

Para expresar (13) en términos de $\theta_i \in \mathfrak{R}$, es necesario utilizar la relación de reducción de engranes en (14).

$$q_{m_i} = \hat{r} \theta_i \quad (14)$$

Para el caso general se sustituye (14) en (13) para cada una de las ruedas y se obtiene (15), donde $\hat{L}_a = \text{diag}(L_a, L_a, L_a) \in \mathfrak{R}^{3 \times 3}$ es la inductancia de armadura del motor, $\hat{K}_e = \text{diag}(k_e, k_e, k_e) \in \mathfrak{R}^{3 \times 3}$, $\hat{R}_a = \text{diag}(R_a, R_a, R_a) \in \mathfrak{R}^{3 \times 3}$, $V = [V_1, V_2, V_3]^T \in \mathfrak{R}^{3 \times 1}$, y $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T \in \mathfrak{R}^{3 \times 1}$.

$$\hat{L}_a \frac{d\hat{i}_a}{dt} + \hat{R}_a \hat{i}_a + \hat{K}_e \hat{R} \dot{\theta} = V \quad (15)$$

Para expresar la ecuación anterior en términos del sistema de coordenadas de movimiento (η_m), se sustituye (3) en (15) y se obtiene (16).

$$\hat{L}_a \frac{d\hat{i}_a}{dt} + \hat{R}_a \hat{i}_a + \hat{K}_e \hat{R} \bar{R}^{-1} \bar{A}^T \dot{\eta}_m = V \quad (16)$$

Para representar (16) con respecto al sistema de coordenadas absolutas (η_w) se utiliza $\dot{\eta}_w = {}^w_m R \dot{\eta}_m$, obteniendo (17).

$$\hat{L}_a \frac{d\hat{i}_a}{dt} + \hat{R}_a \hat{i}_a + \hat{K}_e \hat{R} \bar{R}^{-1} \bar{A}^T {}^w_m R^T \dot{\eta}_w = V \quad (17)$$

3.4 Acoplamiento dinámico rueda/motor

Para realizar el acoplamiento de la dinámica de la rueda con la dinámica de la parte mecánica del motor, se sustituye (9) en (12), obteniendo (18).

$$\hat{R} (\hat{K}_m \hat{i}_a - \bar{J}_0 \ddot{q}_m - \bar{b}_0 \dot{q}_m) - \bar{R} F_r = \bar{J} \ddot{\theta} \quad (18)$$

Posteriormente, la relación de reducción de los engranes en (14) se sustituye en (18), obteniendo (19).

$$\hat{R} \hat{K}_m \hat{i}_a - \hat{R} \bar{b}_0 \hat{R} \dot{\theta} - \bar{R} F_r = (\hat{R} \bar{J}_o \hat{R} + \bar{J}) \ddot{\theta} \quad (19)$$

3.5 Acoplamiento dinámico rueda/motor con la plataforma



Para acoplar la dinámica de la rueda/motor (parte mecánica) con la dinámica de la plataforma (7), (19) debe expresarse en términos de η_m . Para ello, se sustituye (3) en (19), resultando (20).

$$\hat{R}\hat{K}_m\hat{i}_a - \hat{R}\hat{b}_o\hat{R}\hat{R}^{-1}\bar{A}^T\dot{\eta}_m - \bar{R}F_r = \hat{R}\hat{J}_o\hat{R}\hat{R}^{-1}\bar{A}^T\dot{\eta}_m + \bar{J}\bar{R}^{-1}\bar{A}^T\ddot{\eta}_m \quad (20)$$

Despejando F_r de (20) se obtiene (21).

$$F_r = \bar{R}^{-1} \left[-\hat{R}\hat{J}_o\hat{R}\hat{R}^{-1}\bar{A}^T\dot{\eta}_m - \bar{J}\bar{R}^{-1}\bar{A}^T\ddot{\eta}_m + \hat{R}\hat{K}_m\hat{i}_a - \hat{R}\hat{b}_o\hat{R}\hat{R}^{-1}\bar{A}^T\dot{\eta}_m \right] \quad (21)$$

Posteriormente (21) se sustituye en la dinámica de la plataforma (7), quedando la expresión que se muestra en (22).

$$\left(\tilde{M} + \bar{A}\bar{R}^{-1}\hat{R}\hat{J}_o\hat{R}\hat{R}^{-1}\bar{A}^T + \bar{A}\bar{R}^{-1}\bar{J}\bar{R}^{-1}\bar{A}^T \right) \ddot{\eta}_m + \left(\tilde{H} + \bar{A}\bar{R}^{-1}\hat{R}\hat{b}_o\hat{R}\hat{R}^{-1}\bar{A}^T \right) \dot{\eta}_m = \bar{A}\bar{R}^{-1}\hat{R}\hat{K}_m\hat{i}_a \quad (22)$$

Entonces (22) puede expresarse en términos reducidos como (23).

$$\bar{M}\ddot{\eta}_m + \bar{C}\dot{\eta}_m = \bar{B}\hat{K}_m\hat{i}_a \quad (23)$$

donde:

$$\bar{M} = \tilde{M} + \bar{A}\bar{R}^{-1}\hat{R}\hat{J}_o\hat{R}\hat{R}^{-1}\bar{A}^T + \bar{A}\bar{R}^{-1}\bar{J}\bar{R}^{-1}\bar{A}^T$$

$$\bar{C} = \tilde{H} + \bar{A}\bar{R}^{-1}\hat{R}\hat{b}_o\hat{R}\hat{R}^{-1}\bar{A}^T$$

$$\bar{B} = \bar{A}\bar{R}^{-1}\hat{R}$$

Para representar (23) en las coordenadas fijas (η_w), se utiliza $\dot{\eta}_w = {}^w_m R \dot{\eta}_m$, donde $\dot{\eta}_m = {}^w_m R^T \dot{\eta}_w$ y $\ddot{\eta}_m = {}^w_m R^T \ddot{\eta}_w + {}^w_m \dot{R}^T \dot{\eta}_w$, obteniendo (24).

$$\bar{M}({}^w_m R^T \ddot{\eta}_w + {}^w_m \dot{R}^T \dot{\eta}_w) + \bar{C}{}^w_m R^T \dot{\eta}_w = \bar{B}\hat{K}_m\hat{i}_a \quad (24)$$

Reduciendo términos en (24) e incorporando la fricción viscosa en las velocidades angulares del motor, la cual se denota como $F_v = K_v \dot{q}_m$, con $K_v = \text{diag}(k_{v_1}, k_{v_2}, k_{v_3}) \in \mathfrak{R}^{3 \times 3}$, se obtiene (25).

$$\hat{M}\ddot{\eta}_w + \hat{C}\dot{\eta}_w + F_v = \bar{B}\hat{K}_m\hat{i}_a \quad (25)$$

donde:

$$\hat{M} = \bar{M}{}^w_m R^T$$

$$\hat{C} = \bar{M}{}^w_m \dot{R}^T + \bar{C}{}^w_m R^T$$

3.6 Dinámica de los manejadores (Servo-amplificadores)

Para estimar el comportamiento de la dinámica de los manejadores del RMO, se considera que su comportamiento se rige a través de un controlador Proporcional-Integral (PI) de corriente como se muestra en (26), donde \bar{u} es la señal de entrada del sistema, $\bar{K}_p = \text{diag}(\bar{k}_{p_1}, \bar{k}_{p_2}, \bar{k}_{p_3}) \in \mathfrak{R}^{3 \times 3}$, $\bar{K}_i = \text{diag}(\bar{k}_{i_1}, \bar{k}_{i_2}, \bar{k}_{i_3}) \in \mathfrak{R}^{3 \times 3}$ son matrices definidas positivas, que representan las ganancias



proporcional e integral del manejador, y $\bar{e} = i_{di} - i_{ai}$ es el error entre la corriente deseada y la corriente obtenida por el RMO. Para esto se debe configurar el manejador de potencia en modo corriente.

$$\bar{u} = \bar{K}_p \bar{e} + \bar{K}_i \int \bar{e} dt \quad (26)$$

La figura 6 muestra detalladamente el funcionamiento del control en cascada, considerando como lazo interno el control PI de corriente, el cual corresponde al manejador de potencia, y como lazo externo el regulador de posición lineal/angular. En la misma figura, \dot{x} representa la dinámica del sistema RMO completo, y e es el error en la posición lineal/angular del RMO (entre la posición deseada en el espacio cartesiano y la posición real del sistema).

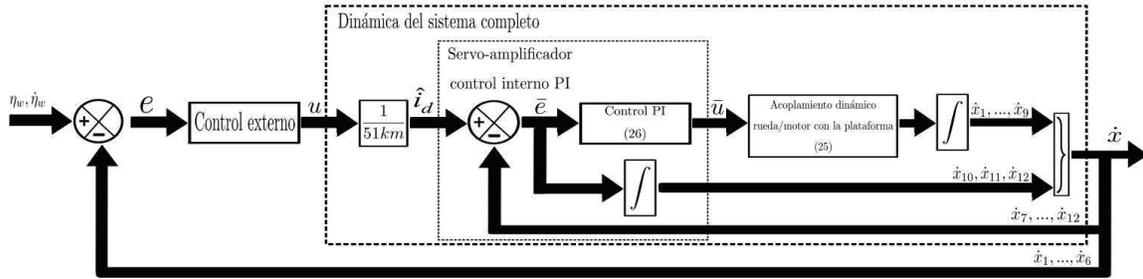


Figura 6. Diagrama de control en cascada.

3.7 Representación en espacio de estados del sistema completo

Sea $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T = [x_w, y_w, \phi_w, \dot{x}_w, \dot{y}_w, \dot{\phi}_w, i_{a1}, i_{a2}, i_{a3}, e_{ia1}, e_{ia2}, e_{ia3}]^T \in \mathfrak{R}^{12 \times 1}$ el vector de estado que incluye la posición, velocidad lineal, velocidad angular y corriente de armadura del RMO en el sistema de coordenadas inercial, y $u = [u_1, u_2, u_3]^T \in \mathfrak{R}^{3 \times 1}$ el vector de voltaje de entrada en los motores de la rueda, el modelo dinámico en (25) representado en el espacio de estados está dado por (26).

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= Cx \end{aligned} \quad (26)$$

donde:

$$f(x) = \begin{bmatrix} [x_4 \ x_5 \ x_6]^T \\ \hat{M}^{-1}(\bar{B}K_M [x_7 \ x_8 \ x_9]^T - \hat{C}[x_4 \ x_5 \ x_6]^T - F_v) \\ LA^{-1}(-K_e \hat{R} \bar{R}^{-1} \bar{A}^T \ ^w_m R^T [x_4 \ x_5 \ x_6]^T - R_A [x_7 \ x_8 \ x_9]^T + \bar{K}_i [x_{10} \ x_{11} \ x_{12}]^T - \bar{K}_p [x_7 \ x_8 \ x_9]^T) \\ [x_{10} \ x_{11} \ x_{12}]^T \end{bmatrix} \in \mathfrak{R}^{12 \times 1}$$

$$g(x) = \begin{bmatrix} 0 \in \mathfrak{R}^{6 \times 3} \\ LA^{-1} \frac{\bar{K}_p}{51K_M} \\ 0 \in \mathfrak{R}^{3 \times 3} \end{bmatrix} \in \mathfrak{R}^{12 \times 3}$$

$$C = I \in \mathfrak{R}^{12 \times 12}$$



4. Resultados

Para realizar simulaciones numéricas se utilizaron los parámetros cinemáticos y dinámicos del prototipo real del RMO que se encuentra en el laboratorio de Diseño Mecatrónico Optimo (D.M.O.) del CIDETEC-IPN en la Ciudad de México, México. Dichos parámetros se muestran en la Tabla 1 y son necesarios para la simulación del modelo dinámico del RMO en MATLAB. Asimismo, se utiliza el método de integración de Runge-Kutta de cuarto orden con un tiempo de integración $\Delta t = 0.00032s$ y el tiempo final $t_f = 130s$.

Para obtener los parámetros del controlador PI que caracteriza la dinámica que representa a los manejadores del RMO, se planteó un problema de optimización con el objetivo obtener un mínimo error en la corriente suministrada por el motor. Dicho problema se resolvió por medio de un algoritmo metaheurístico denominado en la literatura como evolución diferencial en su variante Rand 1 Bin. Los parámetros obtenidos por el algoritmo son los siguientes: $\bar{k}_p = \text{diag}(3.5287, 3.5287, 3.5287) \in \mathfrak{R}^{3 \times 3}$ y $\bar{k}_i = \text{diag}(14637.0980, 14637.0980, 14637.0980) \in \mathfrak{R}^{3 \times 3}$. Además, para compensar las incertidumbres paramétricas no modeladas se proponen, después de realizar diversas pruebas experimentales, los siguientes coeficientes de fricción $K_v = \text{diag}(300, 300, 300) \in \mathfrak{R}^{3 \times 3}$.

Tabla 1. Parámetros cinemáticos y dinámicos del RMO.

Parámetro	Descripción	Valor	Unidades
r	Radio de la rueda	0.0625	m
L	Distancia del centro a la rueda	0.287	m
m	Masa del RMO	16.319	kg
J	Inercia de la rueda	$5.82E-4$	$kg\ m^2$
I_z	Inercia del RMO	0.516	$kg\ m^2$
J_o	Inercia del motor del RMO	$6.81E-6$	$kg\ m^2$
r_g	Relación de reducción de engranes	51	-
L_a	Inductancia de armadura del motor	$0.34E-3$	H
K_e	Constante de fuerza contraelectromotriz	$38.9E-3$	$\frac{v \cdot s}{rad}$
b_o	Fricción viscosa de la flecha del motor	$6.3359E-6$	$N\ m\ s$
R_a	Resistencia de armadura	1.2	Ω
K_m	Contante de torque del motor	$38.9E-3$	$\frac{Nm}{A}$

Por otra parte, para que el RMO realice movimientos en su espacio de coordenadas, es necesario obtener la relación de fuerzas que se debe suministrar a las ruedas del robot. Considerando el análisis de fuerzas estáticas de la sección 3, se puede despejar de $F_m = \bar{A}F_r$ la fuerza necesaria en cada una de las ruedas para realizar el movimiento, obteniendo $F_r = \bar{A}^{-1}F_m$. Además, al proyectarse sobre el marco de referencia W_f , $F_r = ({}^wR\bar{A})^{-1}F_w$, cuyas componentes se muestran en (27).



$$\begin{aligned}
 f_{r_1} &= -F_{wx} \frac{1}{3} (\sin \phi_w - \sqrt{3} \cos \phi_w) + F_{wy} \frac{1}{3} (\cos \phi_w + \sqrt{3} \sin \phi_w) + \frac{1}{3L} \tau_w \\
 f_{r_2} &= F_{wx} \frac{2 \sin \phi_w}{3} - F_{wy} \frac{2 \cos \phi_w}{3} + \frac{1}{3L} \tau_w \\
 f_{r_3} &= -F_{wx} \frac{1}{3} (\sin \phi_w + \sqrt{3} \cos \phi_w) + F_{wy} \frac{1}{3} (\cos \phi_w - \sqrt{3} \sin \phi_w) + \frac{1}{3L} \tau_w
 \end{aligned} \tag{27}$$

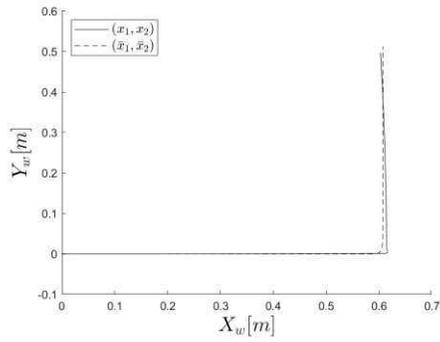
Considerando las condiciones iniciales del robot $\eta_w = [x_w \ y_w \ \phi_w]^T = [0 \ 0 \ 0]^T$, se puede observar que existe una relación con las fuerzas estáticas para efectuar los movimientos con el RMO. Para un desplazamiento del robot sobre el eje X_m , la relación resulta:

$F_r = [-F_{wx}(-\sqrt{3}/3) \ 0 \ -F_{wx}(-\sqrt{3}/3)]^T = [a \ 0 \ -a]^T$, Así mismo, se puede observar que para un desplazamiento del RMO en el eje Y_m , la relación resulta: $F_r = [F_{wy}(1/3) \ -F_{wy}(2/3) \ F_{wy}(1/3)]^T = [a/2 \ -a \ a/2]^T$. Por último, para una rotación sobre el eje Z_m , la relación resulta: $F_r = [\tau_w(1/3L) \ \tau_w(1/3L) \ \tau_w(1/3L)]^T = [a \ a \ a]^T$, en donde se observa que a debe ser una fuerza constante para efectuar los movimientos mencionados.

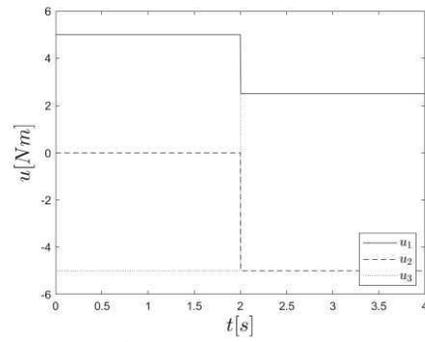
Con el propósito de comprobar el modelo dinámico descrito en la sección 3 y realizar una comparativa entre los resultados en simulación y los resultados en la plataforma experimental, se provee en ambos casos una señal de control constante $u = [5, 5, 5]^T \in \mathfrak{R}^{3 \times 1}$. Lo anterior con el objetivo de que el RMO realice desplazamientos en el sistema de coordenadas $X_m - Y_m$ con respecto al sistema de coordenadas inercial $X_w - Y_w$. Por consiguiente, se proponen dos pruebas. La primera prueba consiste en que el RMO realice un desplazamiento sobre el eje X_w por un periodo de 2 segundos, y al terminar, el robot continúe con un desplazamiento sobre el eje Y_w por un periodo de 2 segundos. El segundo experimento consiste en que el robot realice una rotación sobre el eje Z_w por un periodo de 2 segundos.

En la figura 7 se muestran los resultados obtenidos a través de los experimentos propuestos. En la figura 7a se puede observar una comparación visual del desplazamiento del robot al realizar el primer experimento y en la figura 7e los resultados del segundo experimento, donde los estados de la simulación del modelo dinámico están representados por $(\bar{x}_1, \bar{x}_2, \bar{x}_3)$ y los de la etapa experimental por (x_1, x_2, x_3) , ambos en el sistema de coordenadas inercial (X_w, Y_w, ϕ_w) . Además, se observa que en el desplazamiento en el sistema de coordenadas $X_w - Y_w$, la simulación se aproxima en gran medida a la etapa experimental con un error promedio de $2.767E-2$ m, y para el caso del desplazamiento en el sistema de coordenadas ϕ_w se obtiene un error promedio de $1.255E-1$ m, estos resultados se logran a través de proponer un modelo dinámico que contempla los componentes mecánicos y electrónicos del RMO, y a su vez, compensando las incertidumbres medioambientales con la inclusión de fricción en el modelo. El par suministrado a las ruedas se muestra en las figuras 7b y 7f, en donde también se observan la configuración necesaria y la constante de par de entrada de los motores para realizar los experimentos propuestos.

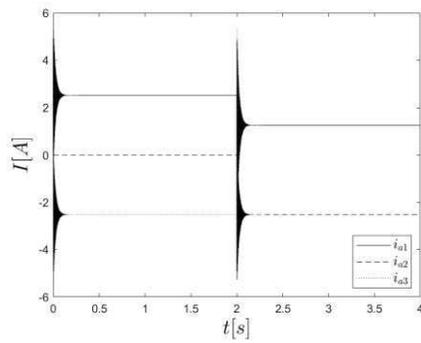
Por último, se pueden observar en las Figuras 7c y 7g las corrientes generadas en la simulación del modelo dinámico propuesto y en las Figuras 7d y 7h las corrientes generadas en el prototipo real del RMO para cada experimento. Además, se puede apreciar en dichas figuras que para romper la inercia del RMO se necesitan sobreimpulsos grandes en la corriente, los cuales oscilan rápidamente hacia un valor constante, ya que la señal de control suministrada es constante en los experimentos propuestos. Para el caso de la simulación del modelo dinámico, esto se debe a la sintonización del controlador PI que caracteriza la dinámica de los manejadores.



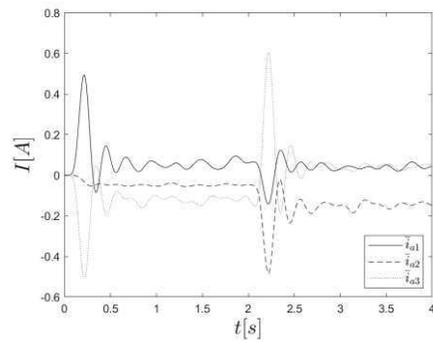
a) Sistema de coordenadas inercial $X_w - Y_w$.



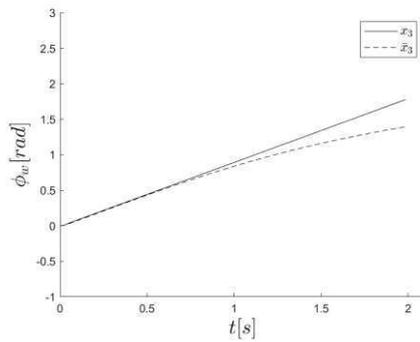
b) Par de entrada en los motores.



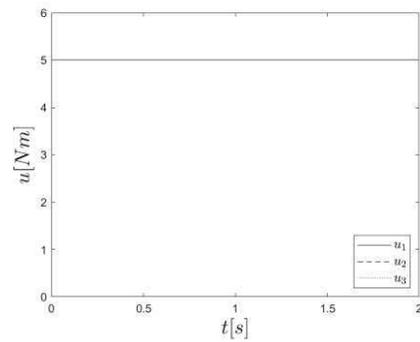
c) Corriente en los motores (simulación).



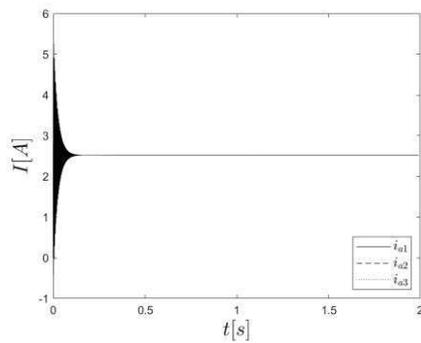
d) Corriente en los motores (experimentación).



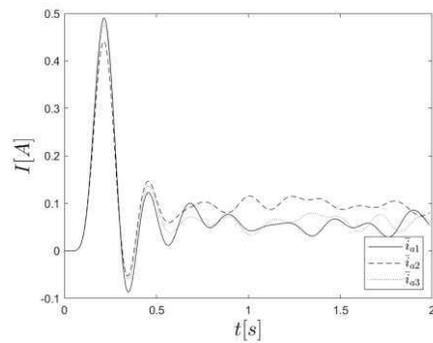
e) Sistema de coordenadas inercial ϕ_w .



f) Par de entrada en los motores.



g) Corriente en los motores (simulación).



h) Corriente en los motores (experimentación).

Figura 7. Comportamiento del RMO con par de entrada constante.



5. Conclusiones

En este trabajo se presenta el análisis cinemático y dinámico del RMO, el cual incluye la dinámica de los manejadores de potencia y de los actuadores presentes en un prototipo real. La dinámica de los actuadores se incluye como la de los motores de corriente directa, y para el caso de los manejadores de potencia se propuso el uso de un control PI de corriente. Para ajustar el control PI, se utilizó un enfoque de sintonización a través de métodos de optimización con el cual es posible encontrar sus ganancias óptimas de manera que se logra caracterizar de mejor forma la dinámica de los manejadores. Además, se incluye fricción en el modelo dinámico, con lo cual se observa que es posible compensar algunas incertidumbres medioambientales desconocidas que no son contempladas en la formulación del modelo dinámico del RMO. Por consiguiente, al incluir la dinámica de los manejadores de potencia, de los actuadores, y la fricción en el modelo dinámico del robot, se observa una respuesta en simulación muy semejante al comportamiento del sistema real.

De igual importancia, se muestra una forma para encontrar la relación necesaria para obtener las señales de control para cada una de las ruedas con el fin de realizar movimientos en este tipo de robots móviles, la cual se consigue mediante un análisis de fuerzas estáticas del sistema. Sin embargo, como trabajo a futuro se propone implementar alguna ley de control para resolver las tareas de regulación y seguimiento, con el objetivo de controlar la velocidad de las ruedas y así probar el desempeño del sistema con otro caso de estudio, afín de garantizar el comportamiento adecuado del modelo dinámico ante cambios en velocidades o aceleraciones, y bajo ciertas condiciones de operación que puedan estar presentes en prototipo robótico real.

Agradecimientos

Proyecto financiado por COFAA y la SIP del Instituto Politécnico Nacional bajo los proyectos 20190239 y 20196226. Los autores reconocen el apoyo de CONACYT, COFAA y BEIFI a través de una beca para realizar estudios de posgrado en el Instituto Politécnico Nacional. Además, se agradece al Fondo de Procuración de Fondos del CIDETEC-IPN por su apoyo para la presentación de este trabajo.

Referencias

- [1] Ortigoza, R. S., et al. *“Wheeled mobile robots: a review”*. IEEE Latin America Transactions, 10(6), p. 2209-2217, 2012.
- [2] Shekhar, S. *“Wheel rolling constraints and slip in mobile robots.”* In Proceedings of International Conference on Robotics and Automation. IEEE, Vol. 3, p. 2601-2607, 1997.
- [3] Ferriere, L., Raucent, B., & Campion, G. *“Design of omnimobile robot wheels”*. In Proceedings of International Conference on Robotics and Automation. IEEE, Vol. 4, p. 3664-3670, 1996.
- [4] Kim, D. S., Lee, H. C., & Kwon, W. H. *“Geometric kinematics modeling of omni-directional autonomous mobile robot and its applications”*. In Proceedings ICRA. International Conference on Robotics and Automation. IEEE, Vol. 3, p. 2033-2038, 2000.
- [5] Loh, W. K., Low, K. H., & Leow, Y. P. *“Mechatronics design and kinematic modelling of a singularityless omni-directional wheeled mobile robot”*. International Conference on Robotics and Automation, IEEE, Vol. 3, p. 3237-3242, 2003.
- [6] Byun, K. S., Kim, S. J., & Song, J. B. *“Design of continuous alternate wheels for omnidirectional mobile robots”*. In Proceedings ICRA. International Conference on Robotics and Automation. IEEE, Vol. 1, p. 767-772, 2001.



- [7] Gracia, L., & Tornero, J. “*Kinematic control of wheeled mobile robots*”. Latin American applied research, 38(1), p. 7-16, 2008.
- [8] Vallvé, J., & Andrade-Cetto, J. “*Potential information fields for mobile robot exploration*”. Robotics and Autonomous Systems, 69, p. 68-79, 2015
- [9] Zhao, Y., Gong, L., Liu, C., & Huang, Y. “*Dual-arm robot design and testing for harvesting tomato in greenhouse*”. IFAC-PapersOnLine, 49(16), p. 161-165, 2016.
- [10] Baca, J., Pagala, P., Rossi, C., & Ferre, M. “*Modular robot systems towards the execution of cooperative tasks in large facilities*”. Robotics and Autonomous Systems, 66, p. 159-174, 2015.
- [11] Lima, R. C. P. “*Sistema de exploración de terrenos con robots móviles: aplicación en tareas de detección y localización de minas antipersonas*”. Universidad Complutense de Madrid, Servicio de Publicaciones, 2010
- [12] Martins, F. N., Carelli, R., Sarcinelli-Filho, M., & Bastos, T. F. “*Modelo Dinámico y Controlador de Seguimiento para Robots Móviles Tipo Uniciclo*”. V Jornadas Argentinas de Robótica-JAR08, 2008.
- [13] Fierro, R., & Lewis, F. L. “*Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics*”. Journal of robotic systems, 14(3), p. 149-163, 1997.
- [14] Yamamoto, Y., & Yun, X. “*Effect of the dynamic interaction on coordinated control of mobile manipulators*”. Transactions on robotics and automation. IEEE, 12(5), p. 816-824, 1996.
- [15] Leow, Y. P., Low, K. H., & Loh, W. K. “*Kinematic modelling and analysis of mobile robots with omni-directional wheels*”. In 7th International Conference on Control, Automation, Robotics and Vision. IEEE, Vol. 2, p. 820-825, 2002.
- [16] Betourne, A., & Fournier, A. “*Kinematics, dynamics and control of a conventional wheeled omnidirectional mobile robot*”. Systems Man and Cybernetics. IEEE, Vol. 2, p. 276-281, 1993.
- [17] Åström, K. J., & Eykhoff, P. “*System identification a survey*”. Automatica, 7(2), p. 123-162, 1971.
- [18] Koh, T., & Powers, E. “*Second-order Volterra filtering and its application to nonlinear system identification*”. IEEE Transactions on acoustics, speech, and signal processing, 33(6), p. 1445-1455, 1985.
- [19] Gotmare, A., Bhattacharjee, S. S., Patidar, R., & George, N. V. “*Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review*”. Swarm and Evolutionary Computation, 32, p. 68-84, 2017.
- [20] Sáenz, A., Bugarin, E., & Santibáñez, V. “*Modelado cinemático y dinámico de un robot móvil omnidireccional de 4 ruedas considerando dinámica de actuadores*”. Repositorio Tecnológico Nacional de México Instituto Tecnológico de la Laguna, 2015.
- [21] Muir, P. F., & Neuman, C. P. Kinematic modeling of wheeled mobile robots. “*Journal of robotic systems*”, 4(2), p. 281-340, 1987.
- [22] Castellanos, J. G., Cervantes, M. V., Santana, J. S., & Martínez, S. R. “*Seguimiento de trayectorias de un robot móvil (3,0) mediante control acotado*”. Revista Iberoamericana de Automática e Informática Industrial, 11(4), p. 426-434, 2014.
- [23] Serrano-Pérez, O., Villarreal-Cervantes, M. G., González-Robles, J. C., & Rodríguez-Molina, A. “*Meta-heuristic algorithms for the control tuning of omnidirectional mobile robots*”. Engineering Optimization, p. 1-18, 2019.
- [24] Kozłowski, K., & Majchrzak, J. “*A backstepping approach to control a nonholonomic mobile robot*”. International Conference on Robotics and Automation. IEEE, Vol. 4, p. 3972-3977, 2002.



Redes Neuronales Físicas Artificiales

Covix Gracia Evelin, Nieves Vázquez José Ángel, Cervantes Morales Emmanuel de
Jesús y Zubiaga Garate Augusto

Instituto Tecnológico Superior de San Andrés Tuxtla.
josenieves@itssat.edu.mx

Resumen

En el presente trabajo se presenta el funcionamiento de una Red Neuronal Artificial (RNA) simulada, así como el diseño de una Red Neuronal Artificial mediante dispositivos electrónicos. Básicamente, el uso de la electrónica y optoelectrónica nos permitirá apreciar la forma en que cada unidad celular interactúan para ejecutar una tarea. Por otro lado, con la finalidad de poder comprender el funcionamiento y la arquitectura que conforma una RNA se presentan los conceptos y modelos matemáticos básicos que las originan. Comprendiendo de esta forma el papel que juega cada elemento electrónico de la RNA física. La idea principal del trabajo es mostrar el diseño y elaboración de la neurona de manera física para que las personas visualicen de esta manera como es el comportamiento de una neurona real.

Palabras clave: RNA (Redes Neuronales Artificiales), optoelectrónica, Biológica

1. Introducción

En la historia de la inteligencia artificial, el modelo cibernético fue pronto relegado a favor del informático. Sin embargo, la computación analógica se revela como un entorno fascinante a la hora de construir modelos altamente intuitivos capaces de emular el funcionamiento de entes biológico y complejos [1].

El cerebro es uno de los órganos más sorprendentes que posee el ser humano, el cual es capaz de coordinar y realizar más de una tarea a la vez. Es debida a esta complejidad que el hombre ha desarrollado sistemas que intente emular el cerebro humano. La IA (Inteligencia Artificial) se ha encardo de estudiar y desarrollar RNA, desde los más sencillos hasta los más complejos, entre otros sistemas inteligentes. Gracia a el desarrollo de estos sistemas inteligentes, como lo son las redes neuronales artificiales; podemos adaptar modelos existentes para que desempeñen actividades, las cuales pueden requerir de un aprendizaje autónomo o supervisado.

En el presente trabajo se emplea la arquitectura de un modelo de una RNA, para llevarla a nivel físico, emulando así las funciones de una red neuronal biológica permitiendo que los jóvenes de ingeniería (mecatrónica, informática, sistemas y a fines) puedan interactuar y adquirir conocimientos en RNA, para posteriormente crear sistemas inteligentes capaces de solucionar algún problema de la vida cotidiana.

La estructura del artículo es el siguiente: En la sección dos se presentan los conceptos básicos relacionados con la neurona biológica y su modelo matemático que la representa, así como el diseño, la simulación y la implementación de la neurona real. En la sección tres se presentan los resultados al implementar la neurona de manera electrónica. Finalmente, en la sección cuatro se presentan las conclusiones y comentarios del trabajo.



2. De la neurona biológica a la neurona artificial

2.1 Neurona Biológica

Una neurona es una célula viva y está constituida por los mismos elementos que conforman las células biológicas. Una neurona consta de un cuerpo (soma) más o menos esférico de 5 a 10 micras de diámetro del que sale una rama principal el axón, y varias ramas más cortas llamadas dendritas. A su vez el axón puede producir ramas en torno a su punto de arranque, y con frecuencia se ramifica extensamente cerca de su extremo. Figura 1 [1].

Una de las características que diferencia a las neuronas de otras células es su capacidad de comunicación [1]. La neurona recibe una señal (impulso) de otras neuronas a las cuales se encuentra interconectada por medio de sus dendritas y posteriormente esta señal generada por medio del axón.

La neurona emplea dos tipos de señales: eléctricas y químicas. La señal eléctrica es aquella que se transmite por medio del axón, mientras que la señal química se genera en las dendritas. Si hacemos una comparación entre una neurona biológica y una artificial, podemos considerar que las señales que llegan a la sinapsis son las entradas a la neurona; estas son ponderadas (atenuadas o simplificadas) a través de un parámetro denominado peso, asociado a la sinapsis. Estas señales de entrada pueden excitar a la neurona (sinapsis positiva) (peso negativo) [2].

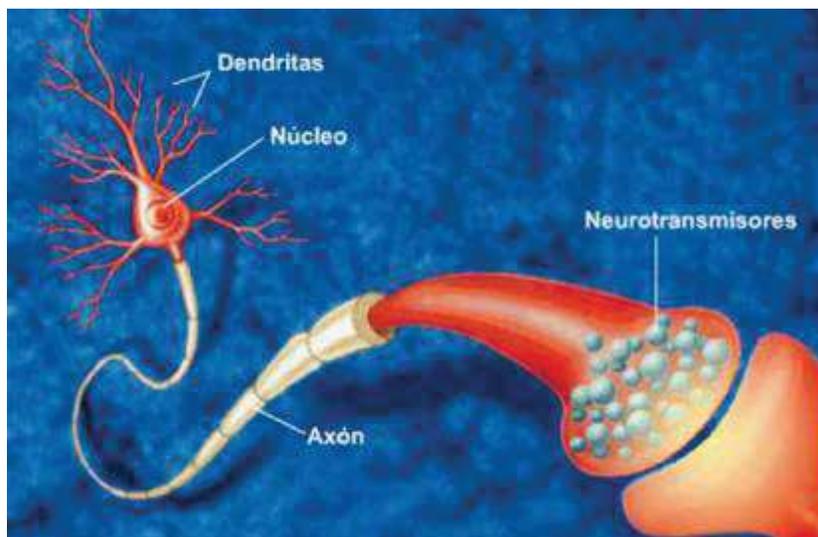


Figura 1. Neurona Biológica

El efecto es la suma de las entradas ponderadas. Si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da salida). Esta es una situación de todo o nada; cada neurona se activa o no se activa. La facilidad de transmisión de señales se altera mediante la actividad del sistema nervioso. La sinapsis es susceptible a la fatiga, deficiencia de oxígeno y la presencia de anestésicos, entre otro. Esta habilidad de ajustar señales es un mecanismo de aprendizaje.

2.2 Neurona Artificial

A partir de los aspectos funcionales de la neurona biológica, se propone un modelo de una neurona artificial como se ilustra en la siguiente imagen. Como en el caso de la neurona biológica, la neurona artificial recibe unas entradas de estímulo que pueden provenir del sistema sensorial externo



o de otras neuronas con las cuales posee conexión. Para el caso del modelo que se propone en la ilustración, la información que recibe una neurona la definimos con el vector de entradas $X = [X_1, X_2, \dots, X_n]$. La información recibida por la neurona es modificada por un vector w de pesos sinápticos cuyo papel es el de emular la sinapsis existente entre las neuronas biológicas. Estos valores se pueden asimilar a ganancias que pueden atenuar o amplificar los valores que se desean propagar hacia la neurona. El parámetro θ_j se conoce como bias o umbral de una neurona. Los diferentes valores que recibe la neurona, modificados por los pesos sinápticos, los sumamos para producir lo que hemos denominado entrada neta. Esta entrada neta es la que va a determinar si la neurona se activa o no.

La activación o no de la neurona depende de lo que llamaremos Función de Activación. La entrada neta la evaluamos en esta función y obtenemos la salida de la red. Si, por ejemplo, esta función la definimos como un escalón unitario, la salida será 1 si la entrada es mayor que cero, en caso contrario, la salida será cero. La salida y_j de la neurona se genera al evaluar la neta en la función de activación. La Figura 2.2 muestra una neurona artificial típica.

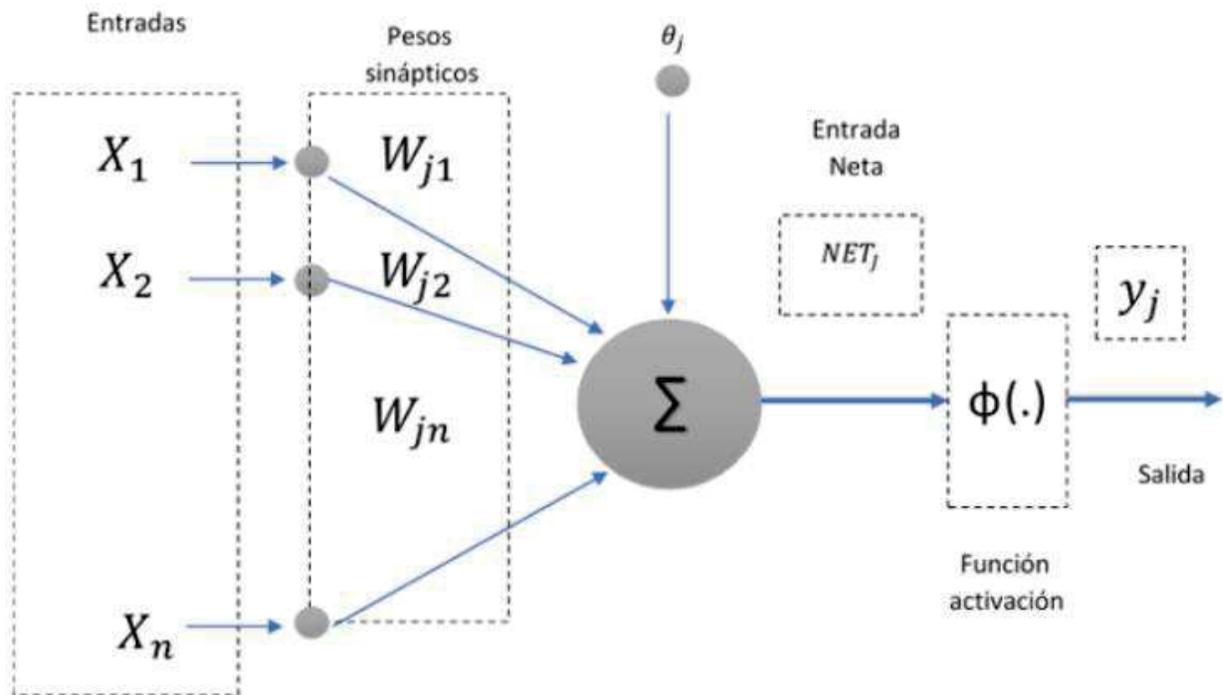


Figura 2. Neurona Artificial

2.3 Historia de las Redes Neuronales

La creación de las máquinas inteligentes junto con la evolución del hombre forman etapas. Es decir, cuando se crearon los primeros seres, primero eran pequeños seres unicelulares, los cuales con el paso del tiempo se fueron perfeccionando, hasta convertirse en unidades más complejas y numerosas.

En aquellos años cuando se crearon las primeras máquinas; la primera máquina de cálculo solo podía realizar una operación, estas no podían ser reprogramadas. Estas máquinas siguieron desarrollándose, hasta convertirse en máquinas reprogramables e inteligentes. Las redes neuronales datan de 1943. En sus inicios estas solo alcanzaban a funcionar como aproximaciones booleanas. A principios de los años 60s las redes neuronales crearon grandes expectativas, las cuales posteriormente decayeron, y significaron pérdidas de tiempo y dinero. En la actualidad se cuenta con



máquinas muchísimo más potente que las de aquel entonces, por lo cual se retomó la estructuración de una red neuronal.

A finales del siglo XIX, el científico español Santiago Ramón y Cajal logra describir por primera vez los diferentes tipos de neuronas en forma aislada. Al mismo tiempo el plantea que el sistema nervioso estaba constituido por neuronas individuales, las que se comunicaban entre sí a través de contacto funcionales llamados sinapsis. Mientras que otros científicos contradecían esta idea, concebían al sistema nervioso como una amplia red de fibras nerviosas.

McCulloch y Pitts (1943) desarrollo de algunos modelos de redes neuronales basados en su conocimiento de neurología, estos modelos se basaban en neuronas simples, consideradas como dispositivos binarios con umbrales fijos. Los resultados de sus modelos fueron funciones lógicas elementales tales como “a or b” y “a and b”. En 1949, Donald Hebb en su libro *The Organization of Behavior*, presenta el principio del aprendizaje no supervisados, conocido como la regla de Hebb [2].

2.4 Redes Neuronales Artificiales

Las neuronas de manera individual poseen una baja capacidad de procesamiento, su alto potencial de procesamiento se puede apreciar cuando dichas unidades se interconectan entre ellas, tal como sucede en el cerebro humano. En la actualidad existen varios investigadores que han estudiado estas estructuras y a su vez han propuestos redes neuronales artificiales [2].

La Agencia de Investigación de Proyectos Avanzados de Defensa (DARPA, por sus siglas en ingles), define una red neurona artificial como un sistema compuesto de muchos elementos simples de procesamiento los cuales operan en paralelo y cuya función es determinada por la estructura de la red y el peso de las conexiones, donde el procesamiento se realiza en cada uno de los nodos o elementos de cómputo

Según Haykin, una red neuronal es un procesador paralelo masivamente distribuido que tiene una facilidad natural para almacenamiento de conocimiento obtenido de la experiencia para luego hacerlo utilizables, se parece al cerebro en dos aspectos:

1. El conocimiento es obtenido por la red a través de un proceso de aprendizaje
2. Las conexiones entre las neuronas, conocidas como pesos sinápticos son utilizadas para almacenar dicho conocimiento.

De forma general podemos decir que una red neuronal artificial es un sistema de procesamiento de información que intenta emula el comportamiento de las redes biológicas.

Las redes neuronales han sido desarrolladas como generalizaciones de los modelos matemáticos del conocimiento humano o de la biología neuronal, con base a las siguientes consideraciones:

- El procesamiento de información se realiza en muchos elementos simple llamados neuronas.
- Las señales son pasadas entre neuronas a través de enlaces de conexión
- Cada enlace de conexión tiene un peso asociado, el cual, en una red neuronal típica, multiplica la señal transmitida.
- Cada neurona aplica una función de activación (usualmente no lineal) a las entradas de la red (suma de señales de entrada pesadas) para determinar su señal de salida.

La distribución de las neuronas dentro de una red neuronal se lleva a cabo mediante niveles, los cuales poseen una cantidad determinada de neuronas. Si un conjunto de neuronas artificiales recibe simultáneamente el mismo tipo de información, lo denominaremos capa. En una red podemos identificar tres tipos de niveles.



Entrada: Es el conjunto de neuronas que reciben directamente información proveniente de las fuentes externas de la red.

Oculto: Corresponde a un conjunto de neuronas internas a la red y no tiene contacto directo con el exterior. El número de niveles ocultos puede estar entre un rango bajo-alto. Las neuronas que conforman cada nivel oculto comparten el mismo tipo de información. Las neuronas de cada capa oculta pueden estar interconectadas de diferentes maneras, lo cual determina, junto con su número, las diferentes arquitecturas de redes neuronales.

Salida: Es el conjunto de neuronas que transfieren la información que la red ha procesado hacia el exterior.

En muchas redes las unidades de procesos tienen respuestas de la forma:

$$y = f\left(\sum_k \omega_k x_k\right) \quad (1)$$

Donde:

x_k : Señales de salida de otros nodos a otras entradas

ω_k : Pesos de ligas de conexión

$f(\cdot)$: Función no lineal simple.

La función f puede ser sigmoideal, tangente hiperbólica, escalón etc. La estructura de una red neuronal puede ser arbitraria. Mientras que la estructura de las capas ya está definida, es decir, las entradas externas son la primera capa, mientras que las salidas son el último cambio, sin embargo, las capas internas que no son nombradas como entradas o salidas, se les llaman capas ocultas.

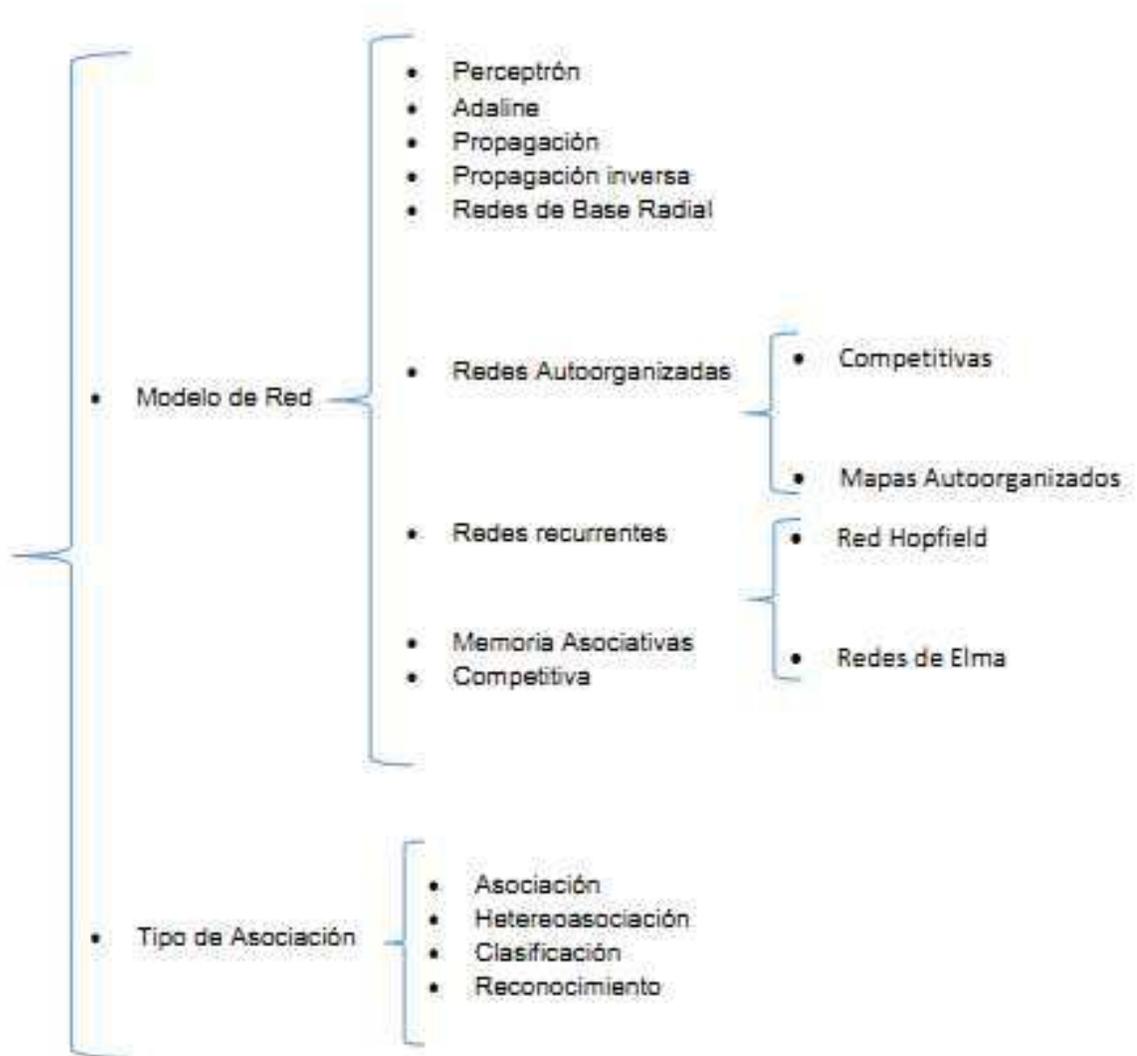
La entrada total u de una unidad K es la suma de los pesos de las entradas conectadas más un bias θ :

$$u = \sum_j \omega_j x_j + \theta \quad (2)$$



2.4.1 Clasificación de las Redes Neuronales





2.4.2 Red del Perceptron

El Perceptron es una red monocapa pues la capa de entrada, al no realizar procesamiento alguno sobre la señal de estímulo, no se tiene en la cuenta y por tanto, esta red posee solo una capa de procesamiento que es la misma de salida. Posee conectividad total, ya que la neurona de la capa de salida está conectada a todas las unidades o neuronas de entrada. Además, por lo general se le implementan unidades de tendencia o umbral para que la superficie de separación no se quede anclada en el origen del espacio n-dimensional en donde se esté realizando la separación lineal, ya que existen algunos problemas de clasificación de patrones que sin el término de umbral no tendrían solución. La función de activación que utiliza la neurona de salida es la tipo escalón binario $[0,+1]$ o bipolar $[-1,+1]$.

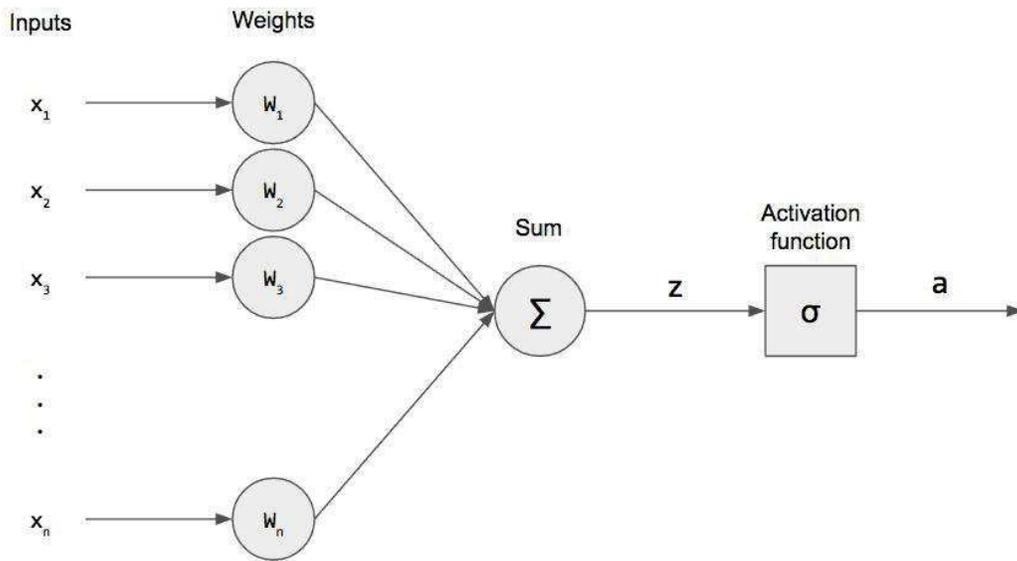


Figure 1: Estructura de la red del Perceptron

Para comprender de mejor manera el funcionamiento de un Perceptron, nos apoyaremos en la figura 5, el comportamiento de esta red cuando la dimensión del espacio de entrada es igual a dos, esto significa que la red tendrá dos entradas y por simplicidad se supondrá una sola neurona de procesamiento.

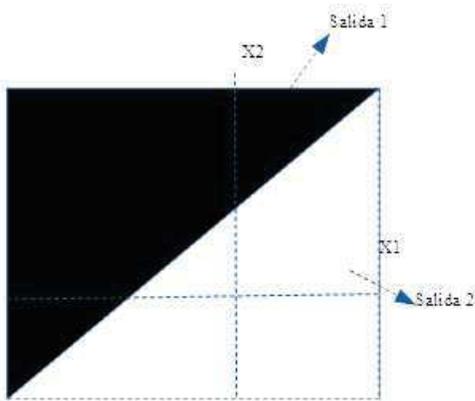


Figure 2: Perceptrón para trabajar puntos en dos dimensiones

Para analizar matemáticamente cuál es la superficie de separación que genera el Perceptron en el plano cartesiano, tomemos la expresión de la entrada neta,

$$Neta = X_1W_1 + X_2W_2 + theta \tag{3}$$



La superficie de separación se tiene exactamente cuando la entrada neta es igual a cero,

$$X_1W_1 + X_2W_2 + \text{theta}=0 \quad (4)$$

Teniendo esta condición, reorganicemos la expresión de la siguiente manera,

$$X_2 = \frac{X_1}{W_1}X_1 - \frac{\text{theta}}{W_2} \quad (5)$$

La superficie de separación generada por el Perceptron corresponde a una línea recta y, este comportamiento, se puede extender a problemas donde tenemos tres entradas y el espacio de trabajo será tridimensional. En este caso la superficie de separación corresponderá a un plano con la siguiente ecuación,

$$X_3 = \frac{W_1}{W_3}X_1 - \frac{W_2}{W_3}X_2 - \frac{\text{theta}}{W_3} \quad (6)$$

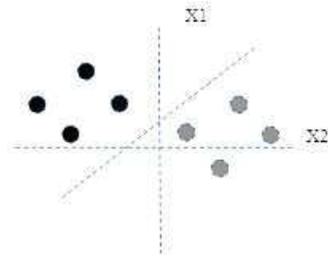


Figure 3: Regiones de separación de un Perceptrón

Extendamos el análisis a problemas donde la dimensión del vector de patrones de entrada del Perceptrones n, ahora la superficie de separación lineal es de dimensión (n-1) que denominaremos hiperplano. Cuando entrenamos un Perceptron modificaremos sus pesos, lo cual trae como consecuencia la variación de la superficie de separación que está generando. En la clasificación de patrones podemos aplicar directamente los conceptos de Perceptron, por ejemplo, tenemos un conjunto de datos con dos clases que representamos en el plano de la figura 6 con círculos negros y grises. El Perceptron propuesto para resolver esta tarea genera una línea que separa las dos clases adecuadamente.

2.5 Circuito de una Neurona Artificial

Antes de llevar a físico nuestro circuito es importante que realicemos una simulación, esto con la finalidad de corroborar que nuestro circuito funciona adecuadamente o ver si existe algún fallo, la simulación es un acercamiento al funcionamiento físico. Para realizar dicha simulación hemos empleado Proteus 8.1, el cual cuenta con los componentes necesarios para llevar dicha simulación. La figura 7 muestra el diagrama esquemático que representa una neurona mediante un circuito. En la figura 8 se observa la simulación del circuito utilizando el software Proteus.

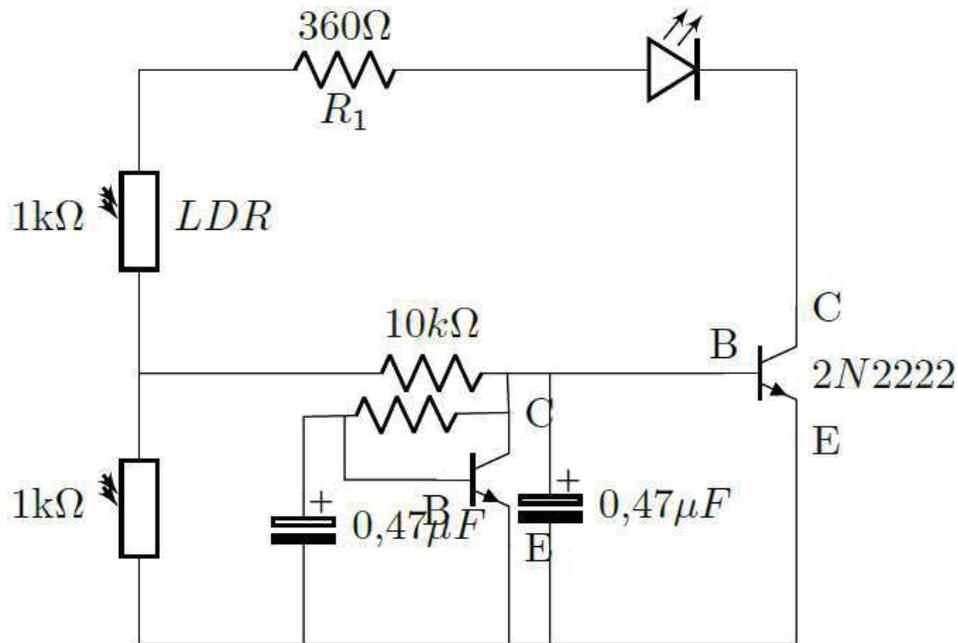


Figure 4: Circuito de una neurona artificial

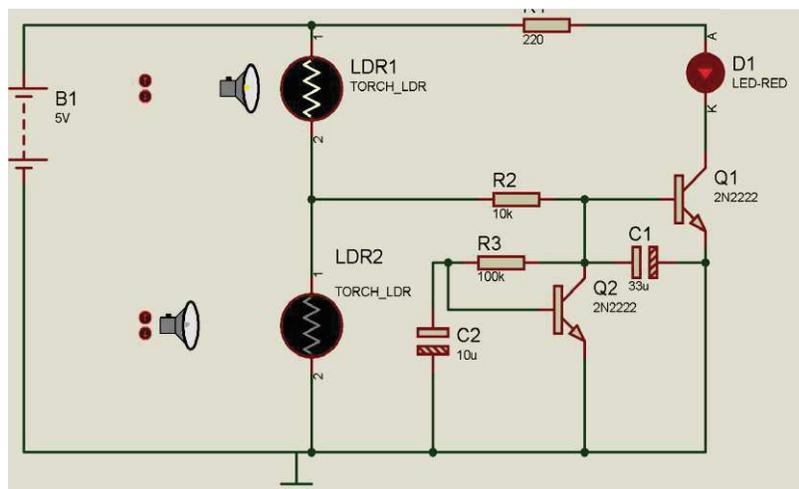


Figure 5: Simulación de una neurona artificial en Proteus



2.5.1 Funcionamiento del circuito

Se aplica un voltaje de 5V el cual es proporcionado por una fuente DC; tras aplicar dicho voltaje una corriente circula por aquellos lugares donde le he más fácil pasar. Entre estos lugares esta la ruta que va hacia el Ánodo del Led, por la foto resistencia la corriente que pasará será muy poco, esto se deberá su alta resistencia; si sobre la fotorresistencia incide un haz de luz lo suficientemente grandes, dicha resistividad disminuirá, permitiendo así la circulación de una corriente por la resistencia de 10k, la que se encuentra ubicada en la base del transistor. La base del transistor es activada cuando alcanza un voltaje mayor o igual a 0.7V, lo cual genera que colector-emisor conmute haciendo encender led.

Al observar el circuito podemos ver que el led se mantiene encendido solo cuando la primera fotorresistencia le incide un haz de luz sobre su membrana. Siempre y cuando la LDR 4 este apagada. Si hacemos incidir luz sobre la membrana de la segunda LDR la corriente ya no circulara hacia la base del transistor Q4, sino por la conexión colector-base, ocasionando el capacitor se cargue y haya fuga de corriente hacia la tierra.

2.5.2 Construcción del circuito

Al armar en físico el circuito y activarlo pudimos observar que las fotorresistencias deben estar lo suficientemente aisladas para que la activación sea más precisa, ya que esta son un poco inestables tras el estar expuestas totalmente a la luz natural. Para poder observar y emular de forma más cerca el funcionamiento de una neurona y la actividad que realiza tras ser actividad, se requerirá de fibra plexy glas, ya que es un tipo de fibra muy especial (en cuanto a características) la cual nos permitirá observar como viaja el haz de luz generado por la salida de la neurona (LED).

De igual forma se deberá realizar un ajuste con respecto a las resistencias y capacitores, la primera con la finalidad de que estas sean capaz de proteger y hacer que el paso de corriente y voltaje sea el requerido para la activación y desactivación. Los capacitores deberán aumentarse (33-10uF) esto con la finalidad de que la velocidad de respuesta de las neuronas se acelere o retarde [3].

A continuación se presenta la tabla 1 que contiene los dispositivos utilizados en la construcción de la neurona. La neurona construida se puede observar en la figura 9 se muestran 3 neuronas en protoboard.

Tabla 1: Materiales para la construcción de una Neurona

Material	Características
2 fotorresistencias	1MΩ
3 resistencias	330Ω, 10kΩ (2)
2 Capacitores	1μF
2 Transistores	2n2222
Led	Rojo $V_{led} = 1.9V$ $I_{led} = 20mA$
Fuentes de voltaje	5V cd
Fibra Óptica	20 cm
Thermofit	30 cm

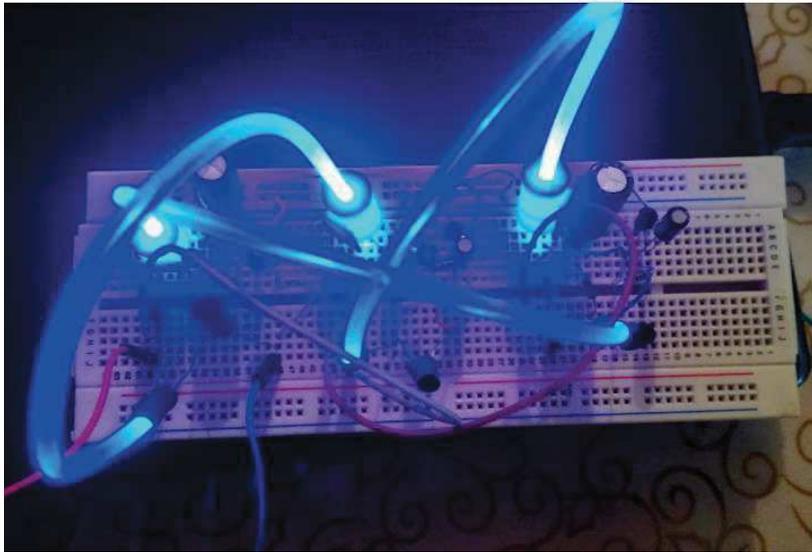


Figura 9. Tres neurona Interconectadas

3. Resultados

Se realizaron solo tres circuitos electrónicos para observar, hacer pruebas de conexión y determinar la forma en que cada unidad que conforma la Red Neuronal estará conectadas. Durante el entrenamiento de la red obtuvimos gráficas las cuales nos permiten ver si está aprendiendo de manera eficiente y el grado de error generado y por mejorar. A continuación se muestra la gráfica generada durante uno de los entrenamientos: La gráfica se ha generado a partir del primer entrenamiento. La figura 10 muestra la gráfica de la evolución del error al progresar las iteraciones. La gráfica de la figura 11 se puede ver cómo se comporta el entrenamiento de la red a medida que evolucionan las iteraciones.

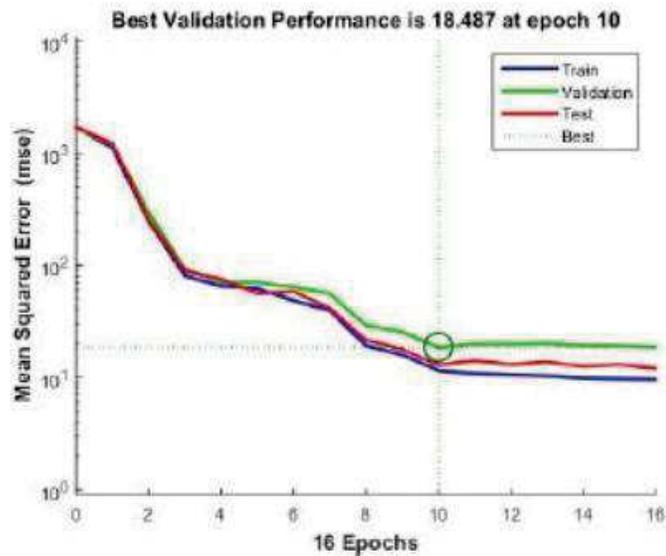


Figura 10. Evolución del error

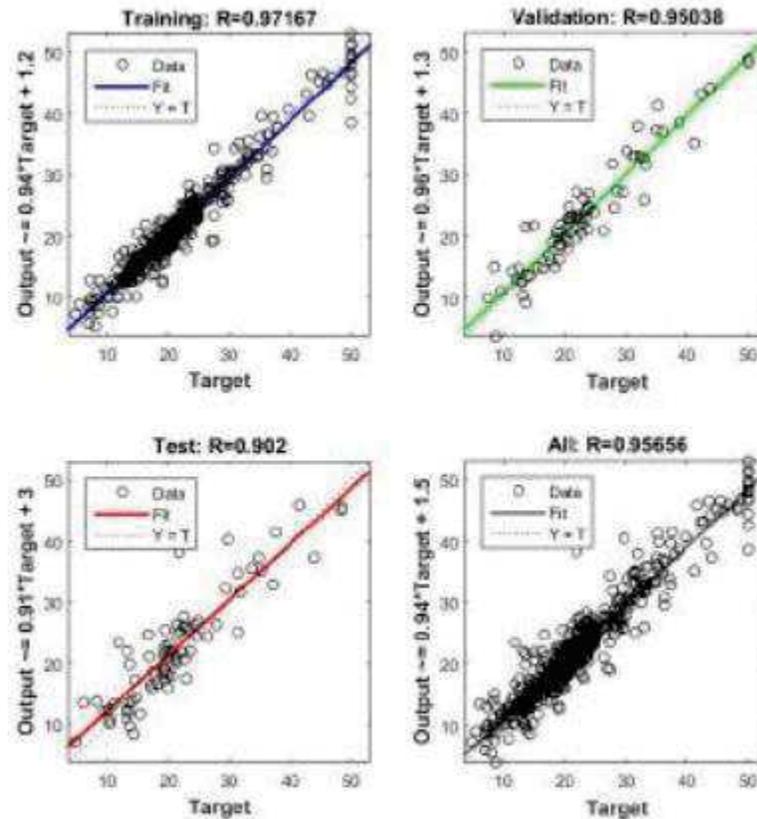


Figura 11: Resultados obtenidos durante el entrenamiento de la red

4. Conclusiones

La emulación del cerebro humano es un tanto compleja, pero el uso de redes neuronales nos permite al menos conocer un poco más de cerca la forma en que las neuronas se comunican, y poder visualizar como es que cada una de ellas trae un pulso eléctrico (que pareciera simple) viaja información hacia otras unidades, para finalmente obtener un resultado, en el caso del programa realizado en el proyecto, el obtener un número tanto en imagen como en la ventana de comando de MATLAB.

Si bien solo se pudo realizar la elaboración de 3 neuronas; debido a que la red consta en su totalidad de 80 unidades aún falta material para poder construir las demás y crear la red. En esta etapa del proyecto se logró comprender e interactuar más de cerca con las funciones de una neurona, así como comprender el papel que juega cada componente electrónico en la comunicación.



La siguiente fase en la que nos encontramos ahora trabajando es el armado de la red total.

Es así como nosotros hemos tenido un acercamiento y lo seguimos teniendo con la IA, y pretendemos que al finalizar por completo este trabajo podamos contribuir a que más jóvenes se interesen por esta área y empleen y apliquen redes neuronales a más proyectos, permitiendo así que dichos sistemas generados cuenten con un mayor rendimiento.

Referencias

[1] Caicedo E., López J., *Un aproximación práctica a las redes neuronales artificiales* 1a ed. Santiago de cali: Programa editorial universidad de valle, pag.218, 2009.

[2] Ponce, Pedro, *Inteligencia artificial con aplicaciones a la ingeniería* 1a ed. México: Alfaomega Grupo editor, pag.378, 2010.

[3] Zubiaga A., Cillerruelo L., *Cuidados intensivos .Redes Neuronales y Computación Conexionista DIY* 2da Art, Science, City International Conference ASC2015 Valencia, Universitat Politècnica de valencia, pag.8, 22-23 octubre 2015.



Máquina Selladora de Bolsas

Fuentes Flores Lorena

Facultad de ingeniería, Universidad Autónoma de Querétaro, Querétaro, México
loreflores90.lf@gmail.com

Resumen

En este artículo se presenta el proyecto “Maquina selladora de bolsas” con el cual se pretende crear un artefacto el cual brinde el apoyo necesario para un proceso de embolsado rápido, eficiente y a un bajo costo donde la producción no se vea afectada, mediante etapas automatizadas y jerarquización de tareas. Este proyecto utiliza el método de sellado térmico que, mediante el calentamiento de una resistencia, elaborada con alambre de nicrom, por medio de un transformador y presión ejercida de un pistón sobre un área cubierta con un polímero de nombre politetrafluoroetileno la bobina de plástico utilizada en el proceso obtendrá un sellado y al mismo tiempo un corte. Las características de la bolsa quedaran a elección del usuario tal como el largo, ancho y calibre de bolsa.

Palabras clave: automatización, sellado, producción.

1. Introducción

En la industria existen diversos procesos los cuales requieren empaquetados de todo tipo, desde empaquetados sencillos que no afectan los agujeros pequeños en el embolsado hasta procesos alimenticios que requieren de un total aislamiento del ambiente exterior y ciertos estándares de calidad específicos. Se requieren de diferentes tipos de bolsas de diversos materiales, calibres y tamaños para los diversos procesos industriales a los cuales van a ser implementadas.

En la industria existe una gran cantidad de procesos que en algún punto de su producción se requiere un embolsado de producto, existen empresas cuyo trabajo es generar “bobinas”.



Figura 1. Bobina de plástico

(Figura 1) de bolsas las cuales son rollos de bolsa los cuales son continuos para generar las bolsas del largo que el cliente desee. La mayoría de estas empresas solo se dedica a generar dichas “bobinas” de plástico y es necesidad de los consumidores de buscar alternativas para procesar este producto y obtener las bolsas que requieren.



Figura 2. Selladoras de bolsa manual

Existen algunas soluciones no muy caras para procesar este tipo de bobinas, pero estos procesos son manuales (Figura 2) y toma al menos a una persona para realizar este trabajo y obtener alrededor de dos bolsas por minuto, lo cual en muchos procesos es una cifra muy pequeña en comparación a lo que consumen diariamente. Las selladoras de bolsa manuales actúan por calor, el cual puede ser regulado dependiendo del plástico de la bolsa y su resistencia. Trabajan solo con este tipo de bolsas ya que no aceptan bolsas de papel o de tela. Estas selladoras de bolsas de plástico y polietileno son de sobremesa.

Con la invención de las máquinas selladoras de plásticos se dio un paso gigantesco para agilizar, en los procesos de fabricación y comercialización, el envasado y empaclado de los productos que admiten este tratamiento. El universo de productos que pueden ser envasados o empacados mediante termo selladoras es inmenso. A continuación, se mencionan algunos ejemplos de ellas:

Selladora eléctrica de pedal. La selladora de pedal utiliza una niquelina que funciona como resistencia eléctrica, la cual se calienta y sella la funda de plástico. Esta máquina selladora funciona con un pedal que permite que el cabezal superior descienda y genere una presión para sellar la funda.



Figura 3. Selladora de pedal

Sistema de funcionamiento:

- Funciona mediante un sellado térmico por niquelina.
- Este sistema de sellador es momentáneo, se calienta y realiza la presión solo cuando se active el pedal para el sello.
- La niquelina posee rugosidades que evitan que el plástico se adhiera.

Máquina continua de banda de sellado horizontal. Este tipo de selladora se utiliza para todo tipo de productos sólidos, ubicando el producto horizontalmente en la máquina.



Figura 4. Máquina selladora horizontal.

Sistema de funcionamiento

- La sujeción y transporte del producto se realiza mediante bandas.
- El sistema de transmisión de movimiento de las bandas es por poleas.
- El sellado se realiza mediante el calentamiento de mordazas, el cual transfiere el calor hacia las bandas que son de níquelina y sella la funda.
- El codificado es por rodillo de tinta sólida que es activado por un sensor fotoeléctrico.
- Tiene un sistema de regulación de temperatura y velocidad de banda.

Máquina continua de banda de sellado vertical. Este tipo de máquinas son diseñadas por el sellado de bolsas que tengan un peso menos a 20 kg y el producto permanezca vertical.



Figura 5. Máquina selladora vertical

Sistemas de funcionamiento

- La sujeción y transporte del producto se realiza mediante bandas.
- Tiene dos motores, el uno se utiliza en sistemas de transmisión de movimiento de la banda transportadora y el otro motor permite la transmisión de movimiento a las poleas que sirven de sujeción y transporte del producto.
- El sellado se realiza mediante el calentamiento de mordazas, el cual transfiere el calor hacia las bandas que son de níquelina y permite el sellado de la funda.
- Tiene un sistema de regulación de temperatura y velocidad de banda.

Máquina selladora tipo L. Esta máquina es utilizada ampliamente para embalajes de contracción de varios productos como comestibles, bebidas, medicamentos, herramientas, cosméticos, juguetes, etc.



Figura 6. Máquina selladora tipo L

Sistema de funcionamiento

- Sellado de la funda es manual y en forma de L. Se necesita un embalaje de doble hoja.
- El sistema de sellado es con niquelina y controlado por un regulador de temperatura y un temporizador.

2. Planteamiento teórico

3.

- La producción de bolsas plásticas generalmente se realiza de manera manual en muchas empresas sobre todo de tamaño pequeño y mediano, generando mano de obra repetitiva y poco productiva.
- Se realizan trabajos repetitivos y poco precisos ya que las distintas medidas de bolsas varían mucho debido a error humano, ya que todo el proceso es manual y no se estandariza una calidad constante en el sellado de las bolsas.
- Se busca automatizar una tarea repetitiva para mejorar y facilitar el proceso de creación de bolsas a partir de bobinas plásticas y al igual proveer de un sistema que genere un sellado de suficiente calidad para los distintos usos industriales.

4. Objetivos

General:

El objetivo general de este proyecto es automatizar el proceso de producción de bolsas a partir de bobinas plásticas (figura 1) generando una máquina productiva y de un precio intermedio entre las máquinas manuales y las máquinas de producción a mayoreo.

Secundarios:

- Automatizar un proceso repetitivo y con diversos errores generados por la imprecisión humana.
- Generar una máquina de mayor precisión y controlable para el proceso de creación de bolsas a partir de una bobina plástica (figura 1).
- Generar una máquina con un precio intermedio, menor a las máquinas que generan este mismo tipo de trabajo, pero con un precio mucho más alto que las máquinas manuales utilizadas en la industria, dándole mayor facilidad a las industrias pequeñas y medianas para poseer este tipo de maquinaria.

Generar un procesos sencillo y preciso para no generar muchas variables que puedan conducir a más errores o precios muy elevados en el proceso de generación de bolsas.



5. Metodología

Se busca una máquina que procese las bobinas de plástico (figura 1) de diversas medidas y genere las bolsas al gusto del consumidor, generando el menor desperdicio posible de producto plástico y otorgando un sellado resistente para los diversos usos a los que las bolsas serán sometidas.

Esto generara un mejor rendimiento de las personas que estén a cargo de generar las bolsas a partir de las bobinas plásticas y un menor desperdicio del material.

5.1 Recursos materiales y humanos

Para la realización de este proyecto y el alcance de su objetivo se requieren de diversos recursos, algunos de ellos son los siguientes:

- Componentes eléctricos y electrónicos.
- Materiales para la reconstrucción de prototipo de maquina selladora de bolsas.
- Bobinas plásticas (figura 1) para pruebas con el prototipo.
- Componentes para la generación y acondicionamiento de suministro eléctrico.

4.2 Ruta Critica

La realización del proyecto se dividirá en una cantidad definida de actividades que son necesarias realizar para llevar a cabo el objetivo del proyecto, buscando la menor cantidad de tiempo posible para llegar al objetivo propuesto

Tabla 1. Ruta crítica de proyecto

No.	Descripción	Entregable	Riesgo
1	Realizar un diagnóstico de la maquinaria existente y observando las diferentes necesidades que se pueden satisfacer	Maquinaria existente con Costos y capacidades de las mismas para realizar su propósito.	Opciones poco viables o poca maquinaria disponible.
2	Diseño de sistema eléctrico del sistema.	Reporte de diseño y funcionamiento.	No se han detectado posibles riesgos para esta actividad.
3	Desarrollo de sistema físico para manejo de las bolsas.	Sistema mecánico para control de las bobinas (Figura 1).	No se tenga a disponibilidad el material necesario.
4	Diseño e implementación de sistema electrónico de control de subprocesos mecánicos de proceso.	Reporte del sistema eléctrico e instrumental de máquina.	La disponibilidad de los actuadores y dispositivos primarios.
5	Implementación de sensores para obtener la medición requerida en las bolsas.	Un sistema capaz de manejar los sensores que serán utilizados con el correcto funcionamiento de los mismos.	Posibilidad de que los sensores no funcionen de la manera esperada o en caso de que su adquisición no sea posible.
6	Unión de sistema eléctrico, mecánico y control para el proceso completo.	Unión de los distintos componentes y sistemas para el correcto funcionamiento de la máquina.	Posible mal funcionamiento de los componentes al trabajar juntos y errores de imprecisión.
7	Pruebas y ajustes para entrega de prototipo funcional.	Manual de usuario y correcto funcionamiento del sistema automatizado.	Los principales problemas que se esperan pueden causar retrasos en la entrega, se piensa anticipar dichos problemas para entregar en tiempo y forma el proyecto presente.



6. Diseño y desarrollo

El circuito diseñado e implementado en el prototipo de máquina selladora de bolsas se conforma particularmente de una resistencia eléctrica a partir de alambre de nicrom. El alambre de nicrom (níquel-cromo – NiChrome – **NiCr**) se ha seleccionado para la aplicación especialmente debido a que es resistente a la corrosión y la oxidación a temperaturas muy altas. Este alambre tiene un equilibrio perfecto de la fuerza de tensión y requisitos de energía, haciéndolo ideal para corte, sellado y también ajustándose a las capacidades de las mayorías fuentes de alimentación.

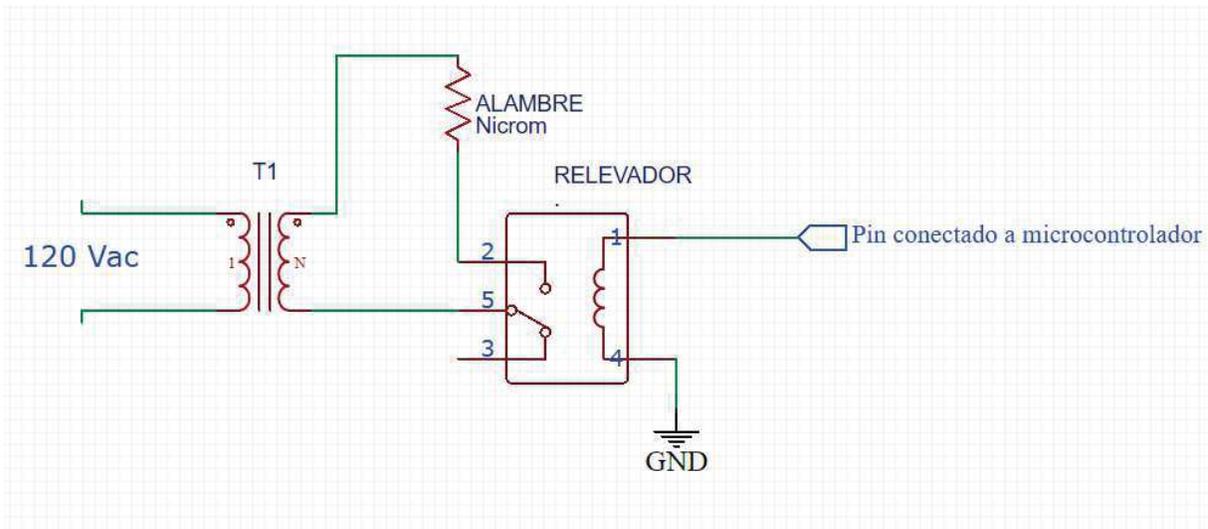


Figura 7. Circuito propuesto para sellado de bolsa

El circuito mostrado en la figura 7 es un sistema que se considera sellado por calor, consiste principalmente en un transformador que es la alimentación para calentar nuestra resistencia, un relevador el cual recibirá la señal del microcontrolador en el momento y durante el tiempo adecuado para el sello y corte de bolsa. Estas señales son considerando el grosor del material a sellar además del tamaño de la bolsa que el usuario desea.

Para el sellado del empaque, en este caso la bolsa de plástico, se ingresa el empaque a la corriente que genera calor debido al material de nuestro alambre (nicrom) que se comporta como una resistencia. Al dar presión en el empaque una vez calentada la resistencia el calor derrite el sello dando una hermeticidad final al sistema de sellado.

7. Resultados esperados, posibles aplicaciones y uso del proyecto.

El desarrollo tecnológico es considerado como el medio para poder llevar a cabo la experimentación y contribución científica.

Se espera poder aportar de manera significativa algún apoyo a la economía de las empresas medianas o pequeñas a poder adquirir maquinaria automatizada a un precio accesible para a producción de bolsas a partir de las bobinas plásticas (Figura 1) para generar materia prima de bolsas para el posterior empaquetamiento de los productos que diversas empresas manejen. Al tener un proceso automatizado se reduciría la mano de obra en tareas repetitivas y automatizables para generar una mayor producción y con una mejor calidad y mayor productividad que la maquinaria manual para el sellado de las bolsas.



Figura 8. Proceso de sellado en prototipo realizado



Figura 9. Bolsa sellada y cortada al finalizar proceso en prototipo realizado

En el prototipo realizado se ha implementado el sello y corte con alambre de nicrom con la finalidad de poder ofrecer un producto de buena calidad y a un costo accesible debido al público al que se quiere dirigir.

Para finalizar, se espera que tenga una aceptación entre el público en general, principalmente en empresas que utilicen bolsas a la medida, empresas de tamaño mediano o pequeño con mayor dificultad a pagar maquinaria de mayor producción.

En resumen, la propuesta aquí planteada se compromete a generar lo siguiente:

- La gestión para obtener los derechos de un modelo de utilidad.
- Un prototipo de máquina selladora de bolsas.

8. Conclusiones

El sistema que se ha propuesto pretende, si bien ya es común en el sector productivo, reducir el costo del trabajo humano teniendo como resultado una baja cifra en los errores en el área de sellado de bolsas (elevar la calidad) y un aumento de seguridad para el operador que manipule la máquina. De la mano se encuentra principalmente la reducción de costes mediante la integración de aplicaciones que sustituyen los procesos manuales, acelerando el tiempo de ejecución de las tareas y eliminando los posibles errores humanos.



Referencias

- [1]"LA JORNADA ECOLOGICA", Jornada.com.mx, 2019. [Online]. Available: <https://www.jornada.com.mx/2013/05/27/eco-m.html>.
- [2]S. Científicas, Agenciasinc.es, 2019. [Online]. Available: <https://www.agenciasinc.es/Multimedia/Ilustraciones/El-plastico-nace-por-accidente>.
- [3]"Breve Historia del Polietileno", Educación en Ingeniería Química, 2019. [Online]. Available: <https://www.ssecoconsulting.com/breve-historia-del-polietileno.html>. [Accessed: 27- Mar- 2019].
- [4]"Colegio Anglo Español", prezi.com, 2019. [Online]. Available: <https://prezi.com/uoapqkrc1ann/colegio-anglo-espanol/>.
- [5]"La importancia de automatizar procesos en una empresa | Ricopia", Ricopia. Soluciones y Servicios Tecnológicos, 2019. [Online]. Available: <https://www.ricopia.com/la-importancia-automatizar-procesos-una-empresa/>.



Robot Neumático de Seis Grados de Libertad

Monter Hernández Adolfo¹, Alegría Anaya Enrique¹, Ramos Arreguín Juan Manuel²✉,
Gorrostieta Hurtado Efrén² y Aceves Fernández Marco Antonio²

¹Universidad Tecnológica de San Juan del Río, Ingeniería Mecatrónica

²Universidad Autónoma de Querétaro, Facultad de Ingeniería

✉jsistdig@yahoo.com.mx

Resumen

Es conocido que en el medio industrial la aplicación de los robots con actuadores eléctricos juega un papel muy importante en las líneas de producción. Sin embargo, presentan algunas desventajas, como es el alto consumo energético, la baja relación potencia-peso, y los altos costos de mantenimiento. A pesar de esas desventajas, los actuadores eléctricos tienen un comportamiento más lineal que un actuador eléctrico. Como contraparte, tenemos los actuadores neumáticos, los cuales son más económicos respecto a los eléctricos, tienen una mayor relación potencia-peso y su mantenimiento es más sencillo y económico. Sin embargo, los actuadores neumáticos tienen la desventaja de tener un comportamiento altamente no lineal, y por lo tanto, son más difíciles de controlar. En este trabajo se presenta el diseño de un robot neumático de 6 grados de libertad, así como el proceso de diseño llevado a cabo. Se muestra también el desarrollo de un controlador básico para un grado de libertad, para demostrar que es posible controlar el comportamiento del robot. Se presentan imágenes del robot diseñado, así como el controlador empleado.

Palabras clave: Actuador neumático, no lineal, diseño, control básico.

1. Introducción

Actualmente, los robots han cobrado una gran importancia, tanto a nivel industrial como a nivel comercial y de entretenimiento. Los robots industriales son muy importantes en las líneas de producción de grandes empresas, por ejemplo, armadoras de autos. En el caso de los robots comerciales, podemos tener de entretenimiento y de aplicación para solución de problemas como salud, aseo, entre otros. De la misma manera, se ha usado los actuadores neumáticos para el desarrollo de robots insectos.

En el caso de los robots industriales, es generalizado el uso de actuadores eléctricos para la generación de movimiento en los robots. Lo anterior, debido a que el control de este tipo de actuadores es más sencillo. Sin embargo, los actuadores eléctricos tienen una baja relación potencia-peso, su mantenimiento es costoso y se dañan con relativa facilidad. Por otro lado, tenemos los actuadores hidráulicos que tienen la mayor relación potencia-peso y su comportamiento es lineal. El problema de los actuadores hidráulicos es que son muy contaminantes, y su mantenimiento es muy caro, además de requerir instalaciones especiales para soportar los niveles de presión que se usa. Finalmente, tenemos los actuadores neumáticos, los cuales tienen una relación potencia-peso media, no se dañan por sobrecarga, son económicos, utilizan aire a presión para generar el movimiento, por lo que no contaminan. El principal problema de los actuadores neumáticos es su comportamiento altamente no lineal. Este comportamiento quedó establecido en [1], donde se establece el modelo termo-mecánico de un actuador neumático.



El desarrollo de este tipo de robots, inicia con Kiyama [1], donde desarrolla el modelo termomecánico aplicado a un robot con actuador neumático de 1 grado de libertad (DOF). El uso de este modelo termo-mecánico ha permitido desarrollar otros trabajos enfocados al uso de actuadores neumáticos en brazos robóticos, que ha resultado en un desarrollo de este tipo de robots. En 2006 se desarrolla un modelo termomecánico simplificado, para una mayor velocidad de procesamiento de la información [2]. Posteriormente, en el mismo año 2006, se presenta un controlador para un robot neumático de 1 grado de libertad [3]. En 2008 se presenta un controlador difuso para el actuador neumático de un robot de 1 grado de libertad [4]. Finalmente, en 2016, se presenta una implementación de controlador embebido en FPGA [5].

Lee aplica actuadores neumáticos en el diseño y construcción de un robot de 3 grados de libertad (DOF), donde se combina con el uso de cámaras y control basado en lógica difusa para el posicionamiento del brazo [6]. Smita trabajó en el uso de un robot neumático en una línea de ensamble [7].

También se ha trabajado en brazos neumáticos de bajo costo. Reportan un buen funcionamiento y similar a utilizar los actuadores tradicionales. Sin embargo, el robot no podría levantar un gran peso [8].

Este trabajo busca establecer las bases del desarrollo de un robot neumático de 6 DOF. Se presenta una aproximación de lo que puede ser el controlador, básicamente para demostrar que el robot es viable, y se debe trabajar aún más en ello.

2. Metodología de Desarrollo

El desarrollo de la construcción del robot neumático se ha realizado de acuerdo a la metodología de la figura 1.

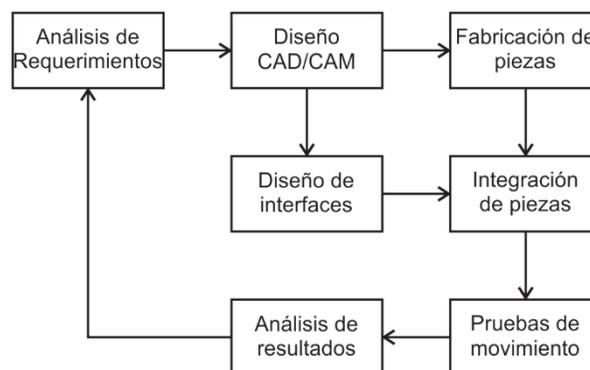


Figura 1. Metodología para la fabricación de un robot neumático.

2.1 Requerimientos.

En esta primera etapa, se ha definido la necesidad de un robot neumático de 6 grados de libertad, para emplearse en tareas de posicionamiento, o también llamado “*pick and place*”. Se requiere que el robot tenga una configuración similar a las utilizadas por robots industriales. Para esto, se consideró configuraciones como paralelo [9], scara [10], cartesiano [11], cilíndrico [12], polar [9] y angular [14].



2.2 Diseño CAD/CAM.

Se diseñó la estructura del robot neumático utilizando el software Solidworks. La figura 2 muestra el diseño de la mesa que va a soportar la estructura completa del robot. La figura 3 muestra el motor neumático utilizado para hacer girar el primer grado de libertad del robot. La figura 4 presenta uno de los actuadores utilizados para desplazamiento lineal. La figura 5 muestra el mecanismo de 4 barras acoplado a la estructura y al pistón neumático para transmitir la fuerza del actuador neumático, y lograr el movimiento θ del brazo. Así mismo, la figura 6 muestra el diseño del efector final (gripper), para que el robot pueda sostener algo y moverlo de lugar.

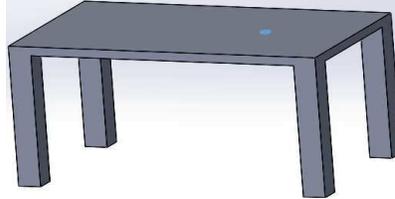


Figura 2. Mesa que soporta al brazo manipulador completo.



Figura 3. Motor neumático CRB2BW15-270SZ, utilizado para movimientos rotacionales.



Figura 4. Pistón neumático CM2YC32-150.

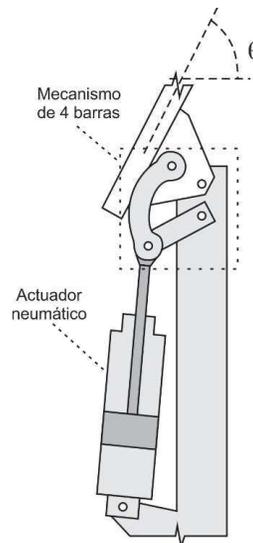


Figura 5. Acoplamiento del pistón neumático y el mecanismo de 4 barras en la estructura del robot.



Figura 6. Efector final para el robot neumático.

Derivado del diseño desarrollado, la figura 7 muestra el diseño conceptual del robot neumático.

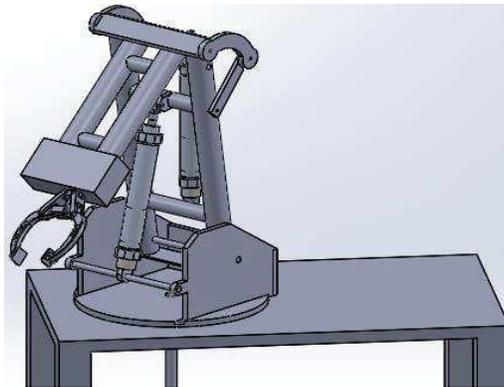


Figura 7. Diseño conceptual del robot neumático.

2.3 Desarrollo de la interfaz electrónica.

La interfaz electrónica ha sido implementada en una tarjeta de desarrollo Basys2, la cual se basa en un FPGA de Xilinx. Se realiza la implementación de control de 1 grado de libertad, para en otro trabajo realizar el control del robot completo. La descripción de hardware ha sido diseñada considerando la entidad de la figura 8.

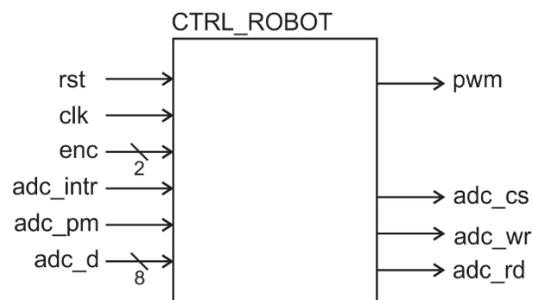


Figura 8. Entidad de la descripción de hardware del control del robot.

La tabla 1 muestra la descripción de terminales involucradas en la figura 8.



Tabla 1. Descripción de terminales de la entidad CTRL_ROBOT.

Señal	Descripción	Valores
rst	Inicializa el sistema digital	Activo en alto.
Clk	Reloj	50MHz, activo en borde positivo
Enc	Información del encoder	Binario
Adc_intr	Interrupción de conversión completa.	Activo en bajo
Adc_d	Dato del dato convertido	Binario
Adc_cs	Selector del convertidor	Activo en bajo
Adc_wr	Habilitación de conversión.	Activo en bajo
Adc_rd	Habilitación de lectura.	Activo en bajo
Pwm	Salida que controla el flujo de aire.	Binario

La figura 9 muestra la arquitectura de hardware correspondiente a la figura 8. En ella se puede apreciar el uso de un controlador PID, un controlador del convertidor ADC0804, el codificador del encoder, un contador ascendente/descendente y el generador de señal PWM.

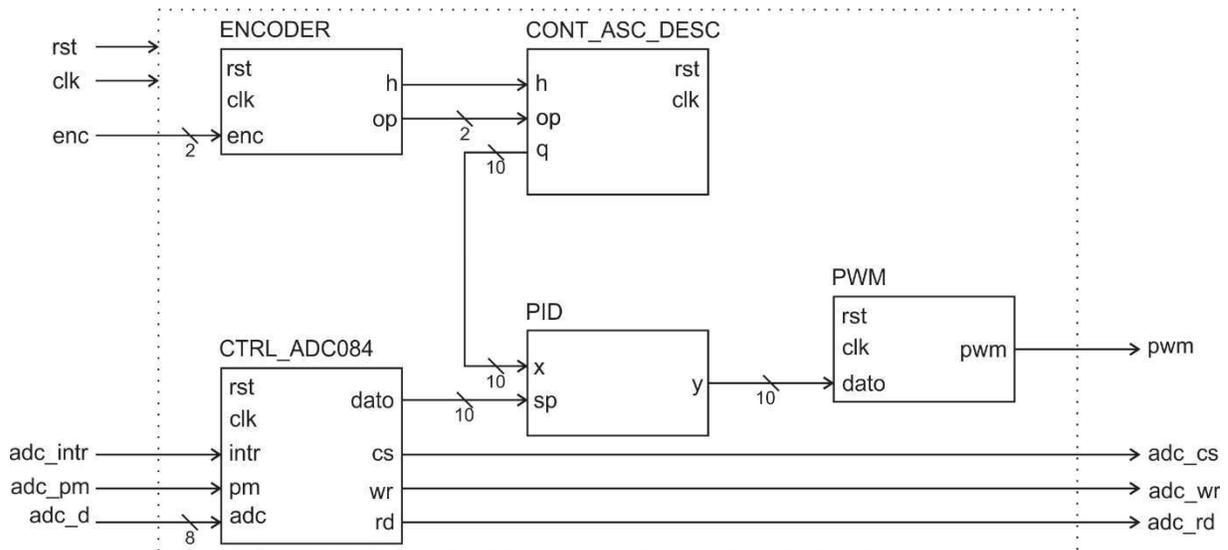


Figura 9. Arquitectura de hardware correspondiente a la entidad del control del robot.

A continuación se describe brevemente cada uno de los bloques involucrados en la figura 9.

Encoder. Está conformado por dos bits, los cuales se usan para interpretar el sentido de giro del motor. Con cada cambio, el encoder entrega una señal de 1 bit (h, activo en alto), el cual se usa como habilitador del contador. También se entrega una señal de 2 bits (OP), el cuál indica al contador si incrementa, decrementa la cuenta o el valor del contador permanece sin cambios. La figura 10 muestra la simulación de este módulo.

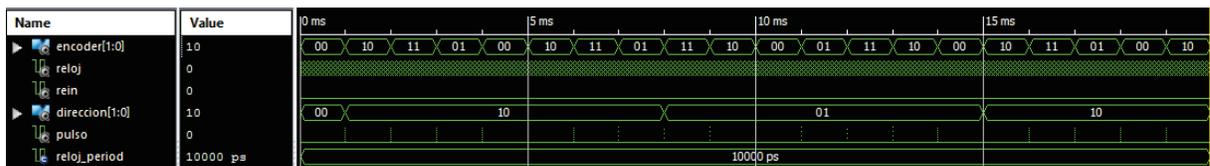


Figura 10. Simulación de la descripción del encoder.



CONT_ASC_DESC. Este módulo es un contador ascendente/descendente de 10 bits. Se usa para conocer la posición actual del motor que controla el flujo de aire hacia el actuador neumático. Utiliza las señales h y OP que provienen del bloque ENCODER. La salida de este bloque es un dato de 10 bits, que representa la posición de un motor para el control de una válvula reguladora de flujo de aire. La simulación se muestra en la figura 11.

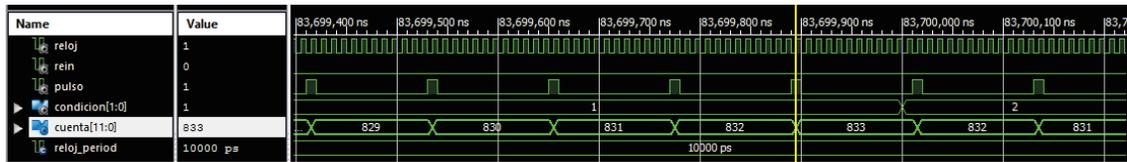


Figura 11. Simulación de la descripción del contador ascendente/descendente.

CTRL_ADC0804. Debido a que se utiliza un sensor resistivo para determinar la posición del primer grado de libertad, se requiere de este módulo para adquirir la posición, y determinar el valor de referencia (SP) que se envía al PID para que se determine la posición del motor.

PWM. Este módulo genera la señal PWM, de acuerdo a la información obtenida del PID. Conforme se acerca el brazo a la posición deseada, el valor PWM tiende a cero, deteniendo gradualmente el giro del motor, y en consecuencia, cerrando el flujo de aire.

3. Resultados

La figura 12 muestra el robot neumático de 6 grados de libertad armado. En la figura se pueden apreciar dos vistas del robot. En este robot neumático, se usa 3 motores neumáticos y 4 pistones neumáticos, para la generación de la fuerza de movimiento.



Figura 12. Robot neumático de 6 grados de libertad.

En la figura 13, se muestra la tarjeta Basys 2 usada, así como las conexiones con el convertidor analógico/digital.

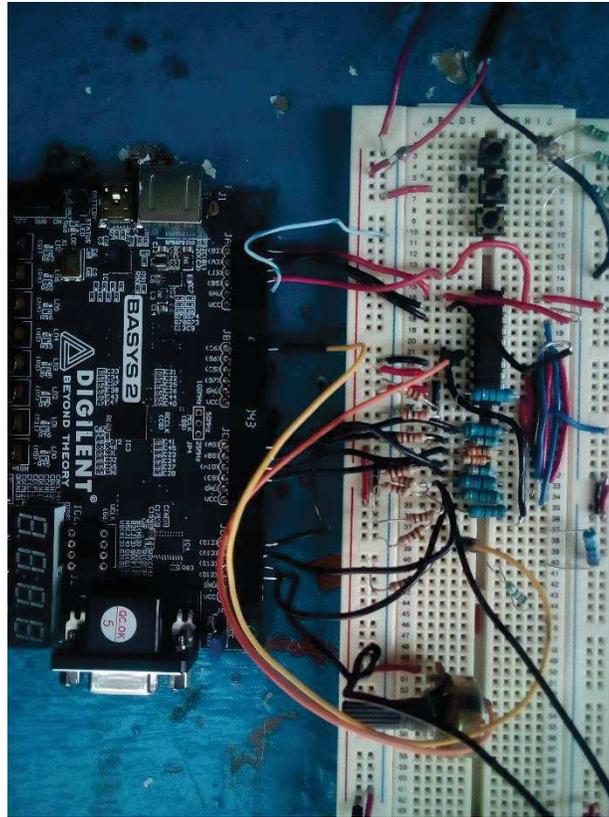


Figura 13. Sistema electrónico usado para controlar 1 grado de libertad del robot.

4. Conclusiones.

El desarrollo de robots neumáticos es un área de oportunidad muy importante, especialmente en México, debido a que la gran mayoría de robots que se utilizan son eléctricos y se adquieren por importación.

A diferencia de los actuadores eléctricos, los actuadores neumáticos son económicos, de fácil mantenimiento, y no se dañan fácilmente por sobrecarga. Debido a esto, es importante tomar en cuenta lo que se pueda realizar con robots neumáticos y generar una base de conocimiento en esta área. Esto puede ser aprovechado en el desarrollo de tecnología, así como en el desarrollo de temas de tesis y proyectos de investigación.

El robot neumático que ha sido presentado en este trabajo aún requiere de más trabajo alrededor. Sin embargo, es importante avanzar en esta línea de trabajo, para poder tener una mayor perspectiva en el desarrollo de tecnología basado en neumática.

Como trabajo a futuro está el desarrollar la electrónica necesaria para que se puedan controlar todos los grados de libertad, así como la implementación de algoritmos de control avanzados enfocados al posicionamiento del robot, pero especialmente, en la generación de trayectorias para que el robot pueda realizar una tarea específica.



Referencias

- [1] Kiyama, F. F., & Vargas, E. “*Modelo Termo-Mecánico para un Manipulador tipo Dieléctrico*”. Información tecnológica, 15(5), 23-31. 2004.
- [2] Ramos, J. M., Vargas, J. E., Gorrostieta, E., & Pedraza, J. C. “*Nuevo Modelo Polinomial del Comportamiento de un Cilindro Neumático*”. Información tecnológica, 17(3), 99-106. 2006.
- [3] Ramos, J. M., Gorrostieta, E., Pedraza, J. C., Romero, R. J., & Ramírez, B. “*Pneumatic cylinder control for a flexible manipulator robot*”. In 12th IEEE International Conference on Methods and Models in Automation and Robotics. pp. 637-641. 2006.
- [4] Ramos-Arreguin, J. M., Pedraza-Ortega, J. C., Gorrostieta-Hurtado, E., de Jesus Romero-Troncoso, R., Vargas-Soto, J. E., & Hernandez-Hernandez, F. “*Pneumatic Fuzzy Controller Simulation vs Practical Results for Flexible Manipulator*”. In *Automation and Robotics*. IntechOpen. 2008.
- [5] Juan-Manuel, R. A., Saul, T. A., Jose-Emilio, V. S., & Marco-Antonio, A. F. “*FPGA embedded pd control of A 1 dof manipulator with a pneumatic actuator*”. International Journal of Robotics and Automation, 31(3), 176-186. 2016.
- [6] Lee, L. W., Chiang, H. H., & Li, I. “*Development and Control of a Pneumatic-Actuator 3-DOF Translational Parallel Manipulator with Robot Vision*”, Sensors, 19(6), 1459, 2019.
- [7] Nadkarni, S., Mahukar, V., Mittal, N. M., Mete, V., & Wadmare, G. “*Assembly of a Product Using Pneumatic Robotic Arm and RFID Module*”. Available at SSRN 3372964. 2019.
- [8] Ahn, H. S., Jin, E., Zhu, G., & MacDonald, B. A. (2018, June). “*Design of Low Cost Pneumatic Robot Arm Control System*”. In 2018 15th International Conference on Ubiquitous Robots (UR) (pp. 151-156). IEEE.
- [9] Staicu, S, “*Dynamics of Parallel Robots*”, Springer International Publishing.
- [10] Boukadida, W., Benamor, A., & Messaoud, H., “*Multi-objective design of optimal sliding mode control for trajectory tracking of SCARA robot based on genetic algorithm*”. Journal of Dynamic Systems, Measurement, and Control, 141(3), 031015, 2019.
- [11] Raheem, F. A., Khaleel, H. Z., & Kashan, M. K., “*Robot Arm Design for Children Writing Ability Enhancement using Cartesian Equations based on ANFIS*”. In 2018 Third Scientific Conference of Electrical Engineering (SCEE), pp. 150-155, IEEE, 2019.
- [12] Alamsyah, F. A., “*The Kinematics Analysis of Robotic Arm manipulators Cylindrical Robot RPP Type for FFF 3D Print using Scilab*”, In IOP Conference Series: Materials Science and Engineering, Vol. 494, No. 1, p. 012100, 2019.
- [13] Reda, H. M., “*Task Allocation and Analysis of Common Manufacturing Processes for Humans & Robots*”, International Journal of Science and Engineering Invention, 5(03), pp. 60-70, 2019.
- [14] Matúšová, M., & Hrušková, E., “*Applying the Computer Aided Systems in Education Process*”, Management Systems in Production Engineering, 27(1), 46-50, 2019.



Control PID-Wavelet para Sistemas Lineales Invariantes en el Tiempo

García Blancas Jesús y Flores Moreno Julian

Instituto Tecnológico Superior del Oriente del Estado de Hidalgo
jgarcia@itesa.edu.mx

Resumen

En este trabajo se presenta un control basado en funciones wavelet (ondeletas) para el análisis de las señales del error a distintas frecuencias como lo son bajas, medianas y altas. El control tiene una estructura similar a un control PID (Proporcional-Integral-Derivativo) y su implementación es similar a dicho control a diferencia que el control PID sus ganancias son constantes y el PID-Wavelet sus ganancias son ajustables con respecto del tiempo bajo las condiciones de escala y traslado de una función wavelet. Los métodos de sintonización de los controles no es una tarea trivial lo cual conduce al desarrollo de controles que minimice esta tarea. Por otro lado, el consumo energético para tener un control robusto ante incertidumbres es alto, incertidumbres tal como medidas de ruido, variaciones de fricción en sistemas mecánicos y torques o tenciones de voltaje externos a diferentes escalas y únicamente se aplican a un grupo de sistemas reducidos o que cumplen con ciertas condiciones por ejemplo la linealidad de los mismos o el orden del sistema. Con las funciones wavelet como entrada de control se puede atenuar este tipo de ruido y mejorar el comportamiento de los sistemas mediante el escalamiento de las frecuencias de las señales del sistema, otra variante que nos da el control PID-Wavelet como positiva es que no se necesita del conocimiento matemático del sistema, únicamente de la entrada y la salida y el tiempo de llegada a la referencia, con todo esto ofrece ventajas que favorecen su implementación en sistemas de control.

Palabras clave: Sistema de Control, Transformada Wavelet Continua, Ruido de Baja Frecuencia.

1. Introducción

Una wavelet es una función cuya energía se encuentra concentrada en el tiempo y sirve como herramienta matemática para el análisis de fenómenos transitorios, no estacionarios y variantes en el tiempo, son funciones oscilatorias de corta duración y de energía finita lo cual las hace factibles para la representación de datos de otras funciones [2]. La idea de representar una señal como una combinación lineal de señales de duración efectiva limitada que se obtiene por traslación y escalado de una función original denominada wavelet madre. La función wavelet madre $\psi(t)$ que verifica la condición de admisibilidad

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (1)$$

Es decir tiene media cero, con lo cual se genera un conjunto de funciones $\psi_{b,s}(t)$ conocidas como wavelet hijas, por su dilatación con un factor s y traslación b , quedando de la forma:

$$\psi_{b,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-b}{s}\right) \quad (2)$$



Su admisibilidad a los cambios de señales permite la implementación en aplicaciones como son en procesos de control industrial. En este artículo se propone un control basado en funciones wavelet, de manera particular un control análogo al ya conocido PID. La utilidad de los controles PID se basa en que se aplica casi de forma general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se puede emplear métodos de diseño analíticos, es cuando los controles PID resultan útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido de los esquemas de control PID básicos y modificados han resultado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo [1]. Existen varios tipos de controles PID tales como son PID no lineales y PID con ganancias no lineales.

Las funciones wavelet ofrecen otra alternativa para el diseño e implementación de controles de alto rendimiento sustentado en los principios de un control PID. De forma general un control PID toma como entrada la señal de error $e(t)$, y actúa sobre la planta para generar una señal de control $u(t)$, como se muestra en la siguiente ecuación:

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_D \frac{de(t)}{dt} \quad (3)$$

Donde K_p , K_i y K_D son ganancias del control PID, estos tres términos representan de forma intuitiva el pasado, el presente y el futuro de la señal de error, de esta manera se puede argumentar el éxito que ha tenido este control dentro de la industria ya que su sintonización puede ser de manera intuitiva [4], este modo de operar los controles PID garantiza robustez pero se pierde optimización de la energía lo cual produce un mayor consumo energético y una corta duración de vida útil de los sensores y actuadores para la implementación del control.

En este artículo se propone de forma generalizada la estructura del control PID, donde sus ganancias K_p , K_i y K_D se proponen como funciones wavelet $\psi_{b,s}(t)$, la idea básica es que la función wavelet sea en función del error $e(t)$ quedando como se muestra en la siguiente ecuación:

$$\psi_{b,s}(e(t)) = \frac{1}{\sqrt{s}} \psi\left(\frac{e(t)-b}{s}\right) \quad (4)$$

Donde $e(t) = y_{ref} - y_{sal}$, con $y_{ref} \in \mathfrak{R}$ un valor constante y $y_{sal} \in \mathfrak{R}$ variable en el tiempo, entonces el control es representado por la siguiente expresión:

$$u(t) = \psi_{b1,s1}(e(t))e(t) + \psi_{b2,s2}(e(t)) \int e(t) dt + \psi_{b3,s3}(e(t)) \frac{de(t)}{dt} \quad (5)$$

Donde $\psi_{b_i,s_i}(e(t))$ son las funciones wavelet por determinar para la sintonización del control PID-wavelet.

En la sección 2 se presenta de forma detallada la teoría wavelet para la implementación del control propuesto en este artículo, en la sección 3 se muestra la estructura y algunos métodos de sintonización del controlador PID, por otro lado, en la sección 4 y 5 se muestran las condiciones que garantizan la admisibilidad del control PID-wavelet con una planta de primer orden y los resultados



obtenidos con el control PID-wavelet comparándolos con un PID clásico y finalmente en la sección 6 se presentarán las conclusiones más destacadas del tema.

2. Teoría Wavelet

Una función wavelet es una función $\psi(t) \in L^2$, que cumple con la condición de admisibilidad mostrada en la ecuación (1). La función wavelet es tradicionalmente llamada wavelet madre, por lo que a partir de ella se puede generar una familia de funciones de doble índice llamadas funciones wavelet hijas.

A continuación, se dan algunos ejemplos de funciones wavelet madre más comunes, entre estas se encuentran: Haar, Mexican hat, Morlet, Meyer, Daubechies, Shannon [3].

Tabla 1. Algunos ejemplos comunes de wavelets madre.

Nombre	Definición
Haar	$\psi(t) = \begin{cases} 1, si, \in [0,0.5] \\ -1, si, \in (0.5,1] \\ 0, en \dots otro \dots caso \end{cases}$
Mexican Hat	$\psi(t) = \frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} (1-t^2) e^{\left(-\frac{1}{2}t^2\right)}$
Morlet	$\psi(t) = e^{-\frac{t^2}{2}} \cos(5t)$
Meyer	$\psi(t) = \begin{cases} (2\pi)^{\frac{1}{2}} e^{\frac{i\omega}{3}} \sin\left(\frac{\pi}{2} \nu\left(\frac{3}{2\pi} \omega -1\right)\right) \\ si \dots \frac{2\pi}{3} \leq \omega \leq \frac{4\pi}{3} \\ (2\pi)^{\frac{1}{2}} e^{\frac{i\omega}{2}} \cos\left(\frac{\pi}{2} \nu\left(\frac{3}{4\pi} \omega -1\right)\right) \\ si \dots \frac{4\pi}{3} \leq \omega \leq \frac{8\pi}{3} \\ 0, en \dots otro \dots caso \end{cases}$
Daubechies	$P(y) = \sum_{k=0}^{N-1} C_k^{N-1+k} y^k$
Sannon	$\psi(t) = \frac{\sin\left(\frac{\pi}{2}t\right)}{\frac{\pi}{2}t} \cos\left(\frac{3\pi}{2}t\right)$



Una función wavelet hija, es una función generada a partir de la dilatación-contracción y traslación de una función wavelet madre y se denota como en la ecuación (2) con $s \geq 0$.

Donde la variable escala s permite hacer dilataciones y las contracciones de la señal, y la variable de traslación b permite el desplazamiento en el tiempo.

El factor de normalización $(\sqrt{s})^{-1}$ ha sido seleccionado tal que $\|\psi_{s,b}\| = \|\psi\|$, $\|\psi_{s,b}\| \forall s, b \in \mathfrak{R}$ por lo que se puede asumir que $\|\psi\| = 1$.

2.1 Análisis Multiresolución

El análisis multiresolución es una técnica que permite analizar señales en múltiples bandas de frecuencia. Consiste de una secuencia de subespacios cerrados V_i en $L^2(\mathbb{R})$:

$$\dots \subset V_2 \subset V_1 \subset V_0 \subset L^2(\mathbb{R})$$

En la Figura 1 se muestra la gráfica de descomposición de una señal referente al tiempo y la frecuencia, la propiedad multiresolución nos permite obtener detalles y características de la señal original.

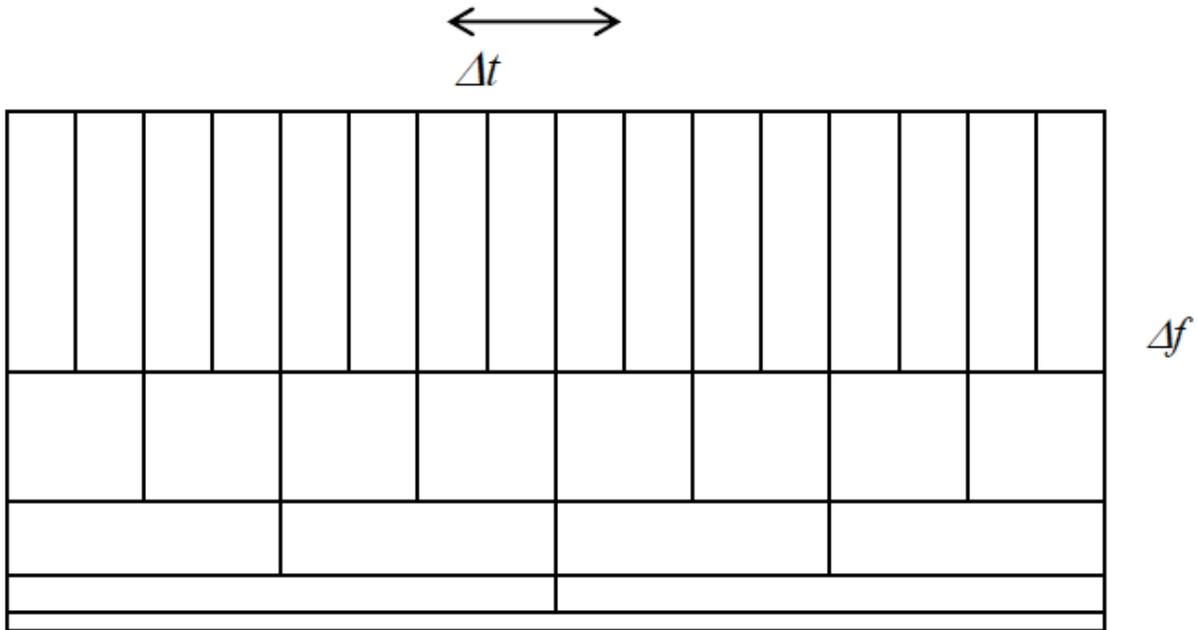


Figura 1. Gráfica de distribución de las frecuencias contra el tiempo de muestreo.

Cuando la resolución se incrementa con i tendiendo a $-\infty$ la función aproximada debería converger a la función original. Esto es:



$$\bigcup_i V_i = L^2(R)$$

Por el contrario, cuando la resolución se decrementa a cero con i tendiendo a $+\infty$, las aproximaciones contienen cada vez menos información y convergen a cero:

$$\bigcap_i V_i = \{0\}$$

Con $i \in Z [2]$.

Una de las características que ha permitido la aplicación de la teoría wavelet en diferentes campos de estudio es la capacidad de realizar un análisis en tiempo y frecuencia de fenómenos estacionarios y no estacionarios. El análisis wavelet se basa, al igual que la teoría de Fourier, en el concepto de aproximación de señales usando la superposición de señales. La diferencia entre la teoría de Fourier y la teoría wavelet radica en que las funciones wavelet varían tanto en frecuencia como en escala.

Una forma general de ver las funciones wavelet es como familias de funciones que tienen una buena localización tanto en frecuencia como en tiempo. Esta característica hace del análisis wavelet una herramienta poderosa para diversas aplicaciones que involucran el procesamiento de señales.

3. Control PID

La estructura de un controlador PID es simple, pero también se convierte en su debilidad, dado que se limita a sistemas simples, por otro lado, tiene gran aplicación en la industria por esa facilidad de implementación.

3.1 Estructura PID

Consideremos un sistema de control en lazo cerrado de una entrada y una salida (sistemas SISO)

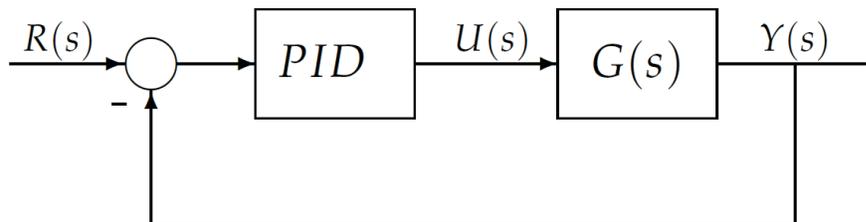


Figura 2. Diagrama en bloques del control PID.

Los miembros de la familia de controladores PID, incluyen tres acciones: proporcional (P), integral (I) y derivativa (D).

Proporcional (P), da una salida del controlador que es proporcional a la señal del error, es decir: $u(t) = k_p e(t)$



Integral (I), proporciona una salida que es directamente proporcional a la integral del error (error acumulado), $u(t) = k_I \int e(t) dt$

Derivativo (D), esta acción tiene carácter de previsión, lo que hace más rápida la acción de control, aunque tiene la desventaja importante que amplifica las señales de ruido y puede provocar saturación en el actuador. La acción de control derivativa nunca se utiliza por sí sola, debido a que solo es eficaz durante periodos transitorios, $u(t) = k_D \frac{de(t)}{dt}$.

Donde k_P , k_I y k_D son ganancias ajustables pero limitadas por el error en un régimen permanente.

Los diferentes métodos de sintonización de los parámetros de un controlador PID, van de acuerdo a la estructura que se utilice del mismo. La sintonización de los controladores Proporcional–Integral–Derivativo o simplemente controladores PID, consiste en la determinación del ajuste de sus parámetros (k_P , k_I y k_D), para lograr un comportamiento del sistema de control aceptable y robusto de conformidad con algún criterio de desempeño establecido.

Para poder realizar la sintonización de los controladores, primero debe identificarse la dinámica del proceso, y a partir de ésta determinar los parámetros del controlador utilizando el método de sintonización seleccionado.

Para la selección de los tres parámetros del controlador PID (K_p , T_i y T_d) puede procederse de muy diferentes formas dependiendo fundamentalmente del proceso a controlar y de la información disponible a priori. Básicamente se dispone de uno de los siguientes métodos y también de la combinación entre ellos:

1. Métodos iterativos de ajuste y error.
2. Métodos directos:
 - a) Por optimización
 - b) Asignación de polos y ceros

Los métodos iterativos de ajuste y error son, en general, de aplicación en aquellos casos en que el conocimiento a priori del sistema a controlar es pobre. Consisten, básicamente en el ajuste iterativo de los parámetros del controlador a partir de la observación de la respuesta temporal del sistema realimentado, y del conocimiento (o experiencia) del operador en referencia a las tendencias de las variables controladas en función de los parámetros de ajuste.

Por optimización, al resolver problemas de sistemas de control óptimo, se tiene por objetivo hallar un procedimiento para determinar la acción de control sujeta a restricciones que disminuyen en alguna medida el comportamiento ideal. Tal medida habitualmente está dada por un criterio de optimización de un índice de rendimiento. Es índice es una función cuyo valor indica hasta qué punto el funcionamiento efectivo del sistema se acerca al funcionamiento deseado.

Método de Ziegler y Nichols, este método basado en ideas de optimización de un índice de rendimiento para sistemas particulares y luego generalizado empíricamente, es probablemente, el de más amplia divulgación.

Ziegler y Nichols, plantearon dos conjuntos de reglas. El primero de ellos es de aplicación a procesos cuya respuesta temporal al escalón tiene forma de "S". Si el proceso a controlar tiene una respuesta al escalón como la indicada, se calcula la pendiente R en el punto de inflexión y la intersección de una recta con esa pendiente y el eje temporal determina el retardo equivalente L .



Los parámetros R y L permiten definir la constante de tiempo y el retardo de un sistema equivalente de primer orden con retardo. A partir de los parámetros R y L , Ziegler y Nichols recomiendan la utilización de los siguientes valores para diseñar al controlador PID.

Métodos de asignación de polos y ceros, A diferencia de los métodos discutidos anteriormente el método de asignación de polos y ceros requiere para su aplicación un buen conocimiento a priori del sistema a controlar, lo cual limita apreciablemente su rango de aplicación.

Los métodos de asignación de polos y ceros se basan en establecer a priori las zonas del plano s donde se desean ubicar las raíces del sistema de lazo cerrado. Posteriormente se eligen los parámetros del compensador de modo tal que las raíces del sistema realimentado se ubiquen dentro de las zonas establecidas.

Los métodos de asignación de polos y ceros se hallan desarrollados en la literatura en forma general para sistemas lineales y sin imponer restricciones a la estructura del compensador.

Puede demostrarse que si:

- el sistema a controlar es de segundo orden y no presenta retardos puros, y si
- el controlador es PID y no presenta restricciones de orden práctico,

entonces los tres polos de lazo cerrado pueden ser localizados en puntos preestablecidos del plano s .

Los valores de las tres constantes k_p , k_I y k_D , que permiten esta localización se obtienen a partir de la resolución de la ecuación característica de lazo cerrado [5].

4. Control Wavelet

En general, un control PID tiene como entrada el error $e(t)$ con el cual genera una señal de control $u(t)$ como se muestra en la ecuación (3). Las ganancias K_P , K_I y K_D son parámetros que actúan sobre el sistema. En términos de frecuencia los términos proporcional e integral absorben las señales de error de baja frecuencia y el término derivativo las señales de error de alta frecuencia [4]. De manera particular en el control PID-wavelet la señal de error, la integral del error y la derivada del error son escaladas por una función wavelet de forma independiente quedando un control como:

$$u(t) = \Psi_p(e)e(t) + \Psi_I(e) \int e(t)dt + \Psi_D(e) \frac{de(t)}{dt} \quad (6)$$

Donde las funciones wavelet $\Psi_p(e)$, $\Psi_I(e)$ y $\Psi_D(e)$ son funciones que actúan sobre el error, la integral del error y la derivada del error respectivamente, donde la estructura de cada una de las funciones queda como:

$$\Psi_{PID}(e) = \Psi_i(E)S_i, \dots i \in [1, n] \quad (7)$$

Con

$$\Psi_i = [\psi_1, \psi_2, \dots, \psi_n] \quad (8)$$

Y



$$S_i = [s_1, s_2, \dots, s_n] \quad (9)$$

Donde n es el número de wavelets con las que se aproximará la señal de error $e(t)$.

Un diagrama esquemático del control PID-wavelet se muestra en la Figura 3, de manera particular para este artículo se implementa una wavelet Morlet mostrada en la Tabla 1 con valores de escala $s \in [0,2]$ y valores de traslación $b \in [0,400]$.

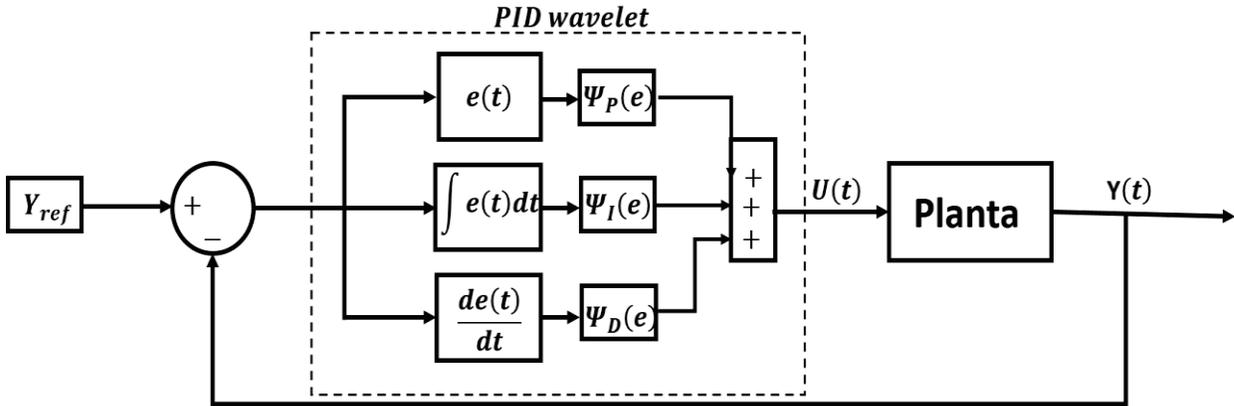


Figura 3. Diagrama esquemático de control PID-wavelet.

5. Implementación de Control PID-Wavelet

La implementación práctica de los controles PID es una tarea no trivial por la sintonización de las ganancias proporcional, integral y derivativa, en cambio la implementación del control PID-wavelet que se propone en este artículo facilita la tarea ya que solo se necesita el conocimiento del tiempo de estabilización del sistema.

Para la implementación de esta técnica de control se trabajó con un sistema de primer orden de la forma:

$$G(s) = \frac{k}{Ts + 1} \quad (10)$$

De manera específica con un sistema de control de presión de aire en un tanque, donde la constante $k = 0.7236$ y la constante $T = 15.7$, que se refieren a la constante de establecimiento y a la constante de tiempo de establecimiento respectivamente (63.2 % de su máximo valor de establecimiento).

Consideremos una entrada sinusoidal $x_f(t) = x \psi(t)$ al sistema de primer orden mostrado en la ecuación (10), la salida en estado estacionario $Y_{ss}(t)$.



$$Y_{ss}(t) = \frac{xk}{\sqrt{1+(T\omega)^2}} \psi(\omega t + \phi) \quad (11)$$

Donde $\psi(t)$ es una función wavelet Morlet mostrada en la Tabla 1 y el elemento ϕ conocido como ángulo de fase y está definido como:

$$\phi = -\tan^{-1}(T\omega) \quad (12)$$

De aquí se tiene que, si $\omega \rightarrow 0 \Rightarrow \phi \rightarrow 0^\circ$, lo cual indica que la amplitud máxima de la salida es k -veces, por otro lado se tiene que, si $\omega \rightarrow \infty \Rightarrow \phi \rightarrow 90^\circ$ entonces se tiene que la amplitud máxima es de $\frac{1}{\omega}$. Con las dos implicaciones anteriores se puede garantizar que el sistema de primer orden con una función de transferencia mostrada en la ecuación (10) es estable en estado estacionario o se mantiene acotada por la constante k [1].

Si la entrada tiene un comportamiento sinusoidal entonces implica que el sistema es estable en estado estacionario, esto conduce que una función wavelet $\psi(t)$ como entrada de control mantiene en estado estacionario a un sistema de primer orden, garantizando que el sistema en lazo cerrado se va a mantener en estado estable ya que una función wavelet por pertenecer a la familia de funciones de energía finita cumple con el principio de admisibilidad, lo cual garantiza que son funciones acotadas y no tienen escape al infinito de sus valores.

6. Resultados

Considerando que el sistema es estable ante una entrada de control del tipo sinusoidal, se puede implementar una ley de control como se muestra en la ecuación (6) bajo el esquema que se ilustra en la Figura 3.

A continuación, se muestran los resultados que se obtuvieron de la simulación del sistema con la implementación del control PID-wavelet que se propone en este artículo, también se muestran los resultados de un control PID. Las ganancias del control PID se calcularon mediante el método de lugar geométrico de las raíces con el cual se obtuvo las siguientes ganancias $k_p = 0.87$, $k_I = 0.262$ y $k_D = 0$.

En la figura 4 se muestra el comportamiento de la señal de control PID contra PID-Wavelet, en el cual podemos observar que con el control PID-Wavelet optimizamos energía que para casos de implementación práctica nos ayuda a mejorar y alargar la vida útil de los actuadores del sistema de control.

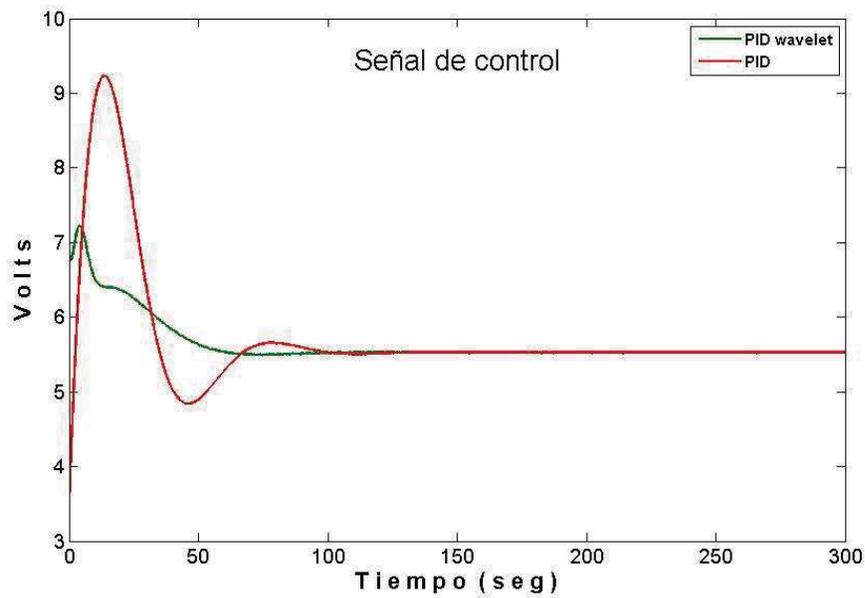


Figura 4. Señal de control para el sistema de presión de aire.

En la Figura 5 podemos observar que el comportamiento de la señal de salida del sistema de presión de aire, en el que se aprecia que la respuesta es suave con el control PID-Wavelet comparado con la señal de salida que produce el control PID la cual tiene un sobre impulso.

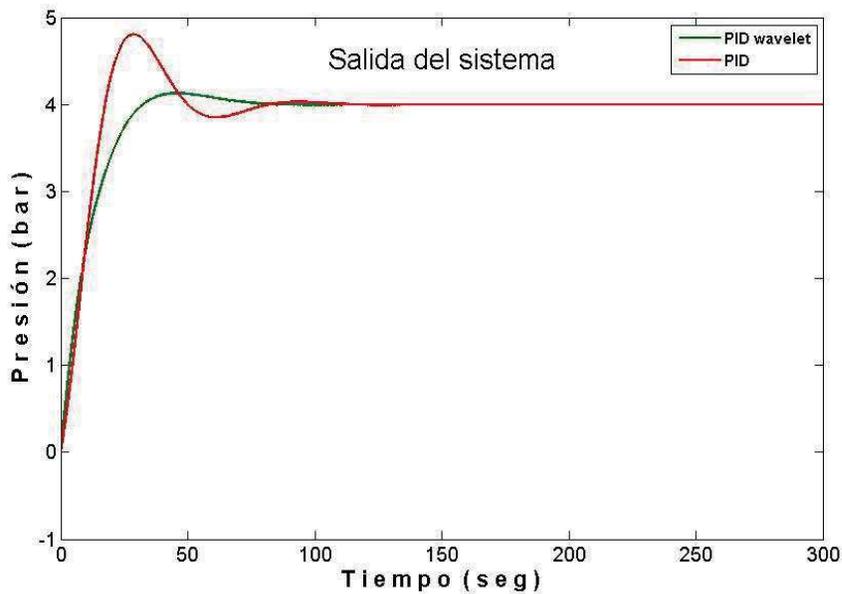


Figura 5. Señal de salida del sistema de presión de aire.

Finalmente, en la Figura 6 observamos el comportamiento de la señal del error del sistema de control de presión.

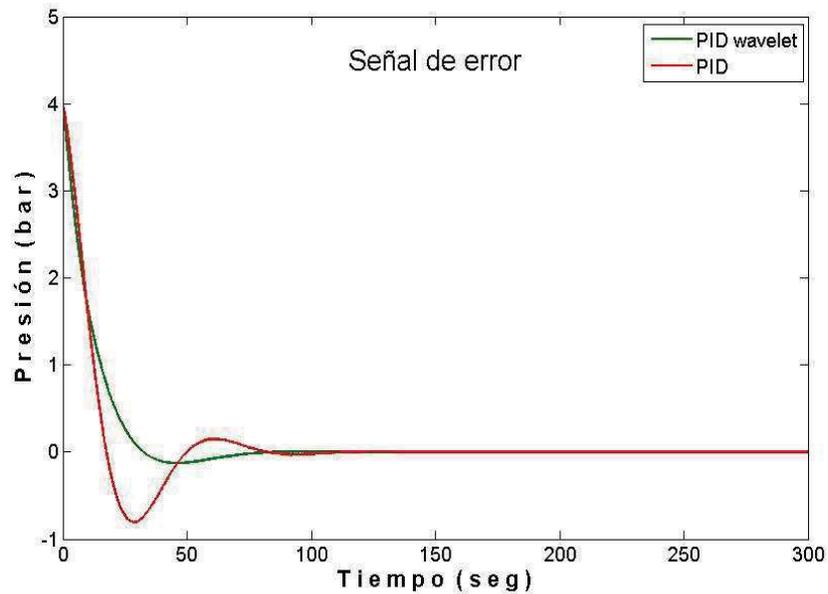


Figura 6. Señal de error del sistema de presión de aire.

Los resultados mostrados son producidos cuando al sistema se le pide una señal de referencia igual a 4 bar de presión, en las cuales se puede observar que existe un ahorro significativo en el consumo energético como entrada de control del PID-Wavelet comparado con un control PID, y la respuesta de salida del sistema tiene un comportamiento sin sobre-impulso el cual garantiza una mayor vida útil de los actuadores.

A partir de los resultados obtenidos en el presente artículo los trabajos que se proponen es la implementación física en sistemas lineales y comprobar la estabilidad para sistemas de orden superior y no lineales para comprobar la admisibilidad general del control PID-Wavelet.

7. Conclusiones

En general el control PID-wavelet propuesto en este artículo presenta grandes ventajas, ya que comparado con un control PID se facilita la sintonización de las ganancias respectivas del control, ya que solo se necesita del conocimiento de la entrada y salida del sistema y del tiempo aproximado de establecimiento del sistema para proponer los parámetros de la función wavelet, por otro lado se observa el comportamiento de la señal de control del PID-wavelet es menor que la señal de control del PID y con ello se concluye que existe una optimización de energía para hacer que el sistema converja a la referencia deseada, el análisis de la frecuencia de los sistemas de control nos favorece en el momento de la sintonización ya que se puede trabajar con bajas, medianas y altas frecuencias de la señal del error y así favorecer al ajuste de las funciones wavelet que multiplicaran los parámetros proporcional, integral y derivativo del control PID.



Referencias

- [1] K. Ogata, Ingeniería de Control Moderna, 5° edición, Madrid: Pearson Educación, 2010, pp.227-399.
- [2] Jaideva C. Goswami and Andrew K. Chan, Fundamentals of Wavelets Theory, Algorithms, and Applications, 2° edition, Texas, Wiley, 2011, pp. 94-237.
- [3] J. A. Cruz, A. Jarillo, L. E. Ramos and O. A. Domínguez, Wavelet PID and Wavenet PID: Theory and Applications, Chapter from the book PID Controller Design Approaches - Theory, Tuning and Application to Frontier Areas, Chapter 2, 2012.
- [4] S. Parvez and Z. Gao, A Wavelet- Based Multiresolution PID Controler, IEEE Transactions on Industry Applications, 41, 2, 537-543.
- [5] Tacconi Eugenio, Mantz Ricardo, Solsona Jorge, Controladores Basados en Estrategias PID, LEICI, Versión electrónica editada por : Tania Salazar y Ana Roquez, 4-17, 2015



Sistema de Control de Acceso Inteligente a través de Cerraduras Electrónicas

Figuroa Soto Israel, Gutiérrez Enríquez David Ramón, Domínguez García Rodolfo Omar y Aguilar Molina Yehoshua

Centro Universitario de los Valles de la Universidad de Guadalajara
odomi@academicos.udg.mx

Resumen

El acceso a los centros de datos o data center (DC) debe tener un indudable grado de seguridad ya que estos son los centros medulares de la información que procesa, utiliza y comparte una institución. El DC principal del CUValles presenta el mismo caso por lo que solo se debe permitir el acceso a éste, a personal autorizado para mayor seguridad.

En este trabajo se aborda el diseño e implementación de un dispositivo que permita el acceso de personas al DC de forma automática y controlada a través de una chapa electrónica accionada por RFID y controlada por una Raspberry Pi 3B+, la cual cuenta con una base de datos para poder registrar a los usuarios autorizados para el acceso al DC, registrar y controlar su acceso.

Palabras clave: Raspberry Pi, Chapa Electrónica, RFID, Data Center.

1. Introducción

En el mundo actual, la tecnología asume cada vez un papel más importante en la sociedad ya que la encontramos aplicada en todas partes pero no siempre es segura, y en ocasiones no se aplica con la velocidad con la que avanza la tecnología para favorecer a la sociedad. Entre las tecnologías que se aplican para mejorar la seguridad existe las que tienen que ver con el acceso seguro a lugares restringidos. Los ambientes que requieren acceso seguro necesitan una serie de elementos para poder garantizar seguridad, como son puertas, chapas de seguridad, cámaras, vigilantes, entre otros. En el día a día las personas se topan con más de una puerta para acceder y salir de algún lugar, si el lugar es importante se necesitará registrar el acceso para poder entrar, ya sea a través del uso de una llave, libreta de registro, junto con un guardia que vigile que no se pase nadie sin autorización. Todo esto requiere de gran tiempo, tener manos libres, el registro, el girar la llave hasta poder abrir la cerradura que en ocasiones se traba. La seguridad y accesibilidad es muy importante, por lo que un sistema que registre el acceso, restrinja a individuos no autorizados y a la vez se desenvuelva con fluidez es de gran utilidad, la solución que se presenta es un sistema de control de acceso inteligente a través de cerraduras electrónicas, el cual funciona dando la entrada a personas registradas que se les entregará una tarjeta RFID, funcionará como identificación para acceder a las puertas permitidas por la base de datos que estará alojada en una Raspberry Pi, la cual por medio de un programa identificará su acceso, lo registrará y si tiene permitido abrir esa puerta le dará el pase, este sistema tiene la ventaja de que no se necesita más de un registro para tener los permisos de acceder a algún lugar con cerradura electrónica, además de que al abrir la puerta solo basta con acercar la identificación al sensor de lectura por lo que es muy rápido, también se ahorra en que haya un vigilante en cada lugar para ver si una persona tiene el permiso o no, el sistema es ágil y seguro, ya que si aquella persona ya no trabaja, vive, o tiene permiso de acceder solo basta con bloquear el acceso de aquel identificador RFID en cualquier momento. Si hay alguna emergencia en la cual se necesite salir, se pueden desbloquear los accesos para evitar un daño grave a la seguridad de las

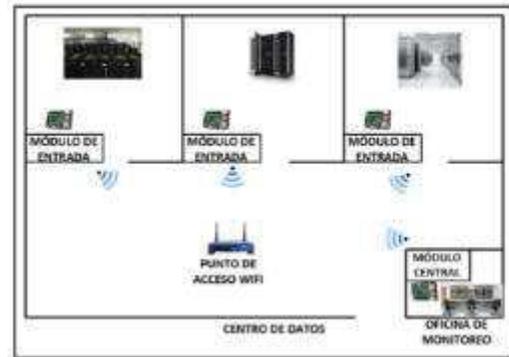
personas, esto a través de una llave normal, así es como se solucionará el problema de tener un ambiente seguro al acceso por medio de una cerradura electrónica con fluidez y un mayor control el mismo en un área local.

Con base a lo anterior, el DC principal del Centro Universitario de los Valles (CUValles) necesita tener el acceso restringido, dicho DC se encuentra ubicado en el Edificio de Servicios Académicos, en donde se implementó dicha propuesta. Por lo que se implementó un sistema automatizado vía una tarjeta Raspberry Pi 3B+ y una chapa electrónica, para acceder al centro de cómputo del CUValles basado en RFID y, para control y seguimiento de los accesos del personal a dicha área.

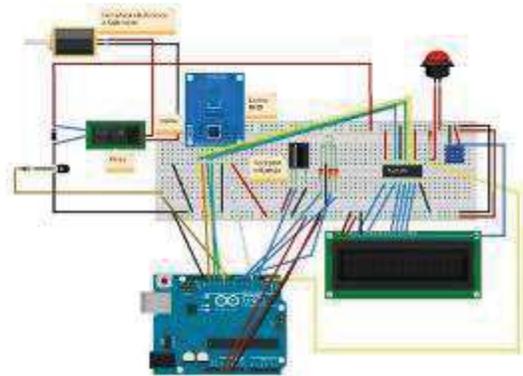
Estado del arte

Existen diversos proyectos que han sido desarrollados con la finalidad de mejorar la seguridad de un lugar importante, tener control del acceso y de más, de entre ellos podemos mencionar algunos de los más destacados, más recientes o más parecidos a continuación:

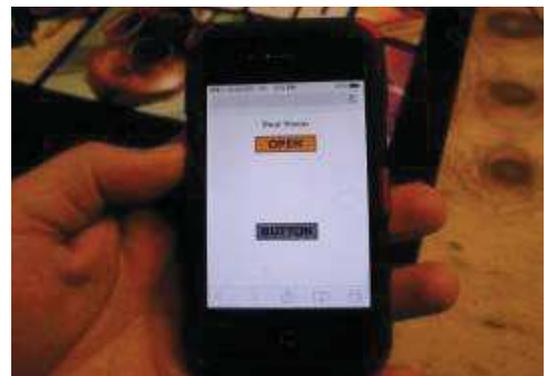
Sistema de acceso usando una tarjeta RFID y verificación de rostro: Desarrollado y aprobado por la Universidad Autónoma Metropolitana-Azcapotzalco en el 2018, este prototipo da el control de acceso a un centro de datos usando una tarjeta RFID y verificación de rostro del usuario con una webcam, la información capturada la envía a una base de datos MySQL que conecta con una Raspberry pi 3 B+ la cual procesa los datos e imprime en una pantalla de LCD al usuario el tipo de acceso, ya sea denegado o permitido este se ve reflejado en la puerta.



Implementación de un sistema de control de entrada y salida empleando el módulo de lectura RFID con la tecnología Arduino: Desarrollado en el 2016 el presente proyecto de investigación consiste en la apertura de una cerradura electrónica por medio de un lector de Radiofrecuencia (RFID) empleando la tecnología de la placa Arduino. El funcionamiento del sistema consiste en pasar la tarjeta o etiqueta RFID al frente del lector, si la tarjeta coincide con el código que se le otorgó al titular de la oficina, se realizará la apertura de la cerradura mostrando en pantalla el nombre del titular de la oficina, de lo contrario se mostrará un mensaje indicando "Acceso no Autorizado".



Control puerta de garaje: Desarrollado en el 2013 este proyecto consiste en una app de celular, a la cual varios usuarios de la casa quienes ya están registrados en la base de datos My SQL, cada que un usuario utiliza la app, Raspberry entra en acción, procesa la información de id de usuario y da acceso a las personas permitidas además envía datos los cuales son registrados cada que abren y cierran la puerta del garaje.





Cerradura electrónica RFID controlada con Raspberry Pi

Pi: Desarrollado en el 2015 Este proyecto consiste en proteger un taller de visitas no deseadas en ausencia del personal. El sistema utiliza un lector RFID y una cerradura eléctrica para dar o no acceso a una puerta a través de una Raspberry Pi que se encarga de registrar todas las aperturas y eventos del sistema para así evitar tener que utilizar un sistema externo.



2. Desarrollo

El proyecto en desarrollo tiene como finalidad brindar seguridad y acceso a usuarios específicos. La configuración que llevará la tarjeta Raspberry Pi, dará como resultado un listado de personal autorizado y toda la información transcurrida será almacenada en una base de datos para su posterior revisión en caso de incidentes.

Los protocolos y los softwares utilizados son opensource, lo que se traduce en menores costos (ya que no es necesario invertir en licencias), la asistencia por parte de la comunidad desarrolladora es muy amplia y sus alcances serán:

- Permitir el acceso al DC vía RFID, a personal autorizado.
- Accionar la chapa electrónica al paso de una tarjeta RFID autorizada.
- Guardar la información de los accesos al DC, a través de la base de datos local de la Raspberry Pi (Sqlite).
- Enviar la información local a la la Nube del CUValles, para el almacenamiento permanente de la información.
- Además estará relacionada con el sistema que monitorea la temperatura ambiental y humedad relativa del DC, ya que es la misma Raspberry Pi quien contiene ambos sistemas.

En la figura 1.1, se muestra un diagrama de flujo del funcionamiento del sistema de acceso, la manera en que decide la apertura o cierre de la chapa:

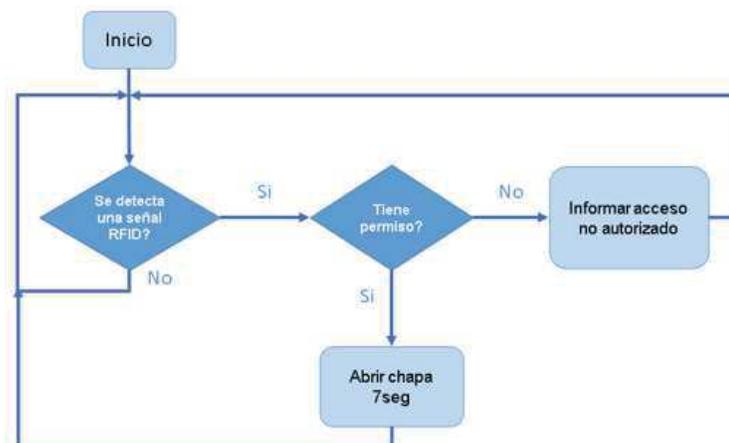


Figura 1.1. Funcionamiento del sistema.

2.1 Requerimientos

Contra eléctrica Mod. 310. La contrachapa utilizada es marca Philips Contra eléctrica Mod. 310 acabados en cromo la cual cuenta con el siguiente diseño:



Figura 1.2. Contrachapa eléctrica.

Características técnicas:

1. Para puertas izquierdas o derechas
2. Medidas: 250 x 30 x 28 mm.
3. Voltaje de activación 12V-16V

Fuente de poder:

La fuente de poder utilizada para la activación de la contrachapa es la siguiente:



Figura 1.3. Fuente de poder.



Características:

1. Voltaje de Entrada: 100-120V AC / 60Hz , 200-220V AC / 50Hz
2. Voltaje de Salida: 12V DC
3. Corriente Máxima de Salida: 2A
4. Potencia Máxima de Salida: 24W
5. Tamaño: 8.5 x 6 x 3.5 cm

RFID RC522

El sensor utilizado para leer la tarjeta de proximidad, fue el RFID-RC522 el cual su diseño es el siguiente.



Figura 1.4. Módulo RFID.

Especificaciones:

1. Voltaje de Operación: 3.3V DC
2. Corriente de Operación: 13-26mA/3.3V DC
3. Corriente de Standby: 10-13mA/3.3V DC
4. Corriente de Sleep: <80uA
5. Corriente pico: <30mA
6. Frecuencia de operación: 13.56 MHz
7. Transferencia de datos: Max. 10Mbit/s
8. Tipos de tarjetas compatibles: Mifare1 S50, S70 Mifare1, MIFARE Ultralight, Mifare Pro, Mifare DESFire.
9. Dimensiones RFID-RC522: 40 mm x 60 mm
10. Dimensiones Tarjeta: 85 mm x 54 mm
11. Temperatura de funcionamiento: -20 a 80 grados centígrados
12. Temperatura de almacenamiento: -40 a 85 grados centígrados
13. Humedad relativa: 5% hasta 95 %
14. La tasa de transmisión por defecto: 9600bps, velocidad de transferencia máxima: 1,228,800bps

Relevador SRD-05VDC-SL-C con modulo HIGH

El actuador que se utilizó para activar la contrachapa eléctrica fue el relevador SRD-05VDC-SL-C el cual cuenta con el siguiente diseño:

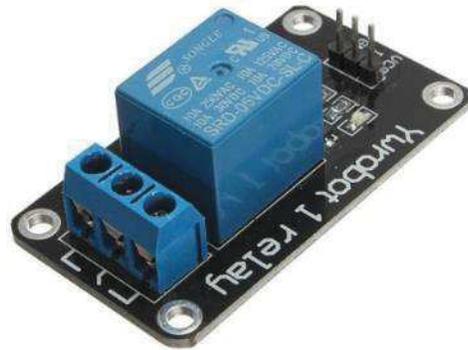


Figura 1.5. Relevador.

Especificaciones:

1. Medida de PCB: 2.0 x 4.3 cm
2. Voltaje de operación 5V
3. PNP transistor

Raspberry Pi 3b +

Es una computadora pequeña y fácil de usar para aprender programación y desarrollar entre otras cosas sistemas embebidos.

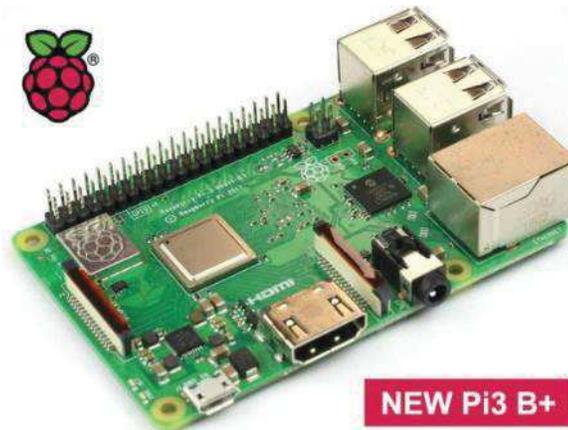


Figura 2.

Es un ordenador del tamaño aproximado de una tarjeta de crédito. Consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Fue lanzado en 2006 por la Fundación Raspberry Pi con el objeto de estimular la enseñanza de informática en las escuelas de todo el mundo.

Raspberry Pi 3 B+

Característica de la Raspberry Pi 3b+:

La Raspberry Pi 3 Modelo B+ es el último producto en el rango de las Raspberry Pi 3.

1. Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit quad-core SoC @ 1.4GHz
2. 1GB LPDDR2 SDRAM
3. 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
4. Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)



5. Extended 40-pin GPIO header
6. Full-size HDMI
7. 4 USB 2.0 ports
8. CSI camera port for connecting a Raspberry Pi camera
9. DSI display port for connecting a Raspberry Pi touchscreen display
10. 4-pole stereo output and composite video port
11. Micro SD port for loading your operating system and storing data
12. 5V/2.5A DC power input
13. Power-over-Ethernet (PoE) support (requires separate PoE HAT).

Implementación

Instalación de la plataforma Raspberry Pi 3B +.
Consulta la página: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

Puesta en funcionamiento

Una vez que tenemos una tarjeta Raspberry Pi 3b+, ver Figura 3, fuente de poder y una micro SD card, un teclado y un ratón, procedemos a lo siguiente:

Tu Raspberry Pi necesita una tarjeta SD para almacenar todos tus archivos y el sistema operativo Raspbian. De al menos 8 GB de capacidad.

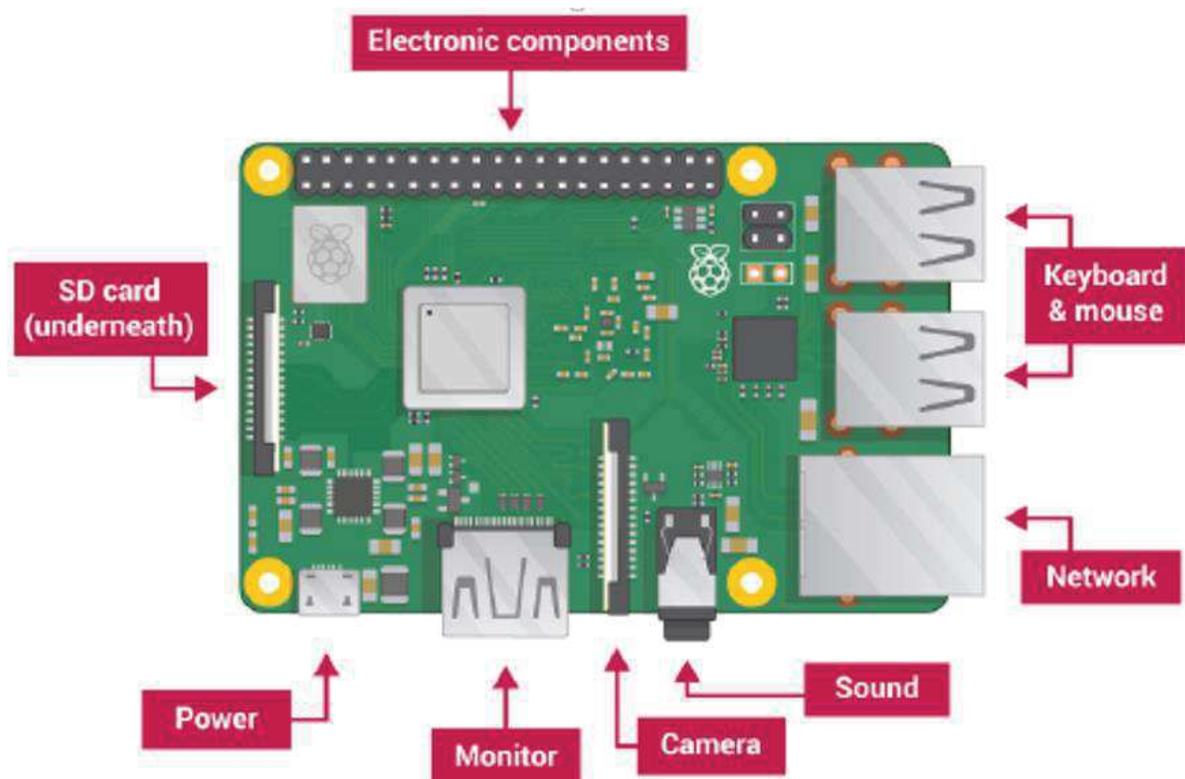


Figura 3.- Raspberry Pi 3B+

Instalando el SO Linux Raspbian de Debian, para la Raspberry Pi 3b+.

Visita la página de descargas de Raspberry Pi. <https://www.raspberrypi.org/downloads>.

Si tienes una tarjeta SD que no tiene el sistema operativo Raspbian todavía, o si deseas restablecer tu Raspberry Pi, puedes instalar fácilmente Raspbian por ti mismo. Para ello, necesitas una computadora que tenga un puerto para tarjeta SD: muchas de las computadoras portátiles y de escritorio tienen una. Ya que desde una computadora será descargado el software e instalado en la tarjeta SD.

Desde esta liga <https://www.raspberrypi.org/downloads>, podremos acceder al software requerido. El uso del software NOOBS es la forma más fácil de instalar Raspbian en su tarjeta SD, ver Figuras 4 y 5.

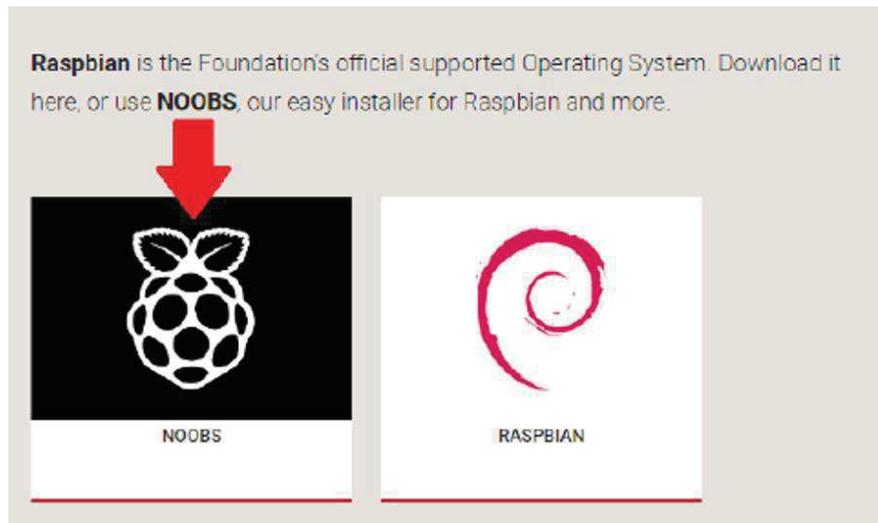


Figura 4.- Descarga del S.O.

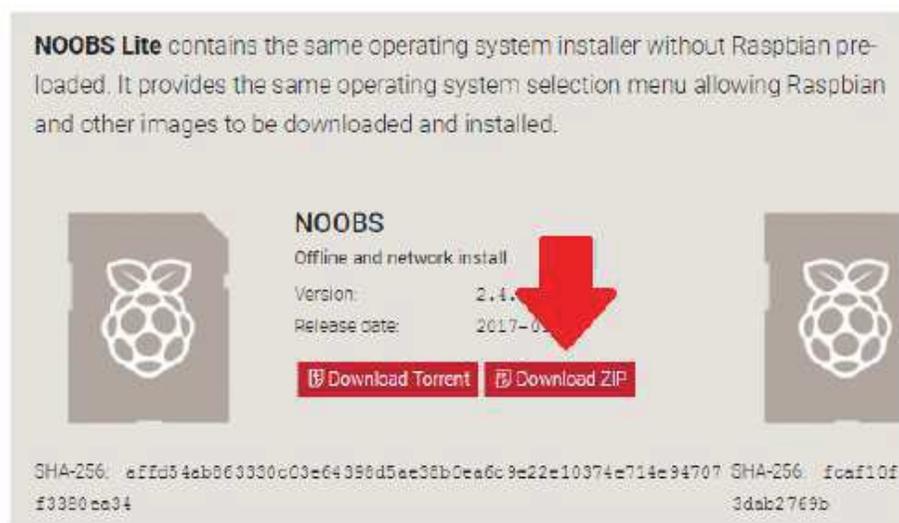


Figura 5.- Descarga del S.O.



Formatear la tarjeta SD

Todo lo que esté almacenado en la tarjeta SD se sobrescribirá durante el formateo. Así que si la tarjeta SD en el que se desea instalar Raspbian actualmente tiene algún archivo, por ejemplo, de una versión anterior de Raspbian, es posible que desee respaldar estos archivos primero para no perderlos permanentemente.

Visita el sitio web de la SD Association y descargue el SD Formatter (https://www.sdcard.org/downloads/formatter_4/index.html) para Windows o Mac. Sigue las instrucciones para instalar el software. Inserta tu tarjeta SD en la ranura para tarjeta SD de la computadora o computadora portátil. En SD Formatter, seleccione su tarjeta SD y formatee la tarjeta, ver Figuras 6 y 7.

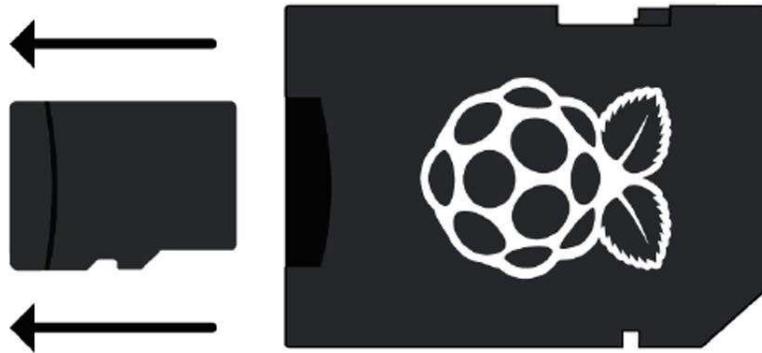


Figura 6.- Porta SD.

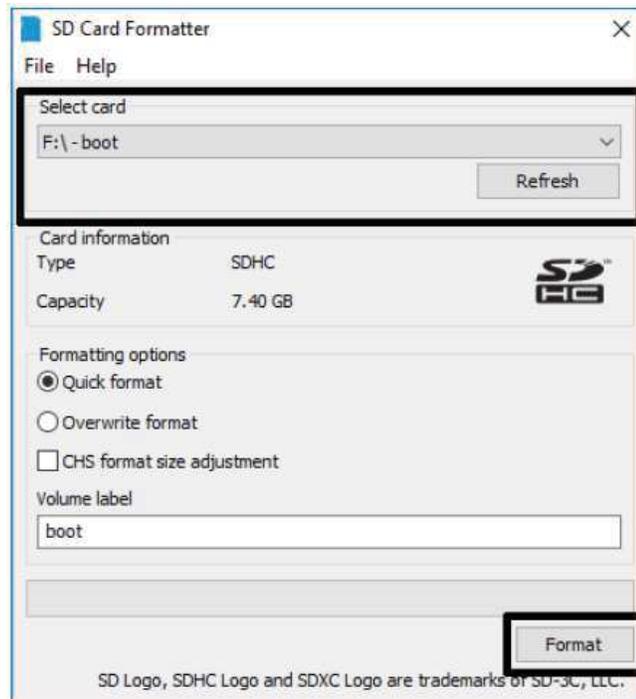


Figura 7.- Formateo de SD.

Extraer NOOBS del archivo zip

A continuación, deberás extraer los archivos del archivo zip de NOOBS que descargó de la Página web de la Raspberry Pi.
Encuentra el archivo descargado: de forma predeterminada, debería estar en su carpeta de Descargas.
Haz doble clic en él para extraer los archivos y mantener abierta la ventana del Explorador / Buscador resultante.

Ahora abre otra ventana de Explorer y navegue a la tarjeta SD.
Selecciona todos los archivos en la carpeta NOOBS y arrástrelos a la ventana de la tarjeta SD para copiarlos en la tarjeta, ver Figura 8.

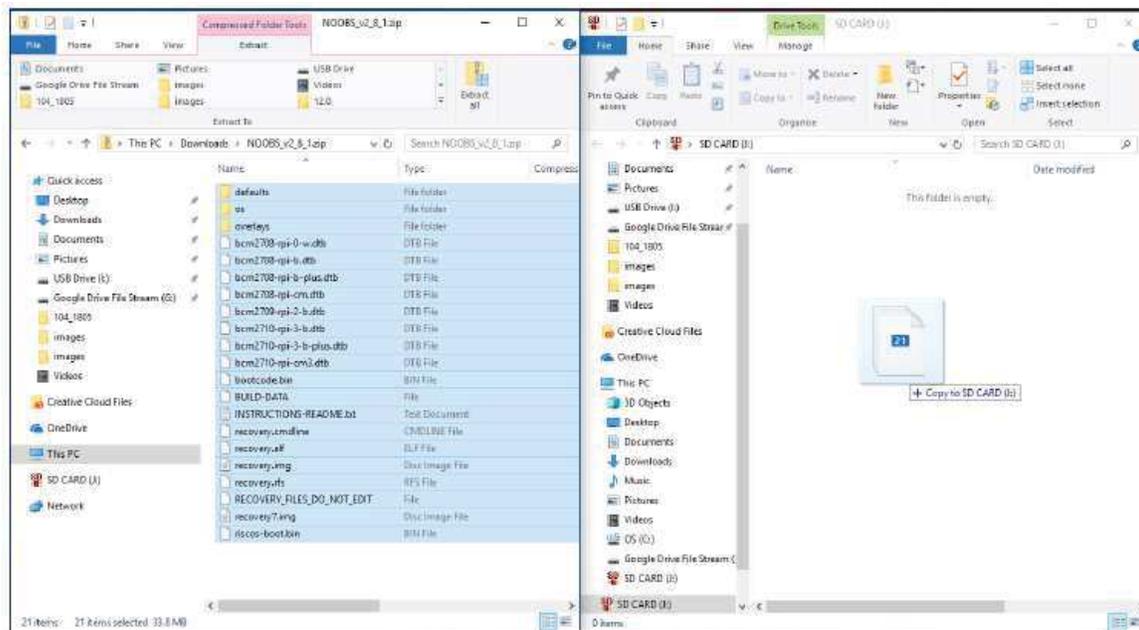


Figura 8.- Copia de archivos a la SD.

Una vez que hayas copiado todos los archivos a la tarjeta SD, puedes expulsar la tarjeta.
Inserta la tarjeta SD que se ha configurado con Raspbian en la ranura para tarjetas micro SD en la parte inferior de tu Raspbian Pi 3b+, ver Figura 9.

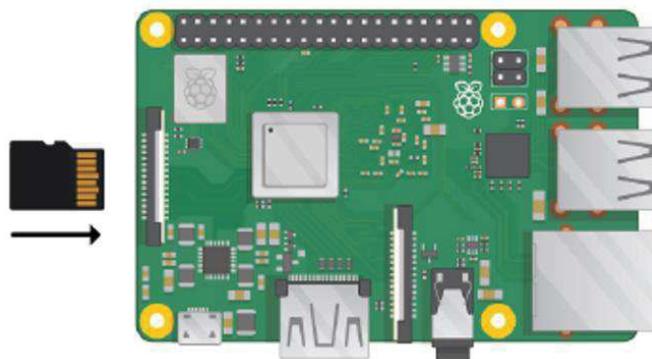


Figura 9.- Inserción de SD en la Raspberry Pi.

Una vez instalada la tarjeta SD en la Raspberry 3B+, procedemos a conectar los demás aditamentos como son: teclado, ratón, monitor vía HDMI como se ve en la Figura 10.

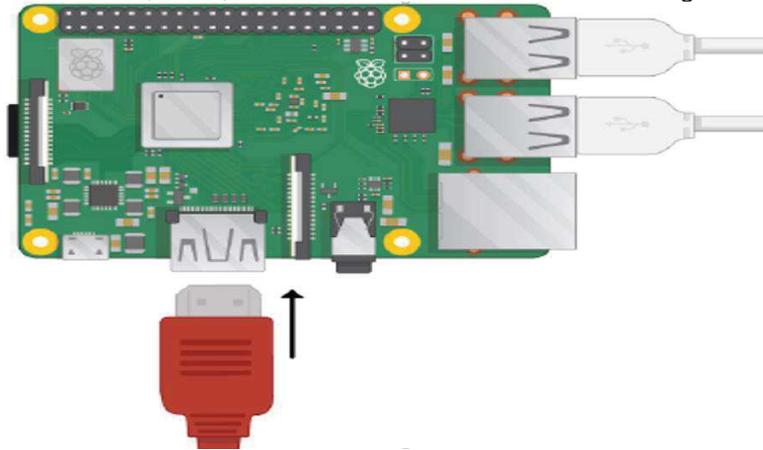


Figura 10.- Conexión de dispositivos externos.

Si deseas conectar la Raspberry Pi a Internet a través de Ethernet, usa un cable Ethernet para conectar el Puerto Ethernet de la Raspberry Pi a un conector Ethernet en la pared o en un enrutador de Internet, ver Figura 11. Note que no es necesario hacer eso si se va a usar WiFi o si no quieres conectarte a internet.

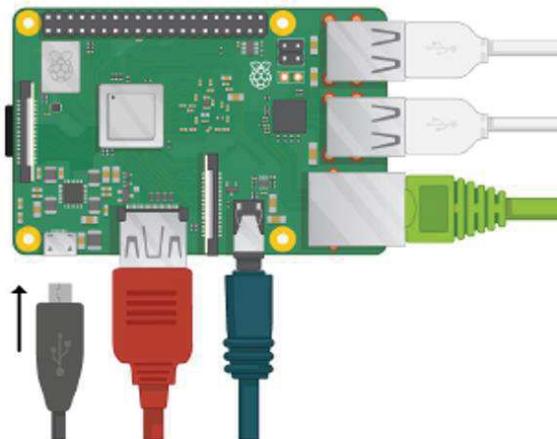


Figura 11.- Conexión de Ethernet.

Tu Raspberry Pi no tiene un interruptor de encendido: tan pronto como lo conectes a una toma de corriente, encenderá. Ten en cuenta que el puerto de alimentación micro USB del Pi tiene un lado más largo en la parte superior. Enchufa una fuente de alimentación micro USB a una toma y conéctala al puerto de alimentación de tu Pi.

Deberías ver un LED rojo encendido en la Raspberry Pi, que indica que la Pi está conectada a la corriente. A medida que se inicia (a esto también se le llama arranque), verás que las frambuesas aparecen en la parte superior izquierda de tu pantalla, ver figura 12.

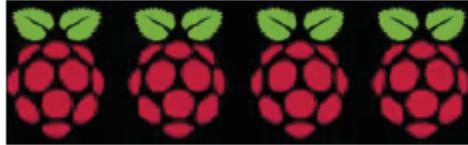


Figura 12.- Frambuesa en pantalla.

Después de unos segundos aparecerá el escritorio de Raspbian en tu pantalla, ver Figura 13.



Figura 13.- Escritorio de Raspbian

Cuando inicies tu Raspberry Pi por primera vez, la aplicación Bienvenido a Raspberry Pi aparece y te guiará a través de la configuración inicial, como se ve en la Figura 14.



Figura 14.- Welcome to Raspberry Pi

Haz clic en Siguiete para iniciar la configuración. Configura país, idioma y zona horaria, ver Figura 15, luego haga clic en Siguiete nuevamente.



Figura 15.- Configuración, idioma, teclado y zona horaria.

Ingresa una nueva contraseña para tu Raspberry Pi, como se ve en la Figura 16 y haz clic en Siguiente.



Figura 16.- Cambio de contraseña

Conéctate a una red WiFi seleccionando su nombre, ver Figura 17, ingresando la contraseña y haciendo clic en Siguiente.



Figura 17.- Conectando una red WiFi

Haz clic en Siguiente para que el asistente compruebe si hay actualizaciones para Raspbian, ver Figura 18, y se instalen (esto puede llevar un poco) mientras.

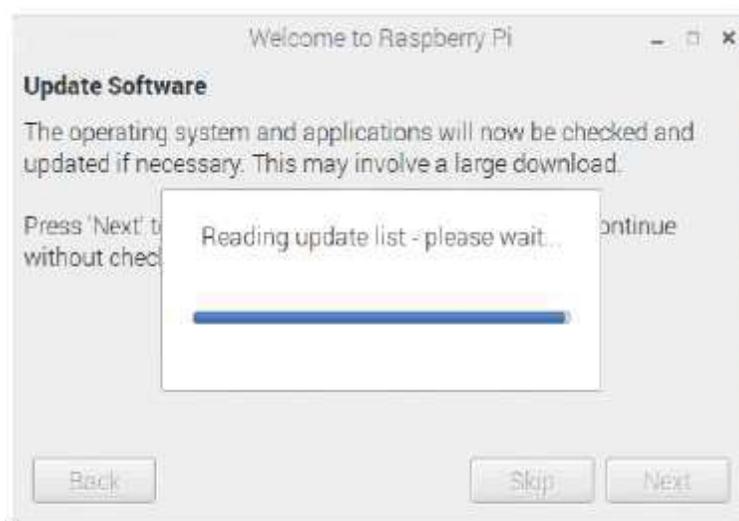


Figura 18.- Actualizaciones

Haga clic en Listo o Reiniciar para finalizar la configuración, ver Figura 19. Solo tendrás que reiniciar si es necesario para completar una actualización.



Figura 19.- Finalización



Implementación

Código Python figura 20, este recibe la información de los pases, en esta sección se marcó 2 etapas para dos distintos tipos de pases:

Pase tipo A:

1. Acceso común al Site

Pases tipo B:

1. Acceso común a el Site.
2. Acceso a registrar/borrar nuevos usuarios

Etapas 1: Registro de los usuarios ya sean tipo A o tipo B.

Etapas 2: Permiso de Accesos.

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522

try:

    id, text = reader.read()
    GPIO.Cleanup()
    if text.startswith('M_'):
        print('Llave maestra')
        nombre = input('Escriba nombre del usuario')
        if nombre.startswith('M_'):
            print('No puede registrar una llave maestra')
        else:
            print('Acerque la tarjeta al sensor')

            id, text = reader.read()
            if text.startswith('M_'):
                print('No se puede sobrescribir una llave maestra')
            else:
                reader.write(nombre)
                print('Registrado')
                print(id, ': ', nombre)
    else:
        print('Usuario: ', text)
        if text.startswith('A_'):
            print('Acceso autorizado')
        else:
            print('Prohibido el paso')
finally:
    print('fin')
```

Figura 20. Código Python para leer ingresos

También se ha implementado el código que habilita a la Raspberry a correr todo el tiempo el programa de acceso y guarde información en la base de datos interna mientras esté corriendo, de esta manera si se va la corriente eléctrica o se apaga por alguna razón la Raspberry, al momento de encender de vuelta, el código estará arrancando junto con los demás programas de inicio de esta, así nos dé una gran ventaja puesto que no debe estar alguien al pendiente de si está en funcionamiento en todo momento el programa y así se vuelve más independiente, como se observa en la figura 21.



```
1 Dir=/home/sensores/dhtxx-rpi-python3AEMON=${DIR}/DHT11_console.py
2 DAEMON_NAME=DHT11_console
3 DAEMON_USER=root
4 PIDFILE=/var/run/${DAEMON_NAME}.pid
5 . /lib/lsb/init-functions
6
7 do_start() {
8     log_daemon_msg "Starting sistema ${DAEMON_NAME} demonio"
9     start-stop-daemon --start --background --pidfile $PIDFILE --make-pidfile --user $DAEMON_USER --chuid $DAEMON_US$
10    log_end_msg $?
11 }
12 do_stop () {
13    log_daemon_msg "Stopping system ${DAEMON_NAME} daemon"
14    start-stop-daemon --stop --pidfile $PIDFILE --retry 10
15    log_end_msg $?
16 }
17
18 case "$@" in
19     start|stop)
20     do_${1}
21 ;;
22
23 restart|reload|force-reload)
24     do_stop
25     do_start
```

Figura 21. Script para iniciar un programa

Una vez guardada la información de acceso en la base de datos interna de la Raspberry, se envían a la nube del CUValles, a través del siguiente link. La dirección de acceso a la Raspberry Pi es la "148.202.89.57" en tiempos predefinidos de esta manera se guarda la información recolectada permanentemente y se evita que algún día la base de datos de la Raspberry se pueda saturar de información.

Como pruebas finales se comprobó el funcionamiento del circuito con la Raspberry figura 22.

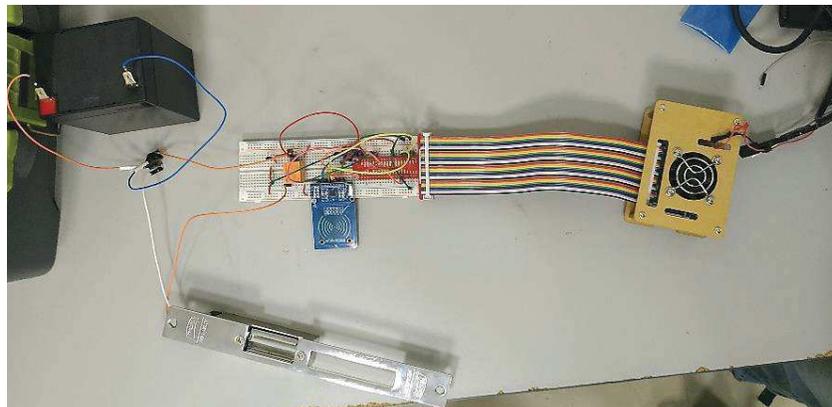


Figura 22. Prueba del circuito electrónico.



3. Conclusiones

La programación en lenguaje Python en este proyecto fue de suma importancia para el desarrollo del proyecto ya que gracias a su sintaxis se logra un código legible por su estructura, además de la utilidad de la librería SimpleMFRC522 la cual ayudo al control del RFID mediante una simplificación de parámetros, el actuador usado (relé) para activar la contrachapa facilitó el funcionamiento al poder separar las corrientes entre los diferentes dispositivos.

En el estado actual del proyecto, se instaló la contrachapa en la puerta del SITE deseado, así como se hizo un modelo en protoboard para verificar el correcto funcionamiento de sus partes, se comprobó que la Raspberry mediante el código y el RFID se comunicaran adecuadamente, se utilizaron 3 tarjetas diferentes, una llamada llave maestra, la cual puede conceder el acceso a mas tarjetas RFID así como editarlas o denegar su pase, verificando el correcto accionamiento de la contrachapa.

Este proyecto se actualizará para darle un funcionamiento más práctico, añadiendo una base de datos así para al requerirse que un individuo entre, ya sea para dar mantenimiento al equipo del SITE, y no hay personal con los permisos adecuados para acceder, el tener su propia tarjeta RFID podría ser inseguro ya que ellos la podrían extraviar o ingresar más tiempo de lo indicado asumiendo distintos riesgos, por lo que se desarrollara una aplicación de celular para dar acceso a los individuos desde larga distancia, tomando el tiempo de acceso al lugar desde minutos a días, el futuro del sistema de acceso inteligente tiene un gran potencial añadiendo todas estas distintas funcionalidades, en un futuro también se aplicara otro método de seguridad, el caso de que alguna vez se llegara a perder la tarjeta RFID o si la llegan a robar, se implementara una cámara de confirmación facial de usuario, está identificara el usuario siguiendo un protocolo seguro, para prevenir si es otra persona que no cumple con los requisitos, y no permitir el acceso, así como permitirlo al cumplirse los requisitos.



Referencias

- [1] Vega J. Sánchez F. Salgado G. Lagos M. "SISTEMA DE ACCESO USANDO UNA TARJETA RFID Y VERIFICACIÓN DE ROSTRO", Ingenius. Revista de Ciencia y Tecnología, Núm. 20, pág.112, 2018.
- [2] Coba A. Samaniego E. "Implementación de un sistema de control de entrada y salida empleando el módulo de lectura RFID con la tecnología Arduino", Journal of Undergraduate Research. RIC, Vol.2- N.º 2, Pág. 9, 2016.
- [3] Freeman M. "SMARTPHONE GARAGE DOOR OPENER". pág.1. 2013
- [4] González O. "Cerradura electrónica RFID controlada con Raspberry Pi", Brico Grec, Pág. 1, 2015
- [5] El Gran Tlapalero. (2019). "Contra eléctrica para puertas cromo 310 Phillips". [online] Disponible en: <https://www.elgrantlapalero.com/contra-electrica-para-puertas-cromo-310-phillips.html> [Accedido 5 Sep. 2019].
- [6] AFEL. (2019). "Fuente de Poder 12V - 2A - 24W - AFEL". [online] Disponible en: <https://afel.cl/producto/fuente-de-poder-12v-2a-24w/?v=2e6507f70a9c> [Accedido 5 Sep. 2019].
- [7] Naylamp Mechatronics - Perú. (2019). "Módulo RFID RC522". [online] Disponible en: <https://naylampmechatronics.com/inalambrico/80-modulo-rfid-mfrc522-.html> [Accedido 5 Sep. 2019].
- [8] Shop.master.com.mx. (2019). "MODULO RELEVADOR DE 5V - Master Electrónicos". [online] Disponible en: <https://shop.master.com.mx/product/detail?id=7605> [Accedido 5 Sep. 2019].
- [9] Static.raspberrypi.org. (2019). "Raspberry Pi 3 Model B+". [online] Disponible en: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf> [Accedido 5 Sep. 2019].
- [10] Raspberry Pi, A small and affordable computer that you can use to learn programming, [On line]. Disponible en: <https://www.raspberrypi.org/>. [Accedido 14-octubre-2018].
- [11] SQLite, [On line]. Disponible en: <https://www.sqlite.org/index.html>. [Accedido 18-abril-2019].
- [12] El Confidencial, Dos millones de razones para saber qué es exactamente Raspberry Pi, [On line]. Disponible en: https://www.elconfidencial.com/tecnologia/2013-11-22/dos-millones-que-es-exactamente-raspberry-pi_56003/. [Accedido 31-marzo-2019].



Máquina Formadora de Hamburguesas a base de Carne Molida

Mascot Ramírez Dulce Janet, Roque González Héctor Manuel y Aguilar Molina
Yehoshua

Universidad de Guadalajara
Centro Universitario de los Valles.
dulcemascot@gmail.com

Resumen

El presente documento describe una maquina cuya acción principal es moldear un producto cárnico previamente preparado que es suministrado manualmente a la máquina, obteniendo como resultado un disco de carne, teniendo la oportunidad de variar la forma del molde.

Este proyecto surge como una respuesta a la tediosa y tardada actividad de generar el disco de hamburguesa manualmente y a los altos costos de las opciones industriales para realizar esta tarea, buscando ofrecer una maquina simple y eficaz situada en un punto entre lo manual y lo industrial.

De esta manera el proyecto se centra en el área de formar de manera automatizada la hamburguesa y transportar, mediante la utilización de una banda, el producto a una etapa posterior como puede ser su empaquetado o su uso inmediato.

Como característica adicional se utiliza un sistema simultaneo de vaciado y llenado de molde haciendo mucho más eficiente y rápido el sistema. Acelerando la producción y abaratando costos.

Palabras clave: compactadora, formadora, hamburguesa, automatizada, productos cárnicos, eficiente.

1. Introducción

La automatización en la industria alimenticia hoy en día es un recurso indispensable cuando se quiere aumentar la producción debido a una alta demanda. Este tipo de procesos esta muy estudiado e implementado a grado industrial. Debido a esto es caro y difícil implementar un sistema automático en el área de las PyME.

Existen máquinas que realizan procesos similares, sin embargo, son costosas e incluso grandes y tienen una alta dificultad de manejo o de mantenimiento, en la mayoría de los casos este tipo de maquinaria es utilizada en grandes empresas y las que se venden en el mercado para empresas medianas y chicas suelen ser manuales.

Por esto se pensó en una maquina la cual cubrieran las necesidades de una institución a bajo precio manteniendo una buena calidad y facilitando el proceso de creación.

Con la implementación de sistemas automatizados se obtiene una mejor supervisión del proceso, mayor homogeneidad y un aumento de producción, además de una mejora en la calidad de los productos.



Algunas de las máquinas de referencia que se encontraron son las siguientes

Nombre	Producción	Características	Desventajas	Maquina	Costo	Imagen
La formadora automática de hamburguesas Castellvall [1]		Distintos tamaños y formas de carne. Regula la presión de llenado.	Es grande y pesada. Manejo y mantenimiento complicado.	Automática	\$130,000	(Imagen 1.1)
Formadora de hamburguesas y nugets Hollymatic Super. [3]	1800 a 2100 porciones por hora.	Removible y con un bloqueo de seguridad. Cuenta con un contador programable digital.	Estándar de 8 Oz. Por porción. Estándar en espesor.	Semiautomática	\$72,000	(imagen 1.2)
La prensa formadora de hamburguesas Garden HP1416. [4]	60 piezas por hora		Diámetro de 12.7cm. Espesor variable.	Manual	\$3,999	(imagen 1.3)
Formadora de carne para hamburguesa Migsa HF130. [5]		Cuerpo de acero y aluminio anodizado	Diámetro y espesor variable.	Manual	\$5,348	(imagen 1.4)
Formadora de carne para hamburguesas Gesame MH-100. [6]	1800 porciones por hora	Tolva automática. Variedad de moldes.	130mm de diámetro. Espesor máximo de 30mm	Semiautomática	\$240,000	(imagen 1.5)
La formadora CRM PLanus. [7]	2100 porciones por hora	Tolva automática. Banda de descarga. Intercaladora de papel	Diámetro y espesor único. Industrial.	Automática		(imagen 1.6)

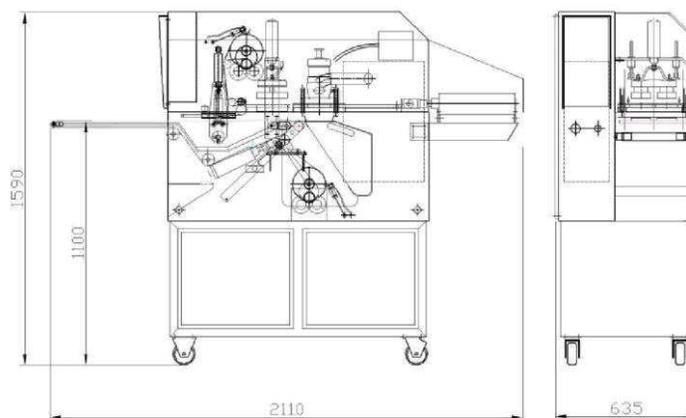


Imagen 1. 1 La formadora automática de hamburguesas Castellvall



Imagen 1. 2 Formadora de hamburguesas y nuggets Hollymatic Super.



Imagen 1. 3 La prensa formadora de hamburguesas Garden HP1416



Imagen 1. 4 Formadora de carne para hamburguesa Migsa HF130.



Imagen 1. 5 Formadora de carne para hamburguesas Gesame MH-100.



Imagen 1. 6 La formadora CRM PLanus

El presente proyecto busca integrar un panel de control en la supervisión, monitoreo y control de una máquina compactadora y moldeadora de carne molida.

Actualmente no se cuenta con una amplia variedad de máquinas con bajo costo que realicen este proceso, con características de autonomía, las cuales son necesarias para el trabajo a gran escala y homogéneo, algunas de las contingencias más comunes en este tipo de sistemas son:

- El tamaño de la carne no es proporcional en todos los empaques.
- No es posible estandarizar el tamaño de cada porción.
- El empaquetado manual implica la inversión de mucho tiempo.
- Se requiere más de una persona realizando esta actividad.

Por lo anterior el monitoreo de los elementos actuadores del sistema tanto de control como de potencia, es de suma importancia ya que, al poder manipular los parámetros antes mencionados, el sistema será capaz de poder detectar las fallas de dichos elementos y reducir tiempos de producción.

2. Desarrollo.

En la primera etapa del desarrollo se comienza con lo referente a la estructura de la máquina. Comenzando el diseño con ayuda de un software CAD en este caso SolidWorks, permitiendo plasmar de una manera gráfica las ideas que se tengan acerca de la forma y medidas de la estructura base y de todas las piezas móviles como se muestra en la Imagen 2.1, ofreciendo un mayor control durante la etapa de diseño, pudiendo cambiar y mejorar piezas y mecanismos antes de su fabricación, además de proporcionarnos la facilidad de hacer análisis de fuerzas de las piezas para poder hacer una correcta elección de los materiales necesarios para la construcción del prototipo.

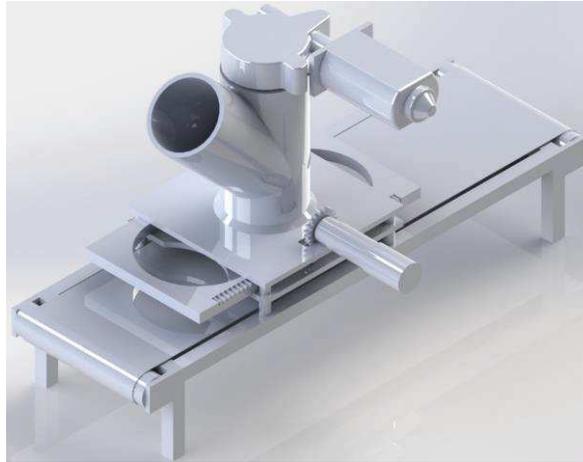


Imagen 2. 1 Prototipo 2

Con esto se tendrá una base sólida para poder continuar con la segunda tarea dentro de la primera etapa que consiste en el análisis del prototipo tarea que se estará desarrollando.

En la imagen 2.2 y la imagen 2.3 se muestra el diseño de primer prototipo en el cual se encontraron ciertos errores en la tolva.

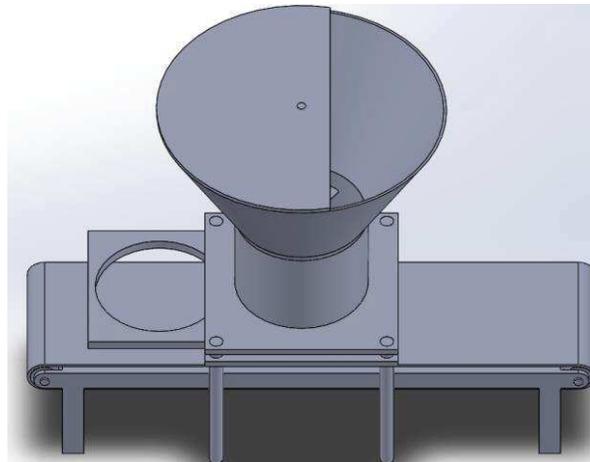


Imagen 2. 2 Prototipo 1: Diseño en solidworks

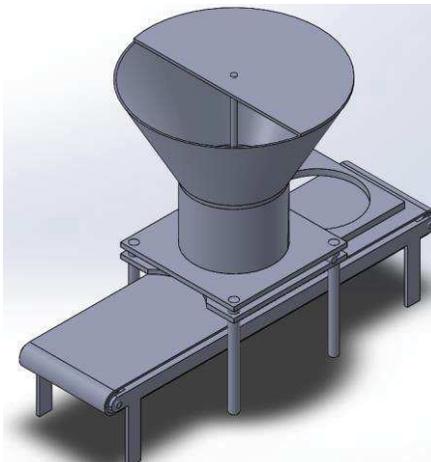


Imagen 2. 3 Prototipo 1: Diseño en solidworks



Como resultado de estos errores se diseñó un segundo prototipo pensando en facilitar la manera de manejo como observamos en la imagen 2.4 y la imagen 2.5, este está diseñado con solidworks para de esta manera predecir errores a futuro o confirmar la estructura del prototipo.



Imagen 2. 4 Prototipo 2



Imagen 2. 5 Prototipo 2

El diseño de la maquina fue pensado y trabajado siguiendo algunas normas NOM, como la NOM 251-5.2.3 “Los materiales que puedan entrar en contacto directo con alimentos, bebidas, suplementos alimenticios o sus materias primas, se deben poder lavar y desinfectar adecuadamente.”

Actualmente existe una norma de inocuidad de los alimentos, los cuales hacen referencia a que los materiales utilizados deben de estar siempre limpios y desinfectados mínimo cuando se comience a utilizar y al finalizar sin importar si están en contacto con el producto o no.



El material utilizado debe de ser inerte que no transmita sustancias tóxicas, olores ni sabores, debe de ser absorbente, resistente a la corrosión y capaz de resistir repetidas operaciones de limpieza y desinfección. Sin grietas ni orificios. En instrumentos de control se requiere de calibración.

En base a la información anterior podemos decir que los materiales ideales para la construcción del prototipo serían, para la estructura metálica aceros inoxidable más específicamente la serie 316 según la norma AISI.

“Acero 316 Piezas que demandan alta resistencia a la corrosión localizada equipo de las industrias química, farmacéutica, textil, petrolera, papel, celulosa, caucho, nylon y tintas; diversas piezas y componentes utilizados en construcción naval; equipos criogénicos; equipos de procesamiento de película fotográfica; cubas de fermentación; instrumentos quirúrgicos”

En la segunda etapa del diseño del prototipo está destinada al diseño e implementación de la parte electrónica y eléctrica que necesitará el prototipo y funcionará como base activa para la programación del sistema de control que permitirá el correcto funcionamiento de las secuencias automáticas y del control manual. Tareas en las que se realizara la elección de un microcontrolador, así como de los actuadores y sensores que se ajusten mejor a las necesidades, además de realizar los circuitos impresos.

En este caso se decidió implementar un sistema de control a base de un microcontrolador arduino gracias a que su estructura es simple, económica. Permitiendo tener un acceso mas simple a los componentes y a su programación. Dentro del sistema de control se incluyen distintos sensores y transmisores que permiten obtener información del proceso, y así mismo tener un registro de actividades y de productividad, además de asegurar la calidad y homogeneidad de las piezas.

Se utilizo como referencia de funcionamiento el diagrama de flujo 2.1 para crear la secuencia de control en el sistema.

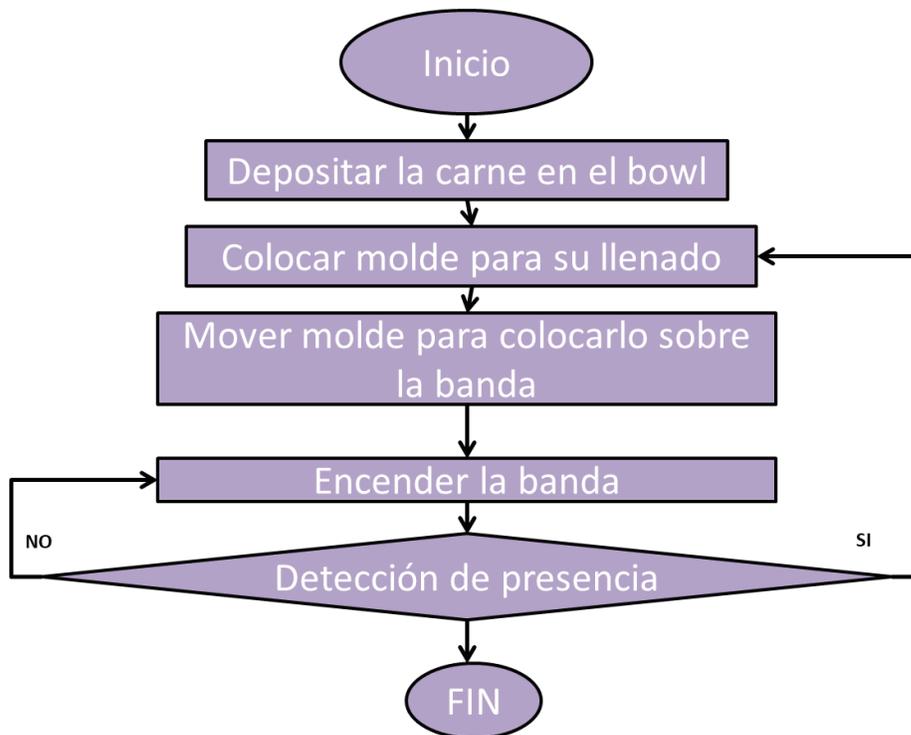


Diagrama de flujo 2. 1



Para este cometido se utilizaron los siguientes componentes

- Arduino nano
- Teclado matricial
- Pantalla LCD
- Sensores de limite
- Encoder
- Motor 100 Kg/cm
- Motor 20 Kg/cm
- Motor 18 Kg/cm

Para la creación del sistema electrónico se utilizó el software DipTrace el cual permite realizar el diseño esquemático y posteriormente la realización del PCB. En esta estará situado todos los componentes electrónicos y de control, así como el arduino, y los componentes de entrada como sensores, además de contar con las salidas preparadas para las etapas de potencia.

En la imagen 2.6 podemos observar el diagrama de los componentes electrónicos y de control esto con la finalidad de observar el diseño y acomodo de los mismos.

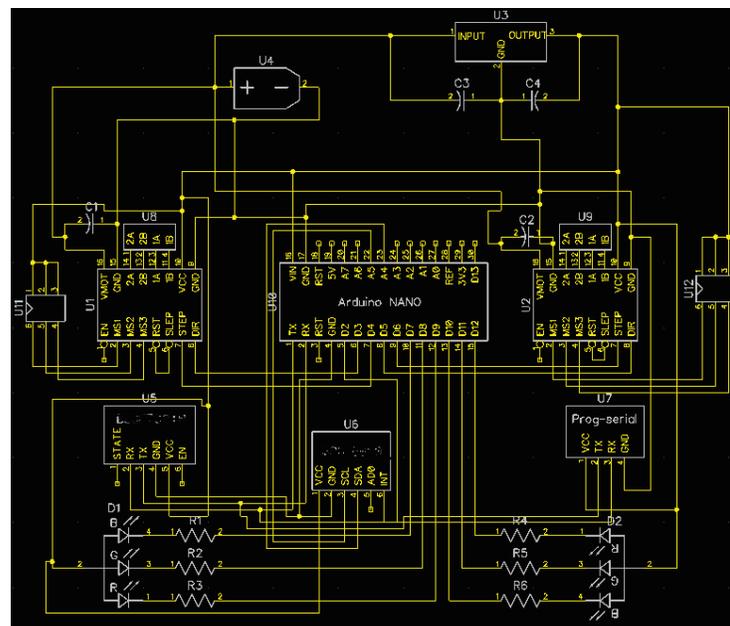


Imagen 2. 6 Diagrama electrónico y de control.

Una vez terminado el diseño anterior se procede al diseño y fabricación de la placa de control en el programa DipTracer, ya que este cuenta con la funcionalidad de convertir los diagramas electrónicos a PCB como observamos en la imagen 2.7.

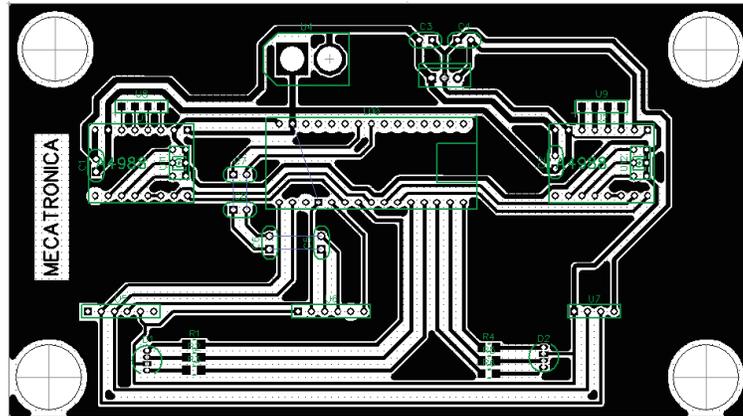


Imagen 2. 7 Placa de control.

En la tercera etapa se realizan dos actividades clave, por una parte, se diseñará el sistema de control de la máquina, que implica desde las secuencias de funcionamiento hasta los sistemas velocidad y posición de los actuadores.

Por otra parte, hará uso el diseño de una interfaz que permita programar secuencias en la máquina, así como realizar un control manual de cada uno de los actuadores, esta interfaz buscará ser sencilla e intuitiva, con el propósito de hacerla amigable a cualquier tipo de potencial operador. Utilizándolo como en la imagen 2.8.

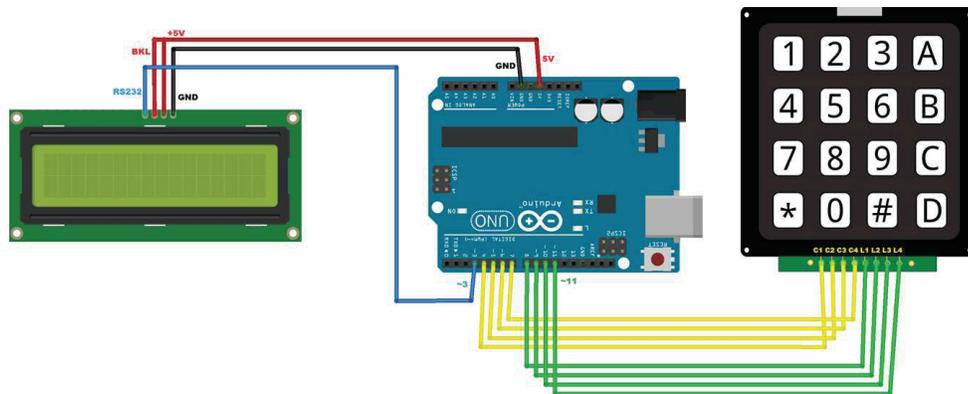


Imagen 2. 8 Interfaz de usuario.

Este proyecto comenzó con la realización de una banda la cual conducirá la carne compactada a su salida.

La máquina consta de una tolva en la cual coloca el producto cárnico previamente molido, mediante un espiral activado por un motor DC este conduce la carne a los discos para su llenado. Los discos son controlados por un motor DC y una cremallera, el cual mediante un relevador es controlado en ambos sentidos, esto con la intención de tener una secuencia de vaciado y llenado mediante dos discos que con ayuda de una banda transportadora tenemos la colocación exacta en ambos sentido sobre esta. El control del sentido de izquierda a derecha depende de la activación de los sensores de límite toando en cuenta un control binario de 2 dígitos.

Esta máquina es capaz de realizar de 10 carnes compactadas por minuto, cabe mencionar que se requirieron de mecanismos para el ajuste de los motores.

3. Resultados

En la imagen 3.1 y la imagen 3.2 podemos ver el primer prototipo de la máquina. En esta se encontraron ciertas deficiencias que reducían la eficacia de la máquina debido al sistema a base de rodillos implementado en un principio, dando como consecuencia demasiada holgura. Este prototipo se realizó con la finalidad de observar su funcionamiento y gracias a esto se descubrió los errores cometidos y no observados en el diseño gráfico.



Imagen 3. 1 Prototipo Físico 1

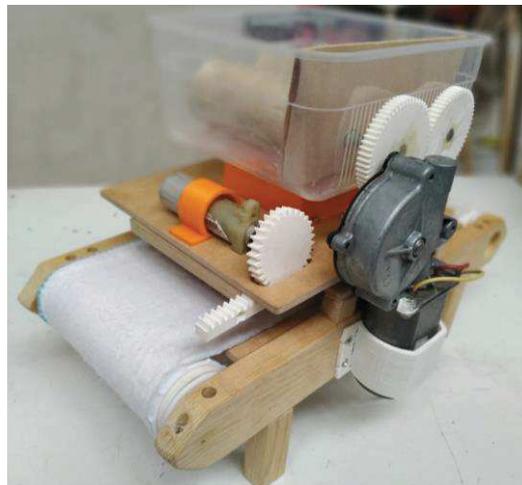


Imagen 3. 2 Prototipo Físico 1

Debido a esto en la segunda versión del prototipo (**Imagen 2.1**) se usaran materiales metálicos y plásticos, más higiénicos. Además de un nuevo diseño. Que podemos visualizar en la imagen 3.3 donde se muestran las vistas de la nueva propuesta.

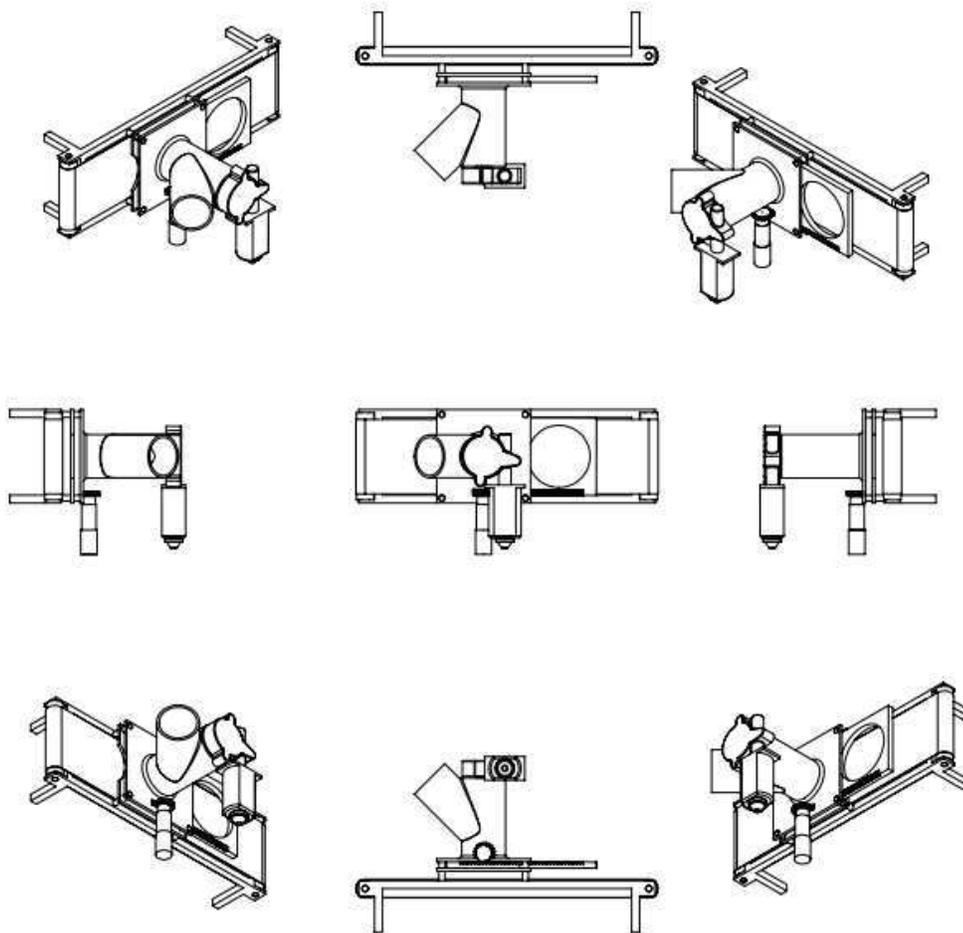


Imagen 3. 3 Dibujo del prototipo 2

4. Conclusiones

Como se ha descrito, la máquina es capaz de realizar una doble actividad la cual es el llenado y vaciado de dos discos esto como opción de reducir los gastos energéticos que podrían causar el hacer solo un disco por vuelta, se implementaron como opciones de ajuste una cremallera la cual mediante un engrane hace que se mueva con más facilidad los discos además de facilitar el control de la misma mediante dos sensores de límite.

Para el diseño de la interfaz se utilizó un sistema opcional es cual mediante información mostrada en el display te guía para poder manejar esta máquina, esta fue hecha mediante programación arduino ya que es más factible y sencilla.

Se optó por cambiar la tolva con rodillos, por una tolva con espiral ya que así se tendrá más precisión al momento de llenado además de esta maneta facilitamos el desmontaje de la misma para su limpieza. El proyecto es de fácil desmontaje en todos sus puntos para poder estar desinfectado y limpiando cualquier residuo que llegue a tener, en el panel de control de igual manera es amigable ya que se pueden reemplazar piezas en caso de fallas.



Referencias

- [1] I. Castellvall, «Formadora de hamburguesas 1FH25,» p. 2, 2006.
- [2] J. V. ANDRADE VILLARREAL, «TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO,» Ibarra-Tesis, ECUADOR, 2015.
- [3] Citalsa, «Formadora de hamburguesas y nugets Hollymatic super,» marzo 2010. [En línea]. Available: <https://citalsa.com/media/talsa-documents/00201003.pdf>. [Último acceso: enero 2019].
- [4] M. libre, «Prensa formadora de hamburguesas manual Garden HP1416,» Garden, [En línea]. Available: https://articulo.mercadolibre.com.mx/MLM-622992801-prensa-para-hamburguesas-de-cuarto-de-libra-_JM. [Último acceso: enero 2019].
- [5] M. libre, «Formador de carne para hamburguesa HF130,» Migsa, [En línea]. Available: https://articulo.mercadolibre.com.mx/MLM-618636802-formadora-de-carne-para-hamburguesas-manual-migsa-ah-hf130-_JM. [Último acceso: enero 2019].
- [6] F. d. hamburguesas, «Carnotex,» Gesame, [En línea]. Available: <http://carnotex.com/?product=gesame-mh-100>. [Último acceso: enero 2019].
- [7] Carnotex, «formadora CRM PLANUS,» CRM PLANUS, [En línea]. Available: <http://carnotex.com/?product=crm-planus>. [Último acceso: enero 2019].



Preparador Automático de Bebidas

Ayón Rubio Cristóbal, Santana Garrafa Ulises, Domínguez García Rodolfo Omar
y Aguilar Molina Yehoshua

Universidad de Guadalajara
Centro Universitario de Valles
Cristo_hitotmail.com

Resumen

El Barman Inteligente es un dispositivo capaz de preparar bebidas profesionalmente, ampliando el número de bebidas que desee agregar, así como tener la capacidad de poder configurar las porciones de manera personal, con una interfaz de dos simples botones. Adaptable en cualquier espacio de nuestro hogar, debido a reducidas dimensiones del Batman y su elegante presentación, así mismo se reduce el precio de una bebida en cualquier bar pues el uso de este producto solo se reduce en el consumo mínimo de electricidad llegando a obtener tu bebida favorita por unos pocos centavos. El dispositivo puede mezclar hasta 6 bebidas diferentes, pudiéndolas mezclar en cualquier tipo porcentaje y contenido, cuidando siempre las configuraciones iniciales para poder tener la bebida solicitada.

El Barman también tiene el potencial para poder ser conectado con Alexa o Google Assistant. El modo de operación a distancia es para beneficio del usuario, el objetivo es poder pedir un pedido inclusive no estando presente.

Palabras clave: automatización, bebidas, mezclador, preparación, barman.

1. Introducción

. Un dispositivo es aquel que está en la barra y cuya función principal de estos es servir, no tiene más complejidad a diferencia de un bar tender el cual tiene una función un tanto más compleja ya que este se encarga de realizar tareas administrativas como inventarios, atender a los clientes y brindarles una experiencia, es por ello que el propósito de este proyecto es presentar un poco de tecnología en cuestión de sabores presentando un barman automático, una máquina que tiene la capacidad de hacer bebidas preparadas con distintos sabores o diferentes sustancias. La idea principal es que a través de un software comandar el sabor o los sabores a mesclar a través de solo un clic.

Una herramienta muy practica y eficiente en todo diseño ya sea mecánico, electrónico, mecatrónico, et., es el uso de metodologías a través de pasos o secuencias que el diseñador sigue la cual se toma como guía para el desarrollo y evolución de una idea, cada diseñador, así como cada diseño tiene esencias únicas, retos, obstáculos y desde luego las necesidades diferentes para llevar a cabo el desarrollo de la idea. Como primer punto es necesario identificar el problema o necesidad a cubrir, una vez previsto el punto anterior se prosigue el desarrollo de la investigación mismo que ayuda a visualizar a y la vez analizar que tanto ya se ha realizado para posteriormente a partir de la información obtenida tomar un punto de partida en el desarrollo del diseño.

El “barman automático” puede hacer bebidas mezclando hasta 6 ingredientes al mismo tiempo (o dependiendo la cantidad de espacios que se configuren o físicamente la cantidad máxima de entradas posibles), para que pueda tener cualquier ingrediente líquido, desde un ron, tequila, wiski, coca cola, Jugo, etc. Se puede personalizar para preparar cualquier cantidad de bebidas, se puede ampliar para mezclar más ingredientes y también tiene el potencial de conectividad con Google



Assistant. Las bebidas se pueden configurar en el barman con dos botones simples y una pantalla, y puede agregar cualquier cantidad de bebidas al repositorio del camarero a través del código del barman.

2. Planteamiento del problema

En esta sección se debe explicar la problemática particular a tratar y como el proyecto presenta una solución a este problema.

Las mil maneras de preparar una misma bebida existen. Actualmente las personas que consumen una bebida cuestan poderla duplicar en una exactamente igual, obtener un trago perfecto en las proporciones correctas es difícil de dominar. Los principales errores es el manejo correcto de las porciones que se necesitan añadir. Las personas necesitamos acudir a lugares especiales en donde las bebidas puedan ser preparadas correctamente, que en ocasiones eleva mucho el dinero invertido para poder tomar un buen preparado de bebida, lo que implica el hecho de tener que salir a buscar los lugares correctos, con las personas y que tengan los instrumentos correctos para poder elaborar una original bebida.

También existe la necesidad de poder cambiar las porciones a preferencia para poder disfrutar un gusto personal. Resulta incómodo pedir estas peticiones a la persona que lo prepara, diversos a varios motivos, como, por ejemplo: el que siga más las indicaciones, elija un ingrediente erróneo etc.

Este novedoso dispositivo iguala los servicios que ofrece un barman despachando bebidas preparadas a partir de otras, basándose en ingrediente anteriores. destaca su compacto diseño para poder moverlo de lugar fácilmente, así como su accesible costo de fabricación que minimiza en gran medida los costes que pueda tener un modelo similar.

Este artefacto tecnológico está pensado para uso personal y domiciliario, llevando la complejidad en un sistema compacto y trayendo consigo resultados profesionales en cada uno de sus servicios de bebida. También añade un sistema de configuración en el cual el usuario pueda cambiar los porcentajes de algún ingrediente de acuerdo a los gustos de cada persona. También tiene un modo de configuración para poder añadir nuevas preparaciones a la memoria de la maquina y esta pueda modificar sus operaciones conforme los requiera., con una interfaz sencilla de solo dos botones y elegante presentación del modelo barman inteligente.



Figura 1. dispensador de bebidas.

En la figura uno se muestra un dispensador, lo cual su control se deriva de maquinas y no de humanos, por lo que es necesario llegar a un modelo similar dentro de su automatización.



Los errores en los humanos con factores de proporción son muy concurrentes, al cambiar la cantidad de porción es indispensable tener medidas y un equipo dedicado exclusivamente al uso y preparación de bebidas. Este equipo suele ser costoso en su adquisición y el personal que pueda laborar en ello se debe pasar un largo tiempo practicando y adiestrando sus habilidades para poder obtener una buena bebida.

El Barman tiene la cualidad de ser un dispositivo de precisión ya que su ejecución se basa en un código de interpretación digital, los dispositivos encargados son siempre monitoreados por el controlador de la raspberry esto nos ayuda a obtener lecturas precisas del tiempo de la ejecución teniendo un resultado muy preciso. Colores de todo este tiempo se ha estado desarrollando diferentes tipos de dispositivos cubren las acciones y trabajos de los humanos volviéndolos más accesible volviéndolos más accesibles hacia las personas, esto nos brinda una ventaja en cuestiones de reducción de precio, una máquina capaz de lograr o superar las acciones que puede realizar un humano tiene gran precio de acción regularmente, pero éste no es el caso ya que equipo montado en la estructura tiene un costo bajo en comparación de la adquisición de otro dispositivo.

2.1 Estado del arte

UXBAN-MONSIEUR- esta empresa estadounidense de desarrollo tecnológicos enfocado en el diseño de la vivienda inteligente trae al mercado electrodomésticos capaces de interactuar con el usuario, conformado por el internet de las cosas. Especialmente el modelo de barman (Monsieur), reúne las características que maneja el internet de las cosas ya que se conecta a la red domiciliar y puede recibir instrucciones mediante un teléfono celular. Limitado al número de configuraciones que tiene, y teniendo que conseguir la materia prima para preparar las bebidas o cocteles.

BAR ROBOTICO BARBOT- un dispositivo creado específicamente para la preparación de mezclas de bebidas y nada más, resalta la simpleza del mecanismo que utiliza para servir, utilizando una banda transportadora y colocándose en el lugar indicado para llenado. Producto 100% mexicano y presentado por primera vez en el programa televisivo shark tank México, Barbot puede conectarse a través de un dispositivo inteligente con conexión bluetooth para realizar pedidos. Desventajas: utiliza un proceso de llenado bastante lento, el consumo de energía es muy elevado debido a sus tantos componente móviles y luminosos, su tamaño hace complicado su movimiento a otro sitio, bastante costoso.

TIPSY_ es un robot "mixólogo" que concita la atención de los visitantes de un centro comercial de Las Vegas en el que se puede pedir un "dirty martini" – sacudido o mezclado – por computadora. Luego de los distribuidores de cupcakes o los restaurantes de pizzas automatizadas, los autómatas desembarcan en las barras de los bares. Monsieur y Tended Bar son aparatos de distribución de bebidas a la manera de una máquina de café: uno puede seleccionar su caipiriña o su bloody mary en un menú digital y unos segundos después tiene su bebida. La idea no se aplica únicamente en Estados Unidos: las start-ups parisinas Realltime Robotics y Kuantom también han creado barmen mecánicos y las máquinas de expedición de bebidas alcohólicas son comunes en las calles de Tokio.

2.2 Justificación

Este modelo ofrece la posibilidad de poder tener la oportunidad de preparar las bebidas que más agraden al cliente en su hogar, siendo accesible en el coste y teniendo un proceso final profesional y de un gusto personal.

No descartando el lujo de poder tener un electrodoméstico inteligente y elegante en su presentación.



2.3 Objetivos

Objetivo general:

lograr un prototipo profesional y compatible para todo usuario por su configuración y agrado a primera vista. En su función principal, preparar bebidas como si de una persona profesional se tratara.

Objetivo particular:

Tener un producto agradable visualmente, poder tener las funciones adicionales como Google Assistant para poderlo manejar remotamente.

2.4 Requerimientos técnicos

El Barman Inteligente requiere una conexión a la red eléctrica domiciliar normal, los gastos energéticos son mínimos, el espacio que requiere no excede el de un microondas, teniendo una dimensión de 16' x 9' x 7.5', la funcionalidad principal es tomar de bebidas base combinarlas en las porciones correctas para obtener una bebida diferente, los componentes clave es el controlador la cual es una raspberry, así como unas bombas peristálticas que tienen un causal constante tolerancia de error muy pequeña.

3. Metodología

La construcción de este dispositivo se tiene en cuenta que el grado de construcción deben, de ser de grado industrial ya que al manipular líquidos o alimentos el cual son de ingesta directa por las personas debe tener cuidado en la higiene y trato de esto.

En la separación de los contenidos debe ser cuidado y monitoreado por el mismo sistema, ya que un error en el cambio de precisión por ejemplo de las bombas puede denotar una bebida mal preparada, dando así productos insatisfechos, por lo que se toma en cuenta un sistema experto capaz de estar cuidando los resultados así mismo como el censo del propio estado de la máquina.

El diseño de las estructuras deberá ser lo suficientemente adaptable a cualquier sitio dentro del hogar de modo que las dimensiones deberán ser lo más pequeños posibles asimismo el material construcción deberá ser ligero y resistente para su soporte. Es bien sabido es tema lleva en su interior circuitos electrónicos que son muy susceptibles a desgastes, cortocircuitos, humedad, etc. Por eso es importante resguardar y colocar en un sitio adecuado y referente a lo electrónico para evitar posibles errores en su funcionamiento.

No menos importante revisar tecnología y productos similares a nuestro barman con el fin de poder obtener ideas y perfeccionarlas, también conociendo sus debilidades errores que puedan ser solucionados el diseño correcto de la estructura y su funcionamiento ahorrará dinero en su elaboración y construcción e incluso en su mejoramiento.



Figura 2. Barman creado por grupo mexicano

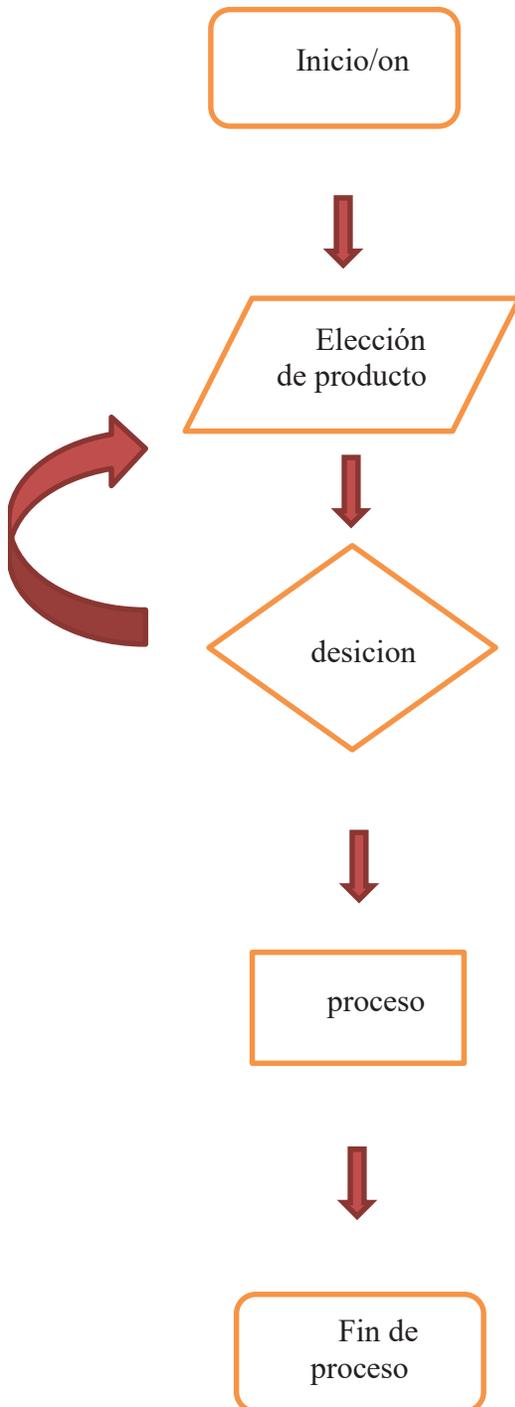
La figura 2 es un producto el cual tiene la misma función a la creada, pues se encarga de preparar bebidas de una forma autónoma, mediante la selección de una de sus elecciones, este diseño es innovador y carece de ciertos errores el cual por ser nuevo se presentan.

3.1 Operación del sistema.



Diagrama de flujo

Barman Inteligente



El dispositivo entra en servicio inmediatamente después de conectarlo a la alimentación eléctrica.

La señal de entrada el cual esta en manos del operador para elegir el tipo de bebida.

La maquina opera con base a su programación la manera de poder ejecutar lo solicitado. De no encontrarla en su memoria se solicitará nuevamente un producto

La ejecución de los actuadores se hace presente, lo que activa el proceso para cumplir la elección. Pone en prueba el diseño como la electrónica interna. Al manejar liquido debemos tener cuidado en su manipulación

El servicio de preparación del trago se ha realizado con éxito, logrando terminar con el proceso satisfactoria.



3.2 Etapas del proyecto

- 1) Construcción de la estructura del Barman: añadir una estructura sólida para poder sostener todo el contenido y colocar en su interior la electrónica.
 - 1.1) forrarlo con un material de calidad para así lograr un buen estilo de diseño en su exterior.
- 2) Diseño de piezas necesarias para la estética y mejor dinámica del Barman, las piezas deberán estar echas a medida e impresas en 3D e instalarse en la estructura inmediatamente de haberse terminado.
- 3) Programación del código sobre la Raspberry, ejecutaremos el código necesario para ejecución lógica del circuito, implementaremos una pantalla en su uso para su fácil interfaz a las personas.
- 4) Montaje y cableado de la placa electrónica, de este modo el acomodo de las piezas eléctricas nos concentraremos en tener cuidado para no causar ningún accidente durante el proceso.
- 5) Presentación del documento en el se reporta todo lo elaborado del proyecto, así mismo se presenta una redacción formal del uso, propósitos y sostenibilidad del proyecto en su uso.

Cronograma

diagrama de Gantt

Muestra el desarrollo de las etapas planeada para la elaboración del preparador de las

TIEMPO/ ETAPAS	Septiembre	Octubre	Noviembre	Diciembre
Construcción de la estructura	[Bar chart showing duration from early Sept to mid-Oct]			
Revestimiento	[Bar chart showing duration from mid-Oct to early Nov]			
Diseño de piezas	[Bar chart showing duration from early Nov to mid-Dec]			
Programación del código	[Bar chart showing duration from mid-Dec to early Jan]			
Montaje y cableado	[Bar chart showing duration from early Jan to mid-Feb]			
Presentación del documento	[Bar chart showing duration from mid-Feb to mid-Mar]			

bebidas, cada una de ellas representado con un el sistema de lapso el cual se sugiere en un lapso de tiempo.

4. Resultados

Después de la investigación realizada se opta por un diseño de barman capaz de generar cocteles o bebidas preparadas a través de una interfaz móvil, cumpliendo así los objetivos planteados o propuestos dentro del proyecto. A través de sensores de presencia se cumplen acciones necesarias como lo son: la posición del portavasos de igual forma la presencia del vaso los cuales son pieza fundamental para el funcionamiento del prototipo.

A través de un diseño mecánico se desarrolla un prototipo que cumple condiciones favorables de confortabilidad, accesibilidad, fiabilidad y bajo costo de fabricación.

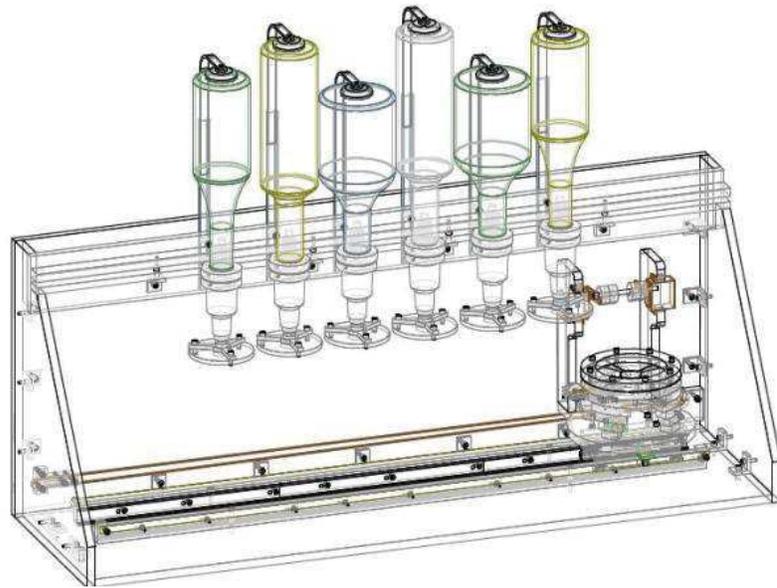


Figura 3. Estructura de un preparador de bebidas

5. Conclusiones

El proyecto se encuentra en una fase de desarrollo y mejora de las piezas, así como su producción, nuestro proyecto habla de cómo poder tener acceso a un instrumento que en el mercado actual es bastante caro tenerlo, y muy complejo sus movimientos, de este modo la aportación es crear algo más accesible a un ciudadano promedio. Pues estos instrumentos solo se ven en países de primer mundo en lujosos o grandes establecimientos, y puede ser un poco contradictorio estar comprándole cosas a una máquina.

La necesidad de crear un dispositivo de un alto nivel y poderlo tener en tu hogar sin la necesidad de poder salir de tu casa y gastar más de lo que se debe por una bebida, o conseguir una persona con conocimientos de esto.

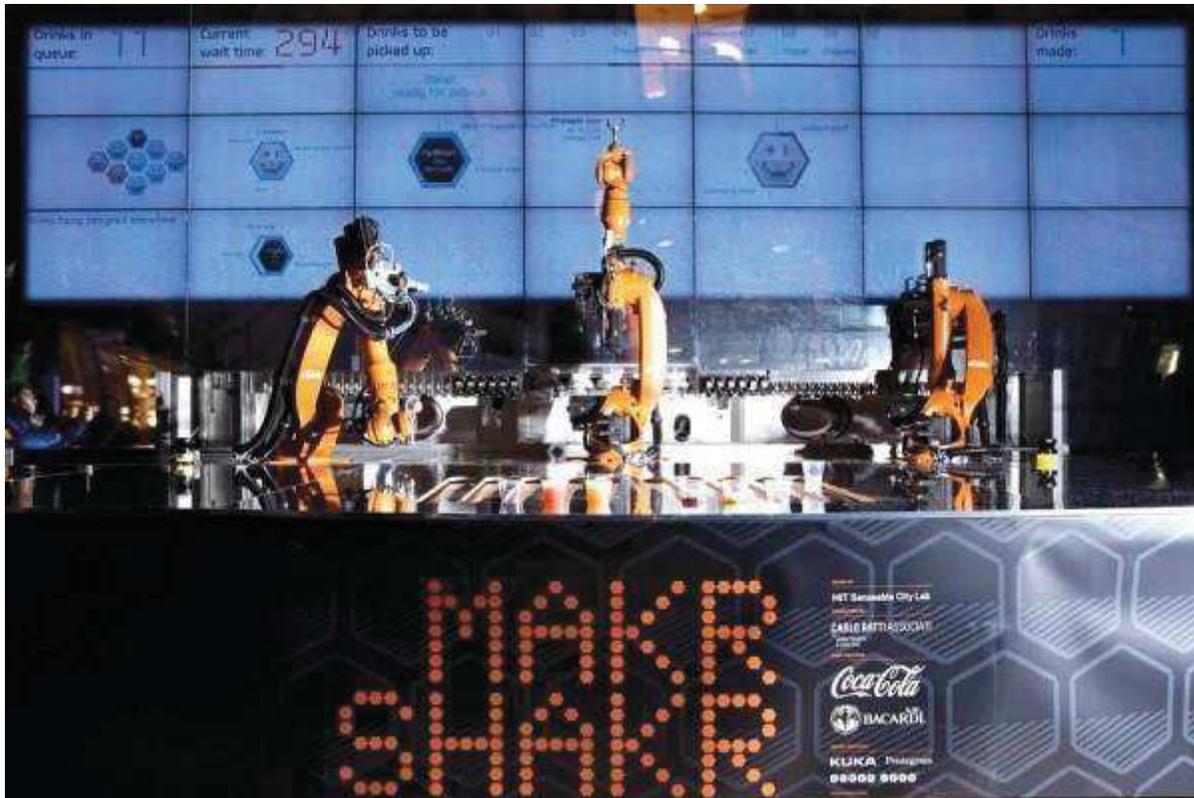


Figura 4. Robot tras de una barra

En la figura 4 muestra como los robots remplazan las actividades de servir, obviamente se ve que es un diseño costoso y poco accesible al público, claramente se ve que el acceso a este tipo de servicio es inaccesible para muchos. El modo de operación de nuestro dispositivo podrá ser accesible, y poderse trasladar fácilmente como cualquier otro electrodoméstico.

El presente proyecto nace a partir de la necesidad que todo individuo cuenta o tiene de servir una bebida para toda ocasión, llámese los distintos tipos de eventos (fiestas, convivios, cumpleaños, etc.) o visto de otra manera; para fines comerciales (bares, restaurantes, antros, etc.) o algún otro momento en el cual es indispensable el uso de bebidas, de tal forma que el diseño de un prototipo de barman automático tiene como consigna cubrir esas necesidades y dar solución a la problemática planteada, realizando el trabajo de servir una bebida sin hacer el mínimo esfuerzo o fatiga de preparación.

El empleo de una metodología de investigación de diseño y desarrollo de prototipos es una herramienta que ayuda a economizar tiempo, espacio y esfuerzos; ya que esta es una guía para todo trabajo de investigación.

El término del diseño de prototipo es de suma importancia ya que este debe de cumplir con todas las especificaciones y necesidades detectadas. La definición del problema es el punto de partida, el punto donde el diseño nace, es un parámetro de evaluación ya que durante el desarrollo del diseño se tendrá presente la esencia del mismo, en pocas palabras **“Cubrir las necesidades y objetivos”**

Una manera de proveer soluciones e ideas de diseño es a través de entrevistas ya que mediante esto se generan ideas y se comparan soluciones existentes de lo que la sociedad opina y sobre esta información se puede hacer una toma de decisión de cada una de las necesidades a cubrir.

Otro punto importante es determinar y obtener los requerimientos y especificaciones los cuales son parámetros fundamentales para el proceso del desarrollo del diseño.



En la etapa de teoría conceptual es importante llevar a cabo un orden desde la generación de conceptos y su evaluación hasta el final del contenido, puesto que ninguna alternativa de solución puede ser clasificada como errónea o correcta hasta el momento de hacer la fase de experimentación o de comprobación.

El desarrollo de un dispositivo de barman automático puede ser de mucha utilidad para cualquier negocio, puesto que a través de él se pueden obtener grandes ventajas como lo es el control de información de servicio (se refiere que a través de este se puede lograr la identificación de la bebida más vendida así como también el registro de las porciones o cantidades de líquidos que se han vendido, y a la vez ayuda a la realización de un inventario el cual es de suma importancia para toda empresa o negocio)

La sinterización de un prototipo de esta índole tiene grandes posibilidades de ser aceptado dentro del mercado puesto que hoy en día es un poco inusual ver este tipo de dispositivos en los diversos lugares, tal vez sea porque el precio es un tanto elevado que ofrecen los diferentes desarrolladores o estos no cuentan con características que satisfagan las necesidades del consumidor, es por ello que a través de la investigación previa se tomaron los diferentes puntos relevantes que forman parte de un aparato novedoso, con calidad y sobre todo la seguridad y confort del usuario. Sin embargo, es de considerarse tomar en cuenta algunos aspectos de Propiedad Intelectual puesto que los dibujos industriales y marcas son parte esencial de los desarrolladores.

Es factible el diseño de un sistema de dispensador de bebidas automático (Barman automático), hoy en día la tecnología es muy importante en la vida diaria de los seres humanos es por ello que la implementación de dispositivos novedosos capaces de resolver las necesidades de una manera más sencilla, pero a la vez más eficiente toma un valor muy importante en visiones futuras. Un sistema automático tiene grandes ventajas a un sistema convencional, hablándose del presente proyecto y haciendo una analogía; el famoso barman (dispensador de bebidas) clásico suele ser el puesto ocupado por una persona el cual tiene la función de preparar las diferentes bebidas pero este tiene la tendencia de agotamiento físico según el trabajo realizado lo cual se convierte en una desventaja en cuestión de producción, por lo contrario un barman automático es aquel aparato que puede realizar la misma consigna que un barman clásico pero con la seguridad pero con un grado más de efectividad en cuestiones laborales.

El mecanismo completo es ideal para ejercer las tareas propuestas. El sistema electrónico que cuenta el prototipo está basado en otorgar eficiencia energética y por ende el consumo de energía eléctrica es un punto importante a destacar sobre el dispositivo, obteniendo así una relación satisfactoria entre costo-beneficio lo cual se convierte en una alternativa especial para todos los usuarios.

Puesto que el barman automático es un dispositivo novedosos, la introducción al mercado podría ser más amplio. Para ello y como trabajo a futuro se tiene que determinar el tamaño de demanda, así como también la aceptación de los usuarios.

Referencias

- [1] A. Egea, «clipset,» [En línea]. Available: <https://clipset.20minutos.es/los-robots-son-los-nuevos-barman/>. [Último acceso: 9 septiembre 2019].
- [2] «<https://www.makrshakr.com/>,» [En línea]. Available: https://www.carloratti.com/wp-content/uploads/2013/12/20130815_MovistarNextMagazine.pdf. [Último acceso: 9 septiembre 2019].
- [3] j. L. G. Friero, «campus training,» [En línea]. Available: <https://www.campustraining.es/noticias/barman-diferencias-barman-bartender-mixologo-camarero/>. [Último acceso: 9 septiembre 2019].



Máquina para la Producción de Pan Caliente

Gamiño Coronado Jesús Antonio, Rosas Garibay Francisco, Aguilar Molina
Yehoshua y Domínguez García Rodolfo Omar

Universidad de Guadalajara
Centro Universitario de los Valles
jesus.gamino@alumnos.udg.mx

Resumen

La Mecatrónica es un campo interdisciplinario de las ciencias de ingeniería. Como disciplina científica, representa la integración correlacionada de las áreas clásicas de la mecánica, electrónica, informática y control. [1]

El presente trabajo expone el diseño e implementación de una máquina totalmente automatizada para la producción de mini hot cakes. Se utilizó como herramienta un software especializado de diseño CAD 3D para modelar las piezas mecánicas, ensamblajes 3D, planos 2D y observar mediante una herramienta integrada en el mismo la simulación del comportamiento en general de las partes mecánicas del prototipo.

El control de la máquina lo realiza un plc Logo! OBA6 de siemens 12/24 rce, con él se controla la lógica del sistema. Además, se emplea el usode Arduino como tarjeta de control de los servomotores debido a que el plc Logo! no cuenta con salidas PWM necesarias para el control. Se realizó un estudio en pequeñas plantas de la industria alimenticia para encontrar alternativas que pudieran servir para llevar a cabo este proceso de una manera más segura, higiénica, confiable, rápida, sencilla y barata.

Palabras clave: plc, ensamblajes, control, PWM, máquina, simulación.

1. Introducción

El término Mecatrónica, ha logrado imponerse a nivel internacional. Es un sello de distinción del trabajo interdisciplinario de desarrollo, producción y comercialización de productos inteligentes. Es una disciplina de gran significado y actualidad, la cual es promocionada como prioridad, en la actividad industrial de los países desarrollados.

Lo que se busca con este proyecto es integrar en un solo prototipo los diferentes campos que integran a la Ingeniería Mecatrónica (Informática, Control, Electrónica y Mecánica), para la producción de pan caliente en serie de una manera más eficaz, rápida y segura.

Para esto se utilizan dos dispositivos para el control de los actuadores utilizados, el primero consta de un PLC Logo! y el segundo en un microcontrolador Arduino.



2. Planteamiento del proyecto

En restaurantes, hoteles, pastelerías industriales y procesadoras de alimentos se necesita agilizar el proceso de producción de ciertos alimentos, ya sea a causa de diversos factores como la necesidad de tener un alcance mayor de producción en un para clientes potenciales. En este proyecto se buscó centrarse en la producción de Hotcackes de una forma automática que solo requiere la intervención de la persona para rellenar el contenedor del prototipo con la mezcla para Hotcackes cada vez que esta se termine. Como perspectiva del proyecto se piensa resolver dicha problemática con la necesidad de ahorrar tiempo debido a que se implementara un sistema automático controlado por un PLC (Programmable Logic Controller) que lleve a cabo el proceso de producción de los Hotcackes de una manera cíclica una vez que el operador inicie el proceso por medio de pulsadores y/o botonera integrada en el prototipo. Nuestro sistema será desarrollado con bases de cada una de las ramas que son parte de la Mecatrónica, (Control, Informática, Mecánica y Electrónica) las cuales a continuación se describe que parte del prototipo corresponde a cada una de ellas:

Control: El sistema de nuestro prototipo estará controlado por un PLC Logo de Siemens, que será el encargado de verificar, dar órdenes, monitorear y tomar decisiones en el cada uno de los pasos del proceso.

Electrónica: Los circuitos electrónicos y de potencia que se utilizan en el prototipo sin duda alguna son parte de esta rama ya que todos estos componentes son el alma del sistema y juntos forman el circuito para que las señales, datos y alimentación fluya de manera que haga funcional la máquina.

Informática: La parte de nuestro proyecto que corresponde a esta rama es porque se utilizan Software especiales para la programación del PLC que se utilizara, la correcta programación es la base para que el proceso se lleve a cabo paso por paso.

Mecánica: los elementos mecánicos más importantes que se utilizan en el proyecto son servomotores que se encargan de mover la pala mecánica para voltear los Hotcackes una vez que se cocinaron de un lado, y además el motor acoplado a la banda tangencial, que es el encargado de hacerla girar.

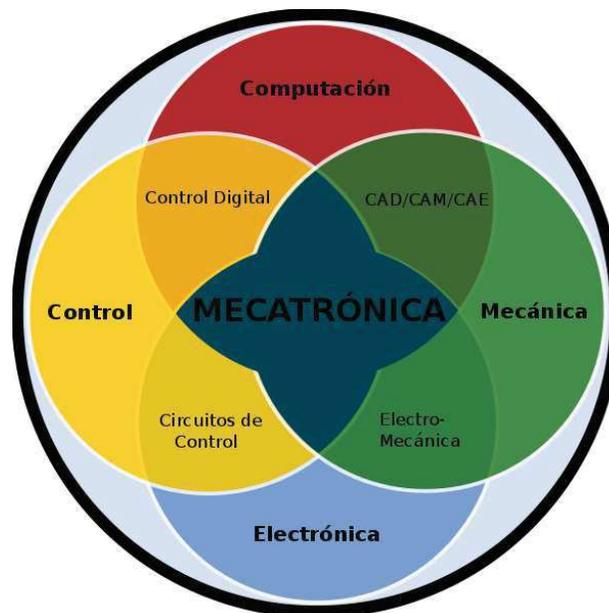


Figura 1: Campos de la Mecatrónica.



2.1 Justificación

La industria alimenticia es la parte de la industria que se encarga de todos los procesos relacionados con la cadena alimentaria. Se incluyen dentro del concepto las fases de transporte, recepción, almacenamiento, procesamiento, conservación, y servicio de alimentos de consumo humano y animal. Es una de las industrias que continuamente sufre cambios respecto a los productos que ofrece ya que constantemente tiene que incorporar nuevos productos con visión de negocio o mejorar los productos potenciales y tradicionales que por mucho tiempo se ha encargado de distribuir. En este caso vemos como producto potencial los Hotcackes, alimento que por años ha sido uno de los favoritos en los hogares de México, por su diversidad de combinaciones de sabores y presentaciones. Es un producto potencial con oportunidad de crecimiento, y para su producción se ha propuesto un sistema automatizado para la producción en serie de este producto.

Además, para la realización de este prototipo se tomaron en cuenta consideraciones muy importantes las cuales se describen a continuación:

Seguridad: La seguridad de la persona encargada de supervisar el proceso debido a que el proceso artesanal es algo riesgoso por las altas temperaturas que se manejan al cocinarlo; este prototipo tendrá su propio suministro de masa e inyección de la misma a una plancha caliente evitando que la persona a cargo quede fuera del alcance de alguna quemadura. Además, el prototipo contará con un pulsador o paro de emergencia en caso de que ocurra alguna falla mecánica o algún otro accidente no previsto.

Higiene: Las piezas o mecanismos que tienen contacto directo con el alimento se fabricarán de acero inoxidable tipo 316 evitando así que el producto sufra algún tipo de contaminación por presencia de óxido ferrítico causado por la limpieza con álgenos.

Confiabilidad: Será un sistema totalmente automatizado y confiable, contando con sensores de temperatura para tener mejor control de la misma y realizar el proceso de cocción de la manera más uniforme posible evitando que el panecillo se retire de la máquina crudo o en caso contrario totalmente quemado.

Rapidez: Debido a que todas las funciones serán programadas, en el proceso de preparación del panecillo no se desperdiciará tiempo porque toda la serie de pasos se realizará uno después de otro tomando un tiempo bien establecido.

Sencillez: Para el uso de este prototipo no se necesitará ser un experto en programación ni tener conocimientos previos, es decir, está diseñado para que cualquier persona pueda operar la máquina.

Durabilidad: Debido a que el prototipo será diseñado a base de materiales resistentes y con una gran resistencia al tipo de entorno de trabajo se estima que su estructura tenga duración acorde a la vida útil de los elementos utilizados, y con un óptimo desempeño según se aplique el mantenimiento preventivo.

2.2 Objetivos

Objetivo general: Obtener un diseño innovador mediante la aplicación de las diferentes áreas de la Mecatrónica para la producción en serie de Hotcackes y que satisfaga los requerimientos de producción industriales y entregue producto de calidad.

Objetivo particular: Una vez concluido el proyecto, se continuará trabajando en el para actualizar el diseño mejorando las áreas de oportunidad en que fallo.

Comercializar el sistema y utilizar el principio de operación del mismo para crear más sistemas útiles en la producción de alimentos.



3. Desarrollo

3.1 Diseño

En esta sección se describen tanto el software como el hardware que fueron empleados para la elaboración de este proyecto y el diseño de este.

SolidWorks: SOLIDWORKS es un software de diseño CAD **3D** (diseño asistido por computadora) para modelar piezas y ensamblajes en 3D y planos en 2D.

Además incluye una gran variedad de herramientas de diseño 3D para el mecanizado de piezas, simulación para evaluación del diseño que lo hacen en uno de los softwares más competitivos actualmente en el mercado. [2]

Equipo de cómputo: En la figura 2 se muestra el equipo de cómputo que fue utilizados para realizar las piezas en 3D para el proyecto, además las características se especifican en la tabla 1.

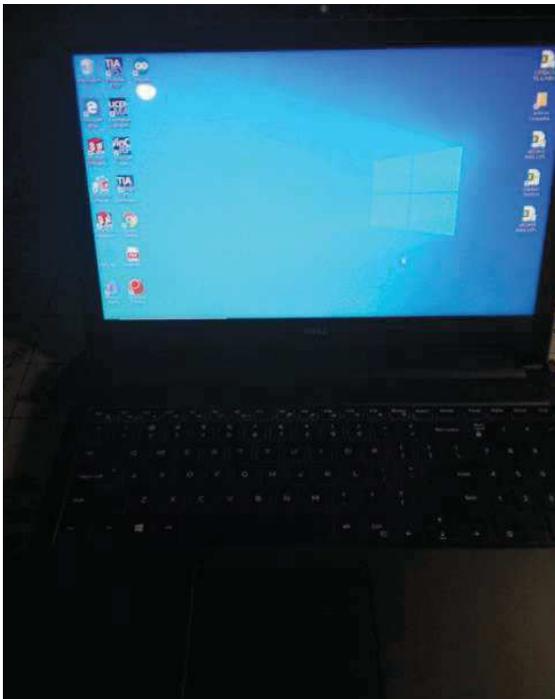
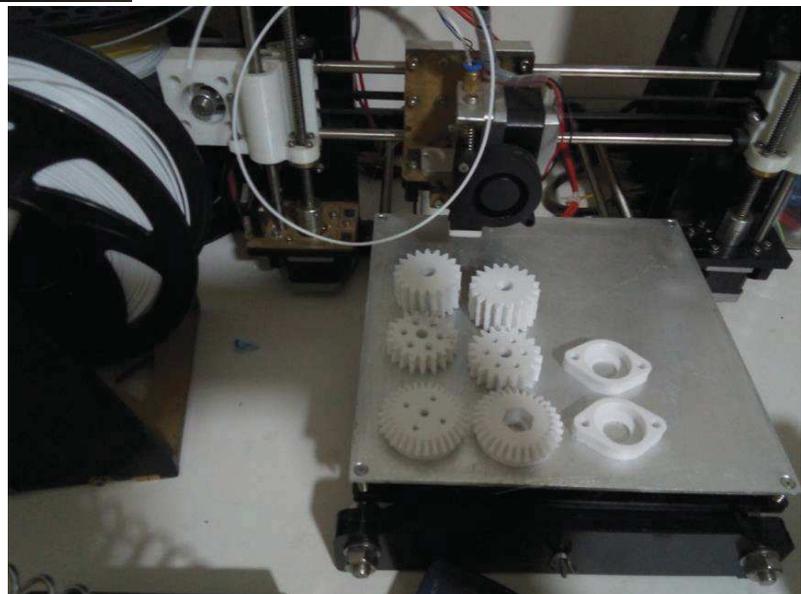


Tabla 1: Especificaciones del equipo de cómputo

Marca	Dell
Modelo	Dell Inspiron 15559
Procesador	Intel Core i5
Memoria instalada	960 Gb 8Gb Ram
Tipo de sistema	Windows
Edición de Windows	10 Pro

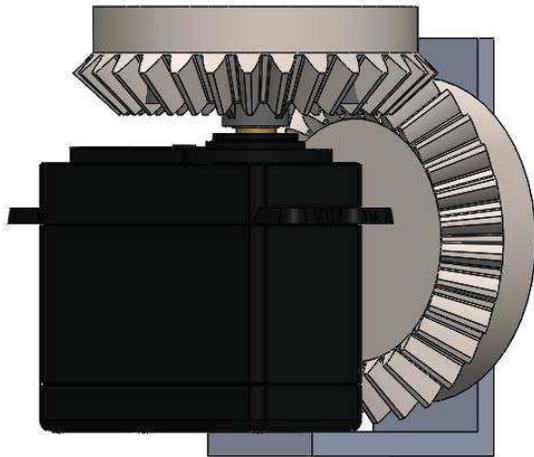
Impresora 3D marca PRUSA K3



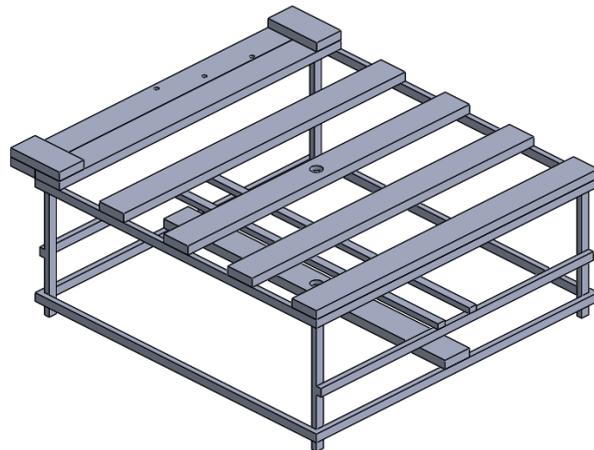


El diseño de todas las piezas mecánicas que componen el dispositivo fueron modeladas en su totalidad mediante el software SolidWorks, con el fin de realizar el ensamblaje de cada una de ellas y ver la simulación en conjunto de todas.

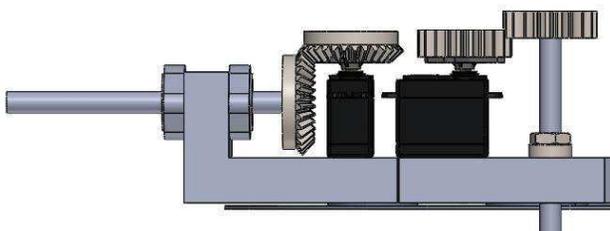
A continuación, se muestran algunas de las piezas realizadas en dicho software:



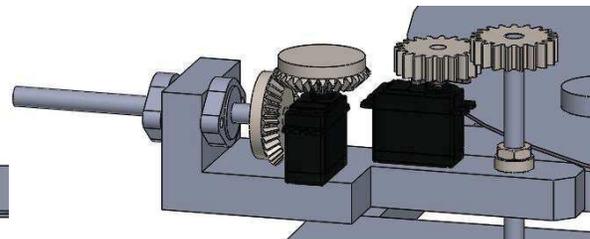
A) Simulación transmisión de movimiento



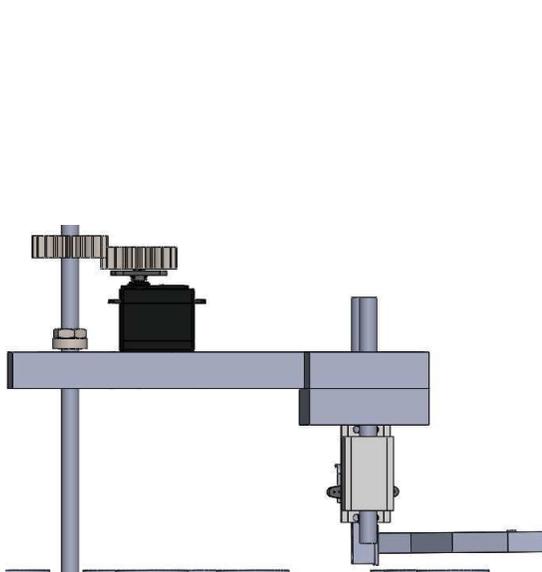
B) Diseño del chasis del prototipo



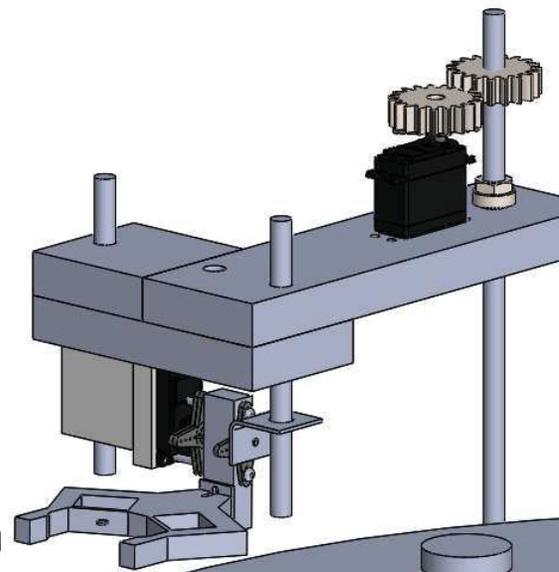
B) Simulación de ensamble brazo robot volteador

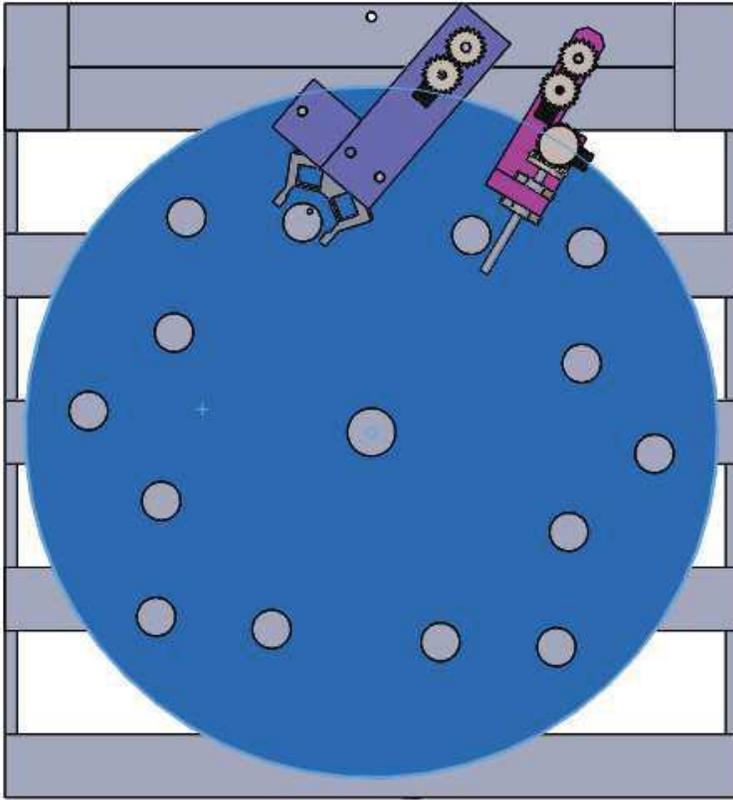


C) Simulación de Ensamble brazo robot volteador

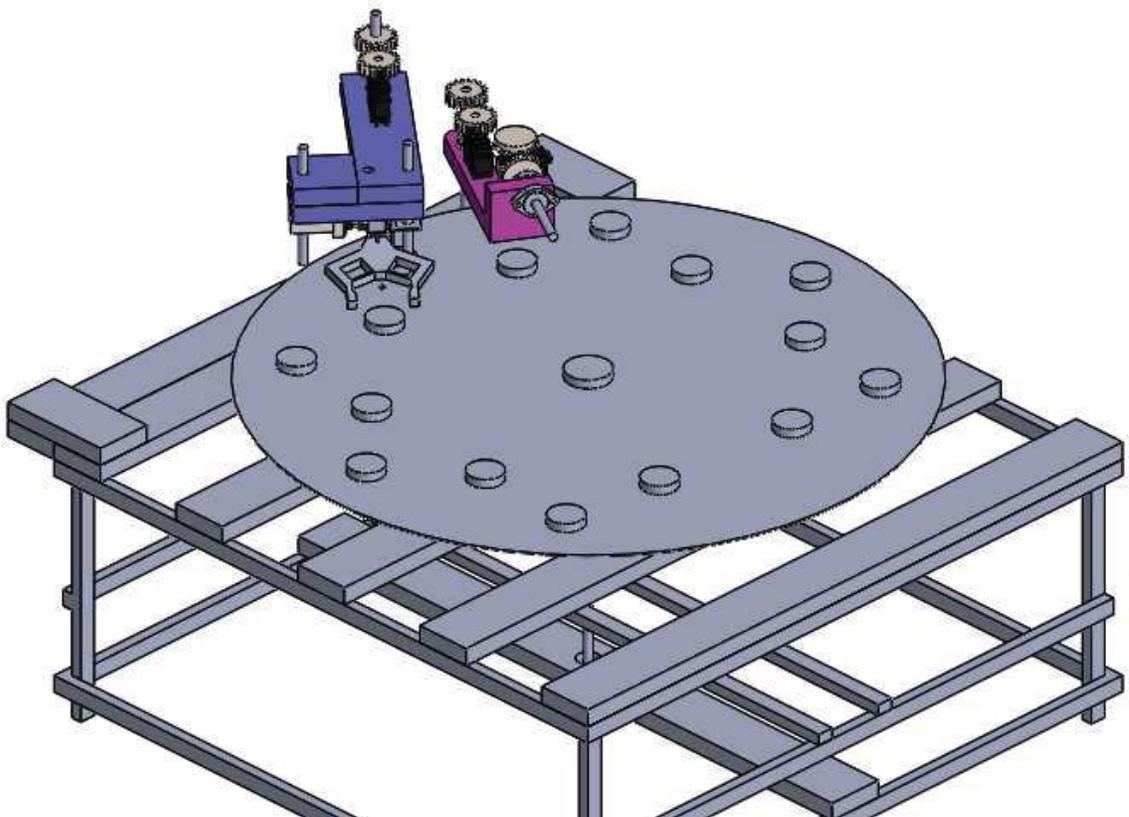


D). Simulación ensamble brazo robot inicial



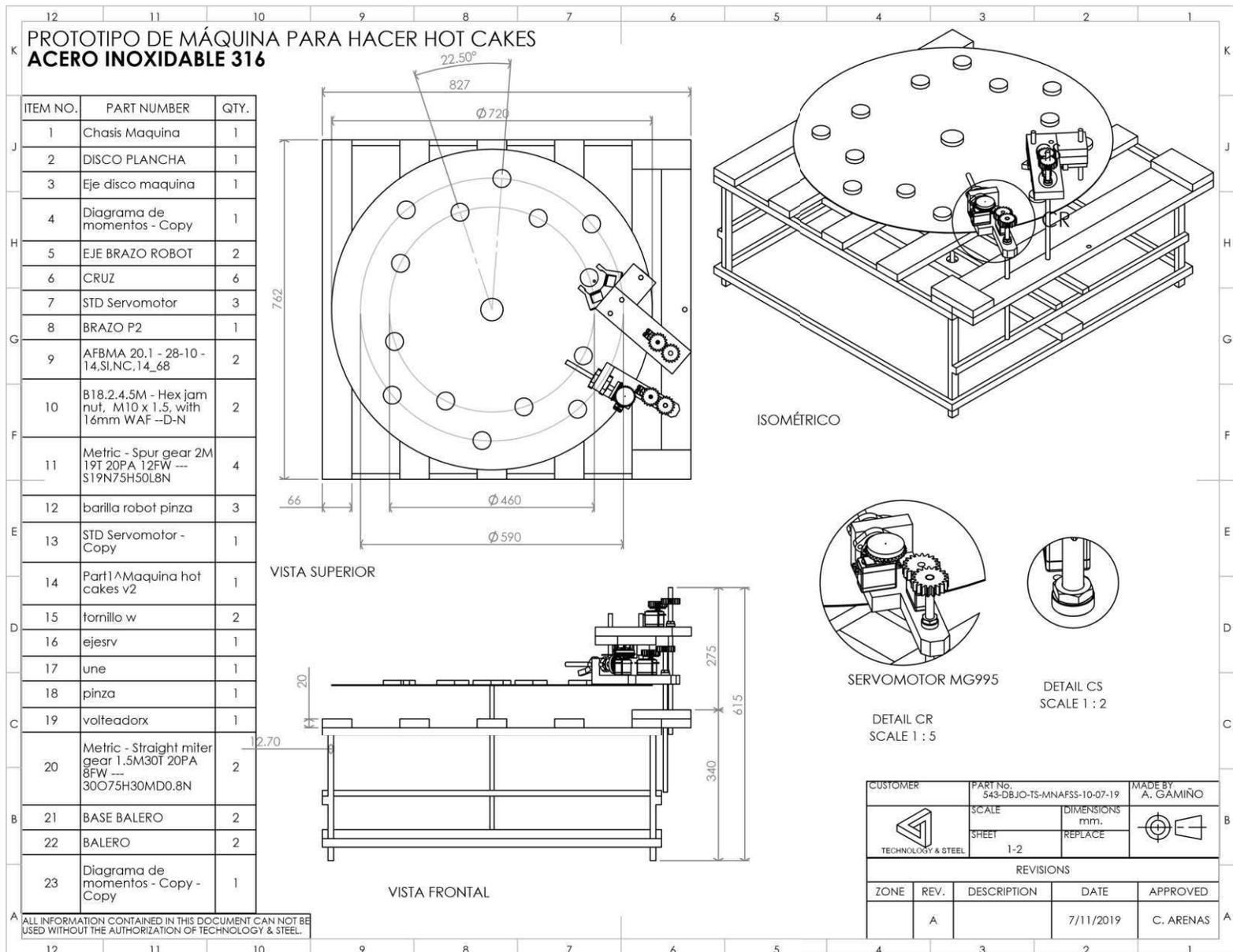


E) Simulación del sistema en operación (Paso calculado, posiciones de brazo robot).





3.2. Planos generales del sistema.





3.2 Requerimientos técnicos

La máquina se alimentará principalmente de una línea de 120 vac, posteriormente el voltaje es rectificado mediante una fuente de 500 watts con salidas de 24vdc, 12vdc, 5vdc y 3.3vdc.

El control de la máquina es basado en un plc Logo! OBA6 de siemens 12/24rce para el control de motores, servomotores y la entrada de los sensores de posición.

Se emplea el uso del Arduino como driver para el control de los servomotores ya que el plc empleado no cuenta con salidas pwm.

Los 24 volts de la fuente de alimentación son utilizados únicamente para el plc y Arduino, ya que cuenta con aislamiento galvánico y mantendrá siempre estable el voltaje en el control para evitar que el sistema se ponga en modo stand by.

Los 12 volts de la fuente son empleados para el control de los motores empleados en el sistema, ya que los motores especificados en la siguiente tabla funcionan a ese voltaje y a una corriente máxima de 8 amp.

Tabla 2: Dispositivos Utilizados

No.	Nombre	Modelo	Voltaje alimentación
1	Logo! Siemens	OBA6	12-24 vdc 100 mA
2	Arduino	Nano R3	5 vdc 50 mA
3	Pololu Motorreductor	N/D	12 Vdc 8 A
4	Servomotor	MG995	5-6 vdc 600 mA
5	Encoder óptico	n/a	5 vdc 50 mA
6	Quemador de gas	n/a	n/a
7	Cilindro de gas	n/a	n/a
8	Válvula de tipo aguja npt 1/4	n/a	n/a
9	Regulador de presión de gas @15PSI	n/a	n/a
10	Arresta llamas	n/a	n/a
11	Bommba peristaltica	Watson Marlow Q552	n/a

3.3 Metodología

Después de un estudio en el área de producción en pequeñas plantas de la industria alimenticia en la Zona Metropolitana de Guadalajara y sus alrededores, encontramos distintas alternativas que nos ayudarían a lograr que el proceso que se quiere implementar sea posible. Como todo sistema de producción, e producto tiene que ser llevado desde un punto inicial donde se recibe como materia prima, hasta el producto terminado listo para su empaque, y en este procedimiento se emplean bandas transportadoras en las que el proceso actúa en sus diferentes etapas.

3.4. Desarrollo

3.4.1. Transporte del producto.

Para nuestro caso y para la optimización de espacio, se optará por utilizar un disco transportador, donde la dimensión del perímetro del círculo se aprovecha para transportar el producto en sus distintas etapas.

En algunas secciones que posteriormente serán evaluadas en el diagrama de Gantt, se implementara el uso de actuadores y elementos finales de control para que el proceso sea ejecutado de la manera que se establece.

3.4.2. Dosificación de la mezcla.

En este caso en específico, se realizaron pruebas de dosificación de la mezcla entre una válvula tipo solenoide npt 1/2" y un pistón dosificador. Primeramente, con la válvula solenoide, debido a la alta viscosidad del producto a dosificar entre un tiempo determinado de apertura de la válvula según la masa/seg, la mezcla no logro pasar desde el lado A de la válvula hasta el lado B, ya que estas válvulas a pesar de tener una conexión de 1/2" NPT en la parte interna el producto es estrangulado por el mecanismo actuador del solenoide y como resultado, el producto nunca circula. La razón de su mecanismo es debido a que la fuerza que la bobina ejerce una vez energizada solo es suficiente para levantar una pequeña cantidad de masa.

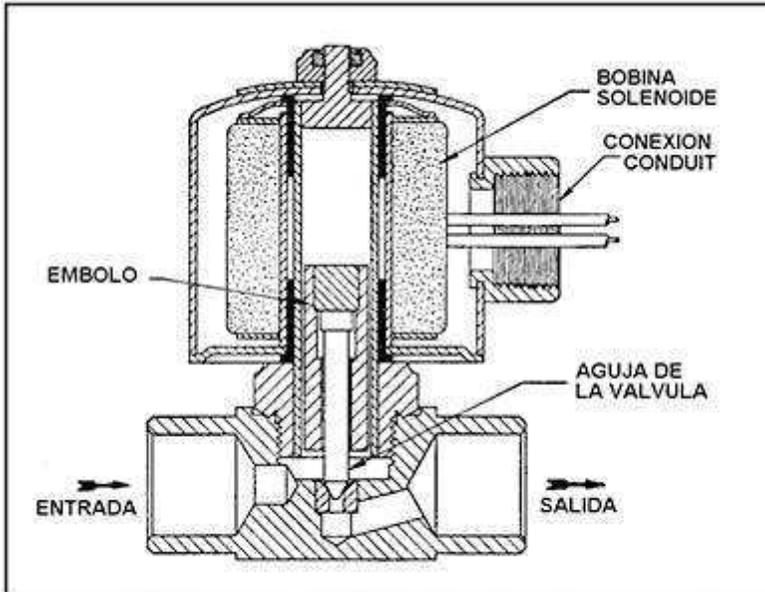


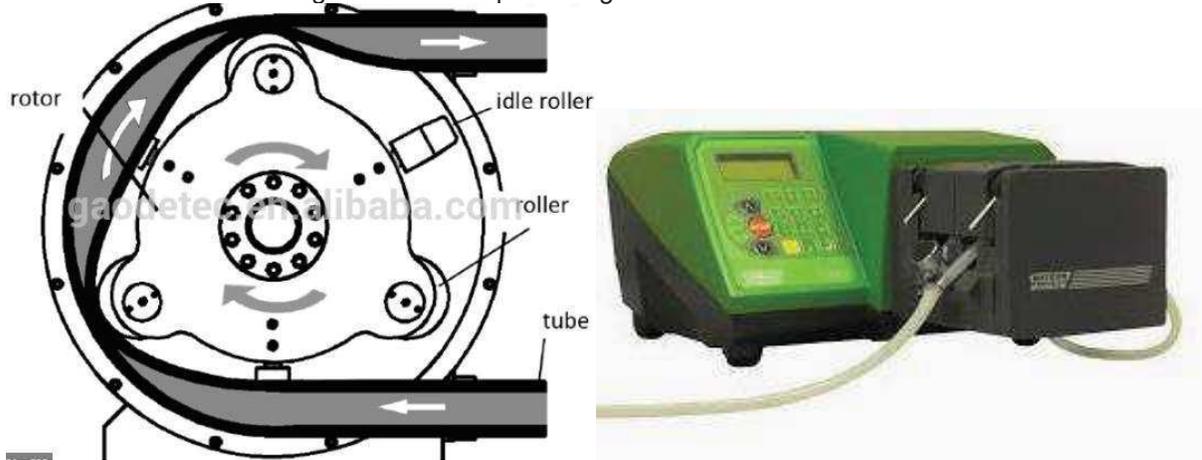
Figura 7.2-Válvula de solenoide típica de acción directa, normalmente cerrada de dos vías.

Para el segundo caso, utilizando un pistón dosificador elaborado por nosotros mediante un tubo de PVC y algunos otros componentes para el sello, se encontraron algunos problemas ya que por falta de maquinaria para realizar un perfecto rectificado cilíndrico de las piezas la mezcla fugaba entre el embolo y la camisa por la presión ejercida.



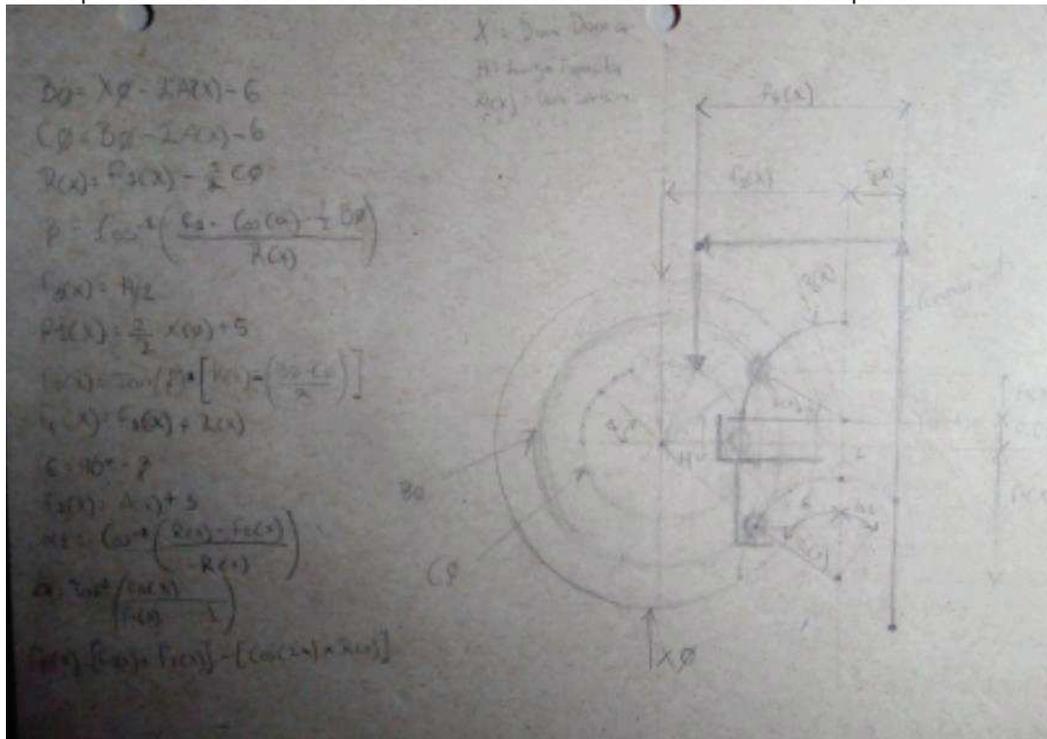


Por último y como resultado final, se empleó el uso de una bomba peristáltica, que acorde a nuestras referencias, una bomba peristáltica es un tipo de bomba hidráulica de desplazamiento positivo usada para bombear una variedad de fluidos. El fluido es contenido dentro de un tubo flexible empotrado dentro de una cubierta circular de la bomba (aunque se han hecho bombas peristálticas lineales). Un rotor con un número de 'rodillos', 'zapatas' o 'limpiadores' unidos a la circunferencia externa comprimen el tubo flexible. Mientras que el rotor da vuelta, la parte del tubo bajo compresión se cierra (o se ocluye) forzando, de esta manera, el fluido a ser bombeado para moverse a través del tubo. Adicionalmente, mientras el tubo se vuelve a abrir a su estado natural después del paso de la leva ('restitución'), el flujo del fluido es inducido a la bomba. Este proceso es llamado peristalsis y es usado en muchos sistemas biológicos como el aparato digestivo.

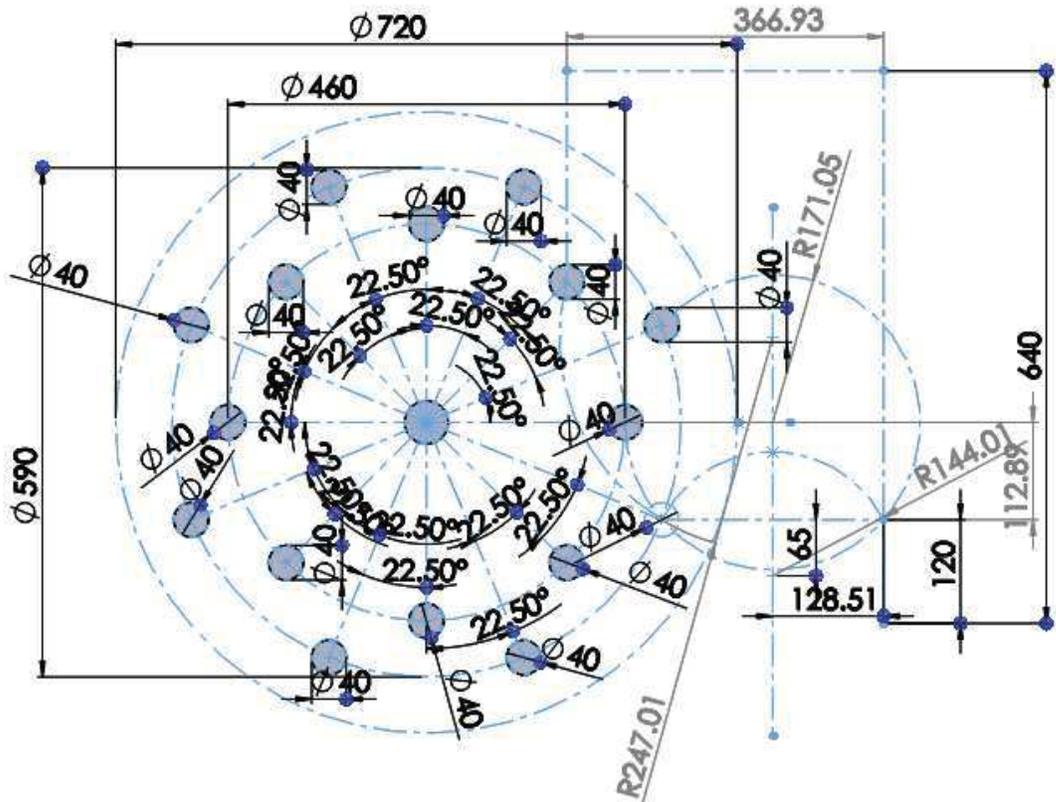


3.4.4. Determinación de movimientos.

Este punto hace referencia a el cálculo de proporciones geométricas para determinar los grados correspondientes a cada uno de los movimientos rotacionales de la máquina.



Cada una de las variables del sistema esta directamente relacionada con el diametro del la plancha rotatoria(Transpotadora). Estas son calculadas para obtener como resultado la posición correcta de cada uno de los componentes de la máquina.



3.4.5. Manufactura de mecanismos

Finalmente se manufacturan cada uno de los componentes del sistema acorde a los cálculos obtenidos y el diseño propuesto en el sistema.

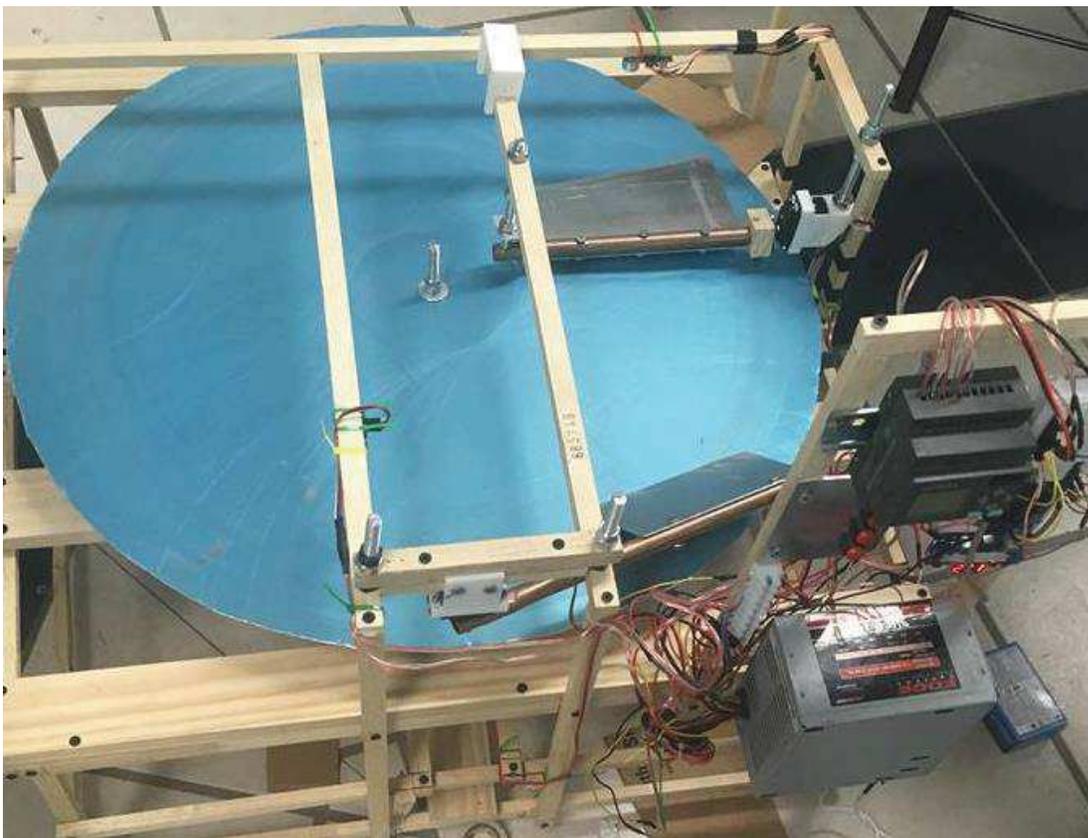
3.4.5.1. Brazo robot volteador.



3.4.5.2. Brazo robot principal.

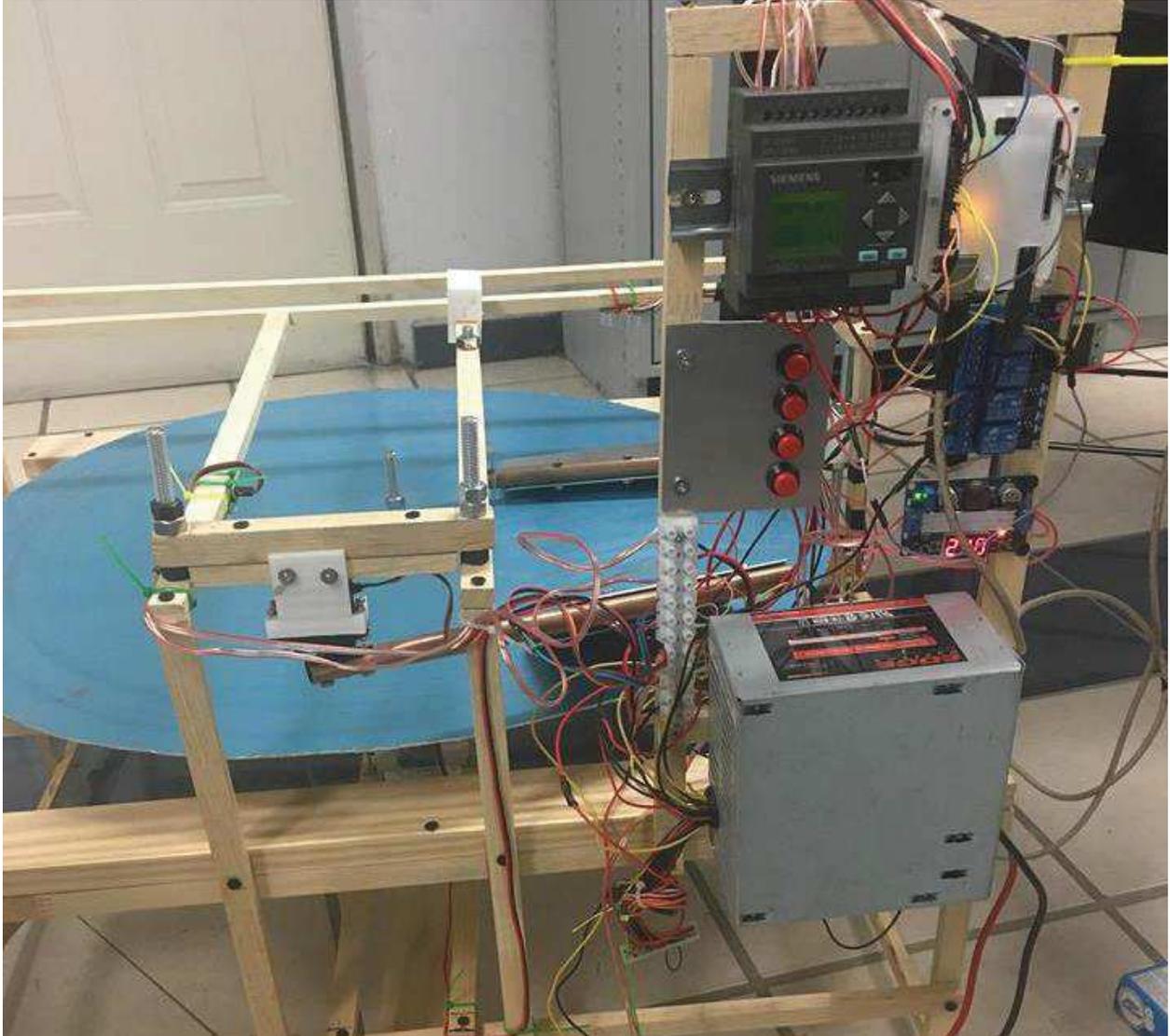


3.4.5.3. Estructura del sistema con plancha-disco rotatorio.





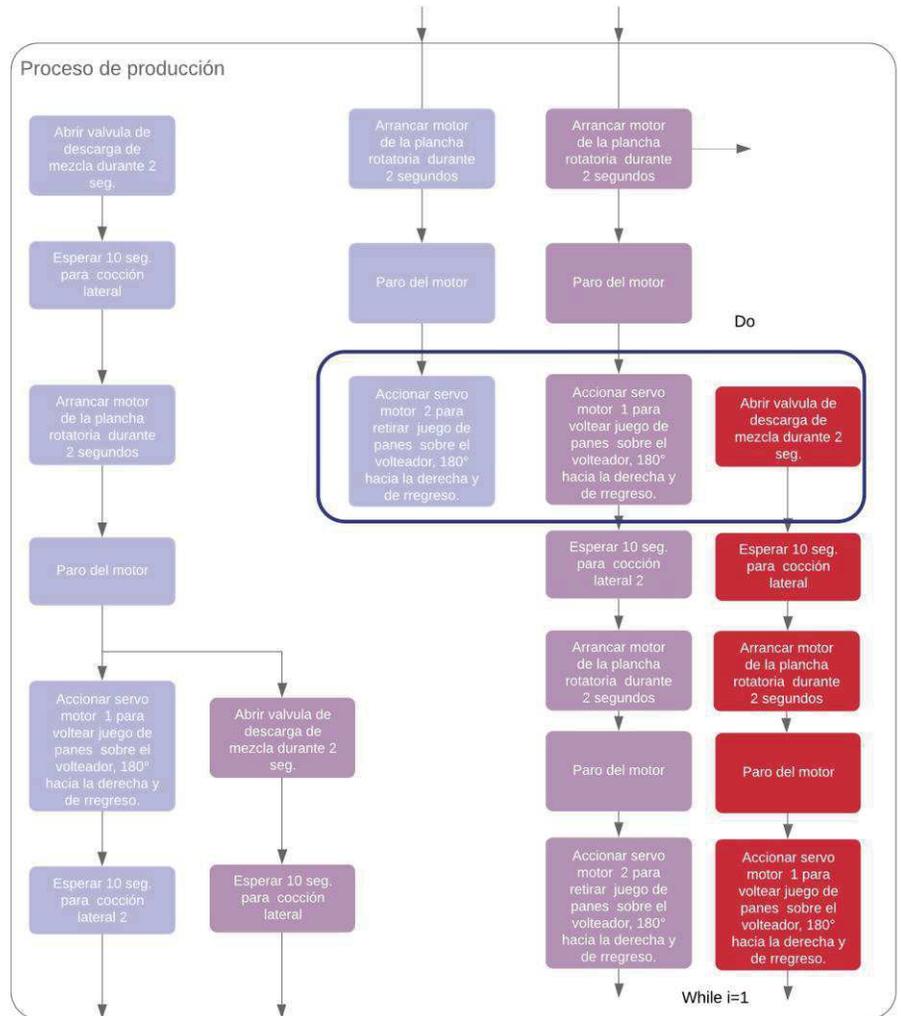
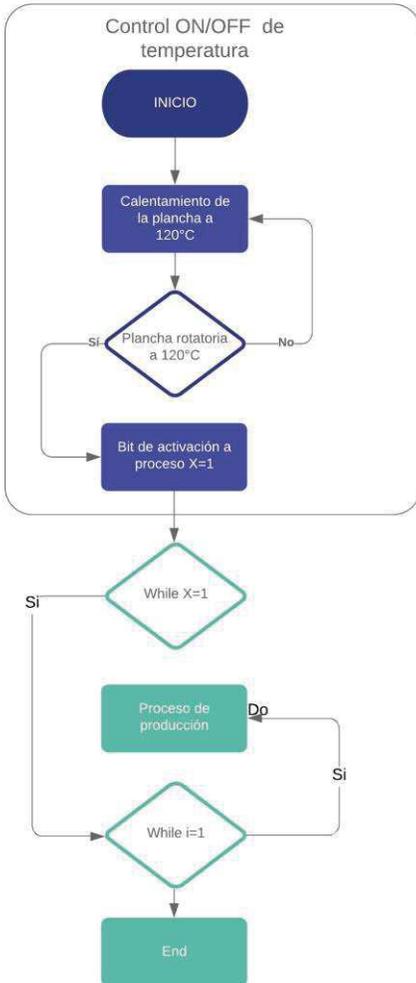
3.4.5.4. Integración de la parte electrónica.





3.5. Operación de sistema

FLUJO DE OPERACIÓN DEL SISTEMA
JESUS ANTONIO GAMINO CORONADO | September 11, 2019





3.6. Cronograma de actividades

Diagrama de Gantt

PROCESO	17/09/2019	20/09/2019	23/09/2019	26/09/2019	29/09/2019	02/10/2019	05/10/2019	08/10/2019	11/10/2019
Planteamiento del problema	█								
Ingeniería básica de la solución	█	█							
Elaboración de diseño (Ergonomía, productividad, producto final)		█	█						
Especificación del material a emplear				█					
Especificación de los actuadores a emplear				█	█				
Especificación del controlador a emplear					█				
Especificación de drivers a emplear					█	█			
Manufactura del sistema					█	█	█		
Cableado de los componentes						█	█		
Generación de diagramas de flujo para programación							█		
Programación del sistema							█		
Supervisión de la operación							█	█	
Corrección de errores								█	
Entrega									█

4. Resultados

En un principio, antes de imprimir todas las piezas en 3D se realizaron pruebas de funcionamiento mediante la herramienta de SolidWorks para revisar que todas las piezas embonaran bien, que tuvieran las dimensiones deseadas y en general que en conjunto trabajaran bien, sin un sobreesfuerzo o desgaste innecesario.

El funcionamiento de la maquina se logró haciendo pruebas conocidas como “a prueba y error” de cada una de sus partes mecánicas, de programación y electrónicas.

Encoder: se hicieron pruebas para tener el mejor lugar de ubicación de los sensores utilizados y ver la calibración necesaria que se necesitaba en cada uno de estos lugares para detección de material, presencia de un objeto etc, debido a que no todos manejan el mismo rango de detección o el modo de detección.

Mecánica: se probaron diferentes tipos de servomotores para la pinza mecánica y al final se optó por el servomotor de 15Kg debido a que los demás que se probaron no nos proporcionaban la fuerza suficiente; además se probaron mediante programación la cantidad de ángulos que se debía de girar en cada uno de los diferentes pasos o etapas del proceso del cocinado del pan.

Programación: Esta parte del proyecto fue donde más se realizaron más experimentos debido a que en un principio no se tenía estimado exactamente la duración completa del proceso, así que fue necesaria la reprogramación varias veces en cuanto a esperas y retardos entre cada uno de los pasos para que el siguiente paso no se accionara una vez terminado el que estaba en ejecución, todo esto para hacer el proceso en el tiempo menos posible, así que calculamos de la manera más aproximada la duración de cada una de las etapas y después lo indicábamos en el código.

Tomando en cuenta el tiempo promedio de espera en el que se cocina uno de los panes y comparándolo con el tiempo que nuestro dispositivo tarda en hacer el mismo proceso, se puede ver que la rapidez es alrededor de 15 y 20 veces mayor gracias a que se ejecuta la misma rutina cada 22.5° grados, que equivale a un nuevo paso de la máquina y a una nueva ejecución del bucle.



5. Conclusiones

Se cree que este dispositivo como primer prototipo desarrollado funciona de una manera óptima, mediante el análisis de diferentes factores se llegó a la conclusión de que si es factible realizar el proceso de fabricación de Hotcakes por medio de nuestro prototipo.

Pero sin duda alguna aún algunos aspectos por mejorar, en un futuro se pretende hacerle modificaciones al prototipo incluyéndole nuevos tipos de tecnologías, como lo es controlarlo de manera remota, así como la visualización en una interfaz del paso que se está llevando a cabo mediante una interfaz hombre máquina.

Visualizando la escalabilidad del sistema propuesto, se cree que el sistema cuenta con el mecanismo que puede solucionar otras áreas de la industria alimenticia, ya que el principio utilizado en este proceso es similar a otros.

El diseño aun no esta finalizado en su totalidad, no se ha aplicado la ingeniería de detalle en el sistema y debido a ello puede que existan áreas donde el mantenimiento y la manipulación de los componentes se vuelve inoperable, sin embargo, las pruebas de funcionamiento del sistema en general fueron un éxito.

Referencias

- Cenidet.edu.mx, 2019. [Online]. Available: <http://www.cenidet.edu.mx/subplan/biblio/seleccion/Tesis/CNAD%20Jaime%20Qui%F1ones%20Rico%202004.pdf>. [Accessed: 01-Sep-2019]
- "Búsqueda --SIGA", Siga.impi.gob.mx, 2019. [Online]. Available: <https://siga.impi.gob.mx/newSIGA/content/common/principal.jsf>. [Accessed: 05-Sep-2019].
- Cocinatipo.com, 2019. [Online]. Available: <http://www.cocinatipo.com/recetas/postres/origen-y-receta-de-hotcakes-diferentes>. [Accessed: 02-Sep-2019]
- NOM_059, 2017. [Online]. Available: http://dof.gob.mx/nota_detalle.php?codigo=5424575&fecha=05/02/2016. [Accessed: 02-Sep-2019]
- Peristalsis, 2014. [Online]. Available: https://www.bombasboyser.com/Productos/bombas-peristalticas.html?gclid=EAlalQobChMIrriBq9yT5QIVeR-tBh1diA9aEAAYASAAEgJ_FfD_BwE. [Accessed: 06-Sep-2019]



Separador Inteligente de Residuos

Cervantes Morales Emmanuel de Jesús, Nieves Vázquez José Ángel, Covix Gracia Evelin, Camacho Torres José Luis¹ y Martínez Domínguez Ángel Guadalupe¹

Instituto Tecnológico Superior de San Andrés Tuxtla.
josenieves@itssat.edu.mx

Resumen

En la actualidad existen una cantidad diversa de problemas debido a la forma de tratar con los residuos. Existen esfuerzos de diversos organismos gubernamentales e incluso internacionales que están trabajando en ideas de cómo resolver esta problemática. Como resultado de estos esfuerzos conjuntos, se han establecidos algunos sistemas que permiten clasificar los residuos y reutilizar los reduciendo de esta forma la contaminación provocada por su uso. En el presente trabajo se muestra la implementación de una Red Neuronal Convolutiva utilizada para la clasificación de residuos orgánicos e inorgánicos, permitiendo que empresas dedicadas a la transformación de estos desechos, destinen menor tiempo a la preparación de dichos residuos. De igual forma se presenta información que fue fundamental para el desarrollo de la Red Neuronal Convolutiva; junto con las herramientas de desarrollo informática.

Palabras clave: Red Neuronal Convolutiva, Visión Artificial, Residuos.

1. Introducción

En México se recolecta diariamente 86 mil 343 toneladas de residuos diariamente, en Veracruz se genera 4,451 toneladas de residuos, del cual solo se recolecta un 5.2%, quedando en 4to lugar a nivel nacional. La gran parte del estado de México no tiene la cultura de un correcto reciclado o no conocen cómo se clasifica, así que uno de los propósitos de proyectos de este tipo es crear la cultura de separar los residuos.

Estos residuos, al no ser tratados por sus propiedades, liberan gases que traen consigo enfermedades, estos residuos también tienen un contaminante llamado lixiviado, el cual, al entrar en contacto con la tierra, ésta la deja infértil, contaminando mantos acuíferos.

El uso de la Inteligencia Artificial para solucionar diversos tipos de problemas no es nada nuevo. Gran parte de sus fundamentos teóricos y tecnológicos fueron desarrollados en los últimos 70 años por científicos de la talla de Alan Turing, Marvin Minsky o John McCarthy. En la actualidad, este término hace referencia a diversas tecnologías que se pueden combinar de distintas formas. Gracias a esas formas de combinarse han resultado avances innovadores que hoy en día hacen frente a nuevos problemas, uno de ellos y de suma importancia en los últimos años es el cuidado del medio ambiente [1].

Las aplicaciones de la Visión Artificial enfocadas al medio ambiente están ligadas a las imágenes aéreas, tanto de superficies extensas obtenidas por satélites, como de áreas más acotadas capturadas por drones, y a las imágenes estáticas de captura de un entorno determinado, como cámaras fijadas en soportes. Otras aplicaciones y usos de la Visión Artificial se pueden encontrar de manera más detallada en [2].



En gestión de residuos, por una parte, puede emplearse la Visión Artificial para la clasificación de residuos en reciclaje, automatizando el proceso de clasificación de los desechos en función de los materiales en su composición, y separándolos con brazos robóticos que dividan los residuos conforme a su naturaleza. Esto sustituye a la metodología actual, con imanes y personas encargadas de clasificar, que hacen el proceso lento y poco efectivo, por lo que con esta tecnología se mejora notablemente la tasa de reciclado. Por otro lado, también pueden detectarse residuos en el medio ambiente, como polución, gases, espumas, basuras marítimas, o escombreras ilegales.

2. Redes Neuronales Convolucionales

Para comprender mejor las Redes Neuronales Convolucionales (RNC) primero veremos el modelo de las Redes Neuronales Artificiales (RNA). Una RNA es una red que contiene nodos, cada uno de ellos son modelados como una neurona artificial; esta red puede representarse con diversas configuraciones conocidas como arquitecturas. En la figura 1 se observa la arquitectura correspondiente a las RNA's, donde cada círculo representa una neurona. Estas neuronas se presentan en capas de la siguiente forma:

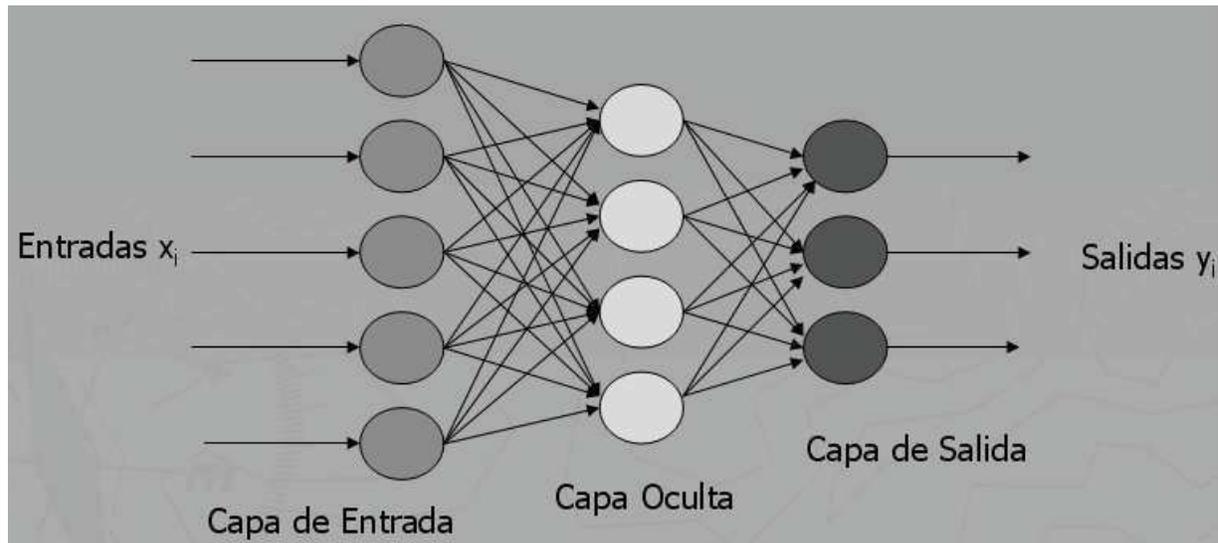


Figura 1. Arquitectura típica de una Red Neuronal Artificial

- Capa de entrada: Estas son las neuronas encargadas de recibir la información que será procesada por la red.
- Capa oculta: Está conectada con otras capas de neuronas en la entrada y la salida conteniendo los pesos intermedios de la red.
- Capa de salida: Es la capa que contiene los resultados finales de las operaciones que se hayan realizado en la red.

Normalmente los parámetros que definen la arquitectura de una RNA son los siguientes:

- Número de capas
- Número de neuronas
- Grado de conectividad



- Tipo de conexiones entre las neuronas

Para el caso del modelo de las RNA's el vector de entrada a la red está determinado por $X = [X_1, X_2, \dots, X_n]$. La información recibida por la neurona es modificada por w que es el vector encargado de proporcionar los pesos sinápticos a la neurona. Los pasos que se realizan en el proceso son los siguientes:

- La neurona recibe un vector que contiene un conjunto finito de información en forma de señales $X = [X_1, X_2, \dots, X_n]$,
- La intensidad sináptica correspondiente a la señal x_i se designa como el peso w_i que puede ser positivo (excitación o reforzamiento) o negativo (inhibición o debilitamiento),
- La neurona realiza una suma ponderada considerando todos los estímulos de entrada así como sus pesos correspondientes y responde mediante una función de activación que decide en base a un umbral θ si la neurona se dispara o no.
- Las funciones de activación introducen la no linealidad en el proceso computacional del modelo artificial y entre las más usuales se encuentran el limitador discontinuo de umbral o función escalón de Heaviside y la función sigmoide.

La Figura 2 muestra el modelo básico de una neurona artificial.

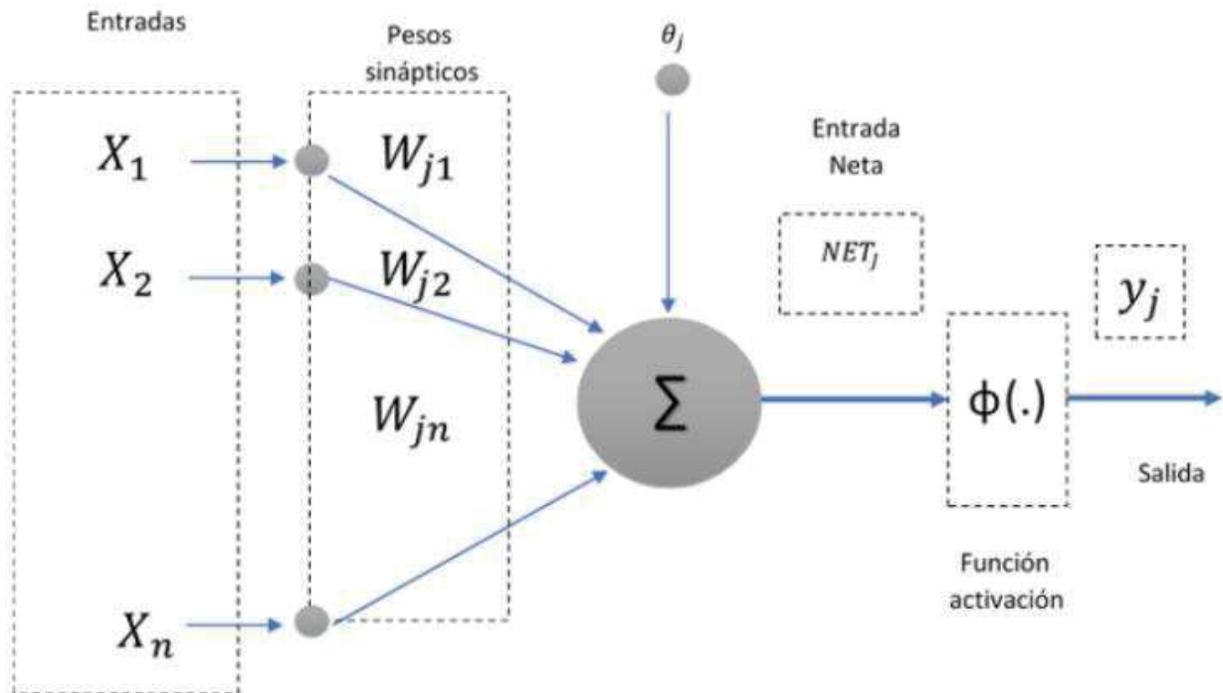


Figura 2. Neurona Artificial.

El parámetro θ_j se conoce como bias o umbral de una neurona. Los diferentes valores que recibe la neurona, modificados por los pesos sinápticos, los sumamos para producir lo que hemos denominado entrada neta. Esta entrada neta es la que va a determinar si la neurona se activa o no.

La activación o no de la neurona depende de lo que llamaremos Función de Activación. La entrada neta la evaluamos en esta función y obtenemos la salida de la red. Si, por ejemplo, esta función la definimos como un escalón unitario, la salida será 1 si la entrada es mayor que cero, en



caso contrario, la salida será cero. La salida y_j de la neurona se genera al evaluar la neta en la función de activación.

Dentro de las Redes Neuronales Artificiales (RNA) existen una gran cantidad de modelos y aplicaciones. Uno de los modelos que están siendo utilizados ampliamente en la actualidad son las Redes Neuronales Convolucionales (RNC), este tipo de redes se basan en el principio de funcionamiento del perceptrón multicapa. La figura 3 muestra el esquema de un perceptrón con una sola neurona. El nodo principal representa la neurona que integra las señales de entrada en forma ponderada de acuerdo a sus pesos respectivos de modo que:

$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1}, \quad (1)$$

Donde $d(x)$ es la función de decisión y w_i, w_{n+1} son los pesos correspondientes a cada entrada. Es importante comentar que el umbral denotado por θ se ha identificado con el peso extra w_{n+1} con señal de entrada virtual unitaria, $w_{n+1}=1$, con la finalidad de uniformar la expresión general.

Este perceptrón responde con uno de dos valores de salida que son -1 o $+1$ dependiendo de la función de activación y del umbral de disparo de la neurona. La salida toma el valor $+1$ si la suma de los pesos cumple que $d(x) > 0$ indicando que el patrón x se clasifica como perteneciente a la clase $C1$; de lo contrario, si se cumple que $d(x) < 0$, el patrón es clasificado como perteneciente a la clase $C2$. Si $d(x) = 0$ existe una indeterminación debido a que x se encuentra justo sobre la frontera de decisión que separa las dos clases; el “empate” se elimina asignando ese patrón a cualquiera de las clases [3].

La expresión de la ecuación (1) igualada con cero nos proporciona la frontera de decisión, es decir

$$d(x) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_{n+1} = 0 \quad (2)$$

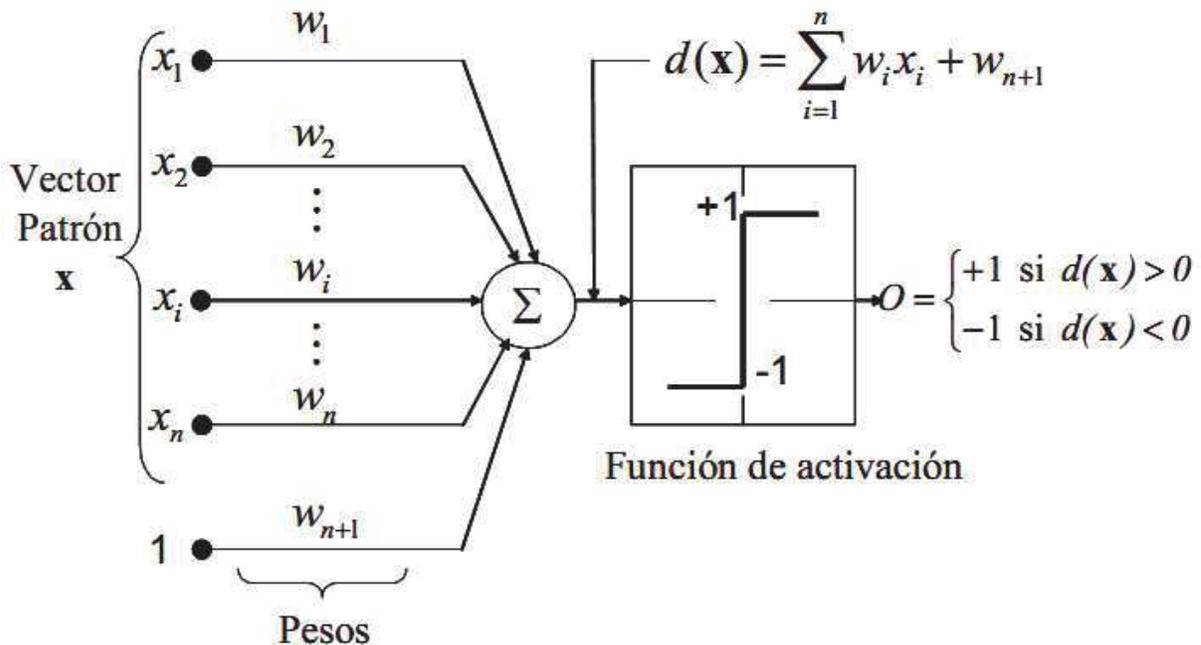


Figura 3. Perceptrón con una sola neurona



Después de haber dado una pequeña introducción de lo que son las RNA, pasamos ahora a tratar las redes neuronales convolucionales. Las Redes Neuronales Convolucionales (RNC) también conocidas como **ConvNet** son redes neuronales que forma parte de las redes neuronales profundas (deep neural network) y son especialmente utilizadas en el área de la visión artificial ya que sus neuronas funcionan de manera similar a las neuronas de la corteza visual de los cerebros biológicos [4,5,6,7]. Las neuronas tienen pesos, los cuales reciben una entrada con la que realiza un producto escalar y sobre la que se aplica una función de activación, tienen una función de pérdida. Sin embargo, este tipo de redes han encontrado aplicaciones en uno de los principales problemas que enfrentan las redes neuronales ordinarias: el tratamiento de imágenes. Pese a que con las redes neuronales convencionales es posible manejar imágenes, una vez las imágenes aumentan su tamaño y calidad las redes neuronales convencionales empiezan a tener problemas. Al tratarse de neuronas totalmente conectadas, provocaríamos un desperdicio de recursos así como un rápido sobre ajuste [8].

Las redes neuronales convolucionales trabajan dividiendo y modelando la información en partes más pequeñas, y combinando esta información en las capas más profundas de la red. Por ejemplo, en el caso del tratamiento de una imagen, las primeras capas tratarían de detectar los bordes de las figuras. Las siguientes capas buscarían combinar los patrones de detección de bordes para conseguir formas más simples, y aplicar patrones de posición de objetos. Por último, en las últimas capas la red intenta hacer coincidir la imagen con todos los patrones descubiertos, para conseguir una predicción final de la suma de todos ellos. Así es como las redes neuronales convolucionales consiguen modelar una gran cantidad de datos, dividiendo previamente el problema en partes para conseguir predicciones más sencillas y precisas [8].

En general, todas las redes neuronales convoluciones están formadas por una estructura compuesta por 3 tipos de capas.

2.1 Capa convolucional

Esta es la capa principal y es la que le da el nombre de la red. En vez de utilizar la multiplicación de matrices como en las redes neuronales estándar, se aplica una operación llamada convolución. Esta operación recibe como entrada una imagen, y sobre ella aplica un filtro que nos devuelve su mapa de características, reduciendo así el tamaño de sus parámetros. La convolución utiliza tres ideas que ayudan a mejorar cualquier sistema sobre el que se aplique machine learning: el primero es la Interacciones dispersas: Al aplicar un filtro de menor tamaño sobre la entrada, reducimos bastante el número de parámetros y cálculos. El segundo tiene que ver con los Parámetros compartidos: Compartir parámetros entre los distintos tipos de filtros ayuda a mejorar la eficiencia del sistema. Finalmente, Representaciones equivariantes: Si las entradas cambian, las salidas cambian de forma similar. Además, la convolución proporciona herramientas para trabajar con entradas de tamaño variable, lo cual es muy útil cuando se trabaja con imágenes (cada imagen tiene un número distinto de píxeles). Se basa en un operador matemático que transforma dos funciones f y g , en una tercera, que en cierto sentido, representa la magnitud en la que se superponen f y una versión trasladada y rotada de g [8].

2.2 Capa de reducción o de pooling

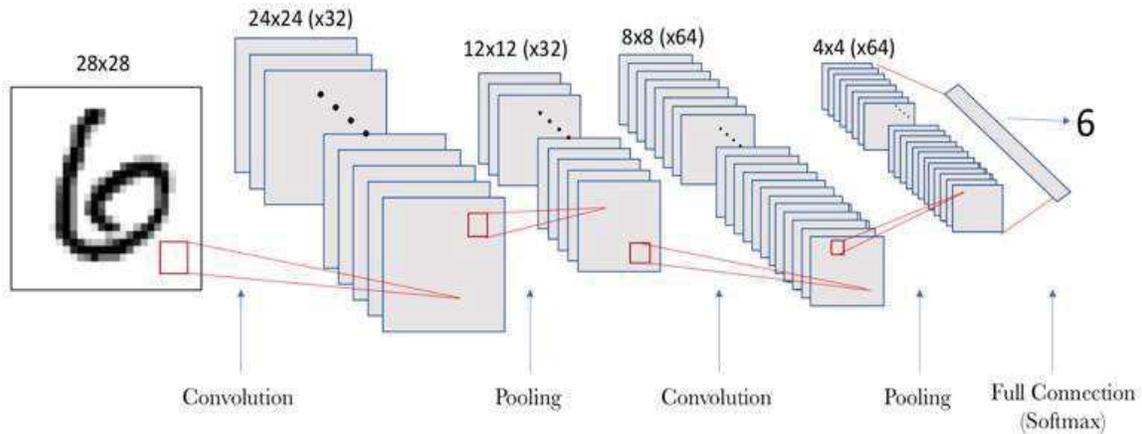


Figura 1. Estructura de una RNC

Esta capa se coloca generalmente después de la capa convolucional. Es la encargada de reducir la cantidad de parámetros a analizar reduciendo las dimensiones espaciales (ancho x alto), quedándose de esta forma con las características más comunes. La operación que lleva a cabo esta capa también se llama reducción de muestreo o submuestreo, ya que la reducción de tamaño implica también una pérdida de información. Sin embargo, para una red neuronal, este tipo de pérdida puede ser beneficioso debido a lo siguiente:

- La reducción del tamaño provoca una menor sobrecarga de cálculo en las próximas capas de la red.
- Reduce habitualmente el sobre ajuste

La operación que se suele aplicar en esta capa es "max-pooling", que divide la imagen de entrada en un conjunto de rectángulos, y respecto a cada uno de ellos, se queda con el valor máximo [8].

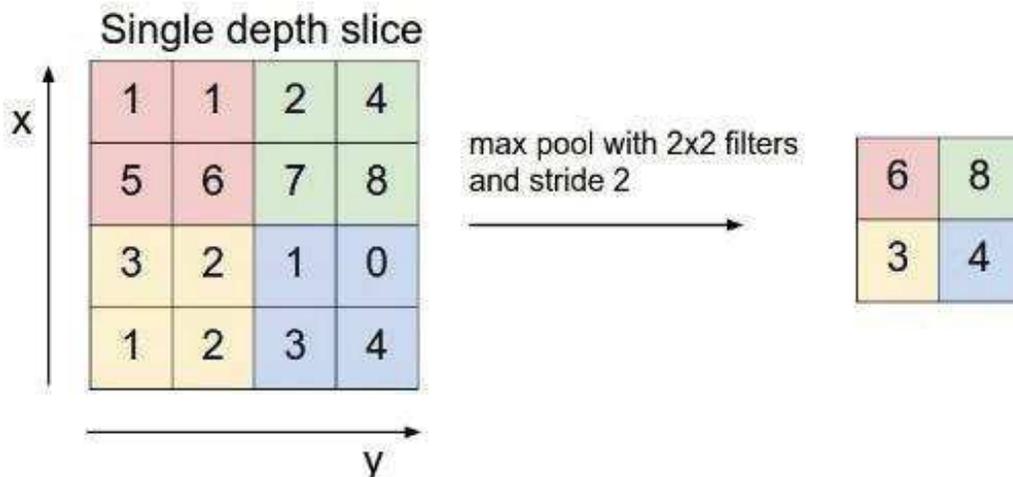


Figura 2. Aplicación del Maxpooling



2.2 Capa Clasificadora.

Una vez que la imagen ha pasado tanto por las capas convolucionales como las de pooling y se han extraído sus características más destacadas, los datos llegan a la fase de clasificación. Para ello, las redes convolucionales normalmente utilizan capas completamente conectadas en las que cada píxel se trata como una neurona independiente. Las neuronas de esta fase funcionan como las de un Perceptrón multicapa, donde la salida de cada neurona se calcula multiplicando la salida de la capa anterior por el peso de la conexión, y aplicando a este dato una función de activación [8].

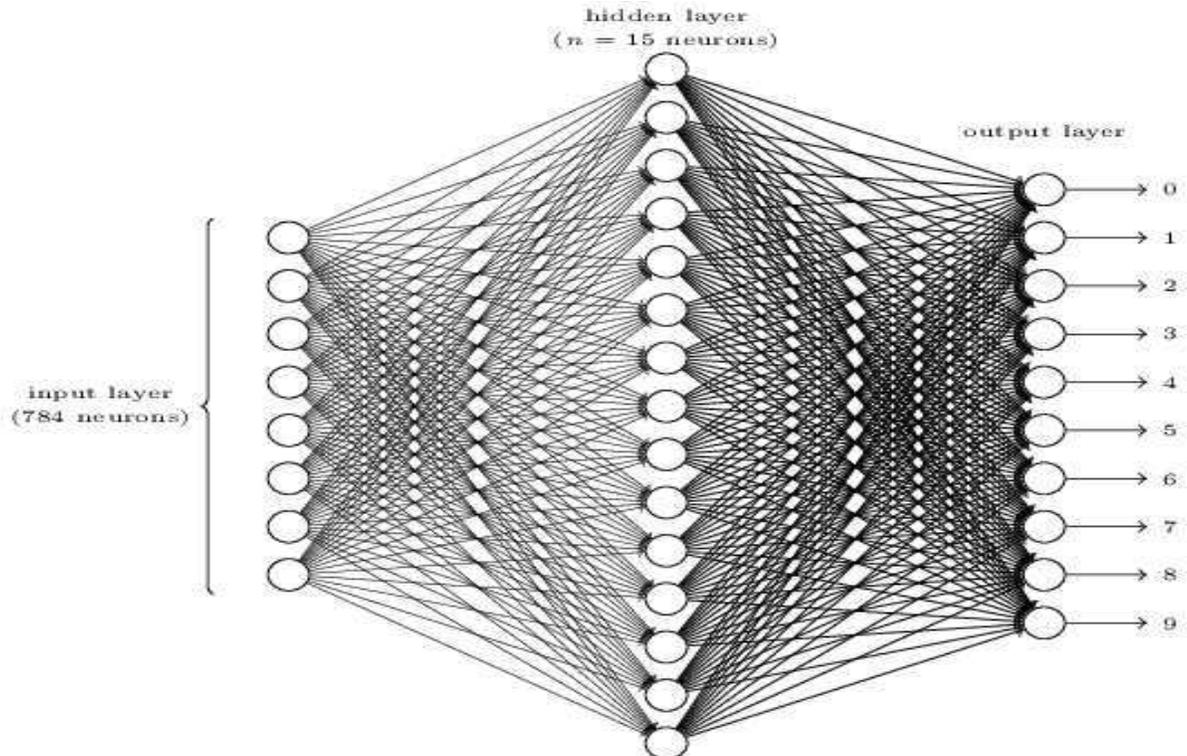


Figura 3. Representación de Capa de Conexión

3. Programación

Para poder desarrollar el programa, el cual contiene a la red neuronal convolucional, se emplean las librerías de TensorFlow y Keras, las cuales permiten programar modelos de redes neuronales.



Figura 4. Librerías de TensorFlow

Keras es una interfaz de programación de aplicaciones (API, application programming interface) para Redes Neuronales de código abierto la cual puede ser usada en Python y a su vez es capaz de ejecutarse sobre TensorFlow. Con estas herramientas se ha desarrollado la Red Convolutiva, la cual ha sido puesta a prueba para garantizar no solo el funcionamiento del clasificador sino también su eficiencia. Por otro lado, TensorFlow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas [9].

3.1 Programa.

Se realizó en Python, un programa de redes neuronales convolucionales la cual necesita la librería TensorFlow y la API Keras para facilitar la programación de la red. A continuación se enumeran los pasos a seguir:

1. Instalar Python (se recomienda la versión 3.5.x a 3.6.x debido a cuestiones de compatibilidad con TensorFlow)
2. Instalar Numpy (versión 1.14.3) y matplotlib (versión 2.2.2)
3. Instalar TensorFlow (versión 1.10) y Keras (versión 2.2.2)

Una vez instalados los aditamentos necesarios para el desarrollo de la red iniciamos con la agrupación o recabación de información.

En primera estancia se debe crear una carpeta la cual contendrá a otras. A continuación se explica de manera detallada y concisa lo que cada una debe llevar.

- Una Carpeta de entrenamiento para nuestra red convolutiva, la cual contendrá dos carpetas más, estas tendrán las clases a manejar (Orgánica e inorgánica), dichas carpetas contendrán imágenes alusivas a su clase.

Una carpeta de validación la cual al igual que la de entrenamiento contendrá dos carpetas (Orgánica e inorgánica) solo que preferentemente no deben contener las mismas imágenes.

Una vez hecho los pasos del 1 al 4 se realiza el programa de entrenamiento y el de predicción, dicha red deberá estar entrenada y su modelo se guardará en una carpeta a la que llamaremos modelo, evitando así que esta acción sea realizada cada vez que se ejecute el programa.



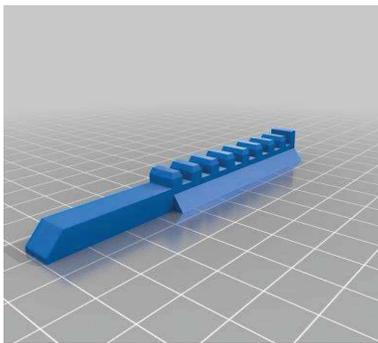
• Para iniciar las pruebas de la Red Neuronal Convolutiva se crea un programa en python el cual comenzará a realizar predicciones de la clase a la que pertenecen las imágenes, comprobando la certeza del programa.

Todo lo mencionado anteriormente permite la construcción de la Red Convolutiva, es hasta este punto donde se ha concluido la primera parte de lo constituirá el proyecto.

3.2 Prototipo

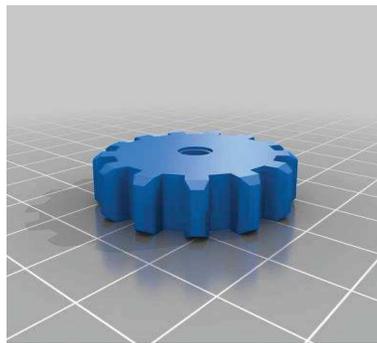
Para la realización del prototipo se deben realizar las siguientes actividades:

- **Diseño de los botes:** En el programa de SolidWorks se hizo el diseño de los botes considerando el siguiente aspecto: El espacio (público y/o privado) donde está destinado, ya que a partir de este factor se determinan las medidas de los botes, y a su vez se contempla la cantidad de residuos que se genera en dicho espacio.
- **Sistema de apertura:** Se emplea un sistema lineal basado en la transmisión de movimiento por medio de engranes, los cuales son puestos en marcha a través del movimiento rotatorio generado por un servomotor. Para controlar dicho sistema, se emplea un microcontrolador, el cual enviará señales que se encargan de manipular la posición de la flecha, permitiendo la apertura y cierre de los botes.

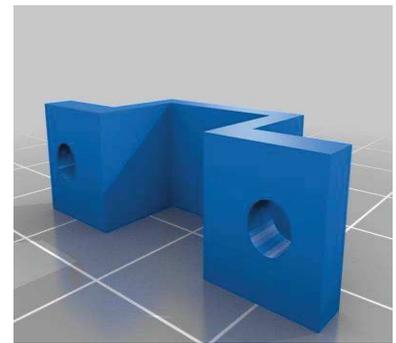


(a)

de

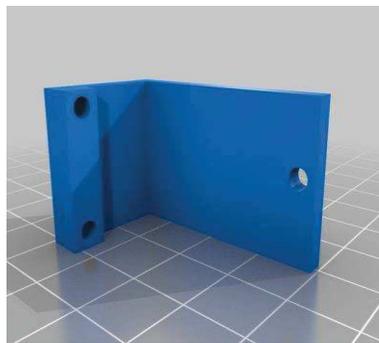


(b)



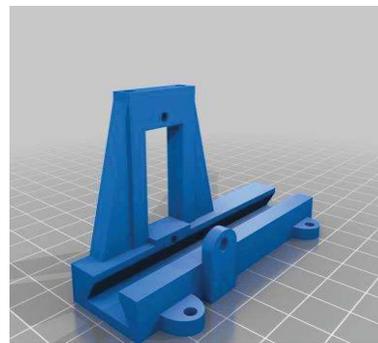
(c)

de la



(d)

del



(e)

y

Figura 5. Modelos 3D del sistema de apertura



3.3 Indicador luminoso.

En los botes se ha integrado indicadores luminosos (tira de led RGB) cuya finalidad es dar un previo aviso de la apertura de alguno de ellos, facilitando así, que el usuario identifique el contenedor donde deberá depositar sus residuos.

Para la activación de los indicadores es necesario emplear un módulo con dos relevadores, dichos dispositivos permitirán la conmutación de uno de sus contactos NA (normalmente abiertos).



Figura 6. Instalación de los Leds al prototipo

4. Resultados

Los resultados que se presentan son obtenidos con una base de datos de 193 imágenes por las 2 clases para entrenamiento y 53 de validación obteniendo los siguientes números que se muestran en la figura 7.

Al realizar pruebas con 10 residuos orgánicos solo seis clasifíco como tal y 4 los clasifíco como inorgánicos, y al realizar pruebas con 10 residuos inorgánicos 5 clasifíco correctamente y los otros 5 los marco como orgánico. El porcentaje de error es algo elevado, pero dichos resultados son derivados de la toma de la imagen, la cual no tenían una calidad adecuada. Así mismo, es importante mencionar que conforme se mejore la calidad de la imagen y la red se entrene más veces, el porcentaje de error disminuirá permitiendo que la clasificación de basura sea casi 100% eficiente.



```
C:\ProgramData\Anaconda3\python.exe
C:\ProgramData\Anaconda3\lib\site-packages\h5py\_init_.py:36: FutureWarning: Conversion of the second argument of issu
dtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type
.
from .conv import register_converters as _register_converters
Found 193 images belonging to 2 classes.
Found 53 images belonging to 2 classes.
Epoch 1/4
2019-09-09 18:58:12.934287: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:141] Your CPU supp
orts instructions that this TensorFlow binary was not compiled to use: AVX2
250/250 [=====] - 321s 1s/step - loss: 0.3408 - acc: 0.8561 - val_loss: 0.4734 - val_acc: 0.811
3
Epoch 2/4
250/250 [=====] - 222s 889ms/step - loss: 0.1322 - acc: 0.9519 - val_loss: 1.2473 - val_acc: 0.
8113
Epoch 3/4
250/250 [=====] - 532s 2s/step - loss: 0.0488 - acc: 0.9795 - val_loss: 1.1080 - val_acc: 0.830
2
Epoch 4/4
154/250 [=====>.....] - ETA: 2:26 - loss: 0.0350 - acc: 0.9838_
```

Figura 7. Resultados obtenidos de la RNC.

5. Conclusiones

Aunque cada vez es mayor la conciencia ciudadana sobre el reciclaje, todavía existe una gran porción de residuos mezclados que siguen sin ser reciclados. Es por ello que varias empresas de innovación se han dado a la tarea de crear sistemas inteligentes que sean capaces de realizar dicha actividad con mayor eficiencia y rapidez.

México se encuentra dentro de aquellos países que reciclan un pequeño porcentaje de sus residuos sólidos, es por ellos que se ha dado a la tarea de incorporar un sistema inteligente (separador de residuos) que contribuya no solo a que más personas se sumen a la causa, sino que comprendan de forma más sencilla las dos clases de residuos que en este sistema se manejan. Gracias a la red convolucional y a todos aquellos elementos que la conforman, la red es capaz de aprender e identificar los residuos, clasificarlos y decidir cuál es el contenedor donde se deberán depositar; para poder reciclar residuos nuevos y que la red aprenda a clasificarlos, es necesario agregar nuevos datos de referencia y someter a un nuevo entrenamiento a la red.

Como trabajo futuro aún queda mucho por hacer, algunas de las tareas pendientes es aumentar la robustez del sistema empleando mejores sistemas de adquisición de imágenes, así como trabajar con los diferentes tipos de iluminación, por otro lado se pretende realizar modificaciones al diseño de los botes para obtener mejores resultados en la parte de apertura y cierre. Por otro lado, se pretende usar las aplicaciones móviles para incorporar la información que proporcionen los botes al internet de las cosas y controlar el sistema.

Referencias

- [1] Purdy, M. and P. Daugherty. “*Inteligencia Artificial, El Futuro Del Crecimiento*”, Accenture Institute for High Performance, [e-Book] 2016.



- [2] Minsait. Una imagen vale más que mil palabras, Visión Artificial. Recuperado de <https://www.minsait.com/es/actualidad/insights/una-imagen-vale-mas-que-mil-palabra>., 2018
- [3] Gonzales, R., Woods, R. “*Digital Image Processing*”, Prentice Hall, New Jersey, 2nd. Ed, 2002.
- [4] Fukushima, K. "Neocognitron". *Scholarpedia*.**2**(1): 1717.
- [5] Hubel, D. H.; Wiesel, T. N."Receptive fields and functional architecture of monkey striate cortex". *The Journal of Physiology*.**195**(1): 215–243. 1968.
- [6] Fukushima, Kunihiko."Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" *Biological Cybernetics*, **36**(4): 193–202, 1980.
- [7] Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda "Subject independent facial expression recognition with robust face detection using a convolutional neural network", *Neural Networks*, **16**(5): 555–559, 2003.
- [8] Antona Cortés, C. “*Herramientas modernas en redes neuronales: La librería keras*” . Universidad Autónoma de Madrid, Madrid, 2017.
- [9] López, R. E. “*Redes neuronales convolucionales con TensorFlow. [Matemáticas, análisis de datos y python]*”. Recuperado de <https://relopezbriega.github.io/blog/2016/08/02/redes-neuronalesconvolucionales-con-tensorflow/>, (2 de Agosto del 2016).



Mecanismo para rehabilitación (MEC-RE)

García González Gladis, Chi Rodríguez Yesenia del Carmen*
y Hernández Gutiérrez Ramón

Instituto Tecnológico Superior de Las Choapas. Departamento de ingeniería electrónica. Carretera Las Choapas-Cerro de Nanchital Km. 6 Colonia J. Mario Rosado Las Choapas, Veracruz, C.P. 96980.

Tel: 01 (923) - 32-32010

chi_yese24@hotmail.com

Resumen

El proceso de rehabilitación física en extremidades inferiores en pediatría se lleva a cabo por medio de apoyo fisioterapéutico que aplican técnicas específicas que ayuden a los niños a desarrollar al máximo su motricidad funcional, utilizando su potencial cerebro-motriz [1]. En el municipio de las Choapas Veracruz se lleva a cabo la atención antes mencionada en el DIF (Desarrollo Integral Familiar) y para estimulación temprana en Centro de apoyo Neverland, en dichas instituciones carecen de aparatos para el proceso de fortalecimiento en extremidades inferiores infantiles. Por lo tanto MEC-RE pretende ser apoyo a la fisioterapia de niños con Síndrome de West que presentan un retardo mental que ocurre en el 90 % de los casos y con frecuencia se asocia con déficit motor parálisis cerebral infantil, trastornos de conducta y rasgos autísticos [2]. Además mediante un microcontrolador se tiene a la disposición 3 niveles precargados de ejercicios terapéuticos para ser utilizados bajo supervisión y diagnóstico de especialistas, de acuerdo a opiniones de los mismos el mecanismo presenta características que mejoran el plan de tratamiento del paciente para una mejor calidad de vida.

Palabras claves: Síndrome de West, Fisioterapia, Trastorno motriz, Neuroplasticidad, Mecanismo.

1.-Introduccion

En la actualidad las personas con discapacidad representan el 15% de la población mundial [3], según el Instituto Nacional de Estadística y Geografía (INEGI), en México el 5.1%; porcentaje que representa más de 5 millones de persona. La discapacidad motriz (problemas para caminar, manipular objetos y de coordinación para realizar actividades) es la más frecuente en el país, superando a la sensorial, de la comunicación y la mental.



La falta de atención por parte de la sociedad a este problema conlleva a desigualdades sociales o impedimentos para el sano desarrollo en personas con dichos padecimientos. La niñez es la etapa en la cual un pequeño avance puede convertirse en un progreso en la calidad de vida de una persona con capacidad diferente. Sin embargo, existen diversos factores ambientales que impiden llegar al punto máximo del desarrollo de un niño; la estimulación, afectividad, factores culturales y socioeconómicos al no ser favorables resultan un impedimento para el desarrollo óptimo del ser humano.

En el municipio de las Choapas, Veracruz existe 1 centro de rehabilitación público que ofrece diferentes tipos de terapias (física, neurodesarrollo, rehabilitación, lenguaje entre otros) para las personas que presentan capacidades diferentes.

La terapia de neurodesarrollo es una de las más solicitadas ya que el desarrollo infantil es un proceso dinámico que involucra a la vez la evolución biológica, psicológica y social, durante los primeros años de vida; etapa donde se forman las habilidades perceptivas, motrices y cognitivas que posibilitaran la interacción en su medio en un futuro. Dicha terapia se imparte por un especialista que con ayuda del padre o acompañante que asiste con el niño, se realiza una rutina de ejercicios y maniobras terapéuticas que a base de repeticiones activan los patrones sensoriomotores del neurodesarrollo, los cuales son específicos del ser humano similares a las actividades adultas [1]. Estas terapias requieren de constancia, participación de la familia y de otras personas, ya que el infante es incapaz de lograr un progreso por sí mismo y en ocasiones las limitaciones físicas le impiden poder sostener su cabeza y tronco por lo que para poder estar en posición vertical requieren de la intervención de varias personas.

El centro de rehabilitación ofrece de dos a tres terapias por semana, lo que incluye terapia física por medio de métodos y técnicas fisioterapéuticas y las indicaciones al acompañante para que aprendan las rutinas de tratamiento las cuales se ejecutan en casa como parte del proceso de rehabilitación.

Muchas veces la familia del niño tiene ciertas limitantes que afectan la constancia de la terapia, estas pueden ser: falta de seguimiento, falta de material de apoyo para una buena ejecución, espacios no adecuados para la realización de los ejercicios y la falta de tiempos para la terapia en el hogar. De esta manera, la estimulación se vuelve irregular, disminuyendo el beneficio que estas podrían tener si la constancia fuera una prioridad.

Parte de las terapias son complementadas por objetos auxiliares que apoyan el en proceso de rehabilitación. Dependiendo de la patología del niño y su lesión motriz, es la estimulación que se le brinda, trabajando con diversos elementos que favorecen el desarrollo de las habilidades buscadas.

Los elementos auxiliares tienen diversos grados de complejidad, van desde elementos simples como pelotas hasta instrumentos mecánicos grandes. Por lo general estos últimos poseen costos elevados, su complejidad impide el uso en lugares pequeños, son poco amigables con el usuario y contexto y sobre todo están dirigidas a personas adultas por lo que los niños no tienen posibilidad de poder utilizarlos.

Cabe enfatizar, las ventajas que padecen los niños con capacidad diferente son un detonador para promover, desde la perspectiva de diseño, dispositivos auxiliares que permitan aminorar esta brecha y brindarles una mejor calidad de vida.



En este artículo se aborda la propuesta del diseño de un Mecanismo de Rehabilitación física que será de uso en unidades de rehabilitación y doméstico en pediatría como apoyo en la fisioterapia dirigido a niños con síndrome de West.

2. Síndrome de West

El síndrome West (SW) se caracteriza por una triada clásica de espasmos infantiles causando retraso o detención del neurodesarrollo, por lo que se podría considerar como un factor post-natal, ya que es el síndrome epiléptico más frecuente en la época de lactante, los que debutan dentro del primer año de vida constituye el 47% de estas. La edad de inicio es en el primer año de vida, con mayor frecuencia entre los 4-10 meses, siendo el pico máximo a los 5 meses de edad. Su incidencia ha sido estimada entre 1 de cada 4,000 a 1 de cada 6,000 nacidos vivos y predominio en los varones 1.5 a 1, anualmente se estima 2 millones de nuevos caso [4].

El SW con frecuencia se acompaña de parálisis cerebral infantil (PCI), la prevalencia global de PC se sitúa entre 2 y 3 por cada 1000 nacidos vivos y se considera como un grupo de trastornos del desarrollo del movimiento y la postura, causantes de limitación de la actividad, que son atribuidos a una agresión no progresiva sobre un cerebro en desarrollo, en la época fetal o primeros años. Estos niños en ocasiones presentan debilidad muscular, condiciona posturas anormales que interfieren en la sedestación. Estas posturas pueden evolucionar hacia curvas de la columna desarrollando deformidades estructurales fijas. Las posturas compensadoras anormales interfieren con actividades funcionales como las de alcanzar objetos y manipular. Por tanto, es crucial prevenir las alteraciones posturales en sedestación y proporcionar un alineamiento correcto para una adecuada calidad de ajustes posturales y movilidad de las extremidades.

La falta de control postural y equilibrio es un factor limitante para participar en actividades de la vida diaria, identificándose como la mayor limitación para el desarrollo motor de los niños con parálisis cerebral [5].

El síndrome de West (SW) hace parte de las encefalopatías de comienzo temprano en la lactancia y la niñez y se caracteriza por la tríada: espasmos, retraso psicomotor y electroencefalograma hipsarrítmico (ondas al azar de alto voltaje), ligado probablemente a factores del desarrollo neurológico. La prevalencia estimada es de 0.15 a 0.2 por 1000, en niños menores de diez años. Cuando las causas son prenatales los espasmos se presentan antes de los tres meses, los de causas postnatales entre los nueve meses y dos años de vida. Son contracciones breves en flexión, extensión o mixtas, generalmente son bilaterales, simétricas y bifásicas. La duración de cada espasmo varía de medio a dos segundos, al comienzo suelen ser aislados, pero posteriormente son salvas de hasta cien espasmos. Las crisis epilépticas son la primera manifestación del síndrome, pero el retraso psicomotor puede precederlas, al iniciar los espasmos hay regresión del desarrollo. [6]

El tratamiento para el trastorno motor causado por SW, está fundamentado en cuatro pilares básicos: fisioterapia, ortesis, fármacos y tratamiento quirúrgico (cirugía ortopédica, tratamiento neuroquirúrgico) [7].



2.1. Ventajas de la rehabilitación física temprana.

En un estudio retrospectivo realizado en el Centro de Restauración Neurológica de La Habana, Cuba, se revisa el rol de la estimulación temprana en 20 pacientes con PC y retraso del desarrollo psicomotor. Al comparar su desempeño en pruebas de desarrollo psicomotor antes y después del período de intervención, todos los pacientes presentan una mejoría, con adquisición de nuevas habilidades. Aunque la estimulación temprana no mejora la parálisis cerebral, puede ayudar al desarrollo integral de los pacientes [8].

En un estudio se evaluó el efecto funcional de un curso de terapia de seis semanas en un grupo de 15 pacientes con PC. Dentro de los aportes de este estudio destaca que la introducción de objetivos funcionales en la terapia, mejora los resultados, pues los aspectos que fueron definidos como objetivos de la intervención (Ej. subir escaleras) fueron aquellos en que se obtuvieron más avances [9].

Basado en los escritos anteriores se opta por buscar alternativas como mec-re que sirvan de apoyo para incrementar las terapias a edades más tempranas posibles por lo que nuestro enfoque es a infantes de edades de 2 a 5 años que es cuando los pacientes con parálisis cerebral infantil tienen mayor plasticidad en su cerebro y da mejores resultados en el desarrollo psicomotor de los niños, es por ello la importancia de terapias intensas, esto sugerido por la Dra. Maricela Hernández Hernández, especialista de neurología pediátrica en el Hospital de Alta Especialidad en la Ciudad De México.

Por el contrario, se realiza una búsqueda bibliográfica de las complicaciones que tienen las personas con parálisis cerebral una vez que han llegado a la adolescencia con el fin de visualizar las consecuencias de la ausencia de rehabilitación física temprana, ya que suelen tener problemas para ser atendidas debido a que su cuerpo ha finalizado su desarrollo y alcanzan hasta 1.80 m de estatura y pueden superar los 50 Kg de peso. Todo esto torna muy difícil su movilidad, locomoción y traslado, sobre todo tratándose de personas con parálisis severa y que por lo tanto no ayudan para llevar a cabo estas actividades y se le llama comúnmente peso muerto.

En su mayoría estas personas dependen físicamente de sus familiares, terapeutas o cuidadores quienes los mueven, los cargan, los trasladan y sufren una alta demanda de esfuerzo físico. [10].

2.2. Neuroplasticidad

La Organización Mundial de la Salud (1982) define el término neuroplasticidad como la capacidad de las células del sistema nervioso para regenerarse anatómica y funcionalmente, para minimizar los efectos de las alteraciones estructurales o fisiológicas, en otras palabras, es la capacidad del cerebro para adaptarse y compensar los efectos de la lesión, aunque sólo sea de forma parcial, es mayor en los primeros años de la vida que en la etapa adulta. Los mecanismos por los que se llevan a cabo los fenómenos de plasticidad son histológicos, bioquímicos y fisiológicos, tras los cuales el sujeto va experimentando una mejoría funcional- clínica, observándose una recuperación paulatina de las funciones perdidas.

La utilización de técnicas físicas abre la posibilidad de aumentar la excitabilidad de la corteza que interese, facilitando su entrenamiento y posibilitando un incremento de la capacidad de aprender



aquello que se entrene en las horas subsiguientes, siendo esta una de las razones por la que se considera importante que el infante realice por medio del mec-re múltiples repeticiones que le estimulen el aprendizaje de este movimiento.

3. Método.

Mediante la utilización del software Solidworks se realiza el diseño del mecanismo en el cual se toma en cuenta las medidas correspondientes al tamaño real del prototipo, así como también el material a utilizar, complementariamente al modelo mecánico se usa un microcontrolador que por medio de la ejecución de un firmware ejecuta 3 niveles de rutinas que se pueden seleccionar acorde a las indicaciones del fisioterapeuta y detener en cualquier instante de ser necesario e inclusive por medio de un temporizador realizar la tarea solo el tiempo estipulado por el especialista.

Aunado a lo mencionado con anterioridad se realiza el diseño la tarjeta de control para que el sistema funcione correctamente y sea usado en lugares públicos de rehabilitación o privados con el principal objetivo de aprovechar la etapa de mayor neuroplasticidad de los niños.

3.1 Registro de dimensiones, para el diseño del mecanismo.

Con datos obtenidos en el artículo “Dimensiones antropométricas de población latinoamericana ,2007”, se extraen los datos de la tabla 1, tomando en cuenta las dimensiones de niños y niñas en edad preescolar teniendo un rango de 2- 5 años.

Tabla 1. Dimensiones antropométricas de niños latinoamericanos.

Parámetros	2 años (niñas)	5 años (niños)
(h) Altura hombro sentado	26.5 cm	41.0 cm
(i) Altura codo sentado	11-1 cm	19.5 cm
(j) Longitud nalga-poplíteo	19.8 cm	33.3 cm
(k) Altura poplíteo	18.2 cm	31.1 cm
(l) Altura rodilla sentado	22.3 cm	37.1 cm
(m) Anchura codos	24.8 cm	37.0 cm
(n) Anchura cadera sentado	16.5 cm	26.2 cm

Analizando las estaturas de los niños se nota que las niñas son más bajitas que los niños por lo que para obtener los datos mínimos se tomó a consideración las mediciones de niñas de 2 años y para los máximos a niños de 5 años, esto con el fin de abarcar la mayor cantidad de posibilidades de medida de los niños que pudieran requerir de dicho equipo.



Con la siguiente información se tiene un rango de tolerancia del uso del mecanismo para niños(as) de 2-5 años de edad teniendo en cuenta las características físicas de cada posible usuario.

Es importante destacar que la columna denominada parámetros de la tabla 1 se encuentra representado en la figura 1, cada inciso representa una dimensión: (h) Altura hombro sentado, (i) Altura codo sentado, (j) Longitud nalga-poplíteo, (k) Altura poplíteo, (l) Altura rodilla sentado, (m) Anchura codos, (n) Anchura cadera sentado.

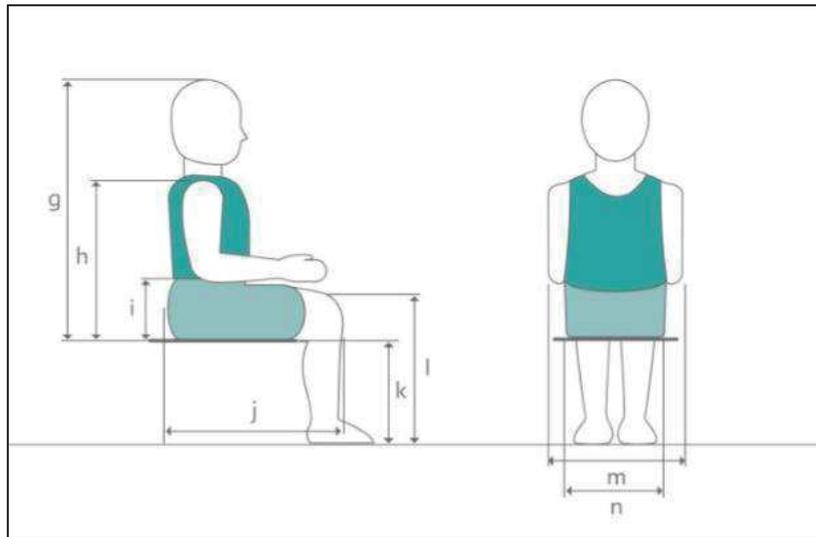


Figura 1. Parámetros de dimensiones antropométricas [1].

3.2 Diseño del prototipo en el software SolidWorks.

De acuerdo a los datos mostrados en la tabla 1, se realiza el diseño propuesto en el software Solidworks, correspondiente a cada una de las piezas del mecanismo como se muestra en la figura 2. Con la finalidad observar la funcionalidad que realiza, también se elaboró el ensamble de las piezas para inspeccionar y verificar coincidencias en las dimensiones y formas de las piezas.

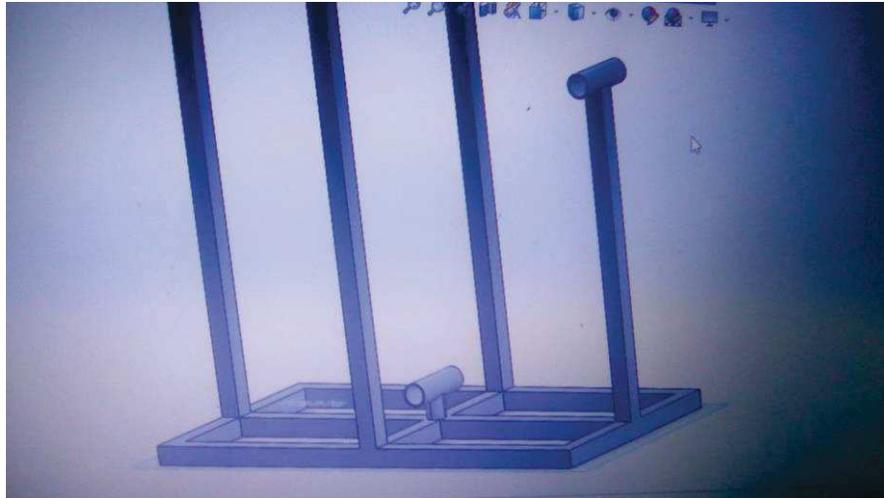


Figura 2. Diseño del prototipo en Solidworks

3.3 Corte y ensamble de las partes mecánicas.

De acuerdo al diseño que se tiene inicialmente se realizan los trazos y cortes de cada uno de las piezas del prototipo basándose en las medidas a escala del mismo realizado en el software como se muestra en la figura 3.



Figura 3. Corte de las piezas en CPVC



Se procede con la unión de las piezas para el ensamble de las partes del mecanismo, ilustrado en la figura 4. Cabe destacar que el ensamble que se realiza es un prototipo (a escala) y no la maquina real, ya que se utiliza CPVC que es un material que no soportaría el peso de los niños que se tienen contemplados en este análisis, que en el caso del peso mínimo sería tomado en cuenta las características de niñas de 2 años que es de 13.2 kg y una estatura de 87.7 cm.

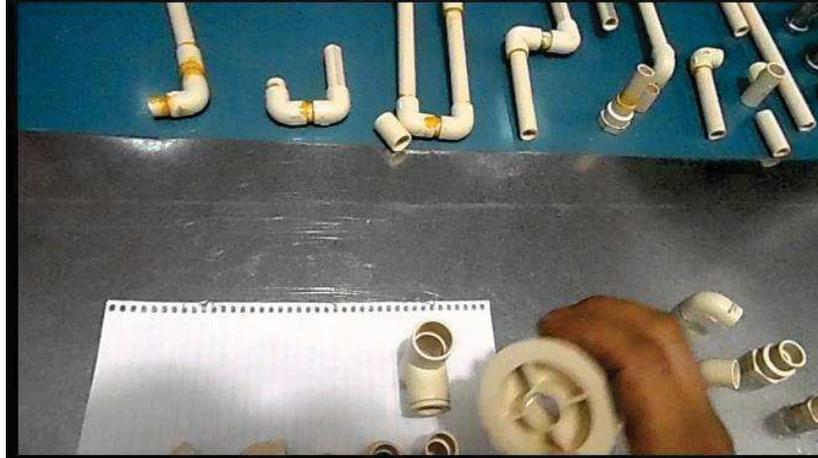


Figura 4. Ensamble de las piezas CPVC para el mecanismo

3.4 Diseño de la tarjeta de control en el software Proteus.

Mediante la utilización del software de diseño esquemático y circuitos impresos Proteus, se realiza el diseño de la PCB (printed circuit board) que es la tarjeta del circuito impreso que tiene como finalidad ser el circuito de control principal del mecanismo.

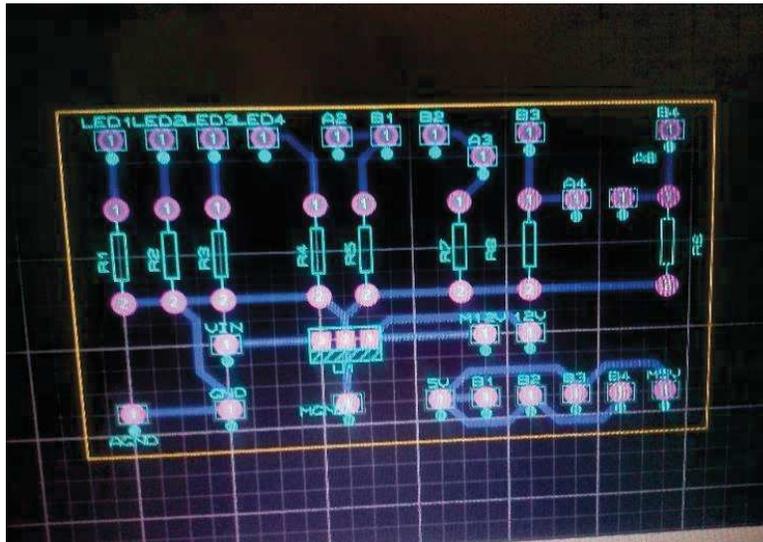


Figura 5. Diseño de la PCB del circuito de control.

En la realización del diseño se toma en cuenta las dimensiones reales proporcionada en la hoja de datos del fabricante de cada uno de los componentes electrónicos a utilizar, los grosores y formas de las pistas o conexiones del circuito que en la figura 4 se muestran como líneas de color azul. En esta misma figura se visualizan círculos de color rosa que representan los pad de cobre que se utilizan



para soldar los componentes a la tarjeta terminada. Los círculos de color gris en el centro de los pad (círculos color rosa) indican el diámetro de perforación de la placa, para la inserción de los componentes electrónicos.

3.5 Elaboración de la tarjeta de control.

Con base al diseño realizado en el software Proteus se lleva a cabo el proceso de transferencia del circuito en la placa de baquelita de una capa de cobre de 1 onza estándar, por medio de transferencia térmica (proceso similar al de sublimación), el diseño transferido en la placa se muestra en la figura 6, y una vez terminado el proceso anterior se procede a perforar acorde a las dimensiones antes seleccionadas en dicho software, con un taladro de mano en cada uno de los pads de cada componente de inserción. Posteriormente se realiza el proceso de soldadura en cada uno de los pines de los componentes electrónicos.

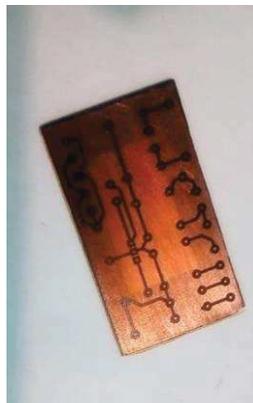


Figura 6. Tarjeta con circuito transferido.

Al término de este proceso ya con los componentes ubicados y en posicionamiento correcto en la tarjeta de circuito impreso, se corta los excedentes de los componentes de inserción, para después comprobar el funcionamiento de circuito de la PCB terminada.

3.6 Desarrollo del código de programación.

Mediante el entorno de desarrollo integrado del acrónimo en inglés “Integrated Development Environment - IDE” se elabora el código compatible con el microcontrolador de 8 bits seleccionado, en este caso se optó por utilizar el Atmega328P de la empresa Atmel – ahora parte de la empresa Microchip- y mediante el entorno IDE de Arduino (el cual es netamente compatible con la mayoría de los microcontroladores de 8 y 32 bits) se desarrolla la programación para controlar el accionamiento del motor base, como se muestra en la figura 7.

```
Arduino IDE - Programador de Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda

MOTOR_01.ino

int boton1= 2;
int boton2= 3;
int boton3=4;
int led1=11;
int led2=10;
int led3=9;
int led4=8;
int led5=7;
int led6=6;
int led7=5;
int led8=4;
int led9=3;
int led10=2;
int led11=1;

void setup()
{
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  pinMode(led5,OUTPUT);
  pinMode(led6,OUTPUT);
  pinMode(led7,OUTPUT);
  pinMode(led8,OUTPUT);
  pinMode(led9,OUTPUT);
  pinMode(led10,OUTPUT);
  pinMode(led11,OUTPUT);
}

void loop()
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,HIGH);
  digitalWrite(led7,HIGH);
  digitalWrite(led8,HIGH);
  digitalWrite(led9,HIGH);
  digitalWrite(led10,HIGH);
  digitalWrite(led11,HIGH);
}
```

Figura 7. Código muestra en el IDE de Arduino.

3.7 Ensamble del prototipo para pruebas pre-operacionales.

Se ensamblan las piezas correspondientes de acuerdo al diseño del modelo realizado en el software de modelado en 2D y 3D Solidworks de Dassault Systemes, y las partes electrónicas del sistema, de igual forma se verifican las conexiones eléctricas entre las partes que forman el sistema como se muestra en la figura 8.

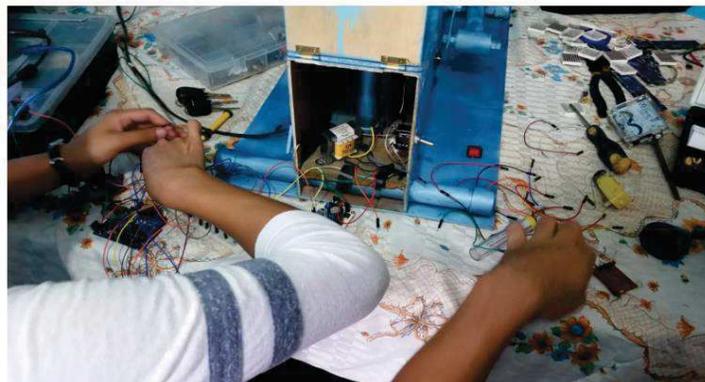


Figura 8. Ensamble del prototipo.

3.8 Pruebas de funcionamiento.

Se realizan los test básicos de las partes eléctricas y/o electrónicas, así como de la parte mecánica, elaborando firmwares individuales para las pruebas operacionales en el microcontrolador de 8 bits, asignando variables en cada uno de los puertos de entrada y salida, así como al elemento final de control se anexaron las rutinas pertinentes adaptándolo a cada caso en particular de acuerdo



al estudio previo del tema, se modificaron en gran medida los parámetros encargados del movimiento mecánico principalmente haciendo las pruebas respectivas para verificar el correcto funcionamiento del prototipo, de igual manera la detección de posibles fallas en el desarrollo del algoritmo planteado, mediante un indicador visual. En la figura 9 se muestra el proceso de optimización las conexiones eléctricas.



Figura 9.- Verificación de conexiones y funcionamiento.

Es importante hacer mención que no solamente se elabora el sistema físico y del hardware, esencialmente se hace énfasis a la creación del algoritmo de programación, se han considerado casos especiales en las rutinas de operación, modificando los tiempos creando una librería especial (creando los archivos *.cpp, *.h mediante el IDE de Arduino – posteriormente se podrá acceder a las librerías de forma gratuita) para la generación de los tiempos y los valores correspondientes en las interrupciones y activaciones del sistema.

Cabe destacar que esta librería se pretende emplear en otros prototipos del área de mecatrónica para el accionamiento de sistemas mecánicos y electrónicos para articulaciones de 2 grados de libertad.

Además, del hardware y firmware se anexaron los indicadores visuales permitiendo tener un mejor control en la activación de las rutinas y visualización del sistema indicando dentro de cada parte del proceso la activación de un diodo emisor de luz “LED” acorde al corrimiento del algoritmo, como se muestra en la figura 9.

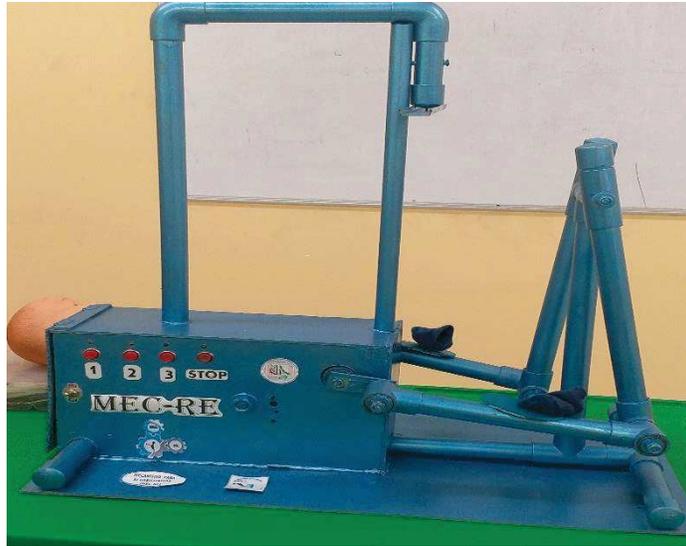


Figura 9.- Indicadores visuales del prototipo.

3.9 Prototipo terminado.

En la figura 10 se muestra el prototipo terminado con el muñeco en el que se realiza la prueba para visualizar el movimiento en piernas que tendría el usuario del mecanismo de rehabilitación.



Figura 10. Prototipo terminado.



4. Resultados

Tomando en cuenta un estudio retrospectivo en un Centro de Neurociencia en la Habana Cuba se atienden a 26 pacientes con PCI (Parálisis Cerebral Infantil), que en las actividades en el cual se observaron mayor resultado fue en subir y bajar escaleras ya que se tiene mayor movimiento en las extremidades inferiores.

El prototipo fue puesto a prueba con un muñeco en el cual se observa que realiza el movimiento en piernas esperado, que es parecido al subir y bajar escaleras.

En el funcionamiento del mecanismo se tiene la siguiente variación de velocidad: en el nivel 1 de 25 RPM (revoluciones por minuto), esto indica que asemeja 25 pasos en 1 minuto; es decir que da un paso cada 2.4 segundos. Este primer nivel de velocidad produce un caminar lento.

El nivel 2 de 35 RPM, indica que asemeja 35 pasos en un minuto, en otras palabras; un paso cada 1.7 segundos aproximadamente. Este segundo nivel de velocidad produce un caminar intermedio.

El nivel 3 de 45 RPM, representa 45 pasos en un minuto, es decir que realiza un paso cada 1.33 segundos aproximadamente, este nivel es el más rápido.

Los niveles de velocidad funcionan correctamente y serán usados de acuerdo al avance que el infante desarrolle a través del proceso de rehabilitación.

Se cuenta con un botón de paro de emergencia que será utilizado en caso de presentar problemas inesperados durante la rutina, de tal manera que el proceso pueda continuar en cuanto el paciente se encuentre en condiciones favorables.

5. Conclusiones

Se pretende que MEC-RE sea elaborado en tamaño real para realizar las pruebas directamente con infantes de 2 a 5 años.

Durante la construcción del prototipo se visitó en varias ocasiones a especialistas en el caso con el fin de que el prototipo cumpliera con los requerimientos necesarios para esta tarea.

El proyecto les parece muy interesante a las personas que se dedican a realizar fisioterapia ya que se está tomando en cuenta a una parte de la sociedad al que no se le da mayor importancia y sin



embargo es vulnerable, además que el enfoque de MEC-RE contribuye a la etapa de mayor alcance en el proceso de rehabilitación física que es la etapa en la que el niño está en desarrollo neuronal.

Es importante contribuir en ayudar en la rehabilitación física de niños que padecen discapacidad motriz debido a que les es benéfico a largo plazo porque se desarrollan de una mejor manera en comparación con niños de la misma edad y pronósticos parecidos.

Este mecanismo puede ser utilizado también para estimulación temprana, siempre y cuando el bebé cuente con la estatura apropiada al diseño del mecanismo.

Referencias

- [1] B. A. Correa González, «Diseño de dispositivo auxiliar para la estimulación motriz y sensorial en niños con discapacidad,» México, 2013.
- [2] H. S. Arrazola y C. A. Berazain, «Síndrome de West: etiología, fisiopatología, aspectos clínicos, diagnóstico, tratamiento y pronóstico.,» *Med-Cient "Luz Vida"*, 2014.
- [3] OMS, «OMS,» [En línea]. Available: <https://www.who.int/disabilities/technology/es/>. [Último acceso: 23 Agosto 2019].
- [4] G. Zaldivar Pascua, «Síndrome de West,» *Acta pediátrica Hondureña*, vol. 4, nº 1, pp. 272, 274, septiembre 2013.
- [5] C. Fernandez Gutierrez y M. D. Apolo Arenas, «Fisioterapia,» vol. 37, pp. 135-139, 2015.
- [6] M. Barbosa, «Estado del arte en el tratamiento del síndrome de West,» *Acta Neurol Colomb*, vol. 22, 2006.
- [7] P. Póo Argüelles, «Parálisis Cerebral Infantil,» *Asociación Española de Pediatría*, 2008.
- [8] M. Garcia Navarro, M. Tacoronte, I. Sarduy, A. Abdo, R. Galvizu y A. Torres, Influencia de la estimulación temprana en la parálisis cerebral, *Rev Neural*, 2000, pp. 716-719.
- [9] V. Knox y A. Lloyd Evans, Evaluation of the functional effects of a course of Bobath therapy in children with cerebral palsy: a preliminary study, *Dev Med Child Neurol*, 2002, pp. 447-460.
- [10] G. Ceballos Caldera, «Sistema de movilidad y transferencia para adultos con parálisis cerebral y cuadriparecia espástica.,» Mexico D.F., 2012.
- [11] R. Hrachovy y J. Frost, Espasmos Infantiles, *Clinica Pediatrica del Norte America*, 1989, pp. 337-356.
- [12] P. P. Argüelles, «Parálisis Cerebral Infantil,» *Clinica Pediatrica del Norte America*, pp. 337-356,



1989.

[13] . A. J. Pozo Alonzo, d. Pozo Lauzan y D. Pozo Alonso, SINDROME DE WEST: ETIOLOGIA, FISIOPATOLOGIA, ASPECTOS CLINICOS Y PRONOSTICOS, REV CUBANA PEDIATR, 2002.

[14] V. Ruggieri, Epilepsias de comienzo y la infancia temprana, Rev. Neuro, 2004, pp. 251-262.

[15] J. Narbona y E. Schlumberger, Retraso Psicomotor, Asociación Española de Pediatría, 2008.



Diseño y Simulación de un Prototipo de Alarmas con Ubicación en Tiempo Real para Objetos Móviles Utilizando un Microcontrolador PIC 18f4550

Ramiro Ramiro José¹, Tacuepian Damian Oscar³, Hernández García Edgar¹
y Méndez Alonso José Miguel².

¹ Ingeniería Mecatrónica - Instituto Tecnológico Superior de Zacapoaxtla
Correo de contacto: semirajo@msn.com

² Ingeniería Informática – Instituto Tecnológico Superior de Zacapoaxtla

³ Alumno de Ingeniería Mecatrónica – Instituto Tecnológico Superior de Zacapoaxtla

Resumen

El presente trabajo describe el diseño y la construcción de un prototipo de alarmas y/o alertas en forma de mensajes de texto (SMS) que son enviadas a un teléfono móvil o Smartphone. Dichas alarmas y/o alertas son generadas por la lectura que adquieren distintos sensores conectados a un microcontrolador, hablando específicamente del PIC18F4550 del fabricante Microchip. Cuando un sensor detecta un cambio en la variable del cual se encarga de registrar, éste genera una interrupción de tipo externa en el microcontrolador el cual se encargará de procesar y enviar en forma de alarma y/o alerta lo obtenido a un módulo GSM (SIM900) en particular. Finalmente, el módulo GSM se encargará de establecer comunicación con el teléfono móvil o Smartphone enviándole las alarmas y/o alertas en forma de mensajes de texto SMS acompañados de una llamada para asegurar que el usuario sea notificado de lo acontecido. Cabe mencionar que el prototipo cuenta con un módulo GPS adicional que permite al usuario obtener la ubicación exacta en tiempo real del sistema de alarmas y/o alertas en caso de usarlo en un sistema móvil.

Palabras clave: Microcontrolador, GPS, SMS, GSM, Alarmas, Seguridad, objetos móviles

1. Introducción

Uno de los más grandes e inevitables problemas a los cuales se enfrenta día a día la población en general es sin duda la delincuencia, se nombran secuestros, homicidios y feminicidios, por ejemplo, que son situaciones nada deseadas por las familias mexicanas debido a que la pérdida de un familiar es muy dolorosa, y marca de por vida la situación en la que se deja a la misma. Se tienen también delitos que de una u otra manera no separan a la familia ni la desintegran, pero si causan pérdidas materiales, tales como el despojo de pertenencias, por ejemplo; de un automóvil, que con esfuerzo, sacrificio y trabajo duro se obtiene. Todas las personas de forma directa o indirecta se encuentran inmiscuidas en esta situación, principalmente los comerciantes, transportistas, ejecutivos.

Según estadísticas de la Asociación Mexicana de Instituciones de Seguros (AMIS), en 2018 se registraron números de robo de vehículos de más de 93 000 unidades, las cuales superan las 90 000 registradas en el 2017, realizando la comparación es muy notable el aumento de robo de vehículos que se ha dado en estos años recientes a nivel nacional. [1] Ubicando al estado de Puebla dentro de los primeros 5 estados con mayor tasa de robo vehicular. Son cifras alarmantes, tomando en cuenta que nadie está exento de ser víctima de tal delito, es por ello que se hace necesario algún tipo de sistema que se pueda adaptar al vehículo y que permita el monitoreo del mismo y darle al usuario mayor confianza de dejarlo estacionado en las calles.



Un sistema de alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella, cumpliendo así, una función disuasoria frente a posibles problemas, además; son capaces de reducir el tiempo de ejecución de las acciones a tomar en función del problema presentado, reduciendo así las pérdidas [2]. Una vez que la alarma comienza a funcionar, dependiendo del sistema instalado, este puede tomar acciones en forma automática. Para esto, la alarma tiene que tener conexiones de entrada para los distintos tipos de detectores y conexiones de salida para activar otros dispositivos.

Con la implementación de un sistema de alarmas y/o alertas en un sistema móvil, se notifica al usuario sobre alguna falla o cambio de estado en el sistema, permitiéndole tomar decisiones antes de que las fallas o cambios de estado afecten en gran medida al sistema. Así, se le puede llamar sistema móvil a un automóvil, una motocicleta, una bicicleta, un robot móvil e incluso algo que llevamos a todas partes como lo es una mochila. El sistema de alarmas cuenta con sensores de temperatura, humedad, llama, lluvia y un módulo GPS que, en cada cambio de estado, notifica al usuario vía SMS añadiendo las coordenadas geográficas para ubicar al sistema móvil y tomar la decisión con respecto a la alarma y/o alerta recibida.

Para iniciar con el desarrollo del prototipo y que a su vez este nos enviara información en tiempo real, fue necesario investigar sobre el término “comunicaciones móviles”, el cual se describe como cualquier enlace de radiocomunicación entre dos terminales, de los cuales al menos uno está en una localización indeterminada, en movimiento, o parado, pudiendo el otro ser un terminal fijo. Esta definición es de aplicación a todo tipo de enlaces de comunicación, ya sea de móvil a móvil o entre fijo y móvil. De hecho, el enlace móvil a móvil consiste muchas veces en un enlace móvil-fijo-móvil. [3]

El término móvil puede referirse a vehículos de todo tipo -automóviles, aviones, barcos, trenes- o, sencillamente, a personas en movimiento, o paradas, que hacen uso de un terminal “móvil”. El enlace de radiocomunicación utiliza parte del espectro electromagnético, concretamente, la banda conocida como de radiofrecuencias comprendida, aproximadamente, entre 100kHz y 10 GHz. [4]

Por ello, es importante mencionar que, en los últimos años, los mayores esfuerzos de la investigación en la robótica han sido enfocados en mejorar la actuación de robots móviles individuales utilizando sensores, actuadores y algoritmos de control avanzados. Como resultado, los robots individuales se han vuelto sofisticados, capaces de realizar tareas arduas o difíciles requeridas por aplicaciones del mundo real. [5] No obstante, muchas de estas tareas aún siguen siendo complicadas para éstos. Sin embargo, siguen siendo limitados para informar sobre su estado de servicio.

Detrás de cada sistema que interactúa con el mundo real, se encuentra el proceso de adquisición de señales. El propósito de la adquisición de datos es el medir las variables físicas como voltaje, corriente, presión o sonido. Mientras que cada sistema de adquisición de datos es definido por los requerimientos de su aplicación. Todos los sistemas comparten un mismo objetivo final de adquisición, análisis y procesamiento de información. Los sistemas de adquisición de datos incorporan sensores, actuadores, acondicionadores de señal, dispositivos de adquisición de datos y aplicaciones de software [5].

2. Metodología

A lo largo de este trabajo desarrollado que integra un prototipo de sistema de alarmas. Para ello se ha dividido el sistema en cuatro bloques para una mejor comprensión, tal como se muestra a continuación según su importancia de aplicación: Bloque de Sensado, Bloque de Procesamiento de señales, Bloque de Transmisión y Bloque de visualización. Se trabajó bloque a bloque para dar un sentido de confianza al sistema, una vez listos los bloques, se unificaron para dar forma a todo el sistema. A continuación, se detalla lo realizado en cada bloque (ver figura 2.1).

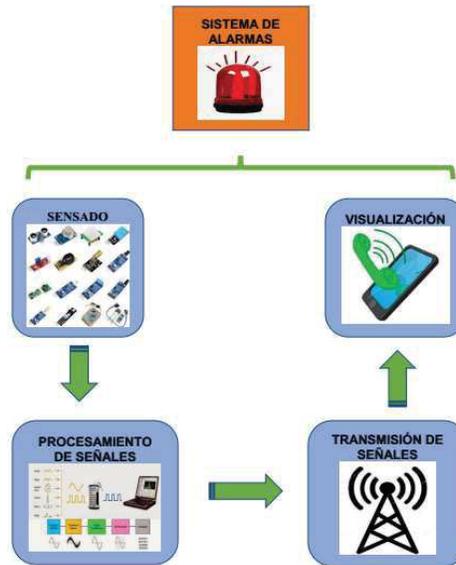


Figura 2.1. Esquema general del sistema por bloques

2.1 Bloque de sensado

En esta sección se anexan los sensores para la medición de variables físicas de acuerdo a la aplicación del sistema de alarmas, estos pueden cambiar de acuerdo a lo que el usuario desee y se adapte mejor al sistema a inspeccionar en nuestro caso se probó con (sensor de flama, sensor de lluvia, sensor de vibración, y módulo GPS).

Para este artículo se describe más a detalle el sensado mediante GPS, utilizando un módulo para establecer comunicación con los satélites y obtener la localización del sistema de alarmas en tiempo real. En este caso se utilizó un **módulo GPS NEO-6M** del fabricante suizo U-blox. Con la información otorgada por el fabricante, se procedió a realizar las pruebas de funcionamiento del módulo, se conectó mediante UART con la PC y los datos obtenidos de los satélites se muestran en la terminal virtual con el software **Tera Term**. Se describen a continuación los pasos seguidos para la prueba.

Se realizó el montaje del circuito en una tarjeta experimental (protoboard): Como se observa en la figura 2.2, se utilizó un adaptador TTL a USB, ya que la PC utilizada no cuenta con puerto DB9 que incluye pines para el protocolo RS232 (TX y RX), de hecho, las nuevas PC's ya no la incluyen, así que con el adaptador TTL a USB se emigra a la tecnología USB.

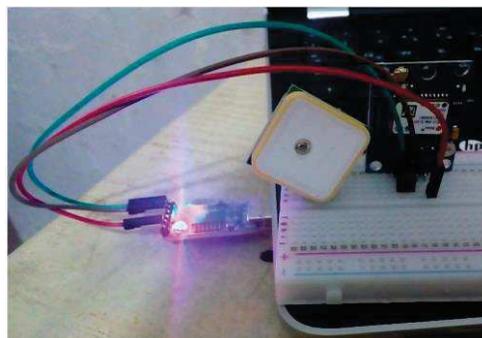


Figura 2.2. Montaje del m modulo GPS Neo6M

El pin TX del módulo GPS NEO6M se conectó con el pin RX del adaptador, el pin de GND con el del adaptador (para cumplir con el protocolo RS232), se omitió el pin RX del módulo NEO-6M



debido a que por el momento no es necesario enviar datos al módulo, solamente recibirlos. Ver el diagrama de conexión en la figura 2.3

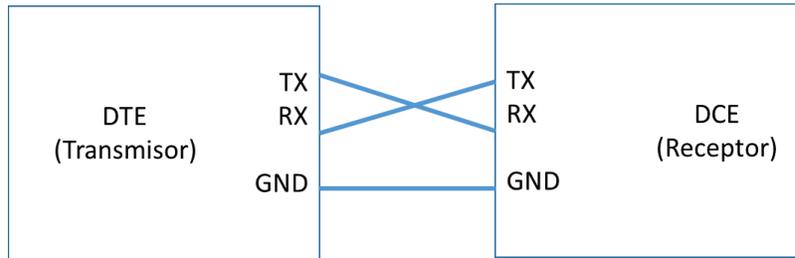


Figura 2.3 Protocolo de comunicación RS232.

1. Se seleccionó la opción de puerto serial y en el cuadro de selección a la derecha se buscó el puerto asignado al módulo, en este caso es el puerto COM8: (Ver figura 2.4)

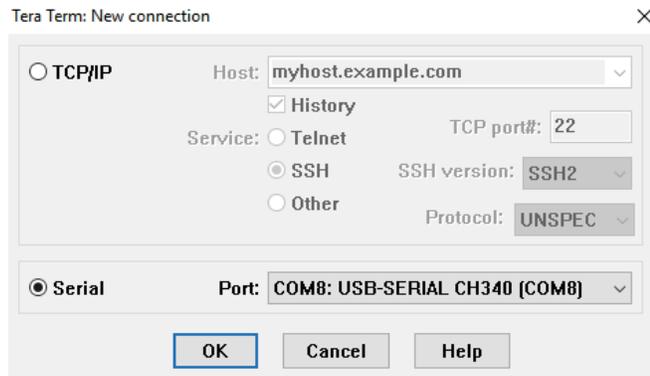


Figura 2.4 Selección de puerto asignado al módulo por la PC.

2. Se configuraron los parámetros de comunicación módulo-PC: (ver figura 2.5)

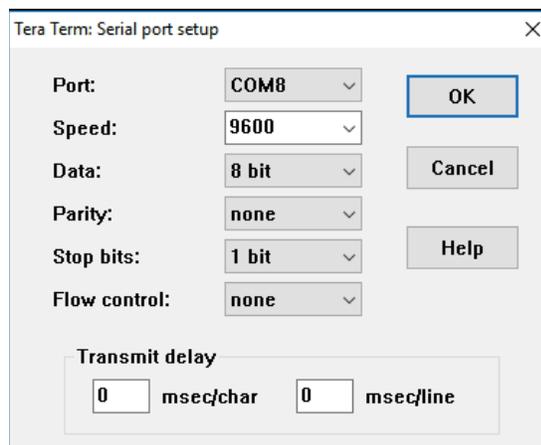


Figura 2.5 Ventana de configuración de parámetros de comunicación.

3. Una vez hecho el montaje y las configuraciones, se muestran en la terminal virtual los datos recibidos por el módulo. En la Figura 2.6, se muestran los datos recibidos por el módulo GPS NEO-6M al haber establecido una comunicación exitosa con los satélites, cabe mencionar que



con dichos datos aún no se cuenta con la información completa, es recomendable realizar la prueba a la intemperie y con un clima despejado para que el tiempo de espera disminuya.

```
COM7 - Tera Term VT
File Edit Setup Control Window Help
$GPTXT,01,01,02,u-blox ag - www.u-blox.com*50
$GPTXT,01,01,02,HW_UBX-G60xx_00040007 FF7FFFFFp*53
$GPTXT,01,01,02,ROM CORE 7.03 (45969) Mar 17 2011 16:18:34*59
$GPTXT,01,01,02,ANTSUPERV=AC SD PDo$ SR*20
$GPTXT,01,01,02,ANTSTATUS=DONTKNOW*33
$GPRMC,V,,,,,N*53
$GPVTG,,,,,N*30
$GPGGA,,0,00,99.99,,,,*48
$GPGSA,A,1,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,00*79
$GPGLL,,,,,V,N*64
$GPRMC,V,,,,,N*53
$GPVTG,,,,,N*30
$GPGGA,,0,00,99.99,,,,*48
$GPGSA,A,1,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,00*79
$GPGLL,,,,,V,N*64
$GPTXT,01,01,02,ANTSTATUS=INIT*25
$GPRMC,V,,,,,N*53
$GPVTG,,,,,N*30
$GPGGA,,0,00,99.99,,,,*48
$GPGSA,A,1,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,00*79
$GPGLL,,,,,V,N*64
```

Figura 2.6 Trama de datos recibidos por conexión hecha exitosamente.

- Después de algunos minutos de espera, el led indicador de señal del módulo GPS NEO-6M comienza a parpadear (ver figura 2.7), lo cual indica que ya se cuenta con información completa, esto se verifica en la terminal virtual que muestra más datos que la figura anterior.

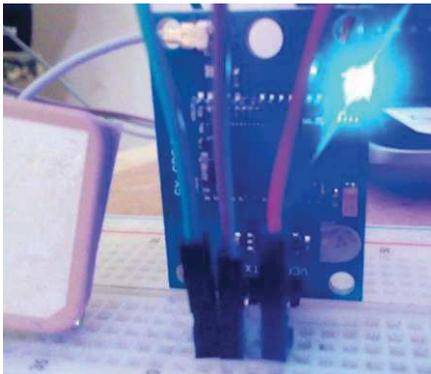


Figura 2.7 Parpadeo de LED indicador en módulo GPS NEO-6M.

En la Figura 2.8 se muestran los datos recibidos de los satélites en el módulo GPS NEO-6M, los datos son recibidos secuencialmente cada segundo. La trama de datos que se muestran en la terminal virtual siguen el protocolo NMEA 0183, que es un medio a través del cual los instrumentos marítimos y también la mayoría de los receptores GPS pueden comunicarse los unos con los otros. Ha sido definido, y está controlado, por la organización estadounidense National Marine Electronics Association.[6]



```

COM7 - Tera Term VT
File Edit Setup Control Window Help
$GPGGA,012509.00,1906.07162,N,09812.18697,W,2.05,3.87,2160.9,M,-7.9,M,,0000*6B
$GPGSA,A,3,19,30,07,13,06,,,,,,,,,6,76,3.87,5.54*05
$GPGSV,3,1,12,02,01,208,24,03,06,098,,06,30,193,27,07,34,146,27*7E
$GPGSV,3,2,12,11,14,040,20,13,26,275,25,15,02,298,,19,49,269,26*70
$GPGSV,3,3,12,28,45,016,,30,70,148,38,46,68,179,37,51,65,206,39*7A
$GPGLL,1906.07162,N,09812.18697,W,012509.00,A,D*7E
$GPRMC,012510.00,A,1906.07262,N,09812.18670,W,5.772,18.13,140219,,D*48
$GPVTG,18.13,T,,M,5.772,N,10.690,K,D*3A
$GPGGA,012510.00,1906.07262,N,09812.18670,W,2.05,3.86,2160.8,M,-7.9,M,,0000*69
$GPGSA,A,3,19,30,07,13,06,,,,,,,,,6,76,3.86,5.54*04
$GPGSV,4,1,13,02,01,208,20,03,06,097,,06,30,193,27,07,34,146,32*77
$GPGSV,4,2,13,11,14,040,,13,26,275,26,15,02,298,,19,49,269,26*77
$GPGSV,4,3,13,20,,25,28,45,016,,30,70,148,39,46,68,179,39*4F
$GPGSV,4,4,13,51,65,206,41*4D
$GPGLL,1906.07262,N,09812.18670,W,012510.00,A,D*7C
$GPRMC,012511.00,A,1906.07352,N,09812.18677,W,4.730,19.09,140219,,D*41
$GPVTG,19.09,T,,M,4.730,N,8.760,K,D*00
$GPGGA,012511.00,1906.07352,N,09812.18677,W,2.05,3.86,2161.3,M,-7.9,M,,0000*67
$GPGSA,A,3,19,30,07,13,06,,,,,,,,,6,75,3.86,5.54*07
$GPGSV,3,1,12,02,01,208,14,03,06,097,,06,30,193,27,07,34,146,32*76
$GPGSV,3,2,12,11,14,040,,13,26,275,27,15,02,298,,19,49,269,27*71
$GPGSV,3,3,12,28,45,016,,30,70,148,39,46,68,179,40,51,65,206,42*77
$GPGLL,1906.07352,N,09812.18677,W,012511.00,A,D*78
    
```

Figura 2.8 Trama de datos recibidos con información completa.

Para su aplicación en este trabajo solo bastaron las tramas GPGGA y GPRMC, los cuales cuentan con los datos de latitud y longitud que bastan para ubicar al receptor GPS en un mapa de acuerdo a la NMEA0123 mostrados en la figura 2.9

RMC Recommended Minimum Navigation Information

```

1 2 3 4 5 6 7 8 9 10 11 12
| | | | | | | | | | |
$--RMC,hhmmss.ss,A,llll.ll,a,yyyy.yy,a,x.x,x.x,xxxx,x.x,a*hh
    
```

- 1) Time (UTC)
- 2) Status, V = Navigation receiver warning
- 3) Latitude
- 4) N or S
- 5) Longitude
- 6) E or W
- 7) Speed over ground, knots
- 8) Track made good, degrees true
- 9) Date, ddmmyy
- 10) Magnetic Variation, degrees
- 11) E or W
- 12) Checksum

Figura 2.9 Formato de datos obtenidos con la trama GPRMC.

Se muestra a continuación el despliegue de los datos más relevantes obtenidos en la trama GPRMC para nuestro caso:

```

$GPRMC,002218.00,A,1906.06209,N,09812.19624,W,0.309,,110119,,A*61
    
```

- 1.) Hora: 00:22:18.00 UTC
- 2.) Estado: A=OK
- 3.) Latitud: 1906.06209



- 4.) N o S: Norte (North "N")
- 5.) Longitud: 09812.19624
- 6.) E o W: Oeste (West "W")
- 7.) Rapidez: 0.309 nudos (572.268 m)
- 8.) ...
- 9.) Fecha UTC: 11/01/2019

Para corroborar la información de los datos obtenidos, se realizó la búsqueda de la latitud y longitud previstos por el módulo GPS en el navegador de Google Maps, teniendo en cuenta que: Latitud: 19°, 06.06209 minutos Norte; se convirtieron los minutos a grados dividiendo $06.06209/60 = 0.10103483$, ya que un grado tiene 60 minutos, obteniendo así **19.10103483°** positivos ya que es en el hemisferio Norte. Longitud: 98 grados, 12.19624 minutos Oeste, se convirtieron los minutos a grados dividiendo $12.19624/60 = 0.20327067$, obteniendo así **98.20327067°** Oeste, en este caso - **8.20327067°** por tratarse del Oeste que se encuentra del lado izquierdo del meridiano de Greenwich. De esta manera, con la información correcta, se procedió a ubicar el módulo GPS NEO-6M en el mapa de Google Maps, según los datos proporcionados, tal y como se muestra en la figura 2.10

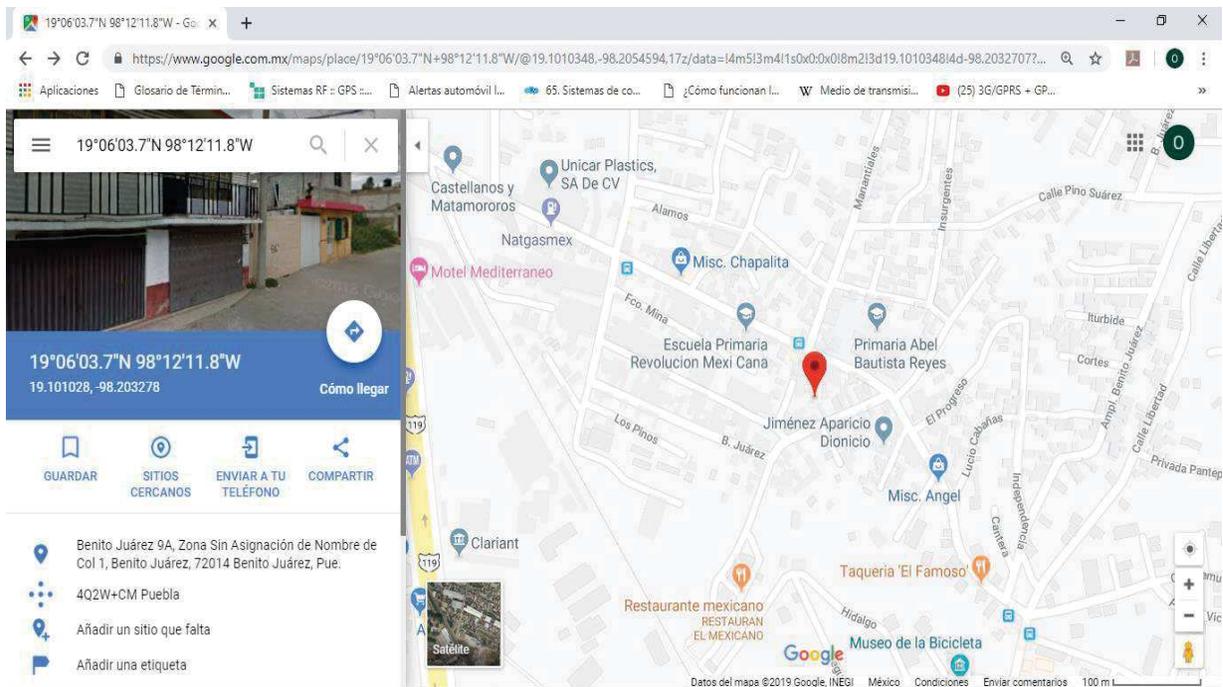


Figura 2.10 Coordenadas previstas por el módulo GPS NEO-6M ubicadas en Google Maps.

2.2 Bloque de procesamiento de señales

En este bloque se realiza todo lo consecuente a la recepción de señales emitidas por los sensores (cambio de estado) y localización por parte del módulo GPS. El microcontrolador recibe la señal de cada sensor (entrada/lectura), se realiza la programación pertinente y al final se entrega una (salida/escritura) para el módulo encargado de enviar las alarmas al usuario (bloque de transmisión de señales). A continuación, se presenta en la figura 2.11 el diagrama de flujo que resume este bloque:



Figura 2.11. Diagrama de flujo del bloque de procesamiento de señales

Aquí es donde el microcontrolador PIC18F4550 (MCU de aquí en adelante) tiene su parte del trabajo, debido a que es el medio por el cual las señales son procesadas y manipuladas conforme a los requerimientos del usuario para su aplicación al sistema de alarmas. Se cuenta entonces con tres fuentes de interrupción independientes (aunque podemos habilitar a otras) en el puerto B del MCU los cuales son: RB0, RB1 Y RB2. Siendo estas: INT0 para RB0, INT1 para RB1 e INT2 para RB2 respectivamente, llamadas cada una interrupción externa.

Cabe mencionar que el puerto cuenta con ocho pines, de los cuales siete forman parte de alguna fuente de interrupción, quedando así el pin RB3 sin usar debido a que no está asociado a ningún tipo de interrupción, limitando así el uso de solamente siete pines para entrada de señales externas de sensores, siendo suficiente esta cantidad para nuestro prototipo.

2.2.1 Programación y simulación de interrupciones externas

Se decidió trabajar con las interrupciones externas con flancos de bajada, es decir, comenzando en estado inicial ALTO y al detectar un estado BAJO se procediera a generar la interrupción.

Para llevar a cabo la programación de las rutinas de interrupción se utilizó el software CCS PICC Compiler en su versión 5.061 del cual ya se tenía noción. Es un software que implementa el lenguaje de programación C para su aplicación en los microcontroladores PIC de Microchip, incluye gran variedad de funciones integradas que permiten la manipulación de los registros de los microcontroladores, además del uso de todas las familias de PIC's.

La programación y simulación de interrupción por cambio de estado en los pines RB4 a RB7 se describió el uso de interrupciones por flanco de bajada en los pines RB0, RB1 Y RB2 que son fuentes de interrupción externa que activan a las interrupciones del MCU INT0, INT1 e INT2. Con ellos llevamos hasta el momento tres interrupciones usadas, las cuales podemos cambiar por sensores que tengan salida digital, se dijo también que el pin RB3 del MCU no formaba parte de ningún tipo de interrupción por lo cual queda descartado. Restan entonces cuatro pines que van del pin RB4 al pin RB7, estos cuatro pines tienen la particularidad de generar una interrupción cuando en cualquiera de ellos se detecta un cambio de estado, sea tanto como de bajada o de subida.

Como en los pines RB0, RB1 Y RB2 se trabajó con interrupciones por flanco de bajada, es decir, cambiando de estado lógico ALTO a estado lógico BAJO en esta sección y con las fuentes de interrupción de los pines RB4 a RB7 se usan flancos de subida, es decir, cambios de estado lógico BAJO a cambios de estado lógico ALTO. Esto con el fin de dar al usuario dos opciones de aplicación y el uso de sensores con ambos cambios de estado.



El estado de las salidas del puerto D correspondientes a las entradas en el puerto B están inicialmente en BAJO (LED's apagados), en la Figura 2.12 se observa el cambio de estado al presionar el pulsador de la entrada del pin RB4, dicho cambio se ve reflejado en el pin D4 del puerto D, el led está encendido.

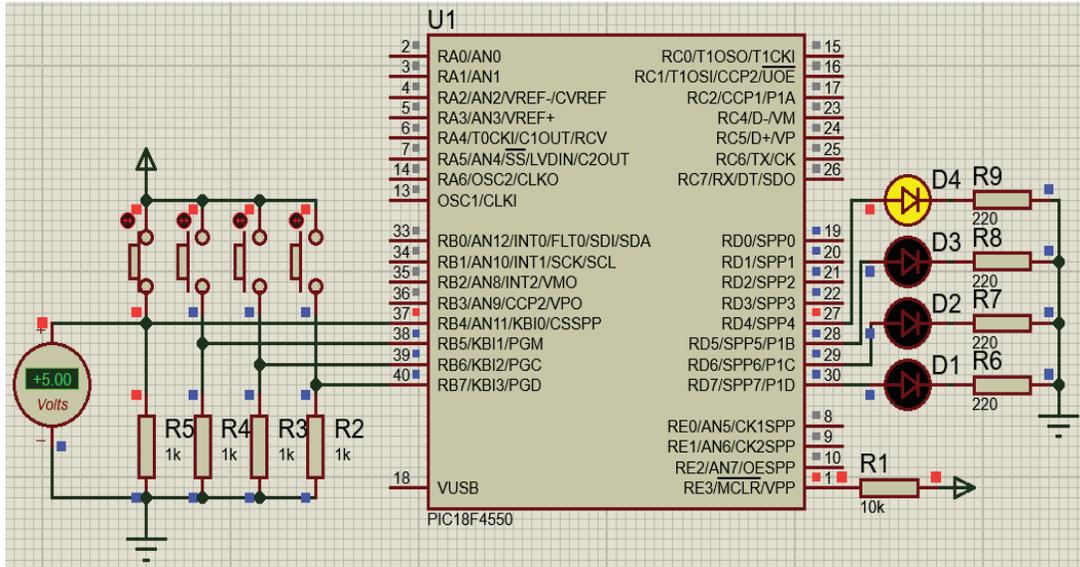


Figura 2.12 Interrupción en puerto RB4 activada, LED en pin RB4 encendido.

2.2.2 Unificación de interrupciones en el puerto B

Los circuitos expuestos anteriormente se hicieron a modo de prueba para evaluar el funcionamiento de cada fuente de interrupción por separado en el puerto B del MCU, todo esto con el fin de evitar confusiones en caso de presentarse anomalías en el circuito, pero como los resultados fueron favorables, es momento de unificar todas las fuentes de interrupción para que de este modo se vaya acercando más al circuito que dará vida a este prototipo de alarmas.

El circuito de la figura 2.13 muestra a grandes rasgos los pines del puerto B utilizados para las fuentes de interrupción y los pines del puerto D como indicadores o respuesta a cada fuente de interrupción:

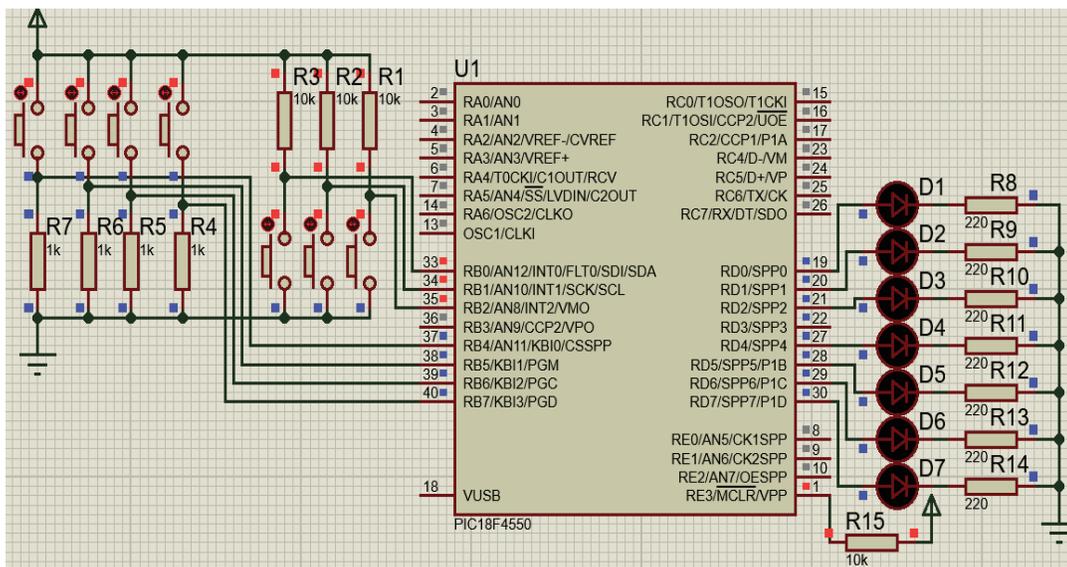


Figura 2.13. Circuito con fuentes de interrupción en Puerto B.



2.2.3 Decodificación de trama de datos GPS

Como se vio en el bloque anterior, el módulo GPS aporta información muy valiosa para trabajar con móviles, básicamente para tener monitoreado el sistema móvil, para tener un margen de maniobra en caso de salirse de los límites permitidos. La trama de datos que entrega el módulo es un tanto confuso para el usuario final, por lo tanto se hizo necesario el procesamiento de las tramas de datos de acuerdo al protocolo NMEA 0183, debido a la complejidad de la programación, se optó por hacer uso de una librería que ya se encarga por si sola de realizar el procesamiento de los tramas y entregar la información en formato más amigable para el usuario, la librería cuya fuente es: (<https://simple-circuit.com/neo-6m-gps-pic18f4550-interfacing/>) permite al usuario visualizar los siguientes datos de la trama GPRMC: hora, fecha, latitud, longitud, altitud, numero de satélites, rapidez y la inclinación.

Desde luego que para este prototipo no es necesario toda esa información, de esta trama se tomaron solamente la hora, la latitud y la longitud, considerados más relevantes para la aplicación, en Proteus como en la realidad, los datos se visualizan por igual en la terminal virtual de Windows. En la Figura 2.14 se observa en la terminal virtual de Proteus los datos requeridos y tomados del módulo GPS NEO-6M de acuerdo a lo que nos permite la librería.

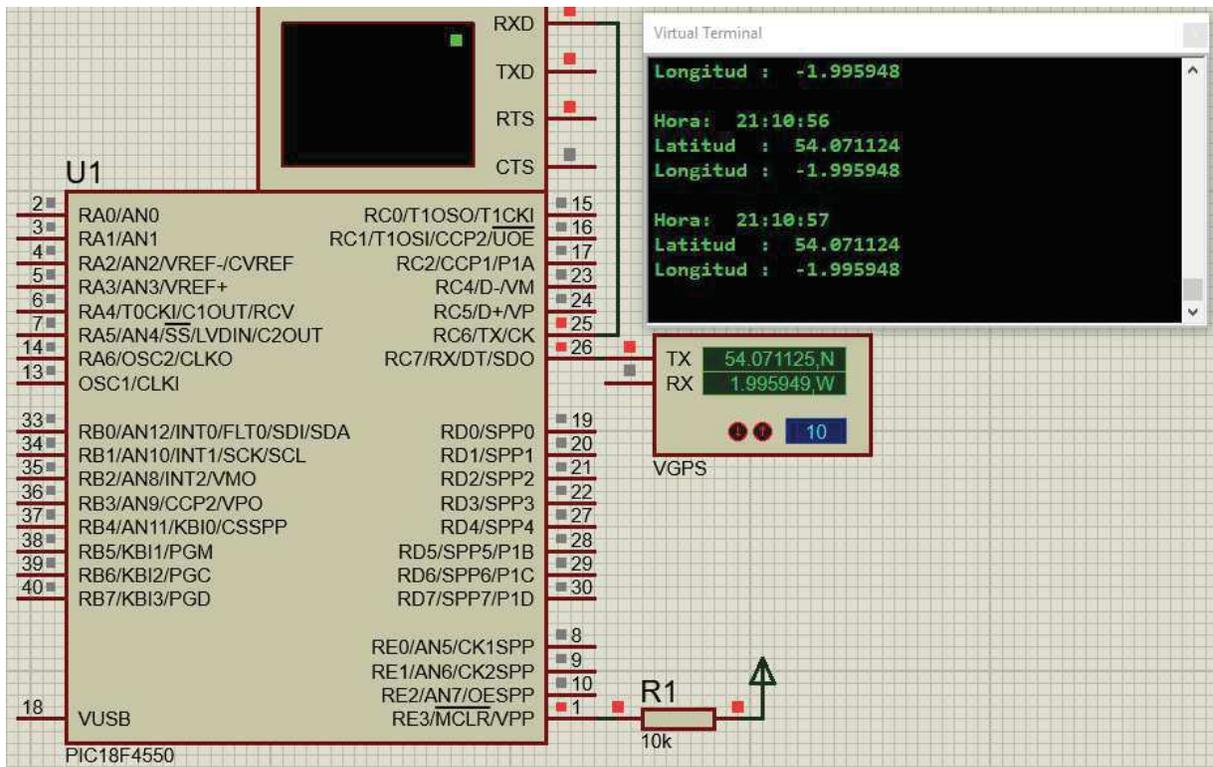


Figura 2.14. Datos entregados por el módulo GPS mostrados en la terminal virtual de Proteus.

A continuación, se muestra en la figura 2.15 la aplicación del código en el MCU PIC18F4550, los datos son mostrados en la terminal virtual del software Tera Term. La hora es de mucha importancia en la aplicación ya que al haber detectado un cambio de estado en algún sensor esta es enviada vía SMS junto a la latitud y longitud para indicar al usuario el lugar y el momento en que se detectó algo inusual.



```
COM7 - Tera Term VT
File Edit Setup Control Window Help
Latitud : 19.100920
Longitud : -98.203232
Hora: 22:25:40
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:41
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:42
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:43
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:44
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:45
Latitud : 19.100922
Longitud : -98.203232
Hora: 22:25:46
Latitud : 19.100922
Longitud : -98.203232
```

Figura 2.15 Terminal virtual del software TeraTerm, se muestran: hora, latitud y longitud, entregados por el módulo GPS NEO-6M.

Para demostrar la fiabilidad de los datos entregados por el módulo GPS (longitud y latitud), se copian los datos obtenidos a la barra de búsqueda de Google Maps para visualizar la ubicación, como puede verse en la Figura 2.16, la ubicación del Smartphone (color azul) coincide con los datos obtenidos por el módulo GPS (marcador rojo), mostrando así la confiabilidad de la localización.

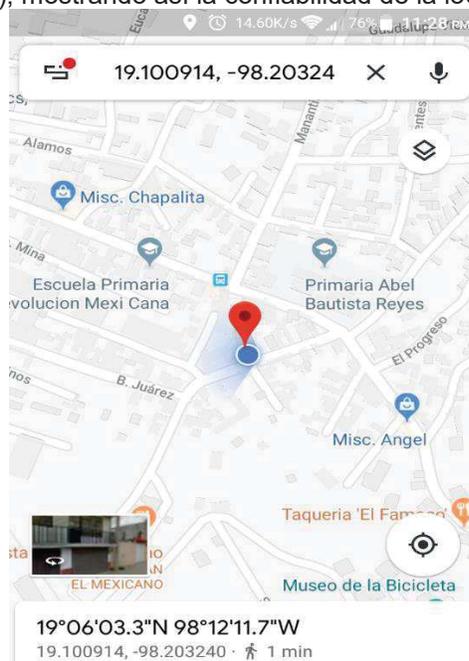


Figura 2.16. Latitud y Longitud mostrados en Google Maps desde un Smartphone (marcador rojo).

Con el uso de la librería “LibreríaGPS” que decodifica los tramas NMEA 0183 del módulo GPS se reducen considerablemente las líneas de código, se optimiza el uso de memoria y hace que la programación de todo el código resulte entendible para el usuario, la Figura 2.17 muestra el código



implementado para la obtención de la hora, latitud y longitud del módulo GPS utilizando el PIC18F4550:

```
1  #include <18F4550.h>
2  #device PASS_STRINGS = IN_RAM
3  #fuses hs,nowdt
4  #use delay(crystal=8MHz)
5  //DIRECTIVA DE CONFIGURACIÓN DEL PUERTO SERIAL MODULO GPS-PC
6  #use rs232(baud=9600, bits=8, xmit=pin_c6, rcv=pin_c7, parity=n, errors)
7  #include <LibreríaGPS.c> //LIBRERÍA PARA DECODIFICACIÓN DE TRAMA GPS NMEA0183
8
9  void main()
10 {
11     while(TRUE)
12     {
13         if(kbhit())
14         {
15             if(GPSRead())
16             {
17                 //FORMATO DE IMPRESIÓN DE LA HORA
18                 printf("\rHora: %02u:%02u:%02u\r\n", GPShour(), GPSMinute(), GPSSecond());
19
20                 // FORMATO DE IMPRESIÓN DE LA LATITUD
21                 printf("Latitud  : %.6f\r\n", Latitud());
22
23                 //FORMATO DE IMPRESIÓN DE LA LONGITUD
24                 printf("Longitud : %.6f\r\n", Longitud());
25             }
26         }
27     }
28 }
29
30
31 }
```

Figura 2.17 Código empleado para obtener la hora, la latitud y longitud del módulo GPS.

3. Resultados

Como resultado se obtuvo el diseño y construcción de una tarjeta de alarmas con un PIC18F4550 como se muestra en la figura 3.1, la cual envía un mensaje de texto a otro dispositivo móvil con la ubicación exacta y el nombre de la alarma por la que fue activado.

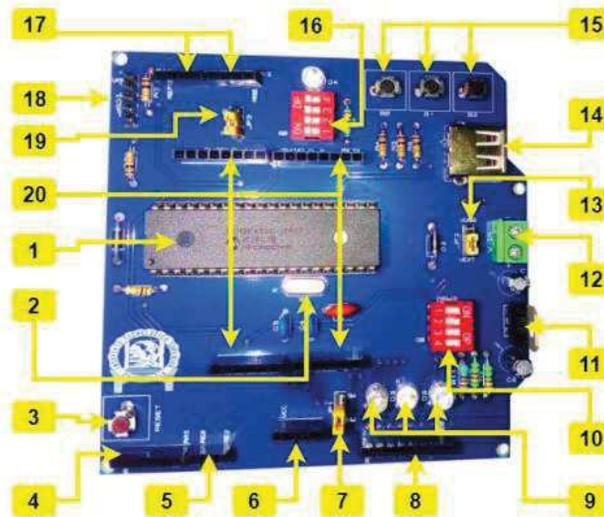


Figura 3.1 Tarjeta de desarrollo construida para el sistema de alarmas.



Los componentes de esta tarjeta se describen a continuación.

1. Microcontrolador PIC18F4550 con encapsulado DIP.
2. Oscilador de Cristal de 20 MHz.
3. Pulsador de Reset.
4. Header hembra para Puerto A, pines RA0 a RA5.
5. Header hembra para Puerto E, pines RE0 a RE2.
6. Header hembra para módulo GPS NEO6M (opcional).
7. Jumper para selección de modo de trabajo para puerto serial.
8. Header hembra para Puerto D, pines RD0 a RD7.
9. LED's de propósito general conectados a los pines RD0, RD1 y RD2 del Puerto D.5
10. Dip Switch para habilitación/inhabilitación de LED's de propósito general en los pines RD0, RD1 y RD2.
11. Circuito integrado L7805 para regulación de voltaje de entrada al circuito de la tarjeta.
12. Terminal block de dos entradas para conectar suministro de voltaje.
13. Jumper para selección de alimentación, por USB o por alimentación externa.
14. Conector USB tipo A hembra.
15. Pulsadores de propósito general conectados a los pines RB0, RB1 y RB2 del Puerto B.
16. Dip Switch para habilitación/inhabilitación de pulsadores de propósito general y LED conectados a los pines RD0, RD1, RD2 y RD3 respectivamente del Puerto B.
17. Header hembra para Puerto B, pines RB0 a RB7.
18. Header hembra para programación ICSP.
19. Jumper para selección de modo de trabajo en puerto serial.
20. Headers hembra destinados a módulos de expansión o Shields de Arduino (opcional).

Se han realizado pruebas con la tarjeta en distintos lugares, con distintos horarios y con distintas alarmas predefinidas (ver figura 3.2). Las cuales han enviado la información como se había programado. Cabe señalar que el mensaje de texto es enviado en tiempo real y a su vez genera una llama a un dispositivo móvil ya definido, aunque se le pueden proporcionar más números telefónicos para que haga la misma operación al mismo tiempo y esto a su vez se vinculen entre todos los dispositivos.

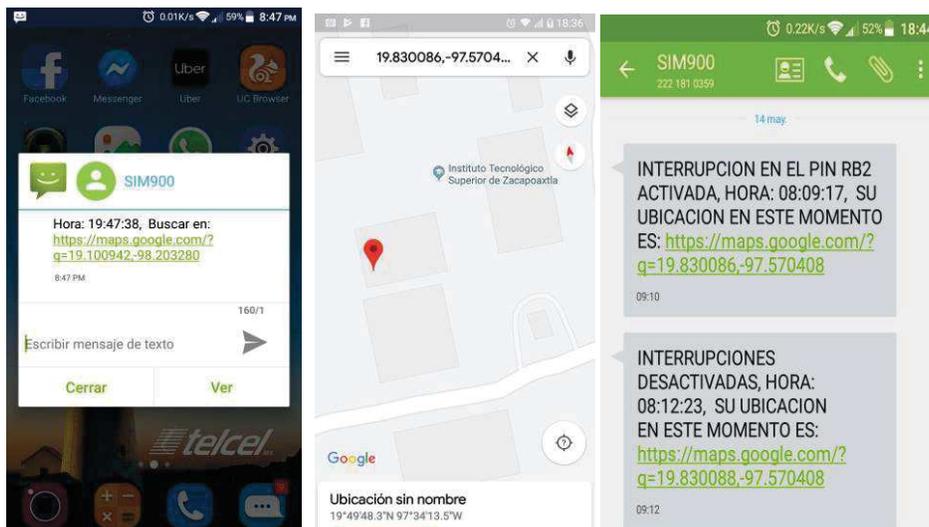


Figura 3.2 Mensaje con ubicación del dispositivo móvil en tiempo real

4. Conclusiones



En este trabajo se logró la comunicación con el microcontrolador PIC 18f4550, y un módulo GPS diseñado para arduino enviando información de la ubicación y de la alarma a un dispositivo móvil (Smartphone), sin embargo; para trabajos posteriores se pretende intercambiar por otro dispositivo GPS. Además, el diseño de la tarjeta quedó demasiado espacioso, por lo que en una segunda edición se considera reducir el tamaño, ya que por el momento se obtuvo una tarjeta de 10cm x 10cm.

Por otro lado, se puede mencionar que este diseño puede funcionar para otros objetivos, tales como seguridad personal, ya que debido al aumento de secuestros y extravío de personas se puede implementar en alguna prenda u objeto que porte la persona y colocando algún sensor para dicho fin, puede ayudar en dar una ubicación real y determinar mediante una alarma mediante un mensaje de texto que ocurre. Recordemos que todo lo que se mueve se considera como un móvil y que no necesariamente debe ser un robot.

Se pretende desarrollar más pruebas en distintos dispositivos móviles. Además de analizar los márgenes de error con respecto a la ubicación y a tiempo de respuesta.

Referencias

- [1] By AMIS in Comunicados, 2018. Consulta en: <https://www.amis.com.mx/amiswp/noticias/informes/cifras-anuales-de-robo-de-autos-asegurados-2017/>
- [2] Mora, P. *“UF1347- Instalación y configuración de los nodos de redes privadas con públicas”*, Editorial Elearning S. L, España, 2017
- [3] Huidobro, J.M, *“Radiocomunicaciones: Viajando a través de las ondas- Ciencia divulgativa”*, Creaciones Copyright, S.L, España, 2011.
- [4] Falcone, F. Huidodro, J.M. Millán, R. *“Manuales Plan Avanzada: Ciudadanía Móvil”*. Editorial Red.es y COIT, España, 2008.
- [5] García, F.J., *“Desarrollo y construcción de una tarjeta para la adquisición de datos de control de robots cooperativos”*. Facultad de Ingeniería, Laboratorio de Robótica, UNAM, México 2016.
- [6] NMEA 0183, *“The NMEA 0183 Protocol, standard defines an electrical interface and data protocol for communications between marine instrumentation”*, Estados Unidos de América, 2000.



Implementación de Sistema Mecatrónico para Simulación de Movimientos de la ATM

Cortés Ramírez Ulises¹, Castañeda Espinoza Angélica¹, Ascanio Balderas Martín², Juárez Rosete y Manuel E.¹

¹Laboratorio de Mecatrónica, Universidad Tecnológica de Huejotzingo, Santa Ana Xalmimilulco, Huejotzingo, Puebla, México
ulises@cortes.mitmx.net, angelica.castaneda@uth.edu.mx, manueleloy.juarez@uth.edu.mx

²Facultad de Estudios Superiores de Iztacala, Universidad Autónoma de México, Los Reyes Iztacala Barrio de los Árboles/Barrio de los Héroes, Tlalnepantla de Baz, México

Resumen

La articulación temporomandibular (ATM) del ser humano tiene tres funciones básicas: hablar, masticar y deglutir; y está formada por estructuras óseas, musculares y articulares, cuenta con tres grados de libertad, dos son angulares y uno lineal, El estudio de esta articulación es atendido por la profesión de Cirujano Dentista y durante el proceso de masticación entran en función los tres grados de libertad que la conforman, de lo que, uno de los problemas a resolver con este proyecto es crear un sistema mecatrónico que cuente con la capacidad de reproducir los movimientos de la ATM, esto para que estuantes de la carrera de Cirujano Dentista, dispongan de una herramienta didáctica de un modelo anatómico en el que puedan estudiar y observar sus movimientos, impactando directamente a las cátedras de Prostodoncia, Ortodoncia y Periodoncia. Por otra parte, el conocer a detalle la anatomía y fisiología de la ATM puede ayudar al estudio de tratamientos y prevención de enfermedades o disfunción de la misma.

Palabras clave: Articulación Témporomandibular, Simulador mecatrónico, Movimiento Mandibular.

1. Introducción

La estructura curricular del antiguo plan de estudios, de la carrera de Cirujano Dentista impartida en la Facultad de Estudios Superiores Iztacala otorgaba a la oclusión un abordaje exiguo, durante el último año de estudios y dado que los avances científicos y tecnológicos, otorgan un cambio de perspectiva pasando de la odontología restauradora a un nuevo enfoque preventivo. Surge como una necesidad la reestructuración del nuevo plan de estudios, que otorgue las herramientas necesarias a los alumnos para enfrentarse a las situaciones que se presentaran a lo largo de su vida profesional [1].

Esto representa no solamente un cambio en los temas a impartir, también requiere de explorar metodologías del aprendizaje que aprovechen el alto grado de interacción de los estudiantes con la tecnología de uso común tales, como teléfono, tablets y múltiples dispositivos móviles; en respuesta a esto el presente trabajo, se enfoca en diseñar e implementar un modelo anatómico que pueda reproducir los movimientos de la ATM.

Con demasiada frecuencia, en las cátedras de Prostodoncia, Ortodoncia, Periodoncia, entre otras, son enseñados conceptos diferentes y hasta contradictorios sobre oclusión. Estos conceptos deberían ser uniformados en los planes de estudio de todas las escuelas y facultades de la república



mexicana. De otra manera el estudiante se graduará con una impresión confusa e incompleta de los problemas de la oclusión, tal como sucede actualmente [2].

Los movimientos articulares de la mandíbula incluyen cuatro factores fisiológicos básicos que son: oclusión dentaria, periodonto, atm y mecanismo neuromuscular. La alteración de cualquiera uno de ellos afectara directamente a los demás; así mismo la alteración de estos elementos constitutivos alterara las funciones propias de este sistema tales como:

- Masticación
- Deglución
- Respiración
- Fonación
- Postura [2]

Cabe mencionar que este sistema posee a la articulación más compleja del cuerpo humano: la Articulación Témporomandibular. En ella tienen lugar todos los movimientos de la masticación, y ella sufrirá considerables modificaciones según su modo de locomoción y el régimen alimenticio propio de cada ser humano. Durante la masticación la mandíbula realiza lo que llamamos un ciclo masticatorio, este consiste en un movimiento tridimensional resultante de la conjunción de movimientos de apertura, cierre, lateralidad, protrusión y retrusión por eso será preciso conocer cómo se realiza cada uno de estos movimientos y cuál es la actividad muscular necesaria para que tengan lugar [2].

La implementación y diseño de herramientas didácticas de la articulación temporomandibular son muy variadas se tiene el caso como el de [3], donde se implemento una aplicación multimedia mostrada en la Figura 1, que consta de un módulo de contenido, el cual incluye los aspectos correspondientes a las características morfofuncionales de la ATM, donde se trabajan los contenidos desde el origen, y desarrollo de las estructuras que participan en su formación, así como alteraciones del desarrollo y clasificación.

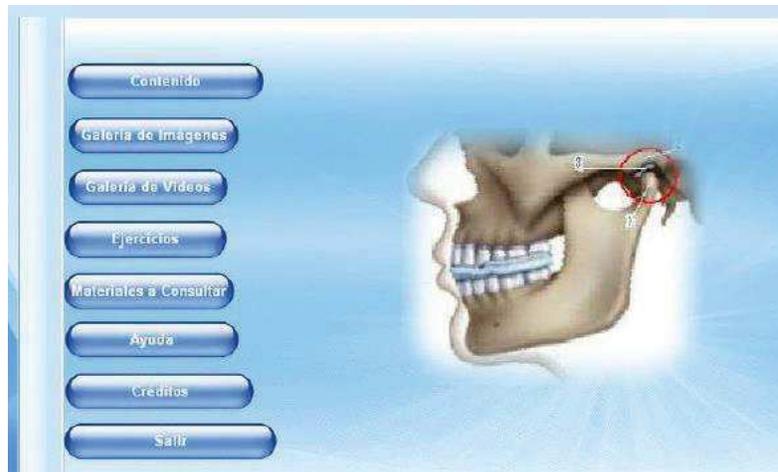


Figura 1. Herramienta multimedia de la ATM [3].

Se tiene también el caso de [4], donde se desarrolla un sistema robotizado para generar los movimientos de la ATM mostrado en la Figura 2, con el fin de contar con una herramienta para prueba de resistencia de materiales de prótesis dentales, en este trabajo los autores consideran las fuerzas ejercidas por la mandíbula y los movimientos de apertura y cierre, lateralidad y protrusiva. Y plantean una apertura de 45mm, lateralidad de 5mm y protrusiva de 5mm, con fuerzas masticatorias promedio de 200 Newtons.



Figura 2. Simulador de movimientos mandibulares [4].

En este trabajo se pretende abordar la implementación de un sistema mecatrónico en el que sea posible replicar de forma animada los movimientos de la ATM tendrá como impacto la facilitación del entendimiento y aprendizaje de los conceptos de Biomecánica, para la mayor comprensión del funcionamiento del sistema estomatognático, en el estudiante de la carrera Cirujano Dentista. Además, ofrecer una herramienta lúdica didáctica y alternativa para la enseñanza en el nuevo plan de estudios de la carrera Cirujano Dentista, con un beneficio tangible a los pacientes que se atienden en las clínicas odontológicas de la Facultad de Estudios Superiores Iztacala (FESI) de la Universidad Autónoma de México.

Como objetivo principal se pretende realizar un sistema físico que cuente con la capacidad de reproducir de manera animada los movimientos que realiza la articulación temporomandibular (ATM) del ser humano durante el proceso de masticación, esto con la finalidad de que pueda ser aprovechada como herramienta didáctica para las carreras de Cirujano Dentista de la Facultad de Estudios Superiores de Iztacala de la UNAM. En esta primera fase del trabajo se presenta los movimientos de lateralidad y protrusiva, que mecánicamente son representados con un movimiento lineal.

2. Anatomía de la Articulación Temporomandibular

Para una descripción de la morfología de la anatomía de cráneo humano se hace uso de un plano de referencia como se ilustra en la Figura 3, en el que se representa el cráneo seccionado por tres planos clave que son:

- Sagital (S)
- Frontal (F)
- y Horizontal (H)

Hay un plano básico llamado SAGITAL, que es el plano de simetría que divide al cuerpo en dos partes o mitades iguales, este también llamado plano medio, por lo que todo lo que está más cercano, se orienta a él, y es catalogado como MEDIAL. Para las referencias anatómicas, el plano SAGITAL es vertical, ya que es la posición del cuerpo de pie. El plano sagital forma con el conjunto de infinitos planos paralelos a él, el sistema de planos SAGITALES, siendo todos ellos verticales y antero-posteriores. En sentido estricto, solo hay un plano sagital: el sagital medial o plano de simetría; los otros planos del sistema, paralelos entre sí, son parasagittales.

El segundo sistema de planos referenciales es el de los FRONTALES. El cuerpo humano en su conjunto está dispuesto en un plano FRONTAL (también llamado coronal), ya que tiene unas amplias superficies aplanadas de delante a atrás, con un frente (hacia delante) y un dorso (hacia atrás), que



contrasta con lo reducido de su perfil. El plano de la frente de la cara marca la frontalidad. Por tanto, todos los planos del sistema **FRONTAL**, son paralelos al plano de la frente.

El tercer sistema fundamental de planos de referencia es el de los planos **HORIZONTALES**, transversales (perpendiculares) a los planos sagital y frontal. En la posición de sujeto de pie (con el eje corporal vertical), las secciones transversales al mismo son horizontales. Se refiere en anatomía a los cortes y secciones. Así, el término transversal se refiere a un corte “de través”, es decir, perpendicular al eje longitudinal de un elemento alargado. En definitiva, el término transversal en anatomía tiene dos acepciones de significado distinto: como una dirección que va (o como un elemento que discurre) de lado a lado, y como plano que secciona “de través”, es decir, como corte perpendicular a la longitud de ese elemento alargado (un elemento en el que predomina la dimensión longitudinal) [3].

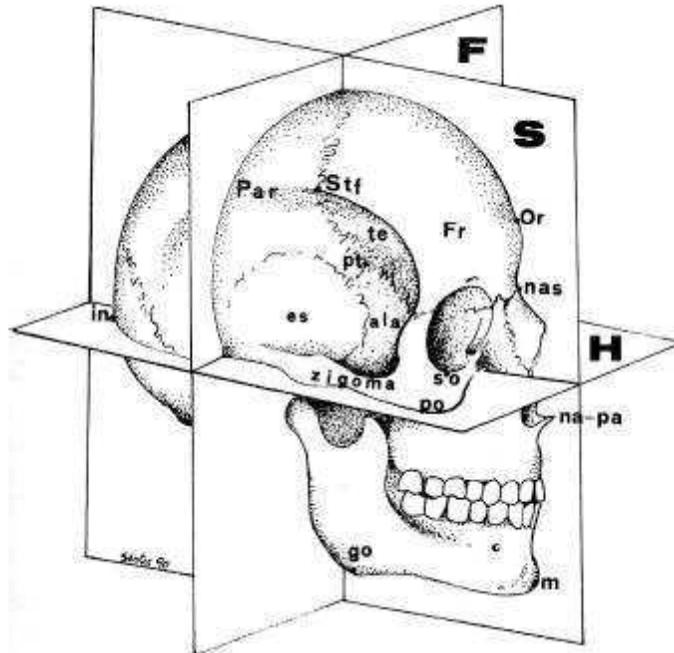


Figura 3. Plano de referencia de la anatomía del cráneo.

En [5] se reportan las medidas de la mandíbula como se ilustra en la Figura 4, denominadas como distancias antropométricas y sus valores se muestran en la Tabla 1.

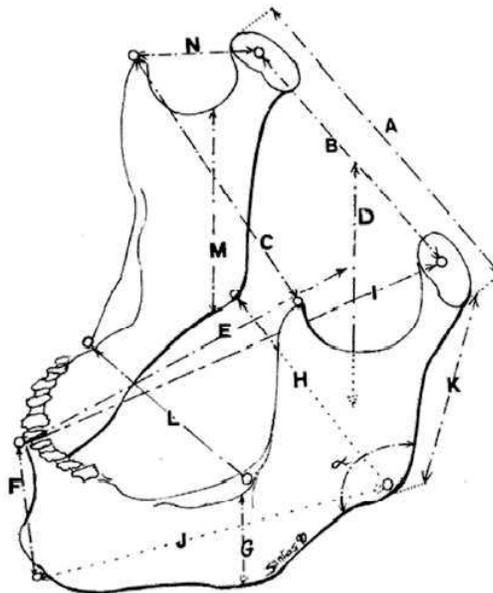


Figura 4. Medidas de la mandíbula.

- A: distancia entre los extremos laterales de ambos cóndilos (distancia intercondílea máxima).
- B: distancia entre los puntos centrales de ambos cúlmenes condíleos (distancia entre los centros geométricos de los elipsoides condíleos).
- C: distancia entre los vértices de ambas apófisis musculares o coronoides.
- D: distancia vertical entre el plano horizontal tangente a ambos cúlmenes condíleos y el plano horizontal pasante por el gonion (teniendo en cuenta que el borde caudal de la mandíbula se considera descansando sobre el plano horizontal).
- E: distancia antero-posterior entre el plano frontal pasante por el eje de bisagra y el punto alveolar anterior Inter incisivo medial (teniendo en cuenta que el borde caudal de la mandíbula, se considera descansando sobre el plano horizontal). En desdentados, el punto oclusal Inter incisivos mediales.
- F: (en desdentados): distancia entre el borde alveolar incisivo y el punto medial anterior del borde mentoniano caudal.
- F: (en dentados): distancia entre el borde oclusal incisivo y el punto medial anterior del borde mentoniano caudal.
- G: (en desdentados): distancia entre el borde alveolar molar y el borde caudal del cuerpo mandibular.
- G: (en dentados): Distancia entre la superficie oclusal molar y el borde caudal del cuerpo mandibular.
- H: distancia intergonial.
- I: distancia entre el punto condíleo culminar central y el punto alveolar anterior Inter incisivos mediales en dentados y similar en desdentados.
- J: distancia entre el gonion y el punto medial anterior del borde mentoniano caudal
- K: distancia entre el nivel culminar condíleo y el gonion.
- L: distancia entre los puntos alveolares disto-linguales del segundo molar.
- M: distancia entre el punto caudal de la escotadura sigmoidea y el borde caudal de la rama ascendente.
- N: distancia entre el punto central del culmen condíleo y el vértice de la apófisis muscular.
- O: Ángulo goniaco (α).

Tabla 1. Distancias antropométricas mandibulares [mm] [5].

Parámetros	1	2	3	4	5	6	7
A	113	103.7	116	111	104	105	110



B	96.5	85.3	98	93	86	89	97
C	85	85	95	93	90	83.5	86
D	50	55	52	43.5	49	42	46.5
E	86	84	84	83	84	91	88
F	30.5	30	27	28	26	34	28.5
G	23.5	23	23	18	21	24	22
H	84.7	84	100	93	91	87	81
I	100	98	106	97	96	106	105.7
J	77.5	79	84	75	75	82	84.5
K	59	55	54	53	58.5	54	53
L	56	53	55	ATR	57	53	61
M	46.7	40	45	51	41	37	33.4
N	30	35	33	28.4	30	37	33.4
α	125	115	122	126	122	125	125

3. Articulación Temporomandibular

Quien se encarga de generar los movimientos de la mandíbula se denomina articulación temporomandibular (ATM). Está formada por el cóndilo mandibular que se ajusta en la fosa mandibular del hueso temporal. Entre ambas superficies articulares se dispone un disco articular fibrocartilaginoso el cual divide la articulación en dos compartimientos (superior e inferior), y como medios de unión presenta la cápsula articular, ligamentos de refuerzo y periféricos.

3.1 Disco Articular

El disco articular está formado por un tejido conjuntivo fibroso y denso desprovisto de vasos sanguíneos o fibras nerviosas. Morfológicamente se describe como un lente bicóncavo, con 2 caras, 2 bordes y 2 extremidades. Su periferia es de mayor grosor (borde posterior 3-4mm y borde anterior 1-2mm) y en el centro es más delgado (1mm). La cara anterosuperior es cóncava en relación a la eminencia articular del temporal y convexa en la porción que queda en relación a la fosa mandibular, formando parte de la articulación temporomeniscal, mientras que la cara posteroinferior, es cóncava en toda su extensión y forma parte de la articulación maxilomeniscal (Ver [6]).

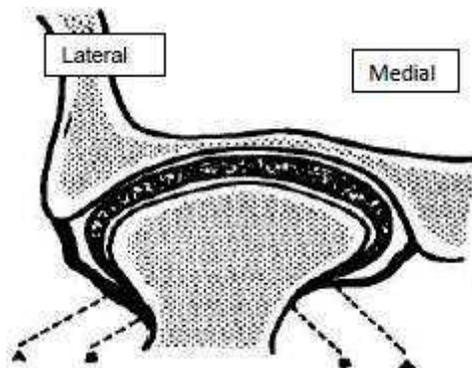


Figura 5. Ligamentos lateral y medial.

3.2 Movimientos de la Articulación Temporomandibular (ATM)



La articulación temporomandibular realiza las funciones de masticación y habla, estas funciones son generadas a partir de los movimientos de:

- **Apertura y Cierre:** En este movimiento mantiene una oclusión céntrica y una relación céntrica, por la contracción de los músculos infrahioideos. La articulación supra meniscal intervine en este movimiento que se realiza en dos tiempos: En el primer tiempo o pequeña apertura los cóndilos del maxilar inferior se encuentran paralelamente de acuerdo a su eje de rotación. El segundo tiempo o gran apertura se debe por la contracción simultanea de los pterigoideos externos, que hacen que los meniscos se deslicen por las vertientes posteriores de los cóndilos temporales [7].
- **Movimiento de Propulsión:** En este movimiento el maxilar inferior debe dirigirse hacia adelante. Por el cual intervienen los músculos geniioideos que deben tener una actividad depresora, este movimiento debe ser regulado por los músculos elevadores para evitar que el maxilar inferior descienda demasiado. Concluido este movimiento los dientes inferiores se encuentran 5 milímetros por delante de los dientes superiores [7].
- **Movimiento de retropropulsión:** El maxilar inferior debe volver a la posición de oclusión céntrica en la que estaba inicialmente. Los primeros en intervenir son los músculos depresores con ayuda de los digástricos, el trabajo en conjunto hace retornar a los cóndilos y a los meniscos a las cavidades glenoideas [7].
- **Movimiento de lateralidad Centrifuga:** El maxilar inferior debe dirigirse hacia uno de los lados laterales para que el punto mentoniano se aleje del plano sagital. En este caso el maxilar inferior debe bajar un poco para que no exista engranamiento dentario, donde el cóndilo del lado opuesto al del movimiento es traccionado por el pterigoideo externo y recorre junto con el menisco el plano de trayectoria condílea, dirigiéndose abajo, adelante y adentro. En este movimiento se requiere, que músculos maseteros y pterigoideos internos mantengan el contacto dentario, donde el punto mentoniano recorre una distancia de 5 a 15 milímetros.
- **Movimiento de lateralidad centripeta:** El cóndilo de maxilar inferior, lateralizado vuelve a la primera ubicación, de oclusión céntrica, por la contracción de las fibras horizontales del músculo temporal.

4. Implementación de Movimientos de Lateralidad, Propulsión y Retropropulsión

Basándose en la revisión de la literatura cita con anterioridad, podemos clasificar los movimientos de la mandíbula como:

- Apertura Cierre.
- Lateralidad.
- Propulsión.
- Y retropropulsión

Para estos tres movimientos se tienen tres componentes principales ilustrados en la Figura 6, uno es el cóndilo mandibular, el disco articular que sirve como cojinete de deslizamiento y la cavidad del hueso temporal.

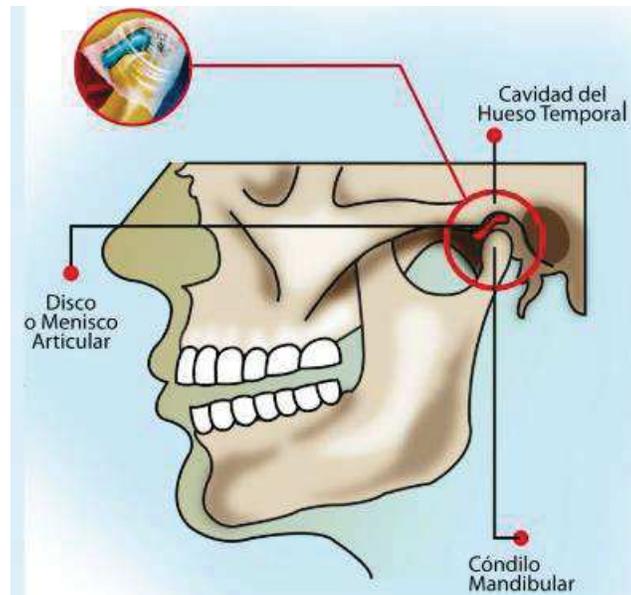


Figura 6. Cavidad del Hueso Temporal, cóndilo mandibular y disco articular [8].

Mecánicamente se puede argumentar que el movimiento de apertura y cierre está definido como un movimiento angular, que como ya se mencionó con anterioridad consta de dos fases, donde la segunda fase de gran apertura está acompañada de un movimiento de Propulsión, es decir que además del movimiento angular también se tiene un movimiento hacia adelante de la mandíbula, este movimiento angular no está contemplado en esta etapa del proyecto.

Para el caso del movimiento de propulsión consta del desplazamiento hacia adelante de los dos cóndilos mandibulares como se ejemplifica en la Figura 7, y el movimiento de retropulsión está determinado como el movimiento hacia atrás de la mandíbula como se señala en Figura 8.

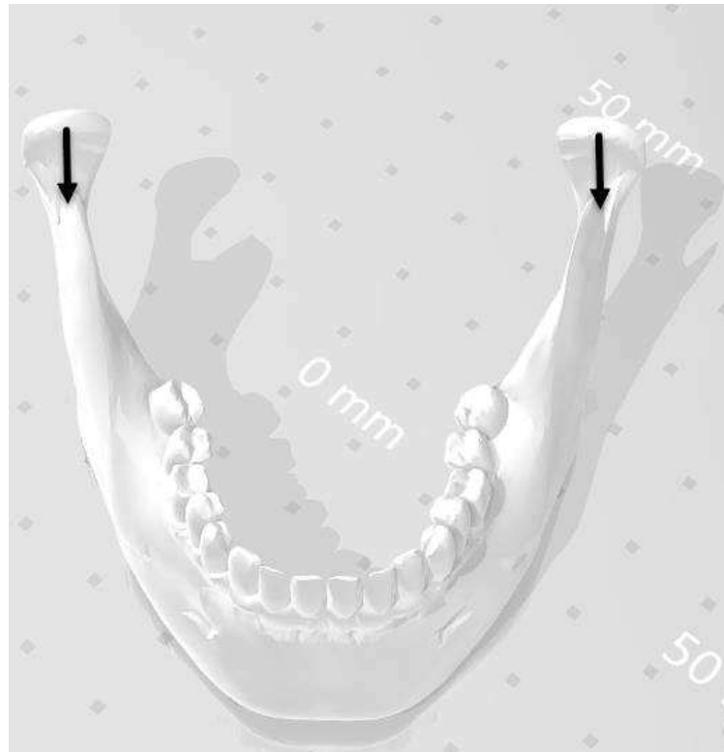


Figura 7. Movimiento de Propulsión de la mandíbula.

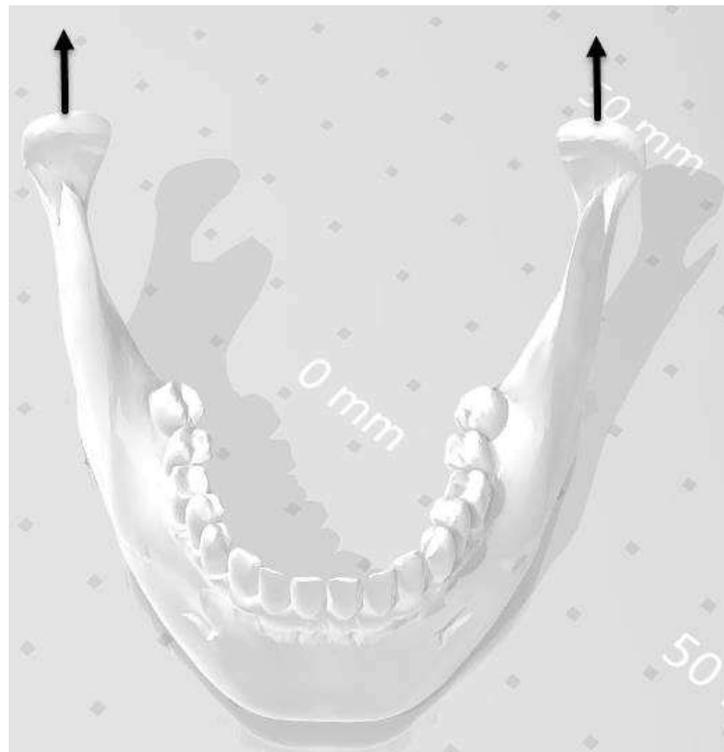


Figura 8. Movimiento de retropulsión de la mandíbula.

4.1 Implementación de articulación lineal para el movimiento de los cóndilos

Para recrear estos dos movimientos se diseñó una articulación lineal, implementada con un tornillo sin fin motorizado como se muestra en la Figura 9, la cual está conformada por:

1. Servomotor de rotación continua.
2. Base unión con servomotor
3. Tornillo sinfín y cilindro.
4. Cardan.
5. Base rotula para unión con el cóndilo mandibular

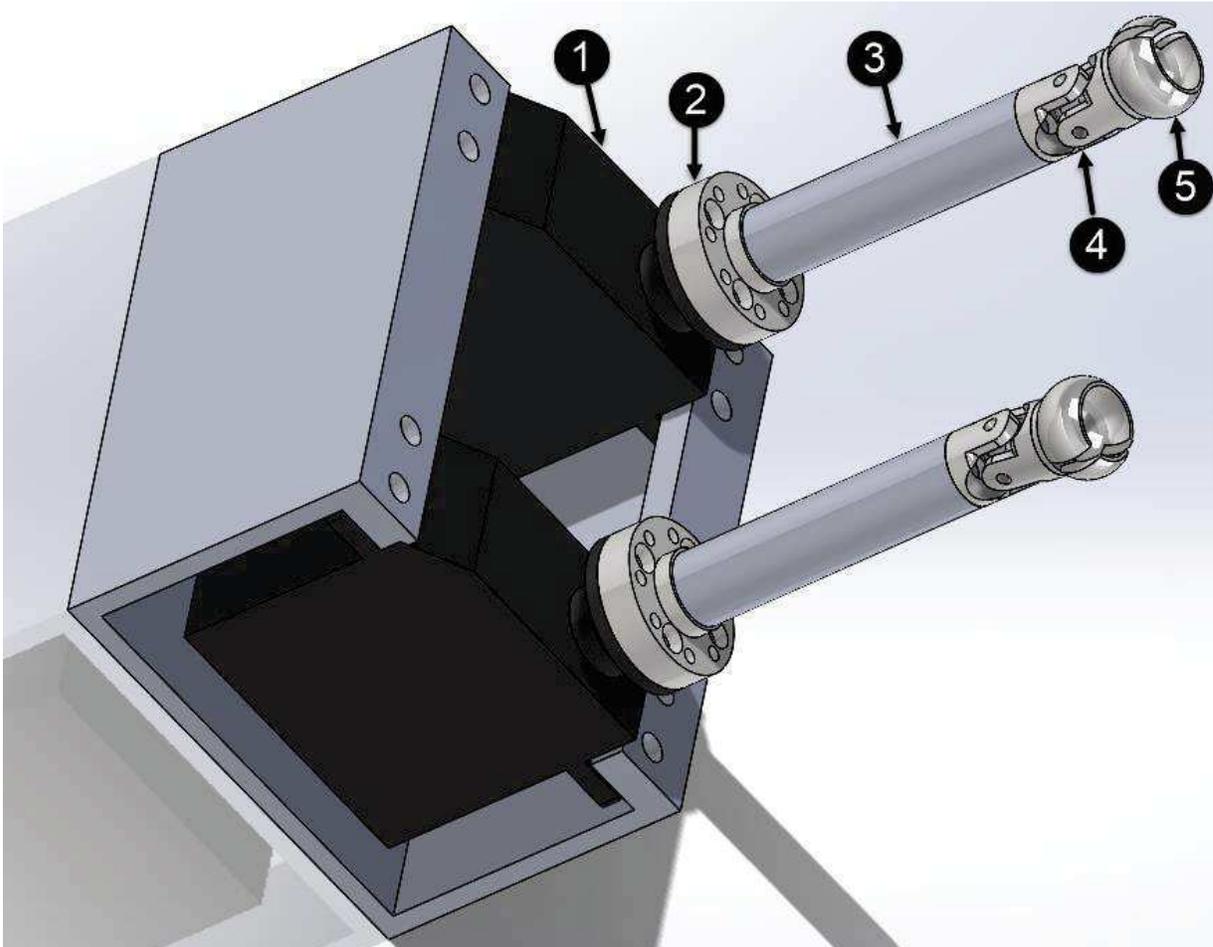


Figura 9. Articulación lineal con tornillo sinfín.

Al girar los dos motores en sentido horario se desplazarán los dos cóndilos hacia adelante, debido a que el desplazamiento del cóndilo es en la cavidad del hueso temporal y esta es curva, se tiene la implementación del cardan y la rótula, con la finalidad de mantener el movimiento lineal y el cóndilo siga la curvatura de la cavidad del temporal. Cuando los dos motores giran en sentido antihorario se generará el movimiento de los dos cóndilos hacia a tras correspondiente a la retropropulsión.

Para el caso del movimiento de lateralidad se genera con el desplazamiento lineal de uno de los cóndilo de forma independiente como se ilustra en la Figura 10 y Figura 11, para el caso del mecanismo implementado en la Figura 9, el cardan y la rótula además de mantener el movimiento lineal a través de la curvatura de la cavidad del hueso temporomandibular, cuando uno de los cóndilo esta en movimiento el que está en reposo permanecerá en su eje de rotación.

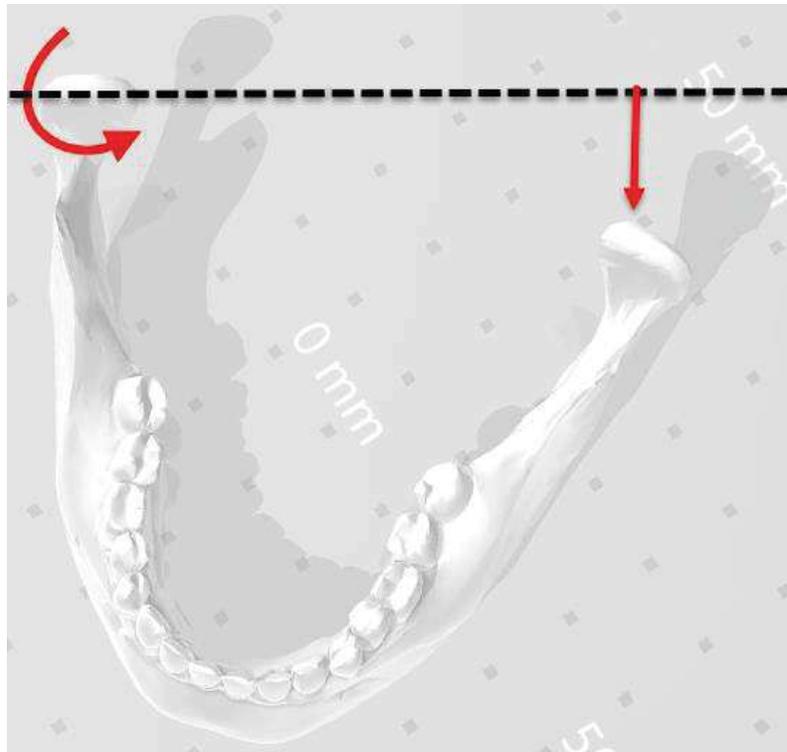


Figura 10. Movimiento de lateralidad derecha.

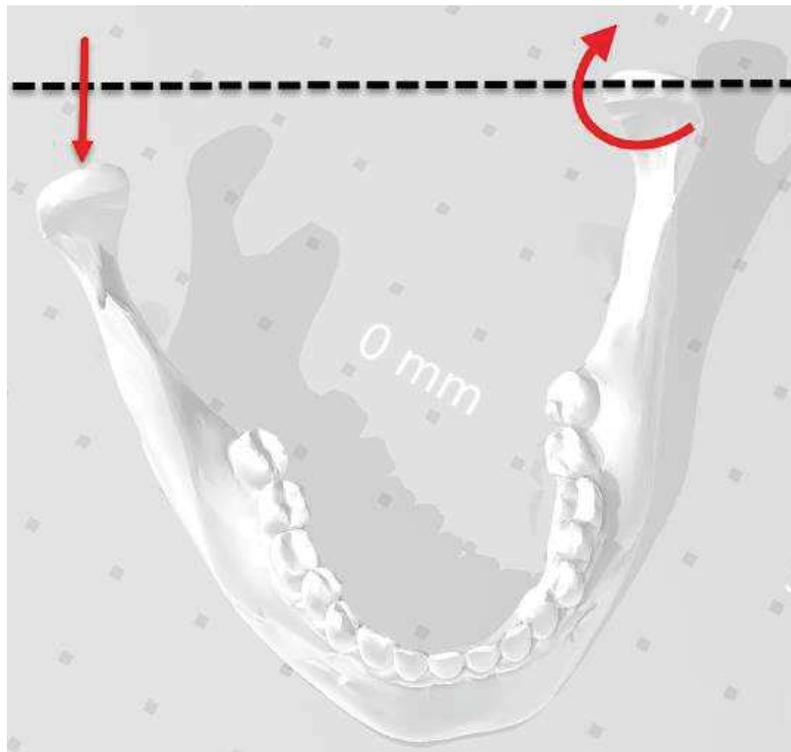


Figura 11. Movimiento de lateralidad Izquierda.



4.2 Programa para generar los movimientos de las articulaciones lineales

Como ya se mencionó previamente las articulaciones lineales se actuaron con servomotores de rotación continua ilustrado en la Figura 12, en el caso de este servomotor la referencia que se le indica es la velocidad angular en un rango porcentual de 0 a 100%, donde el 0% corresponde a la velocidad máxima en sentido anti-horario, 50% a una velocidad cero y 100% a la velocidad máxima en sentido horario, esto es lo que permite que el cóndilo pueda moverse de adelante hacia atrás.

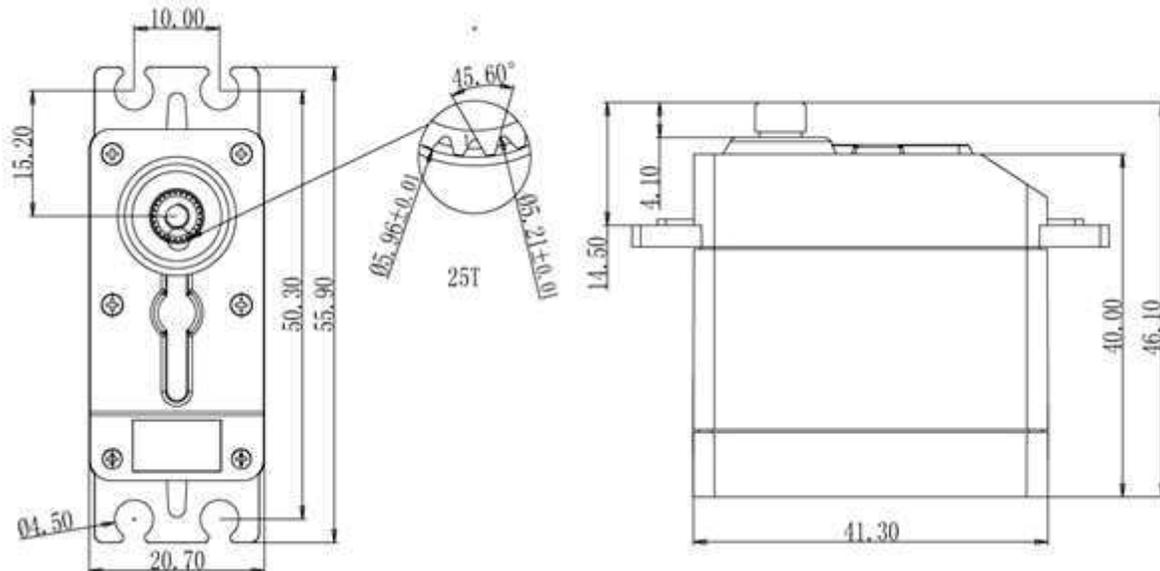


Figura 12. Servomotor de rotación continua SM-S430R-CCW.

Para manipular los servomotores de rotación continua se implementó una interfaz de usuario en el software LabVIEW mostrada en la Figura 13, y como hardware se empleó la tarjeta FRDM-KL20D50M, compatible con el lenguaje de programación ISE-LabVIEW© [9], de la que su código se muestra en la Figura 14.

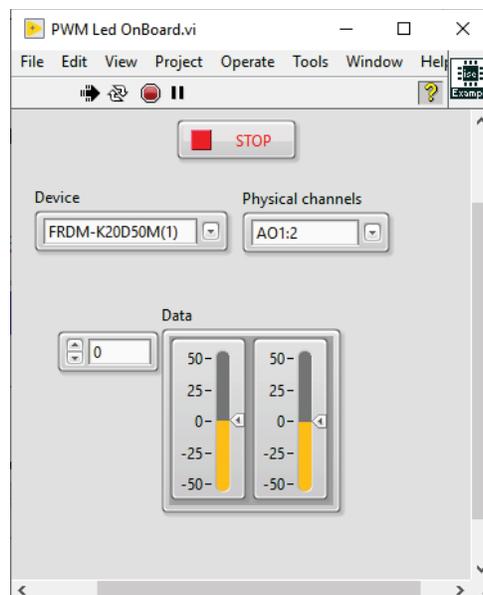


Figura 13. Interfaz de usuario para la manipulación de servomotores.

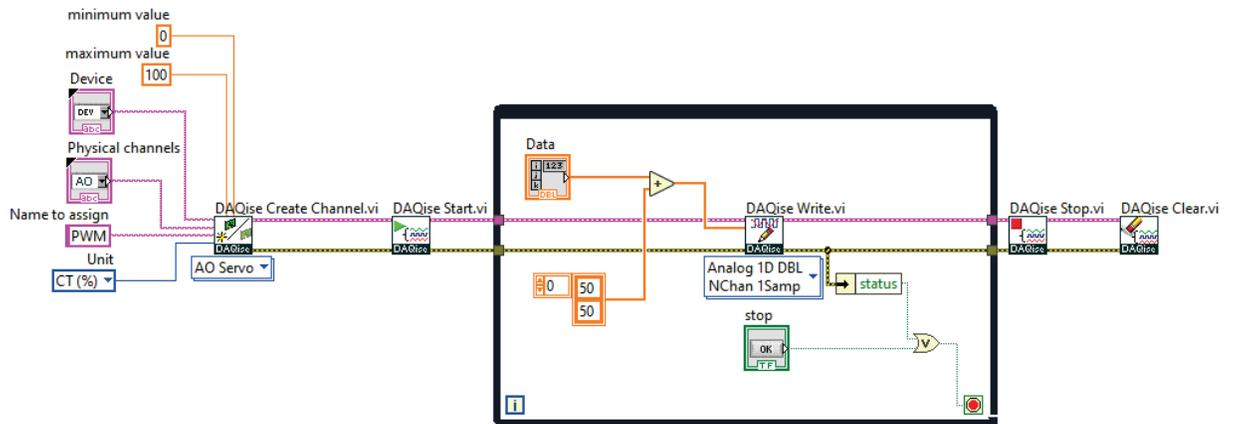


Figura 14. Código para la manipulación de Servomotores.

5. Resultados

El sistema mecatrónico tiene la capacidad de generar tres de los movimientos que genera la articulación temporomandibular que son los de lateralidad, propulsión y retropropulsión, con fines didácticos aplicado en la carrera de Cirujano Dentista, de la Figura 15, se puede observar cómo está conformado el diseño mecánico.

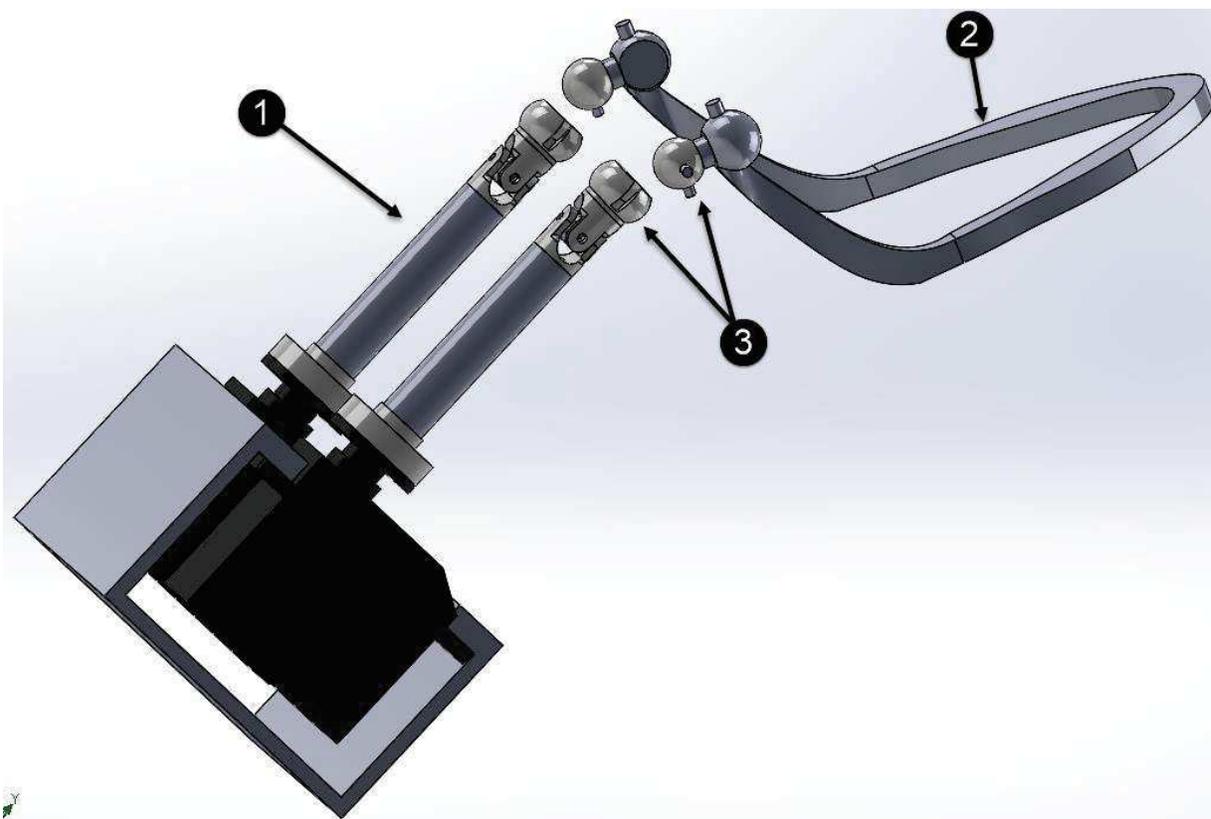


Figura 15. Sistema mecatrónico para generar los movimientos de lateralidad, propulsión y retropropulsión de la ATM.

- De la Figura 15, se tiene:
1. Articulación lineal.
 2. Rotula de cóndilo.



3. Representación de kínder de la mandíbula.

6. Conclusiones

El proyecto planteado pretende generar los movimientos que realiza la articulación temporomandibular para la función de masticación, para esto se requieren de cinco movimientos que son:

- Apertura
- Cierre.
- Lateralidad.
- Propulsión.
- Y retropropulsión

En esta fase del proyecto se tiene implementados los movimientos de lateralidad, propulsión y retropropulsión, generados a partir de dos articulaciones lineales, unidas a la mandíbula a través de un cardan y rotula, esta configuración mecánica posee el movimiento lineal de los cóndilo y además permite seguir la curvatura de la cavidad del hueso temporal, la rotula permite que cuando uno de los cóndilo se mueve linealmente el otro tenga un movimiento rotatorio sobre el eje perpendicular al plano horizontal.

Referencias

- [1] Universidad Autónoma de México, Plan de Estudios de la Carrera de Cirujano Dentista, Iztacala, México: Universidad Autónoma de México.
- [2] A. A. Lugo, Fundamentos de Oclusión, México: Instituto Politécnico Nacional, 2004.
- [3] A. Herrera Forcelledo, L. A. Lazo Herrera y D. León Medina, «Herramientas informáticas educativas sobre las características morfofuncionales de la articulación temporomandibular,» Universidad Médica Pinareña, vol. 14, n° 3, pp. 248-256, 2018.
- [4] C. Machado, J. Castañeda, P. Pineros y L. Suárez, «Desarrollo de un simulador de movimientos mandibulares para pruebas de resistencia abrasiva de biomateriales,» Actas Odontológicas, vol. 3, n° 1, pp. 55-59, 2006.
- [5] P. Moreno Cabello, Estudio de la Dinámica Mandibular Humana en un Articulador Dental Virtual Individualizable, Córdoba: Universidad de Córdoba, 2007.
- [6] P. A. García Garro, Y. S. Meneses Beltrán y G. C. Vega Ávila, Descripción Anatómica y Cinemática del Musculo Pterigoideo Lateral (MPTL), Santiago de Cali, Colombia: Universidad del Valle, 2012.
- [7] A. R. Velarde Huanca, «Fisiología de la articulación Temporomandibular,» Revista de Actualización Clínica, vol. 23, pp. 1075 - 1079, 2012.
- [8] clinicavass, «Tensión en la mandíbula: causas, efectos y ejercicios de relajación,» [En línea]. Available: <https://clinicavass.com/tension-mandibula-causas-efectos-ejercicios-relajacion/>. [Último acceso: 2019 Septiembre 20].
- [9] U. Cortés Ramírez y A. Castañeda Espinoza, «Interfaz de Sistemas Embebidos con LabVIEW (ISE-LabVIEW®),» Congreso Internacional sobre Innovación y Desarrollo Tecnológico, pp. 303-308, 2018.



Implementación de un Algoritmo de Control de Trayectoria en un Sistema de Arquitectura Abierta

Pájaro Adrián J. Domingo¹, Reyes Báez Rodolfo²,
Cid Monjaraz Jaime J.¹ y Reyes Cortés Fernando¹

¹Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla.

²Jan C. Willems Center for Systems and Control, Faculty of Science and Engineering, University of Groningen.

Resumen

En este trabajo se presenta el diseño y puesta a marcha de un sistema electrónico de arquitectura abierta para la implementación experimental de algoritmos de control de trayectoria en robots manipuladores de transmisión directa. El sistema electrónico de arquitectura abierta que se emplea en este trabajo es una tarjeta Arduino Due a la que se le realiza el acoplamiento de las señales con las que se obtiene la posición actual del robot adaptando un sencillo circuito electrónico para este propósito, también se realiza un ajuste a los convertidores digital-analógico de la tarjeta empleada a través de la adaptación de un circuito electrónico que permita enviar de forma correcta la señal de control al robot manipulador de transmisión directa. Para mostrar que las adaptaciones y ajustes realizados a la tarjeta Arduino Due funcionan de forma correcta se realiza la implementación del algoritmo de control de trayectoria backstepping en un robot manipulador de transmisión directa de dos grados de libertad.

Palabras clave: Arduino, Backstepping, Control de trayectoria, Robot manipulador.

1. Introducción

En la actualidad existe una gran variedad de robots manipuladores que pueden realizar numerosas tareas complejas, sin embargo, la mayoría de estos posee una arquitectura cerrada, es decir, que no es accesible su sistema mecánico o electrónico lo que impide desarrollar investigación en temas como el diseño de controles de posición, de trayectoria, de fuerza, entre otros temas de interés para la comunidad científica, por eso resulta necesario contar con sistemas que tengan una arquitectura abierta que permita desarrollar investigación.

A los robots antropomórficos que por diseño poseen eslabones rígidos unidos por articulaciones giratorias compuestas por servomotores de transmisión directa se les denomina robot de transmisión directa como en (Reyes, 2011). Transmisión directa (Direct drive) significa que el servomotor funciona como una fuente ideal de par o torque, no tiene pérdidas de energía como sucede en motores provistos de engranaje. Direct drive elimina el sistema de engranaje tradicional, por lo que el rotor del motor está directamente acoplado al eslabón, además, con esta tecnología se elimina el juego mecánico y se reduce significativamente el fenómeno de fricción.

Un servomotor está compuesto por motor eléctrico, sensor de posición y el amplificador electrónico (electronic driver), los servomotores de transmisión directa tienen tres modos de operación: posición, velocidad y par. El modo posición permite mover al motor a una posición preestablecida con la desventaja de que en este modo de operación no es posible desplazar objetos o realizar control de fuerza. El modo velocidad permite controlar el movimiento del robot a una velocidad deseada, tanto en modo posición y modo velocidad no es posible implementar otro tipo de controladores conservando las características de sistemas con arquitectura cerrada, por otra parte, el modo de operación por par



posee características de sistemas con arquitectura abierta, haciendo posible implementar algoritmos de control, por ejemplo, el control PD+ que se presenta en (Kelly, 2006) el algoritmo de control backstepping que se presenta en (Fossen, 1997) o el algoritmo Slotine-Li (Slotine, 1987) entre otros.

Además de la arquitectura mecánica del robot manipulador es necesario contar con arquitectura electrónica que permita realizar la implementación de algoritmos de control. Arduino es un sistema electrónico de arquitectura abierta con una amplia gama de aplicaciones en robótica y mecatrónica como se presenta en (Cortés, 2015), en robots móviles como en (Martínez Guzmán Sonia, 2017) y diversos autores. El Arduino Due, figura 3, está basado en un microcontrolador ARM Cortex-M3. Tiene 54 pines de entrada-salida digital (de los cuales 12 se pueden usar como salidas PWM), 12 entradas analógicas, 4 UART (puertos seriales), un reloj de 84 MHz, conexión compatible con USB, 2 DAC (convertidor digital a analógico), 2 TWI, un conector de alimentación, un conector SPI, un botón de reinicio y un botón de borrado, además de otras características que pueden ser consultadas más a detalle en (Arduino.cc, 2019). Sin embargo, para su correcto funcionamiento requiere de sistemas que permitan acoplar las señales de entrada a la tarjeta y al mismo tiempo ajustar las salidas que por lo general son las señales que se requiere para el control de un sistema físico.

En este trabajo se realiza la implementación de un algoritmo de control de trayectoria para robots manipuladores en una plataforma experimental de arquitectura abierta tanto en la estructura mecánica del robot y en la electrónica necesaria para la implementación del control, por lo que se realiza el acoplamiento de las señales que proporcionan la posición actual del robot a las entradas de la tarjeta Arduino Due, además, se realiza el ajuste de los DAC que están disponibles en la tarjeta, con lo que se controlan los dos grados de libertad del robot manipulador. Para validar que el sistema completo funciona adecuadamente y que puede ser utilizado a nivel investigación y enseñanza de control de robots manipuladores se implementa el control de trayectoria backstepping.

2. CASO DE ESTUDIO

Considere un robot manipulador con dos eslabones rígidos unidos por articulaciones giratorias compuestas por servomotores de transmisión directa como el que se presenta en la figura 1.



Figura 1. Robot manipulador de transmisión directa de dos grados de libertad



además, considere un encoder incremental y un servoamplificador, a continuación, se presentan los modelos de los servomotores que conforman la estructura del robot manipulador:

Tabla 1. Servomotores que conforman la estructura del robot manipulador

Eslabón	Modelo	Resolución	Torque
1	DMG3-1050A-115B	10260 $\frac{\text{pulsos}}{\text{rev}}$	50 Nm
2	DMG3-1004C-115B	10260 $\frac{\text{pulsos}}{\text{rev}}$	4 Nm

Los servomotores están configurados para operar en modo par, en esta configuración se requiere enviar al servomotor una señal que puede estar en el rango de -5V a 5V para su correcto funcionamiento. Los encoder proporcionan dos señales A y B de 5V las cuales están desfazadas 90° como se muestra en la figura 2.

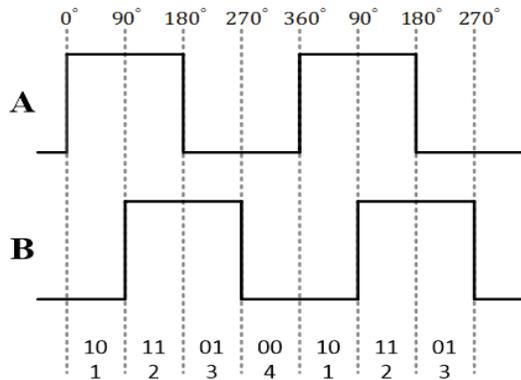


Figura 2. Forma de las señales del encoder.

La implementación de algoritmos de control de trayectoria requiere del conocimiento del modelo dinámico del robot manipulador de transmisión directa de 2 grados de libertad, el cual está expresado como:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + g \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \quad (1)$$

donde

$$M_{11} = m_1 l_{c1}^2 + m_2 l_{c2}^2 + m_2 l_1^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2, \quad (2)$$

$$M_{12} = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2, \quad (3)$$

$$M_{21} = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2, \quad (4)$$

$$M_{22} = m_2 l_{c2}^2 + I_2, \quad (5)$$

$$C_{11} = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2, \quad (6)$$

$$C_{12} = -m_2 l_1 l_{c2} \sin(q_2) (\dot{q}_1 + \dot{q}_2), \quad (7)$$



$$C_{21} = m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1, \quad (8)$$

$$C_{22} = 0, \quad (9)$$

$$g_1 = l_{c1} m_1 \text{sen}(q_1) + m_2 l_1 \text{sen}(q_1) + m_2 l_{c2} \text{sen}(q_1 + q_2), \quad (10)$$

$$g_2 = l_{c2} m_2 \text{sen}(q_1 + q_2). \quad (11)$$

A continuación, se presenta en tabla 2, los valores de los parámetros del robot manipulador de transmisión directa de dos grados de libertad, para los componentes de la matriz de inercia, la matriz de Coriolis y el vector de pares gravitacionales del modelo dinámico de la ecuación (6).

Tabla 2. Parámetros del robot manipulador de dos grados de libertad

Eslabón	Notación	Unidades
Eslabón 1	m_1	6.94 kg
	l_1	0.35 m
	l_{c1}	0.057 m
	I_1	$1.422 \text{ Nm} \frac{\text{s}^2}{\text{rad}}$
Eslabón 2	m_2	1.18 kg
	l_2	0.35 m
	l_{c2}	0.041 m
	I_2	$0.097 \text{ Nm} \frac{\text{s}^2}{\text{rad}}$
	g	$9.81 \frac{\text{Nm}}{\text{s}^2}$

2.1 Tarjeta Arduino Due

El Arduino Due funciona a 3.3V por lo que para la implementación de algoritmos de control se debe considerar que voltajes superiores podrían dañar la tarjeta, también se debe considerar que las salidas de los DAC's incorporados no son ideales ya que estos presentan un offset de aproximadamente 0.500mV por lo que es necesario realizar una etapa de ajuste a 0V, además estos DAC's solo proporcionan voltaje positivo haciendo necesaria la etapa electrónica para la obtención de voltajes negativos y voltajes positivos.

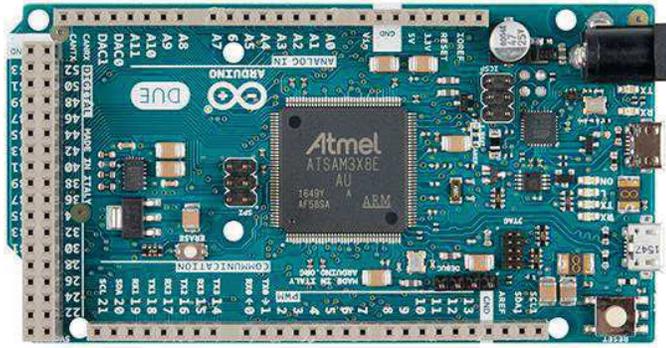


Figura 3. Tarjeta Arduino Due.

3. PLANTEAMIENTO DEL PROBLEMA DE CONTROL DE TRAYECTORIA

Considere un robot manipulador compuesto de n eslabones rígidos unidos a través de articulaciones giratorias con servomotores de transmisión directa y el modelo dinámico que se presenta en la siguiente ecuación:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \quad (12)$$

donde $M(q) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, $C(q, \dot{q})\dot{q} \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrífugas y de Coriolis, $g(q) \in \mathbb{R}^n$ es el vector de pares gravitacionales y $\tau \in \mathbb{R}^n$ es el vector de pares aplicados. Los vectores $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denotan la posición, velocidad y aceleración articular respectivamente. La representación en espacio de estados del modelo dinámico de la ecuación (12) toma la siguiente forma:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1}[\tau - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix}. \quad (13)$$

Algunas propiedades de interés que se presentan en (Kelly, 2006) del modelo dinámico de la ecuación (12) para diseño de controladores y análisis de estabilidad por Lyapunov se muestran a continuación:

Propiedad 1. La matriz de inercia $M(q) \in \mathbb{R}^{n \times n}$ es simétrica $M(q) = M(q)^T$ y definida positiva $M(q) > 0$.

Propiedad 2. La matriz de fuerzas centrípetas y de Coriolis $C(q, \dot{q})\dot{q}$ y la derivada con respecto al tiempo de la matriz de inercia $\dot{M}(q)$ satisfacen:

$$x^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] x = 0 \quad \forall x, \quad (14)$$

es decir, la matriz $\dot{M}(q) - 2C(q, \dot{q})$ es antisimétrica.

El problema de control de trayectoria es formulado en los siguientes términos: Considere el modelo dinámico de la ecuación (12) y una trayectoria de posición deseada, la cual se considera que es suave, continua y al menos dos veces diferenciable. El problema de seguimiento de trayectoria consiste en diseñar una ley de control τ , de tal forma que el vector de posición q siga con precisión a la trayectoria deseada q_d , es decir,

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{q}(t) \\ \tilde{\dot{q}}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2n}, \quad (15)$$



donde $\tilde{q} \in \mathbb{R}^n$ representa el error de posición y se define como $\tilde{q} := q - q_d$ y el error de velocidad $\tilde{\dot{q}} \in \mathbb{R}^n$ se define como $\tilde{\dot{q}} := \dot{q} - \dot{q}_d$.

4. ACOPLAMIENTO Y AJUSTE DE SEÑALES

En esta sección se presenta los ajustes realizados a la tarjeta Arduino Due para poder realizar la implementación de algoritmos de control de trayectoria en un robot manipulador de transmisión directa de dos grados de libertad, primero se presenta la electrónica que permite acoplar las salidas de los encoder's a la tarjeta Arduino Due y después se presenta la electrónica que permite ajustar y amplificar la salida de los convertidores digital-analógico.

4.1 Acoplamiento de los encoders

Para realizar una correcta lectura de la posición actual del robot manipulador a través de la tarjeta Arduino Due se requiere que la señal se conserve lo más posible sin ruido, es decir, libre de señales no deseadas que alteren la señal del encoder para una correcta lectura. En la figura 4, se presentan las señales que salen del encoder antes de la etapa de acoplamiento.



Figura 4. Señales antes del circuito de acoplamiento.

Para ajustar la señal de salida de los encoder se realiza la implementación de un circuito basado en un transistor tipo npn el cual es operado en modo saturación, en el colector del transistor se coloca el voltaje de referencia en este caso corresponde a los 3.3 V que se requieren a la entrada del Arduino Due, en la base del transistor se inserta la señal del encoder lo que nos permitirá tener en el emisor la señal del encoder acondicionada para la correcta lectura del encoder, el circuito se muestra en la figura 5.

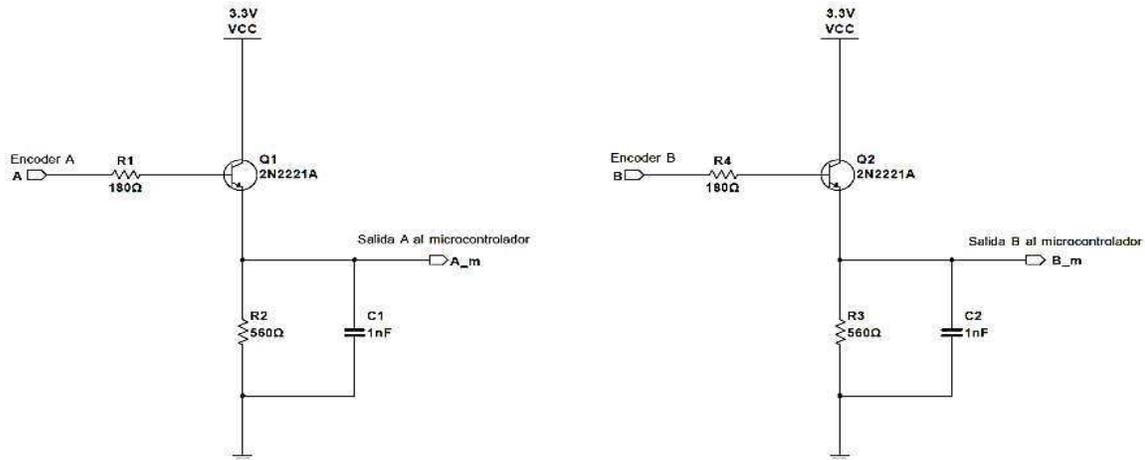


Figura 5. Circuito de acoplamiento del encoder del robot a la tarjeta arduino due.

Después de la etapa de acoplamiento con el circuito propuesto se obtienen las siguientes señales a la salida del encoder como se muestra en la figura 6.

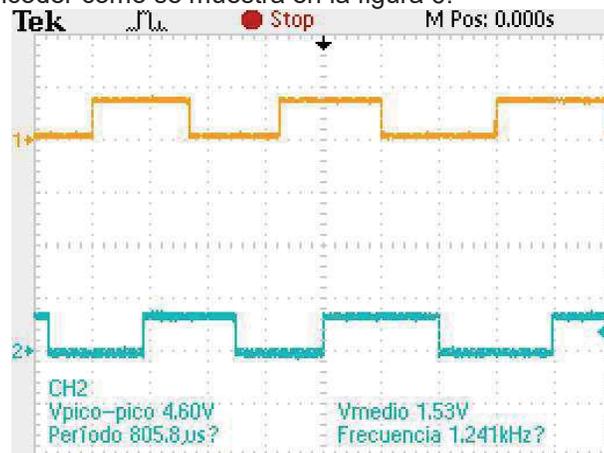


Figura 6. Señales después del circuito de acoplamiento

4.2 Adaptación de los convertidores digital-analógico de la tarjeta Arduino Due

Los convertidores digital-analógico con los que cuenta la tarjeta Arduino Due pueden ser configurados para operar con resoluciones de 10 o 12 bits, por conveniencia en este trabajo se configuran para operar a 12 bits con lo que el voltaje de salida puede tomar 4096 valores distintos. Considerando las necesidades de adaptación para los DAC del Arduino Due se realiza un circuito electrónico que adapte los DAC's del Arduino a los requerimientos necesarios y se presenta en dos partes.

En la primera etapa se realiza el ajuste del offset del DAC interno del Arduino DUE, a través del diseño de un divisor de voltaje con el cual se pueda obtener 500mV los cuales se emplean para ajustar el offset de los DAC's del Arduino DUE, los valores de las resistencias se obtienen haciendo uso de la siguiente ecuación:



$$V_S = \frac{R_2 \times V_E}{R_1 + R_2}. \quad (16)$$

La salida del divisor de voltaje se conecta a un amplificador operacional configurado como seguidor esto con la finalidad de desacoplar impedancias entre el divisor de voltaje y el resto del circuito, a continuación la salida del amplificador seguidor se conecta a la terminal negativa del siguiente amplificador operacional que esta configurado como restador inversor el cual se encarga de realizar la resta del offset que tiene el DAC de arduino DUE, los valores de las resistencias para esta parte se obtienen como se muestra en la siguiente ecuación:

$$V_{S1} = V_{ADAC} \left[\frac{(R_3 + R_1)R_4}{(R_4 + R_2)R_1} \right] - V_S \left(\frac{R_3}{R_1} \right). \quad (17)$$

En la figura 7, se muestra el circuito de la primera etapa.

En la segunda etapa, el voltaje obtenido a la salida del amplificador operacional U4B de la figura 7, es de aproximadamente 2.35V, por lo que es necesario implementar un siguiente amplificador operacional que nos permita ajustar a 5V, esto se realiza con el amplificador operacional U4C el cual tiene un arreglo de resistencias R8 y R7 para obtener una ganancia de 2.15 con la que podremos ajustar al voltaje que se requiere esto a través de la siguiente ecuación:

$$V_{S2} = V_{S1} \left(1 + \frac{R_7}{R_8} \right). \quad (18)$$

El amplificador operacional U4D nos permite ajustar el voltaje del DAC de tal forma que nos permita obtener un voltaje que este entre -5V y 5V, como se muestra en la figura 8, con lo que es posible que los servomotores del robot manipulador puedan trabajar en ambos sentidos de giro.

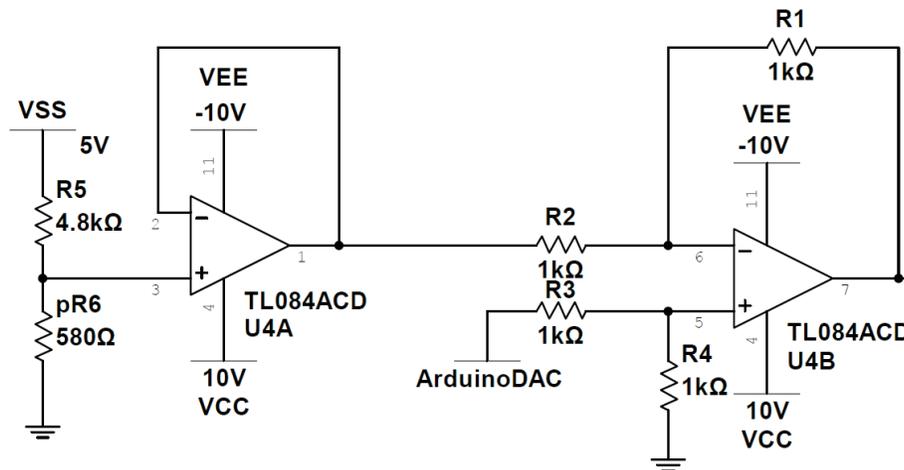


Figura 7. Ajuste de offset del Arduino Due.

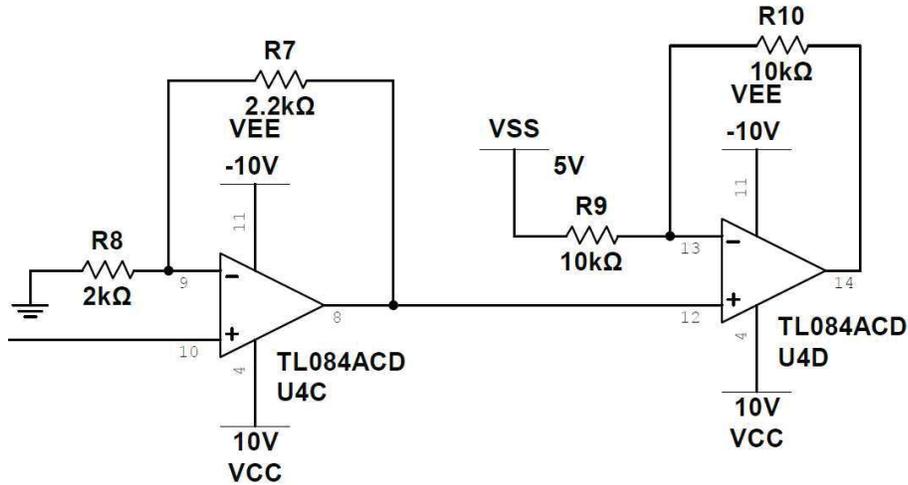


Figura 8. Amplificación de voltaje del DAC de Arduino Due.

Se deben realizar dos circuitos, uno para cada DAC del Arduino Due. Para verificar el correcto funcionamiento del circuito se le realiza un barrido de voltaje, esto con la finalidad de observar la respuesta y asegurar que su comportamiento es lineal. El barrido de voltaje se realiza enviando al DAC del Arduino Due valores en intervalos de 50 con lo que se obtienen 82 muestras de valores analógicos en el rango de -5V a 5V, con un multímetro se toman los valores analógicos obtenidos de cada muestra, este procedimiento se realiza a ambos circuitos con lo que se obtienen las siguientes graficas:

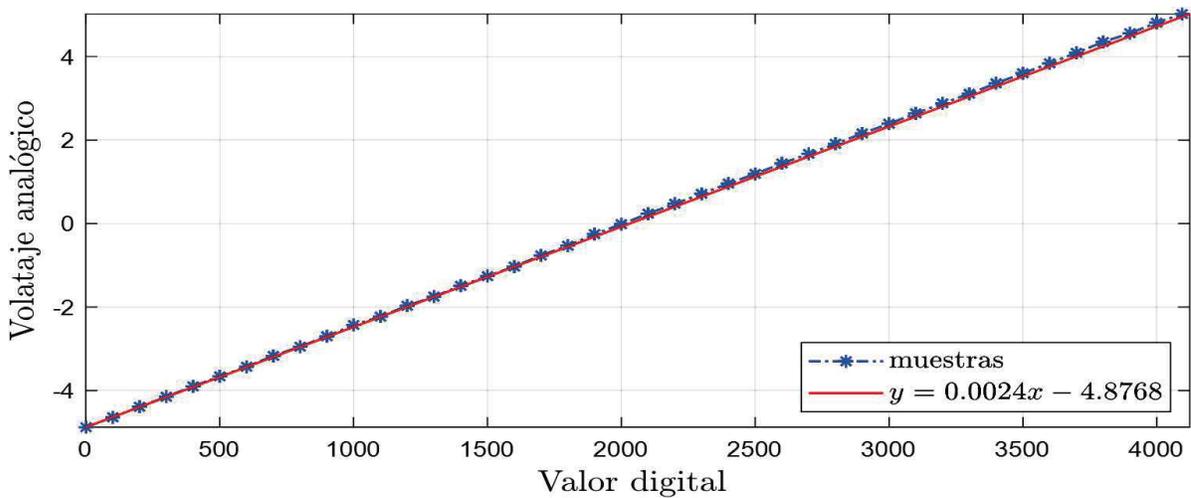


Figura 9. Respuesta del DAC1 de Arduino.Due

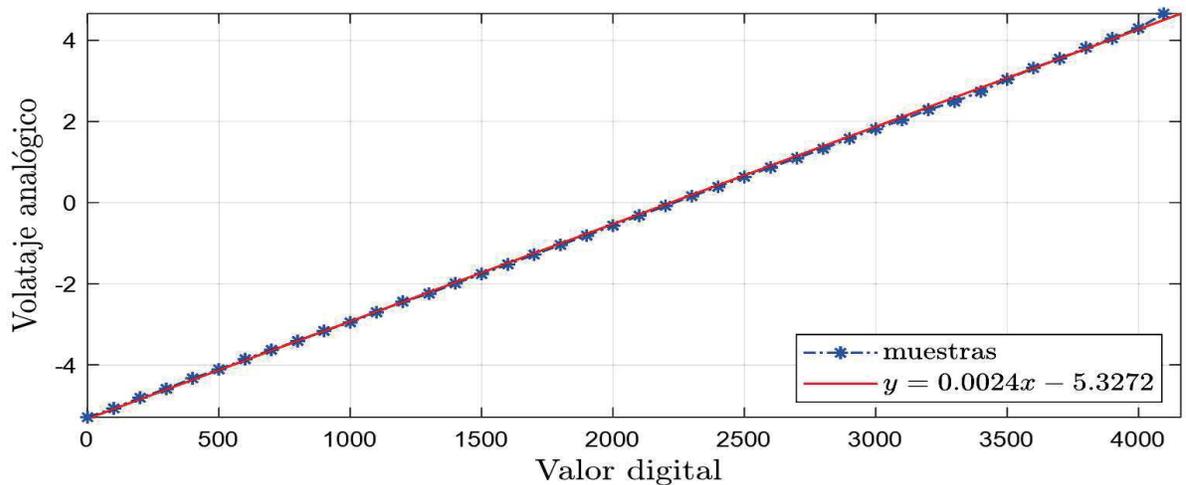


Figura 10. Respuesta del DAC2 de Arduino.Due

En la figura 9, se presenta la respuesta obtenida del circuito implementado al DAC1 del Arduino due, se observa en la línea azul que su comportamiento es lineal, por lo que se realiza la obtención de la ecuación que caracteriza a la recta del DAC1 ya que con esta se realiza el envío de valores con lo que se realiza el control del robot manipulador de 2 grados de libertad, de la misma forma se observa un comportamiento semejante para el DAC2 del Arduino DUE con el circuito implementado por lo que también se obtiene la ecuación de la recta que caracteriza su comportamiento.

5. IMPLEMENTACIÓN Y EVALUACIÓN EXPERIMENTAL

En esta sección se realiza la implementación experimental de un algoritmo de control de trayectoria en el robot manipulador de transmisión directa de dos grados de libertad, en la figura 11, se presenta el diagrama general del sistema, donde se observa la lectura de posición actual del robot y el envío de la señal de control τ , también se presenta la adquisición de datos, la cual se realiza mediante un cable USB conectado del Arduino Due a una computadora en la que se realizan las gráficas con los datos obtenidos de la tarjeta Arduino Due.

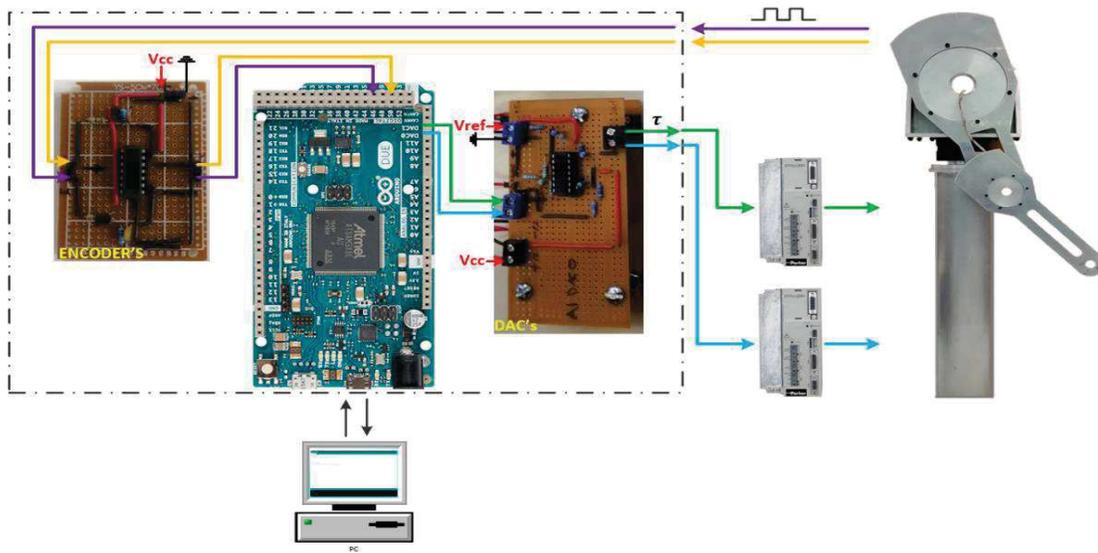


Figura 11. Diagrama general de la planta experimental.

5.1 Trayectoria deseada

Para la implementación del algoritmo de control de trayectoria considere una trayectoria deseada q_d , la cual consiste en un círculo ubicado en el cuarto cuadrante del plano cartesiano y la ecuación que lo describe como se muestra a continuación:

$$x_d = x_c + r \operatorname{sen}(t), \quad (19)$$

$$y_d = -y_c + r \operatorname{cos}(t), \quad (20)$$

donde x_d y y_d representan a la trayectoria deseada en coordenadas cartesianas, r es el radio del círculo con longitud $r = 0.2m$, el centro del círculo en el plano cartesiano se ubica en $x_c = 0.3m$ y $y_c = 0.3m$. Para el diseño e implementación se requiere la primera derivada de las ecuaciones (19) y (20), por lo que se tienen las siguientes expresiones:

$$\dot{x}_d = r \operatorname{cos}(t), \quad (21)$$

$$\dot{y}_d = -r \operatorname{sen}(t). \quad (22)$$

La segunda derivada de las ecuaciones (19) y (20) queda como se muestra enseguida:

$$\ddot{x}_d = -r \operatorname{sen}(t), \quad (23)$$

$$\ddot{y}_d = -r \operatorname{cos}(t). \quad (24)$$

Debido a que es necesario conocer las posiciones deseadas en coordenadas articulares el procedimiento para obtenerlas se realiza a partir de la cinemática inversa del robot manipulador de 2 grados de libertad asociada al cuarto cuadrante como se realiza en (Kelly, 2006), entonces se tiene:



$$q_{d1} = \frac{\pi}{2} + \tan^{-1}\left(\frac{y_d}{x_d}\right) - \tan^{-1}\left(\frac{l_2 \sin(q_{d2})}{l_1 + l_2 \cos(q_{d2})}\right), \quad (25)$$

$$q_{d2} = \left(\frac{\cos^{-1}(x_d^2 + y_d^2 - l_1^2 - l_2^2)}{2l_1 l_2}\right). \quad (26)$$

5.2 Algoritmo de control de trayectoria

Considere el algoritmo de control de trayectoria backstepping que se presenta en (Fossen, 1997) el cual esta dado en la siguiente ecuación:

$$\tau = M(q) \ddot{q}_r + C(q, \dot{q})\dot{q}_r + g(q) - K_v s - K_p \tilde{q}, \quad (27)$$

donde $K_v \in \mathbb{R}^{n \times n}$ y $K_p \in \mathbb{R}^{n \times n}$ son matrices diagonales de ganancias definidas positivas y q_r es la referencia virtual de velocidad y se define como

$$q_r := -q_d - \Lambda \tilde{q}, \quad (28)$$

donde $\Lambda \in \mathbb{R}^{n \times n}$ es una matriz diagonal de ganancias definida positiva y s es la entrada virtual de control que se define como

$$s := \dot{\tilde{q}} + \Lambda \tilde{q}. \quad (29)$$

Las ganancias que se emplean para la implementación del algoritmo de control backstepping se presentan en la siguiente tabla

Tabla 3. Ganancias para el algoritmo de control backstepping.

Ganancias	Valor
K_{p1}	128
K_{p2}	28
K_{v1}	64
K_{v2}	14
Λ_1	1.5
Λ_2	2

La implementación del algoritmo de control backstepping en el ambiente de programación de Arduino (IDE) se realiza siguiendo el diagrama de flujo de la figura 12.

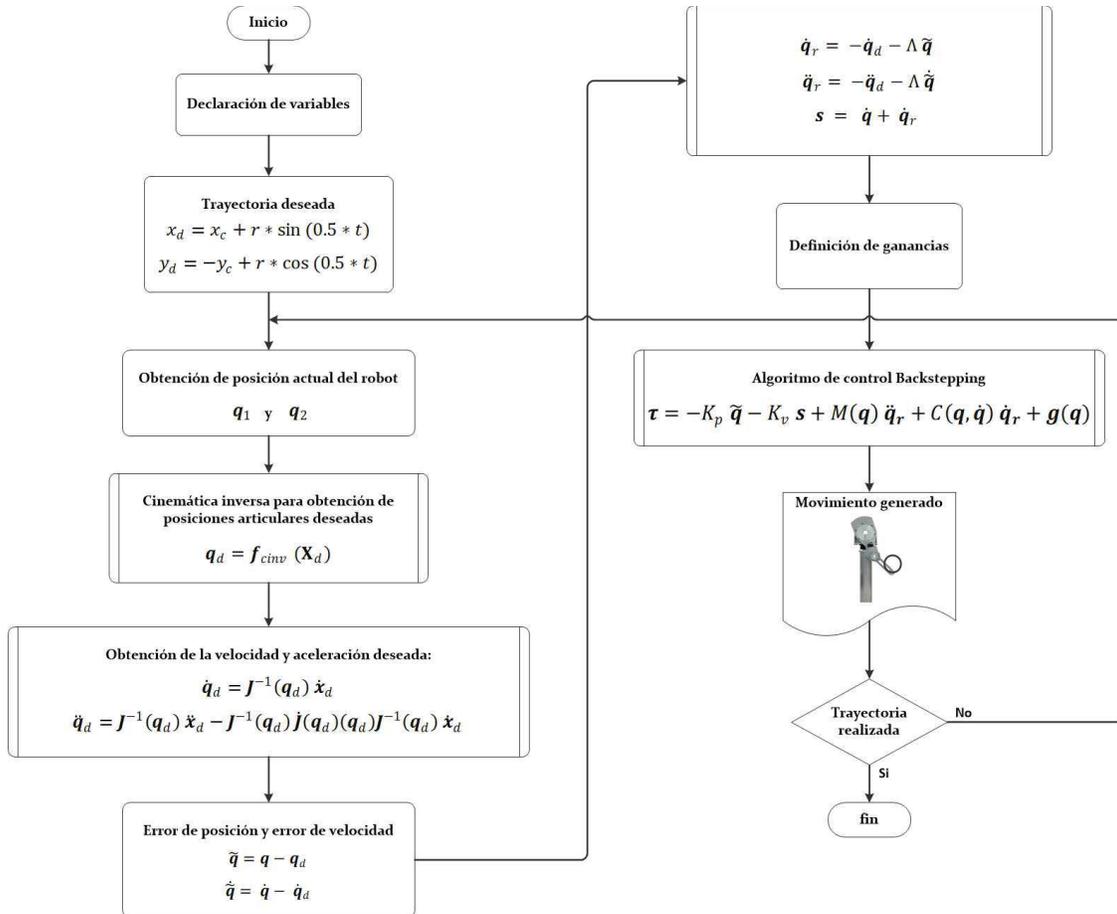


Figura 12. Diagrama de flujo para la implementación del algoritmo de control backstepping.

6. Resultados

En esta sección se presenta los resultados obtenidos de la implementación del algoritmo de control de trayectoria backstepping en la planta experimental de arquitectura abierta para robots manipuladores de transmisión directa de dos grados de libertad propuesta.

Para mostrar que las etapas de adquisición de posición actual del robot a través de la tarjeta Arduino Due con el circuito electrónico de la figura 5 y el envío de la señal de control τ mediante el uso de los DAC's de la tarjeta Arduino Due con la etapa de ajuste y amplificación electrónica de la figura 7 y la figura 8 operan de forma correcta en configuración como se muestra en el diagrama general del sistema de la figura 11, con la trayectoria deseada en el plano cartesiano que se propone y se muestra en las ecuaciones (19) y (20), y el algoritmo de control backstepping de la ecuación (27) y las ganancias propuestas de la tabla 3, se presentan las siguientes gráficas de trayectoria deseada, posiciones articulares y errores de posición y torques empleados.

En la figura 13, se muestra la gráfica de trayectoria deseada en línea azul y la trayectoria realizada por el robot manipulador de transmisión directa de dos grados de libertad en la línea roja, se observa que con las ganancias propuestas para realizar la trayectoria deseada el robot manipulador no alcanza a la posición deseada $q_d(t)$ teniendo un error de posición en su punto máximo de



aproximadamente 5 grados. Aunque la trayectoria deseada no fue alcanzada por el robot manipulador de transmisión directa no significa que para el sistema sea difícil alcanzar a la posición deseada, lo que significa es que se necesita realizar una correcta estrategia matemática de sintonización de ganancias con las que a través del empleo de cotas máximas y mínimas en pares máximos permitidos así como las cotas en velocidad máxima permitida por los servo actuadores que influyen directamente en la estructura matemática de los algoritmos de control de trayectoria se obtendría una mejora en los resultados obtenidos ya que en este caso la elección de las ganancias empleadas se realizó de forma empírica.

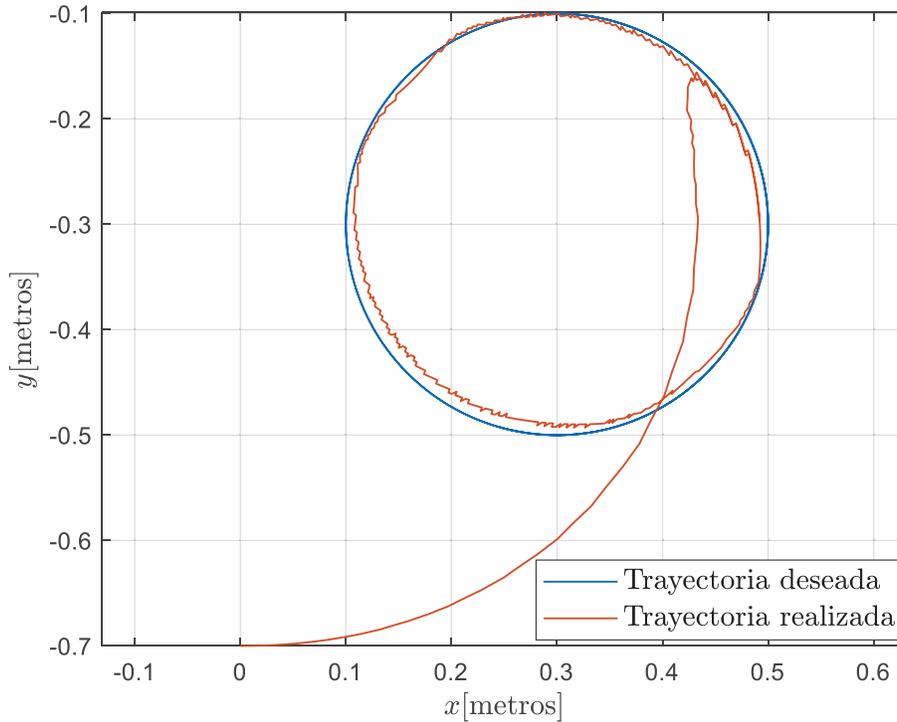


Figura 13. Gráfica comparativa, trayectoria realizada y trayectoria deseada.

En la figura 14, se muestra primero la gráfica de las posiciones articulares que realiza el robot manipulador de transmisión directa con las que se obtiene la trayectoria deseada haciendo uso de la cinemática directa del robot manipulador como se muestra en las siguientes ecuaciones

$$x_r = 0.35 * \text{sen}(q_1) + 0.35 * \text{sen}(q_1 + q_2), \quad (30)$$

$$y_r = -0.35 * \text{cos}(q_1) - 0.35 * \text{cos}(q_1 + q_2), \quad (31)$$

con las que es posible visualizar en el cuarto cuadrante del plano cartesiano la trayectoria realizada por el robot manipulador, después, se presenta la gráfica de los errores de posición para el eslabón 1 (q_1) y el eslabón 2 (q_2), donde se visualiza lo que ya se había evidenciado en la gráfica de la figura 13, que el robot manipulador tiene un considerable error de posición, también se puede visualizar que al robot manipulador le toma 2 segundos en aproximarse a las posiciones deseadas partiendo desde su posición de reposo la cual está ubicado sobre el eje negativo y del plano cartesiano.

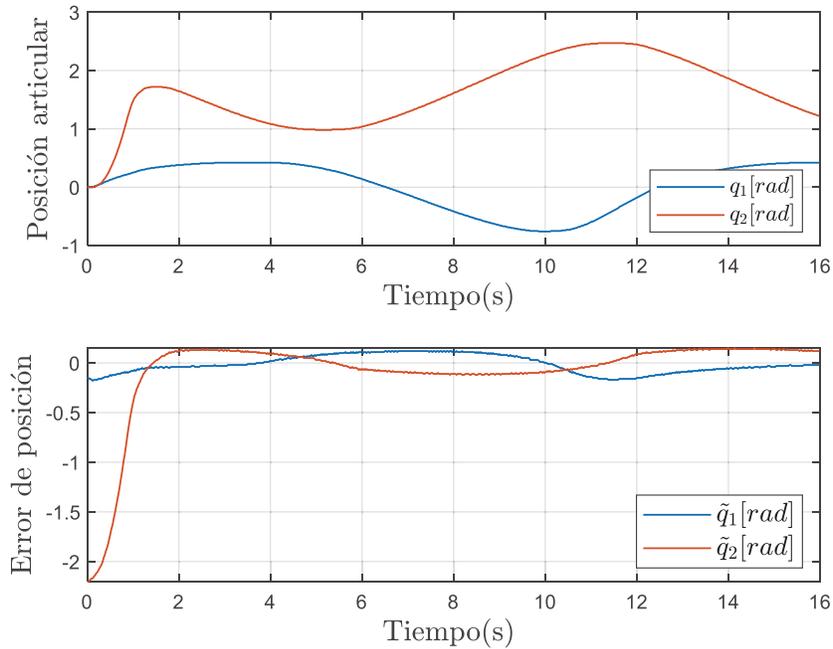


Figura 14. a) Posiciones articulares. b) Error de posición.

En la figura 15, se presentan las gráficas de torques o pares que se envían a los servomotores del robot manipulador como señal de control, en la primera parte se presenta el torque empleado para el servomotor de la articulación 1 se observa que el torque máximo no sobrepasa los 50 Nm permitidos por el servomotor, de igual forma se observa la gráfica de torque para el servomotor de la articulación 2, aunque el torque máximo que puede proporcionar el servomotor es de apenas 4 Nm es suficiente para realizar la trayectoria.

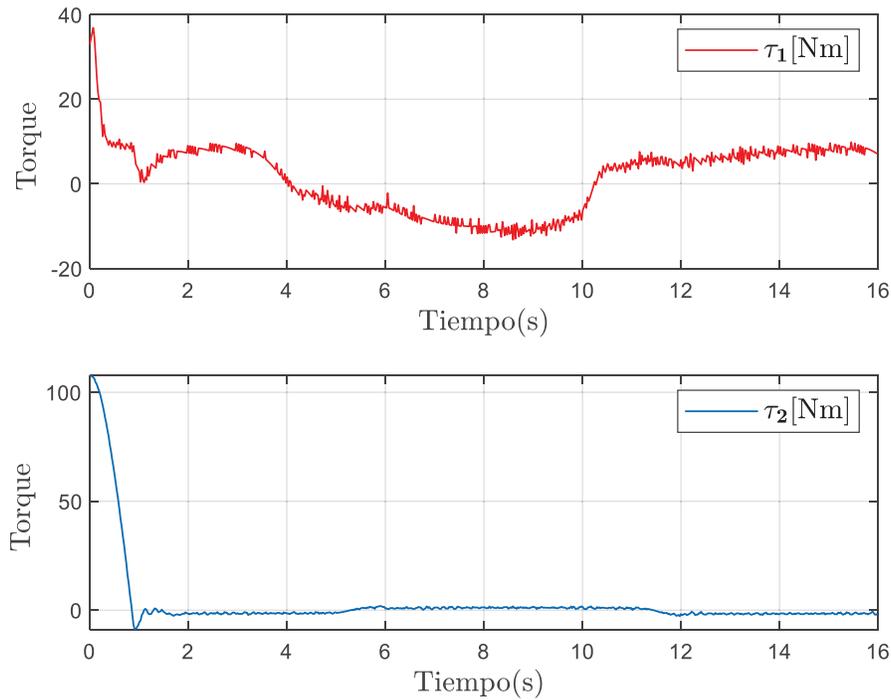


Figura 15. Torques empleados para realizar la trayectoria.



7. Conclusiones

En este trabajo se realizó la implementación de un algoritmo de control de trayectoria en un sistema de arquitectura abierta, se le denomina de esa forma ya que los servomotores que conforman la estructura mecánica del robot manipulador pueden ser configurados para operar como una fuente de par lo que facilita la implementación de algoritmos de control de trayectoria y otros tipos de controladores, además, se hizo uso de la conocida plataforma Arduino, famosa por su arquitectura abierta que permite a diseñadores, estudiantes e investigadores diseñar diversas aplicaciones mediante sus librerías o modificaciones y adaptaciones que es posible realizar a su arquitectura electrónica con lo que se abre la posibilidad de realizar diseños a bajo costo.

En este trabajo se hizo enfoque a utilizar solo dos grados de libertad de los tres disponibles en el robot manipulador de transmisión directa ya que el Arduino Due cuenta con solo dos DAC's, esto no significa que no sea posible implementar el tercer grado de libertad, es posible agregando el un convertidor digital-analógico que se pueda adaptar a la arquitectura Arduino.

Aunque en este trabajo se hizo uso del Arduino Due, actualmente existen tarjetas con mayores capacidades, por ejemplo, el Arduino mkr vidor 4000 el cual incluye dentro de su estructura un FPGA con 16000 elementos lógicos con un reloj que puede ser configurado a 150MHz, el FPGA integrado también puede ser utilizado para operaciones DSP de alta velocidad en procesamiento de audio y video.
y todas las capacidades que incluye la arquitectura Arduino.

El control de trayectoria en robots manipuladores es un tema que ha sido estudiado por diversos investigadores y centros de investigación por lo que hoy en día sigue siendo de interés contar con sistemas que permitan implementar de forma correcta y robusta estos controladores.

Referencias

- [1] Arduino.cc. (15 del 08 del 2019). *Arduino Due*. Obtenido de Arduino Due: <https://store.arduino.cc/usa/due>
- [2] Cortés, F. R. (2015). *Arduino: aplicaciones en robótica, mecatrónica e ingenierías*. Alfaomega.
- [3] Fossen, T. I. (1997). Nonlinear vectorial backstepping design for global exponential tracking of marine vessels in the presence of actuator dynamics. *Proceedings of the 36th IEEE Conference on Decision and Control*, 4237-4242.
- [4] Kelly, R. a. (2006). *Control of robot manipulators in joint space*. Springer Science & Business Media.
- [5] Martínez Guzmán Sonia, C. D. (2017). Aplicación de Arduino para Controlar un Robot Móvil Seguidor de Línea en la Materia de Inteligencia Artificial. En J. E. José Emilio Vargas Soto, *Modelado y Control de Sistemas Mecatrónicos* (págs. 156-166). Asociación Mexicana de Mecatrónica A.C.
- [6] Reyes, F. (2011). *Robótica-Control de robots manipuladores*. Alfaomega grupo editor.
- [7] Slotine, J.-J. E. (1987). On the adaptive control of robot manipulators. *The international journal of robotics research*, 49-59.



Análisis de la Presencia de Actuadores en Sistemas de Amortiguamiento Activo con Controladores en Tiempo-Finito

Rosales Martinez Antonio, Martínez Martínez Jesús, García Trinidad Enrique y Juárez Carrilo César

Tecnológico de Estudios Superiores de Huixquilucan
Departamento de Mecatrónica
Barrio El Río, S/N, La Magdalena Chichicaspa,
Huixquilucan, Estado de México C.P. 52773
jarm@ieee.org

Resumen

El sistema de amortiguamiento en un vehículo es un elemento indispensable para proveer comodidad y seguridad de conducción. Los tipos de sistemas de amortiguamiento existentes son amortiguamiento pasivo, amortiguamiento semi-activo y amortiguamiento activo. De los tres tipos, el amortiguamiento activo resalta por que ofrece un amortiguamiento actuado que garantiza niveles de comodidad de conducción excepcionales. El sistema de control es una parte esencial de los sistemas de amortiguamiento activo, específicamente, algoritmos de control con convergencia en tiempo-finito han sido estudiados, dando como resultado un amortiguamiento rápido y robusto ante perturbaciones y variaciones paramétricas. Sin embargo, la dinámica del actuador del sistema de amortiguamiento activo se omite en el diseño del algoritmo de control. Particularmente, los algoritmos de control en tiempo finito se diseñan considerando el grado relativo del sistema, si no se conoce el grado relativo del sistema o éste es mayor que el considerado, las cualidades de los algoritmos con convergencia en tiempo finito se degradan. En este trabajo, se estudia el efecto de la dinámica del actuador en el diseño de algoritmos con convergencia en tiempo finito. El estudio se lleva a cabo usando herramientas de análisis en frecuencia, como la función descriptiva y la ecuación de balance armónico. El análisis muestra que la dinámica del actuador incrementa el grado relativo del sistema, lo cual resulta en un deterioro del desempeño del algoritmo de control con convergencia en tiempo finito. Los resultados presentados se verifican mediante simulaciones por computadora.

Palabras clave: Sistemas mecatrónicos, Sistemas de control, Amortiguamiento activo

1. Introducción

Durante las últimas dos décadas, los sistemas de control en el sector automotriz se han convertido en un factor crucial en los procesos de innovación. Para garantizar el mejoramiento de requerimientos relacionados al consumo de combustible, emisiones, seguridad y comodidad de conducción, se requiere aplicar sistemas de control [1]. Un componente crucial del vehículo es el sistema de amortiguamiento, debido a que éste se utiliza para garantizar la comodidad y seguridad de conducción. Para la comodidad de conducción, el sistema de amortiguamiento aísla a los ocupantes de la superficie donde se desplaza el vehículo. También, el sistema de amortiguamiento asegura que exista una superficie de contacto entre la llanta y el camino, lo cual garantiza el control(manejo) del vehículo [2].

Existen tres tipos de sistemas de amortiguamiento: los pasivos, semiactivos y los activos. Los pasivos son los sistemas de amortiguamiento convencionales que se componen de un resorte y un



amortiguador. Los semiactivos y activos son sistemas mecatrónicos que modulan el amortiguamiento mediante la integración de actuadores controlados [3]. Los semiactivos emplean amortiguadores con coeficiente de amortiguamiento variable mientras los activos son capaces de proveer fuerzas de amortiguamiento externas que son generadas por actuadores hidráulicos, neumáticos y eléctricos, regulados por algoritmos de control. En un sistema activo, el actuador se conecta en paralelo con un sistema de amortiguamiento pasivo, es decir, se ubica entre la carrocería y el eje de la rueda, con el fin de entregar o disipar energía del sistema [4]. Entonces, el actuador se encarga de controlar la altitud del vehículo con el objetivo de garantizar la comodidad de manejo a pesar de las condiciones del camino.

El algoritmo de control que se encarga de manipular al sistema de amortiguamiento activo es un elemento indispensable. Se debe garantizar un buen control de altitud del vehículo para proveer la comodidad requerida a pesar de las incertidumbres y perturbaciones que se presenten en el sistema. Los algoritmos de control con convergencia en tiempo finito han sido ampliamente adoptados en diversas aplicaciones, incluyendo los sistemas de amortiguamiento activo. La bondad de dichos algoritmos es que si convergen al valor requerido en un tiempo acotado, lo cual es una gran ventaja sobre muchos de los algoritmos de control que presentan convergencia asintótica [5,7]. Además, que presentan insensibilidad a las perturbaciones externas acotadas.

La metodología más común para diseñar algoritmos de control con convergencia en tiempo finito es mediante la identificación del grado relativo [13]. Gran parte de los controladores en tiempo finito aplicados a los sistemas de amortiguamiento activo usan dicha metodología [5,7,8]. Si se obtiene el grado relativo del sistema, alguna de la familia de controladores para diversos grados relativos se puede emplear y asegurar convergencia en tiempo finito y robustez [13].

Sin embargo, existen dinámicas no modeladas que incrementan el grado relativo del sistema, lo cual deteriora el comportamiento de los algoritmos de convergencia en tiempo finito [15]. Particularmente, en el diseño de algoritmos de control en tiempo finito para sistemas de amortiguamiento activo, la dinámica del actuador se omite [5,7,8,9,11]. Entonces, de manera teórica los controladores presentan condiciones ideales de desempeño como convergencia exacta a cero en tiempo finito, pero en el sistema real donde los actuadores están presentes es difícil lograr dicho comportamiento ideal.

Gran parte de los trabajos de control de sistemas de amortiguamiento activo mediante algoritmos de tiempo finito, no analizan ni consideran la dinámica de los actuadores, lo cual es crucial para determinar el comportamiento real del sistema de control. Por tal motivo, en este trabajo se presenta un análisis de la presencia de actuadores en sistemas de suspensión activa controlados por algoritmos en tiempo finito. Durante el análisis se muestra que la presencia de la dinámica del actuador deteriora el desempeño de los controladores, pues surgen oscilaciones de amplitud y frecuencia finita. El análisis se lleva a cabo mediante la aplicación de la Función Descriptiva y la ecuación de balance armónico. Los resultados que se presentan están centrados en el uso de un algoritmo llamado Super-Twisting [17], sin embargo, la metodología de análisis se puede usar para una variedad de algoritmos con convergencia en tiempo finito.

La estructura del trabajo es la siguiente: la sección 2 contiene la descripción y el modelo del sistema de amortiguamiento activo; en la sección 3 se plantea el problema que se aborda considerando el modelo presentado en la sección 2; las herramientas preliminares de análisis en frecuencia de algoritmos en tiempo finito se muestran en la sección 4; la sección 5 contiene el análisis de la presencia de actuadores en sistemas de amortiguamiento activo, mientras que en la sección 6 se valida el análisis de la sección 5 mediante simulaciones; finalmente se presentan las conclusiones del trabajo.

2. Descripción del sistema

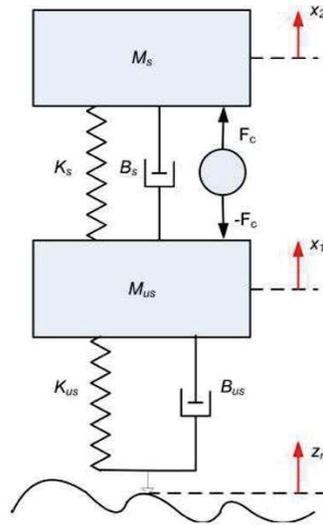


Figura 1. Diagrama de amortiguamiento activo

Si las cuatro ruedas del vehículo tienen una dinámica independiente, se considera el sistema de amortiguamiento activo mostrado en la figura 1, donde se observa que represente como un sistema nada resorte amortiguador con dos masas. El sistema se compone de dos entradas, F_c es la entrada de control y z_r es una entrada que equivale a la superficie del camino. Las variables x_s y x_u representan el desplazamiento de la llanta y el desplazamiento de la carrocería. La masa M_s representa la masa del vehículo y M_{us} es la masa equivalente a la llanta y el eje de la rueda. El modelo dinámico se compone de dos ecuaciones,

$$\ddot{x}_s = -g - \frac{F_c}{M_{us}} - \frac{(B_s + B_{us})\dot{x}_s}{M_{us}} + \frac{B_s \dot{x}_u}{M_{us}} + \frac{B_{us} \dot{z}_r}{M_{us}} - \frac{(K_{us} + K_s)x_s}{M_{us}} + \frac{K_s x_u}{M_{us}} + \frac{z_r K_{us}}{M_{us}} \quad (1)$$

$$\ddot{x}_u = -g - \frac{F_c}{M_s} - \frac{B_s \dot{x}_s}{M_s} - \frac{B_s \dot{x}_u}{M_{us}} + \frac{K_s x_s}{M_{us}} - \frac{K_s x_u}{M_s} \quad (2)$$

donde g es la fuerza de gravedad, K_{us} , K_s son constantes de los resortes y B_s , B_{us} son coeficientes de amortiguamiento.

Definiendo las variables de estado $\dot{x}_s = \dot{z}_{us}$, $\ddot{x}_s = \ddot{z}_{us}$, $\dot{x}_u = \dot{z}_s$, $\ddot{x}_u = \ddot{z}_s$ y sustituyéndolas en la ecuación (1) y (2), se tiene el siguiente modelo

$$M_{us} \ddot{z}_{us} = -B_s \dot{z}_{us} - B_{us} \dot{z}_{us} - F_c + B_s \dot{z}_s + B_{us} \dot{z}_r - (z_{us} - z_s)K_s - (z_{us} - z_r)K_{us} \quad (3)$$

$$M_s \ddot{z}_s = B_s \dot{z}_{us} + F_c - B_s \dot{z}_s - (z_s - z_{us})K_s \quad (4)$$

Definiendo otras variables de estado como $x_1 = z_s - z_{us}$, $x_2 = \dot{z}_s$, $x_3 = z_{us} - z_r$, $x_4 = \dot{z}_{us}$, se obtiene,

$$\dot{x}_1 = x_2 - x_4 \quad (5)$$

$$\dot{x}_2 = \ddot{z}_s \quad (6)$$

$$\dot{x}_3 = x_4 - \dot{z}_r \quad (7)$$

$$\dot{x}_4 = \ddot{z}_{us} \quad (8)$$

y se obtiene la ecuación en variables de estado,



$$\dot{x} = Ax + Bu + B_d d \quad (10)$$

$$y = Cx + D_u \quad (11)$$

donde

$$A = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -K_s/M_s & -B_s/M_s & 0 & B_s/M_s \\ 0 & 0 & 0 & 1 \\ K_s/M_{us} & B/M_{us} & -K_{us}/M_{us} & -B_s + (B_{us}/M_{us}) \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 1/M_s \\ 1 & 0 \\ B_s/M_{us} & -1/M_{us} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ K_s/M_s & -B_s/M_s & 0 & B_s/M_s \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 1/M_s \end{bmatrix},$$

y B_d es la matriz de perturbaciones y d es una perturbación externa acotada. Las variables de estado x_1, x_2, x_3, x_4 representan el movimiento de la suspensión, la velocidad vertical del vehículo, el movimiento de la llanta y la velocidad vertical de la llanta, respectivamente.

3. Planteamiento del problema.

Considere el modelo del sistema de amortiguamiento con perturbación definido por la ecuación (10). Siguiendo la metodología de diseño de algoritmos de control con convergencia en tiempo finito, se define una superficie o salida,

$$\sigma = c^T x \quad (12)$$

donde $c^T = [c_1, c_2, \dots, c_{n-1}]$ son constantes positivas. Derivando σ en (12) y sustituyendo la ecuación (10), se tiene la ecuación

$$\dot{\sigma} = c^T (Ax + Bu + B_d d)$$

donde aparece la señal de control. Por lo tanto, el sistema tiene grado relativo uno con respecto a la salida σ y alguno de los algoritmos con convergencia en tiempo finito se puede implementar para asegurar convergencia en tiempo finito e insensibilidad ante la perturbación d .

Sin embargo, el modelo de la ecuación (10) que se utiliza ampliamente para el diseño de algoritmos de control en tiempo finito, no contiene la dinámica del actuador, sensores, o alguna otra dinámica adicional que aparezca en el lazo de control real. Entonces, el grado relativo del sistema de amortiguamiento real siempre es más grande que el considerado para el diseño del algoritmo de control. Específicamente, la dinámica de los actuadores en sistemas de amortiguamiento activo es al menos de orden 1 [2].

Suponga la dinámica del actuador como [16],

$$\mu \dot{v} = -v + u \quad (13)$$

donde μ es la constante de tiempo, v la salida del actuador y u la entrada de control. Tomando en cuenta que la dinámica del actuador (13) se integra en cascada con la dinámica del sistema de amortiguamiento, la derivada $\dot{\sigma}$ es,

$$\dot{\sigma} = c^T (Ax + Bv + B_d d),$$



donde se observa que la señal de control u no aparece en la derivada, por lo tanto, no se puede concluir que el grado relativo del sistema es uno.

El análisis en el dominio de la frecuencia de controladores en tiempo finito mostró que la presencia de dinámicas no modeladas incrementa el grado relativo del sistema y deteriora el desempeño de los algoritmos de control [13,15]. Como resultado de la presencia de las dinámicas no modeladas, surgen oscilaciones de magnitud y frecuencia finita alrededor de la salida o variable deslizante.

En el presente trabajo, se analiza el efecto de las dinámicas no modeladas en sistemas de amortiguamiento activo controlados con algoritmos de convergencia en tiempo finito. Particularmente, se analiza el algoritmo super-twisting y su desempeño real (considerando el actuador) en sistemas de amortiguamiento activo.

4. Análisis en el dominio de la frecuencia de algoritmos en tiempo finito.

El análisis en frecuencia de algoritmos con convergencia en tiempo finito se lleva a cabo mediante la aplicación del concepto de Función Descriptiva y la ecuación de balance armónico.

4.1 Función Descriptiva (FD)

La FD es una herramienta de análisis de sistemas no lineales cuando estos están sometidos a entradas senoidales. De manera específica, la FD es un tipo de linealización de un elemento no lineal. El tipo de sistemas a los cuales puede aplicarse FD son aquellos que se pueden separar en una parte lineal y una parte no lineal, como se muestra en el diagrama de la figura 2.

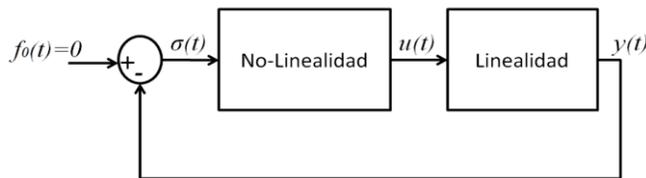


Figura 2. Sistema no lineal

Considerando el sistema no lineal de la figura 2, los requerimientos que se deben cumplir para poder aplicar el método de FD son [14]:

1. Existe solo una componente no lineal. Esto quiere decir que si hay dos o más elementos no lineales en el sistema, estos se deben poder agrupar como una sola no linealidad (como puede hacerse en el caso de dos no linealidades en paralelo) ó solo considerar una de las no linealidades y omitir las restantes.
2. El elemento no lineal es invariante en el tiempo. Esta suposición implica que se consideran solo sistemas no lineales autónomos. La razón de esta suposición es que el método de FD está basado en gran parte en el uso de la traza de Nyquist, la cual se aplica solo a sistemas invariantes en el tiempo.
3. Con una entrada senoidal $\sigma = \sin \omega t$, solo la componente fundamental $u_1(t)$ de la salida $u(t)$ es considerada. El presente requerimiento es la suposición fundamental del método de FD. Esta suposición implica que los armónicos de alta frecuencia son omitidos en el análisis. Para cumplir esta suposición es importante que el elemento lineal que tiene como entrada $u(t)$, o sea la salida del la no linealidad, tenga las propiedades de un filtro pasa-bajas, es decir,



$$|G(j\omega)| \gg |G(jn\omega)|$$

4.2 Cálculo de la FD

Considere que el elemento no lineal de la figura 2 tiene una entrada senoidal con amplitud A y frecuencia ω , $\sigma = A \sin \omega t$, como se muestra en la figura 2.2. La salida de la no linealidad es una señal periódica, aunque no senoidal. Al ser la salida de la no linealidad $u(t)$ una señal periódica, ésta puede representarse mediante series de Fourier como

$$u(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (14)$$

donde los coeficientes de Fourier a_i 's y b_i 's son generalmente funciones de A y ω , y se calculan con las siguientes expresiones:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} w(t) d(\omega t)$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} w(t) \cos(n\omega t) d(\omega t)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} w(t) \sin(n\omega t) d(\omega t)$$

La FD se define como la representación entrada-salida de la no linealidad,

$$N(A, \omega) = \frac{1}{A} (b_1 + j a_1) \quad (15)$$

Una vez calculada la FD de la no linealidad, la FD reemplaza a la no linealidad en el análisis y se estudia como un elemento lineal con función de respuesta en frecuencia $N(A, \omega)$, como se observa en la figura 3.

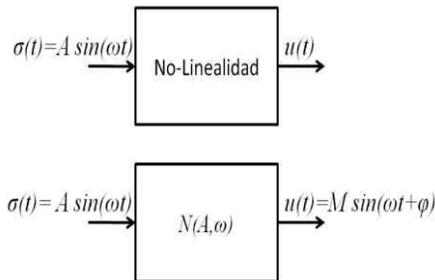


Figura 3. Representación en FD de un elemento no lineal

4.3 Parámetros de oscilaciones usando FD

Considere el sistema mostrado en la figura 4, donde $N(A)$ es la FD y $W(j\omega)$ representa al elemento lineal.

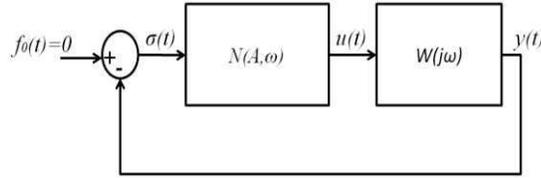


Figura 4. Representación en FD de un elemento no lineal

El análisis en frecuencia corresponde a identificar oscilaciones en el sistema de control. Se identifica la amplitud A y la frecuencia ω de la oscilación por medio de la ecuación de balance armónico,

$$W(j\omega) = -\frac{1}{N(A, \omega)} \quad (16)$$

donde $W(j\omega)$ es la respuesta en frecuencia del elemento lineal y $N(A, \omega)$ es la función descriptiva de la no linealidad. La solución de la ecuación de balance armónico tiene una interpretación gráfica como la mostrada en la figura 5, donde la solución se encuentra en la intersección de la traza de la respuesta en frecuencia $W(j\omega)$ y el inverso negativo de la FD ($N^{-1}(A, \omega)$).

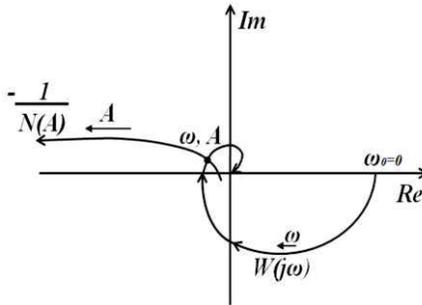


Figura 4. Representación en FD de un elemento no lineal

4.4 Algoritmos con convergencia en tiempo finito en el dominio de la frecuencia

Los sistemas de control con algoritmos con convergencia en tiempo finito se pueden llevar a la forma del diagrama de la figura 2, donde el sistema se puede separar en una parte lineal y no lineal [13,16,]. La parte lineal es la respuesta en frecuencia de la planta que se controla y la parte no lineal es el algoritmo de convergencia en tiempo finito, por tal motivo se requiere el cálculo de la FD del algoritmo de convergencia en tiempo finito a analizar. Una amplia variedad de algoritmos con convergencia en tiempo finito ha sido representada mediante FD. Por lo tanto, de manera general el análisis en frecuencia de sistemas controlados con algoritmos con convergencia en tiempo finito se puede llevar a cabo con la ecuación de balance armónico (16), considerando que $N(A, \omega)$ es la FD del algoritmo.

Un algoritmo con convergencia en tiempo finito ampliamente utilizado es el algoritmo por modos deslizantes llamado Super-Twisting, el cual se define como

$$\begin{aligned} u &= -k_1 |\sigma|^{\frac{1}{2}} \text{sign}(\sigma) + \dot{\eta} \\ \dot{\eta} &= -k_2 \text{sign}(\sigma) \end{aligned} \quad (16)$$

donde $k_1, k_2 > 0$ son las ganancias del controlador. El análisis en frecuencia de un sistema de control con algoritmo Super-Twisting se lleva a cabo considerando el diagrama de la figura 5,

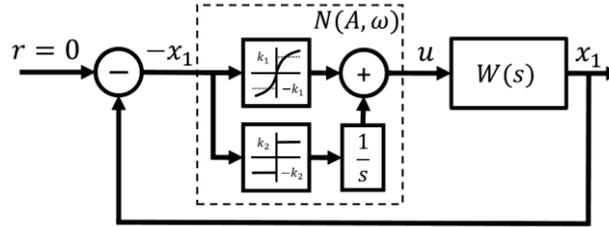


Figura 5. Representación en FD de un sistema con Super-Twisting

La expresión analítica de la FD para el algoritmo Super-Twisting es

$$N(A, \omega) = \frac{2\alpha_1 k_1}{\pi A^2} + \frac{1}{j\omega} \left(\frac{4k_2}{\pi A} \right)$$

donde $\alpha_1 \approx 1.748$, y la representación gráfica de N^{-1} es la mostrada en las curvas de la figura 6. Nótese que las curvas de la FD de Super-Twisting están ubicadas solo en el cuadrante 3 del plano. Entonces, sistemas con grado relativo 1, cuya respuesta en frecuencia se ubica en el cuarto cuadrante, no tendrían intersección con N^{-1} , a este caso se le llama *comportamiento ideal* [LibroSMC], debido a que la ausencia de oscilaciones se asume una convergencia exacta a cero de σ . De manera contraria, un comportamiento no ideal es aquel que presenta oscilaciones cuya identificación se hace con la solución de la ecuación de balance armónico (16) o de manera gráfica con la intersección de N^{-1} con la respuesta en frecuencia del sistema.

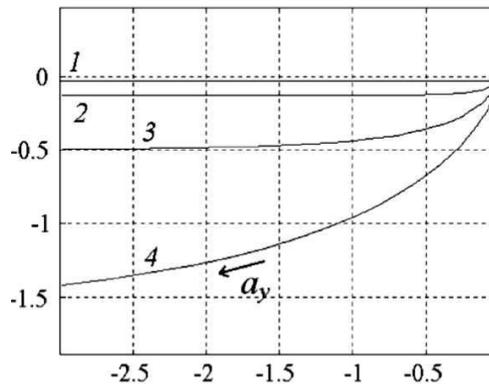


Figura 6. Gráfica de N^{-1} del algoritmo Super-Twisting.

Una vez mostrada la forma en que se analizan los sistemas de control en el dominio de la frecuencia, se procederá en la siguiente sección a estudiar los sistemas de amortiguamiento activos controlados por algoritmos con convergencia en tiempo finito.

5. Análisis de sistemas de amortiguamiento activo considerando actuador

Considerando la dinámica del sistema de amortiguamiento activo de la ecuación (10) y la salida de la ecuación (12). Con la derivada de σ con respecto al tiempo

$$\dot{\sigma} = c^T (Ax + Bu + B_d d)$$

Se concluye que el grado relativo del sistema de amortiguamiento activo (10) con respecto a la salida (12) es uno. Por lo tanto, cualquier controlador con convergencia en tiempo finito para grado relativo uno, lleva la salida σ a cero en tiempo finito sin importar la presencia de la perturbación d .



El diagrama del sistema de control para el sistema de amortiguamiento activo controlado por un algoritmo en tiempo finito es el que se presenta en la figura 7. El bloque $W(j\omega)$ es la respuesta en frecuencia del sistema de la ecuación (10)



Figura 7. Diagrama de bloques sistema de amortiguamiento activo.

5.1 Caso 1: Sin actuador

Considere que el algoritmo en tiempo finito es el Super-Twisting definido en la ecuación (16), cuyas curvas N^{-1} se ubican en el tercer cuadrante. De la derivada de la salida σ con respecto al tiempo, se sabe que el grado relativo del sistema con respecto a la entrada de control es uno. Entonces, la repuesta en frecuencia $W(j\omega)$ que tiene como entrada u y salida σ , es de grado relativo uno y el análisis en frecuencia concluye que se presenta un comportamiento ideal del controlador i.e. σ va converger a cero en tiempo finito sin importar las presencia de la perturbación.

5.2 Caso 2: Con actuador

Considerando la dinámica del actuador, el diagrama de la figura 7 se transforma en de la figura 8, donde se aprecia la dinámica del actuador en cascada con la respuesta en frecuencia de la planta.

Analizando el caso en el que el algoritmo de control es Super-Twisting y realizando el análisis en frecuencia. La planta sobre la que va actuar el control u es una planta en cascada que se forma con la respuesta en frecuencia $W(j\omega)$ y la dinámica del actuador. Por lo tanto, el grado relativo de $W(j\omega)$ no es uno y será al menos de dos.

Si se analiza la FD del Super-Twisting (N^{-1}) que se encuentra en el tercer cuadrante del plano y la cascada entre $W(j\omega)$ y la dinámica del actuador, que al menos será de grado relativo dos. Se concluye que puede existir la intersección entre N^{-1} y $W(j\omega)$. Como resultado surge una oscilación de amplitud y frecuencia finita, y se concluye un comportamiento no ideal del sistema.

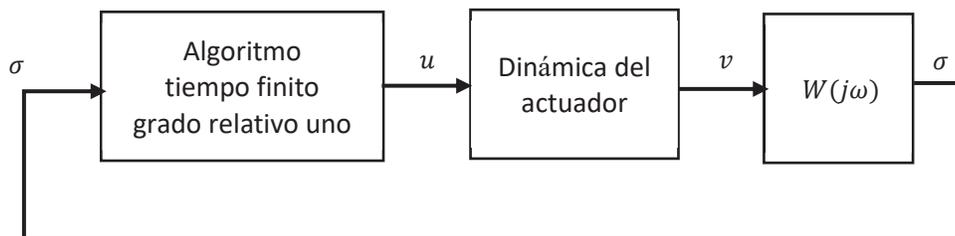


Figura 8. Diagrama de bloques sistema de amortiguamiento activo con actuador.

En la siguiente sección, con ayuda de simulaciones por computadora se verifican las conclusiones y análisis realizado en la sección 5



6. Simulaciones

Para verificar el análisis de la sección 5, se implementó en Matlab-Simulink el modelo del sistema de amortiguamiento activo de la ecuación (10), junto con el controlador de la ecuación (16). La simulación se llevo a cabo a paso fijo, usando el método ODE1 (Euler), con un paso de 0.1 milisegundos.

Los valores de los parámetros del sistema que se usaron en la simulación son: $M_s = 10$, $M_{us} = 2$, $B_{us} = 13$, $B_s = 15$, $K_s = 800$ y $K_{us} = 190$. Las ganancias del algoritmo de control que se usaron son $k_1 = 20$ y $k_2 = 60$. La prueba que se realizo fue partir de condiciones iniciales $x_1(0) = 0.5$ (los otros 3 estados se dejaron en cero) y dejar que el controlador amortiguará al sistema.

Se simularon tres casos: el sistema sin control y sin dinámica del actuador, el sistema con el algoritmo de control en tiempo finito sin dinámica del actuador; y el sistema con la dinámica del actuador y el algoritmo de control en tiempo finito. La dinámica del actuador que se considera es $0.03\dot{v} = -v + u$.

Los resultados que se muestran corresponden al estado x_1 que representa el movimiento del vehículo, el cual debe ser lo más amortiguado posible, y la superficie σ la cual debe ser exactamente cero para que el controlador asegure robustez.

La figura 9 corresponden al caso 1, cuando el sistema se encuentra en lazo abierto. Como se puede observar el amortiguamiento del sistema es lento porque solo esta actuando el sistema pasivo.

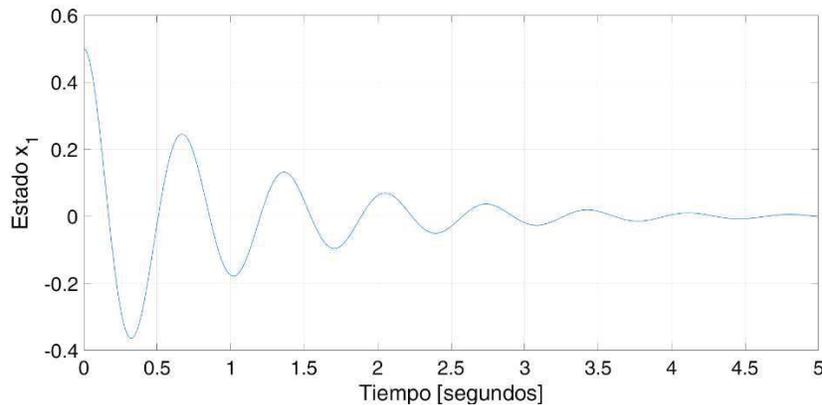


Figura 9. Estado x_1 sin control

El caso 2 es en lazo cerrado empleando el controlador en tiempo finito de la ecuación (16). La figura 10 muestra como el estado x_1 es llevado a cero en tiempo finito y sin oscilaciones. La superficie σ se presenta en la figura 11, donde se observa que la superficie converge a cero en tiempo finito. Tanto los resultados de la figura 10 como de la 11 comprueban que se tiene un caso de comportamiento ideal del algoritmo.

En el caso 3, la dinámica de primer orden de un actuador se integra en cascada con el controlador y la planta. Como se puede observar en la figura 12, el estado x_1 converge en tiempo finito, pero a una región alrededor de cero, donde se queda oscilando. De manera similar la superficie σ también converge en tiempo finito a una región alrededor de cero, donde permanece oscilando, ver figura 13. El comportamiento que muestran las figuras 12 y 13, es un comportamiento no ideal del algoritmo de control debido a la existencia de dinámicas no modeladas como la del actuador.

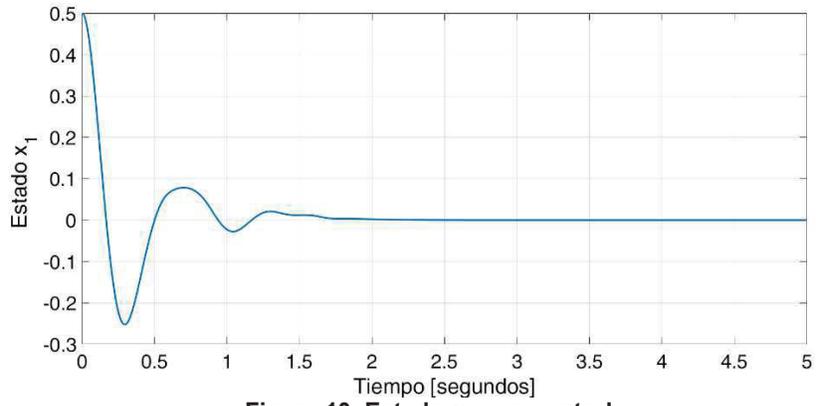


Figura 10. Estado x_1 con control

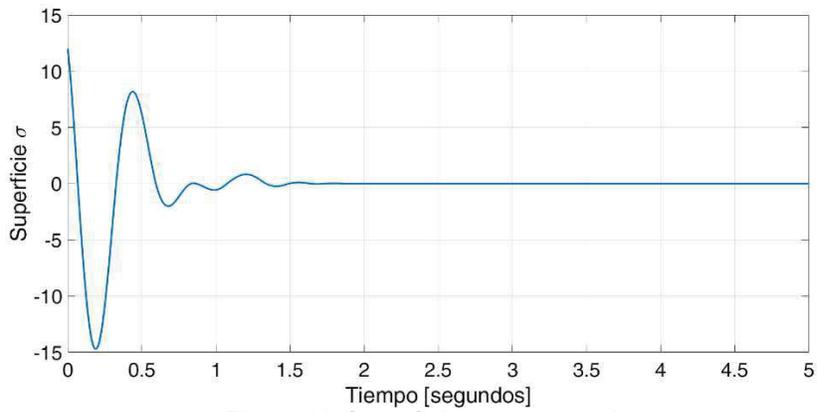


Figura 11. Superficie σ con control

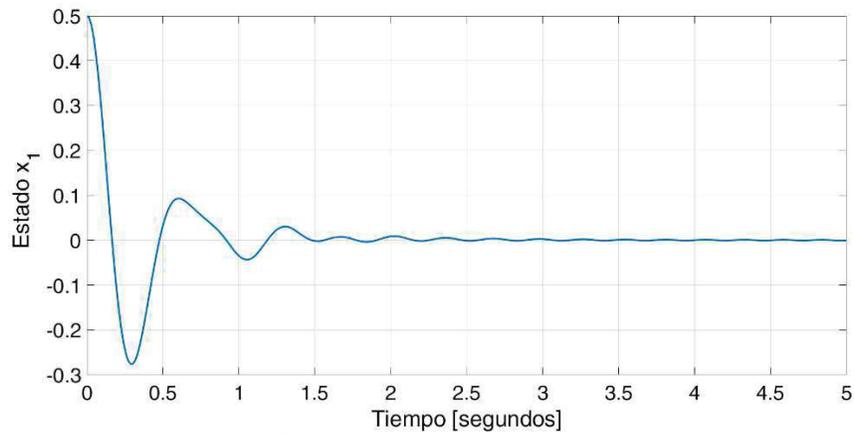


Figura 12. Estado x_1 con control más actuador

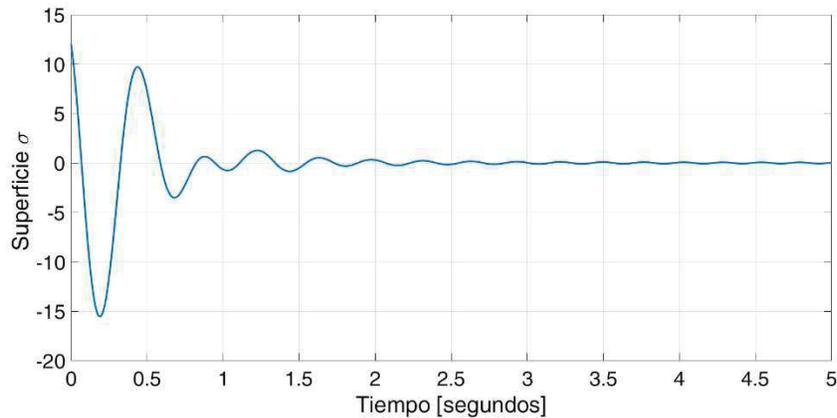


Figura 13. Superficie σ con control más actuador

7. Conclusiones

En el presente trabajo se analizan los efectos de las dinámicas no modeladas en los sistemas de suspensión activa controlados por algoritmos con convergencia en tiempo finito. El hecho de no considerar dinámicas como la del actuador durante el diseño del algoritmo de control degrada el desempeño del controlador cuando se implementa en el sistema real, el cual si contiene todas las dinámicas no modeladas. El fenómeno que se presenta son oscilaciones de amplitud y frecuencia finita en las variables que se desea controlar. El análisis que se presenta en el artículo se llevó a cabo con un algoritmo de control, pero puede ser aplicable a varios algoritmos con convergencia en tiempo finito que cuenten con su representación de Función Descriptiva.

Como se mostró en el análisis y las simulaciones, el omitir la dinámica del actuador en el diseño del algoritmo de control no asegura el desempeño ideal de los algoritmos con convergencia en tiempo fijo. En cambio, se presenta un comportamiento no ideal donde sí existe convergencia en tiempo finito pero a un región alrededor del origen y no exactamente al origen como es el caso ideal.

Referencias

- [1] Kiencke, U., y Nielsen, L. "Automotive Control Systems", Springer, USA, 2da Edición, 2000.
- [2] Cao, J., Liu, H., Li, P. y Brown, D.J. "State of the Art in Vehicle Active Suspension Adaptive Control Systems Based on Intelligent Methodologies," in IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 392-405, Sept. 2008.
- [3] Koch, G., y Kloiber, T. "Driving State Adaptive Control of an Active Vehicle Suspension System," in IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pp. 44-57, Jan. 2014.
- [4] Sun, W., Zhao, Z., y Gao, H. "Saturated Adaptive Robust Control for Active Suspension Systems," in IEEE Transactions on Industrial Electronics, vol. 60, no. 9, pp. 3889-3896, Sept. 2013.
- [5] H. Pan and W. Sun, "Nonlinear Output Feedback Finite-Time Control for Vehicle Active Suspension Systems," in IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2073-2082, April 2019
- [6] J. Zhang, W. Sun and H. Jing, "Nonlinear Robust Control of Antilock Braking Systems Assisted by Active Suspensions for Automobile," in IEEE Transactions on Control Systems Technology, vol. 27, no. 3, pp. 1352-1359, May 2019.
- [7] H. Pan, W. Sun, H. Gao and J. Yu, "Finite-Time Stabilization for Vehicle Active Suspension Systems With Hard Constraints," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2663-2672, Oct. 2015.



- [8] K. Patel and A. Mehta, "Second order sliding mode control of active suspension system with supertwisting algorithm and disturbance observer," IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, 2017, pp. 6526-6531.
- [9] J. D. Sánchez-Torres, A. G. Loukianov, M. I. Galicia, J. Ruiz and J. Rivera, "ABS and active suspension control via high order sliding modes and linear geometric methods for disturbance rejection," 2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control, Merida City, 2011, pp. 1-6.
- [10] V. S. Deshpande, P. D. Shendge and S. B. Phadke, "Nonlinear Control for Dual Objective Active Suspension Systems," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 3, pp. 656-665, March 2017.
- [11] V. S. Deshpande, M. Bhaskara and S. B. Phadke, "Sliding mode control of active suspension systems using a disturbance observer," 2012 12th International Workshop on Variable Structure Systems, Mumbai, Maharashtra, 2012, pp. 70-75.
- [12] M. Avesh and R. Srivastava, "Modeling simulation and control of active suspension system in Matlab Simulink environment," 2012 Students Conference on Engineering and Systems, Allahabad, Uttar Pradesh, 2012, pp. 1-6.
- [13] Shtessel, Y., Edwards, C., Fridman, L., Levant, A.: 'Sliding Mode Control and Observation'. (New York: Springer, 2014)
- [14] Khalil, H.: 'Nonlinear Systems'. (New Jersey:U.S.A.: Prentice Hall, 1996)
- [15] I.M. Boiko , On relative degree, chattering and fractal nature of parasitic dynamics in sliding mode control, J. Frankl. Inst. 351 (4) (2014) 1939–1952.
- [16] Pérez, U., y Fridman, L. "Design of super-twisting control gains: A describing function based methodology" in Automatica, vol. 99, pp. 175-180, 2019
- [17] Levant, Arie. "Sliding order and sliding accuracy in sliding mode control". International Journal of Control, 58(6),1247–1263, 1993.
- [18] A. Rosales, Y. Shtessel, L. Fridman and C. B. Panathula, "Chattering Analysis of HOSM Systems: Frequency Domain Approach", IEEE Transactions on Automatic Control, Vol. 62, No. 8, August, pp. 4109 - 4115, 2017



Prototipo de Embotelladora para Agua en PET de 600 ml para Pequeños Productores

Palma Cruz Victor, Valencia Benitez Roberto, Amador Chagala Héctor M. y Palma Cruz Lorena

Instituto Tecnológico Superior de San Andrés Tuxtla.
victorpalma@itssat.edu.mx

Resumen

En el presente artículo se describe el desarrollo de un prototipo en donde se implementa un sistema automatizado compacto, multifuncional de embotellado de líquidos en envases de PET de 600 ml. Dicho proceso consiste en cuatro etapas: diseño de CAD del sistema de mecanismos, proceso de manufactura, diseño de automatización al sistema, finalmente la fase de implementación, puesta en marcha y operación del prototipo.

Para la etapa de diseño se hace uso de software como SolidWorks para el dibujo, ensamble y modelado de los componentes, se elaboraron algunas piezas mediante una impresora 3D, en la parte de automatización se maneja la programación mediante el software RS Logix, propio del PLC Allen Bradley MicroLogix 1100 implementado en el sistema.

Se encontraron resultados óptimos en la puesta en marcha del prototipo logrando el llenado en promedio de 360 envases por hora. En términos generales, la implementación del sistema automatizado aumenta considerablemente la productividad en el llenado de líquidos en PET de 600ml., disminuyendo la mano de obra invertida y reduciendo el tiempo de producción.

Palabras clave: Embotelladora, automatización, PLC, llenado de líquidos.

1. Introducción

A lo largo de la historia, diversos productos envasados en botellas PET han ido ganando terreno por la practicidad de transportación, por lo que en la actualidad es muy frecuente su uso. Grandes empresas tienen la infraestructura tecnológica y económica a su alcance para embotellar sus productos, optimizando los tiempos de producción mediante procesos de embotellado automatizados.

Existen numerosos métodos para realizar el proceso de embotellado en la industria, a continuación se describen los más significativos en la actualidad:

- Proceso lineal "paso a paso". Éste tipo de distribución corresponde al caso en el que los equipos necesarios para el proceso de llenado se agrupan en una misma zona y se ordena de acuerdo con el proceso secuencial de fabricación. Se emplea usualmente en los casos en que exista una elevada demanda de uno o varios productos más o menos estandarizados, o en la fabricación de productos específicos que tienen como base un producto genérico. Por ejemplo, el embotellado de gaseosas, el montaje de automóviles.



- Proceso "rotativo" continuo, en donde la alimentación de los envases, las válvulas de llenado y la operación de envasado se realiza en un sistema tipo carrusel; para lo cual se utiliza un sistema de estrellas de alimentación y salida de envases las cuales van sincronizadas al tanque porta-válvulas de llenado.

La automatización para la industria, es usar tecnología que integre un proceso de control a través de dispositivos capaces de tomar decisiones e interactuar con otros, basándose en un programa establecido por el integrador para el manejo de algunas variables, mediante su monitoreo y comparación con un valor esperado del proceso; esto se realiza de manera automática, generando en el sistema mayor productividad, confiabilidad, estabilidad y calidad en sus resultados [1].

Los componentes básicos de un sistema de control basado en un PLC se describen a continuación:

- Control Lógico Programable (PLC), llamado también denominado Autómata Programable (AP) [2]. Bajo este nombre se incluyen los módulos electrónicos a través de los cuales todas las funciones de la máquina o sistema a controlar son direccionadas y activadas en una secuencia lógica
- Sensores. Estos componentes están situados directamente en el sistema o la máquina a controlar, y a través de ellos el PLC interroga el estado o posición de la máquina. Son los elementos empleados para captar información del entorno, en este caso, el sistema a controlar.
- Actuadores. Estos componentes están situados directamente en el sistema o máquina a controlar y a través de ellos, se generan fuerzas o movimientos dentro del proceso. Regularmente son actuadores de tipo neumático (operan con aire comprimido), hidráulicos (basados en aceite mineral) y eléctricos (basados en electricidad).
- PC o dispositivo programador. Se utiliza para crear el programa que contiene la lógica del sistema o maquinaria a controlar y transferirla a la memoria del PLC. Así mismo, permite la edición, depuración y puesta a punto del programa del PLC [3].

Los lenguajes de programación para la programación de un PLC según la estándar internacional IEC 61131 son los siguientes:

- Instruction list (IL), texto
- Ladder (LD), gráfico
- Function block diagram (FBD), gráfico
- Structured text (ST), text

El lenguaje LADDER ("escalera"), es un lenguaje de programación gráfico basado en los clásicos esquemas de control eléctricos con relés, de este modo, es muy fácil de entender; además es el más apropiado para utilizar, por ser un software libre.

Mediante una consulta en la página web del IMPI (Instituto Mexicano de Propiedad Industrial), algunos de los trabajos patentados que se encuentran sobre la implementación de un sistema de embotellado son los que se describen a continuación.

En 1997 México, D.F., La Barre Paul patenta una instalación de embotellado en línea que comprende esencialmente una unidad de fabricación de recipientes, especialmente de botellas, de material termoplástico, una unidad de llenado de dichos recipientes y una unidad de transporte de los recipientes que vienen de ser fabricados interpuesta entre la salida de la unidad de fabricación de los recipientes y la entrada de la unidad de llenado de recipientes y la entrada de la unidad de llenado de recipientes ; la unidad de fabricación y la unidad de llenado se disponen a una distancia tan corta



como sea posible una de la otra y la unidad de transporte posee una corta longitud y esta acondicionada para desplazar los recipientes unos detrás de otros sin que los recipientes sufran choques sensibles especialmente unos contra otros; de preferencia, comprende además otra unidad de retención temporal de recipientes que se une selectivamente a la unidad de transporte y acondicionada para recibir y retener cierto número de recipientes.

En el 2010, Clarke Modet y Compañía de Mexico S.A. patenta un dispositivo que consiste en introducir fluido en contenedores, así como una línea de transporte desde el contenedor de almacenamiento al dispositivo de llenado, una válvula en la línea de transporte entre el contenedor de almacenamiento y el dispositivo de llenado que de forma automática conmuta dependiendo de la velocidad de flujo del líquido dentro de la línea de transporte.

En 2013, KRONOS DTA, S.A DE C.V patenta una maquina embotelladora de envases PET que integra desde el proceso de soplado hasta el de empaclado comprende la integración de los procesos de soplado, etiquetado, llenado y enroscado en una misma máquina, tiene la posibilidad de envasar con envases de bajo gramaje, produce 2,500 envases por cavidad por hora, es capaz de soplar envases desde 40ml hasta 3.5 litros y roscas de 16 mm a 70mm, procesa múltiples productos como : refresco, cerveza, agua, alimentos, químicos, productos de aseo personal, farmacéuticos entre muchos más, además de ser una maquina compacta y de fácil cambio de moldes; se caracteriza por el manejo de las botellas por las rosca, es de alta eficiencia y autónoma, lo que le permite grandes ahorros energéticos, bajos porcentajes de desperdicios, tiene la flexibilidad de procesar distintos productos, es muy compacta, lo que significa ahorros de espacio en almacenes de 90%, cuenta con la facilidad de empaclar una variedad de productos por paquete y con bajos requerimientos de mantenimiento.

En 2014, Universidad Tecnológica De Pereira patenta una máquina para el llenado de líquidos naturales sin azúcar endulzados a partir del aloe vera, y pasteurizados a altas temperaturas. La empresa presentaba un inconveniente en el proceso de embotellado, debido a que no existía en ella un equipo para esto y lo debían de realizar manualmente con recipientes plásticos; esto demanda gran dedicación y esfuerzo por parte de los empleados y hacía el proceso muy lento; por otro lado la técnica de pasteurización por calor requiere el embotellado a temperaturas muy altas, lo que supone un alto riesgo para los operarios, que conducen a accidentes por quemaduras muy frecuentes en este proceso.

Dentro de los trabajos realizados relacionados con la automatización del proceso de llenado que se encuentran en los registros del IMPI se mencionan los siguientes.

En 2008, Gonzales Monroy, J.G. del Instituto Politécnico Nacional automatiza la operación de llenado que se realiza mediante un sistema de boquillas que llenan la botella con el producto que previamente ha ingresado a una tolva alimentadora, desde donde se reparte por un juego de tuberías y válvulas hacia las boquillas. Una vez que la botella ha sido llenada, esta sale a un sistema mecánico de colocación de la tapa corona por presión. En este sistema se utiliza el PLC S7-400 para mejorar la automatización.

En 2015, Yon Palomino Puma, José Antonio Manrique Hernández de Lima Perú diseñan un sistema en el cual se llena en botellas, 3 diferentes tipos de líquidos, sin necesidad de hacer algún cambio de automatización o configuración, permitiendo el funcionamiento automático del sistema haciendo que la mano humana no sea necesaria en el proceso. El sistema fue diseñado en el programa MatLab usando redes neuronales para procesar las imágenes en tiempo real. Primero se tiene la banda transportadora que es utilizada para llevar los diferentes tipos de botellas, el recorrido es hasta un soporte de madera para mantenerse fija hasta el inicio de la base circular giratoria, que permite llevar a la botella hacia su etapa de llenado, se retira la botella de la base circular por medio de una de las levas giratorias (enchapado) finalmente, la etapa en la cual la botella llenada de manera correcta es retirada de la base circular por medio de una de las levas giratorias. Dicho sistema fue controlado mediante PLC.



La estructura del artículo se organiza de la siguiente manera: en la primera parte se describe el proceso de embotellado que se requiere automatizar. En la siguiente sección se realiza la descripción de cada una de las cuatro etapas consideradas para el desarrollo del prototipo que consisten en el diseño de CAD del sistema de mecanismos, el proceso de manufactura, el diseño de automatización, por último la implementación, puesta en marcha y operación del proceso. En cada etapa se describen las características de los elementos así como de los mecanismos en operación.

La importancia de la automatización en el procesamiento de envasado radica en mejorar la productividad al reducir la cantidad de operaciones manuales durante el proceso operativo. Aunque los procesadores comparten objetivos comunes como eficiencia operativa, ninguna línea de producción es igual, el diseño dependerá de las necesidades del proceso a automatizar.

Actualmente en la región de los Tuxtlas, las pequeñas empresas realizan manualmente el embotellado de agua para consumo humano exclusivamente en presentaciones de 20 litros, no pueden extender su comercialización en una presentación de 600 ml; debido al bajo acceso a la tecnología utilizada por el elevado costo para adquirirlo así como para la adaptación de instalaciones eléctricas especiales; otro factor importante es carecer del espacio suficiente para la colocación de un equipo robusto.

Por lo antes expuesto, se considera que el prototipo implementado es una aportación importante para que los pequeños productores puedan obtener eficiencia en la producción, reducir la mano de obra, reducir los tiempos empleados en el proceso de envasado; además de que no requiere de instalaciones especiales como son las subestaciones eléctricas; por otra parte aumenta la accesibilidad económica para su adquisición.

2. Desarrollo del prototipo de embotelladora.

2.1 Descripción del proceso de embotellado.

El proceso de envasado de líquidos mediante la embotelladora automatizada realiza las tareas de llenado de líquido en PET, posicionamiento de taparrosca en PET y taponado de la taparrosca obteniendo un producto terminado.

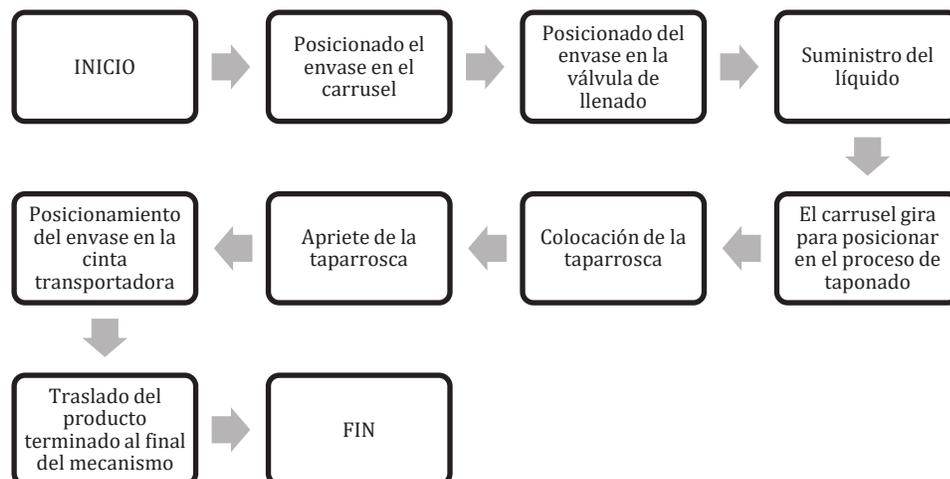


Figura 1 Proceso de envasado de líquido en envases de PET



El sistema de embotellado inicia con la colocación de forma manual de la botella de PET en una banda transportadora accionada por un motor eléctrico que transporta el envase a un carrusel que lo posiciona en el suministro de líquido mediante una válvula hidráulica accionada mecánicamente a través de un cilindro neumático, al llegar al nivel requerido, el carrusel gira posicionando al segundo proceso que es la colocación de la taparrosca, a continuación nuevamente el motor eléctrico se acciona para el apriete de la taparrosca en el PET; finalmente el carrusel vuelve a girar para posicionar el envase en la banda transportadora que lo traslada hasta finalizar su recorrido, en donde una persona almacena manualmente el producto terminado, en la figura 1 se puede observar el proceso mediante un diagrama de flujo.

2.2 Descripción de las etapas de para la implementación del prototipo.

El desarrollo del prototipo se realizó en cuatro etapas: **diseño de CAD del sistema de mecanismos** que consiste en el diseño y selección de los elementos que conforman el prototipo, además de realizar los planos y modelado mediante el software SolidWorks, **proceso de manufactura del sistema de mecanismos** en donde se imprimieron algunas piezas mediante una impresora 3D, además se realizó el montaje de la estructura y el armado de los mecanismos, **diseño de automatización al sistema de mecanismos** en esta etapa, se seleccionaron los componentes eléctricos, neumáticos y de control, se implementó un PLC Allen Bradley MicroLogix 1100 para el cual se elaboró un programa de control en lenguaje Ladder para dicho autómata, además se complementó con el montaje de los elementos que conforman el sistema. **Implementación, puesta en marcha y operación del proceso** que consiste en realizar las pruebas de arranque, analizando que la obtención de respuestas sean las esperadas por los actuadores, generando los movimientos planteados en el diseño del sistema, esta etapa es imprescindible puesto que determina la calidad y eficiencia en el funcionamiento del prototipo.

2.2.1 Diseño de CAD del sistema de mecanismos.

Mediante el software SolidWorks se realizaron los planos, el ensamble y modelado de los componentes, algunos de los elementos más importantes son: la estructura soporte del sistema de embotellado (ver figura 2).

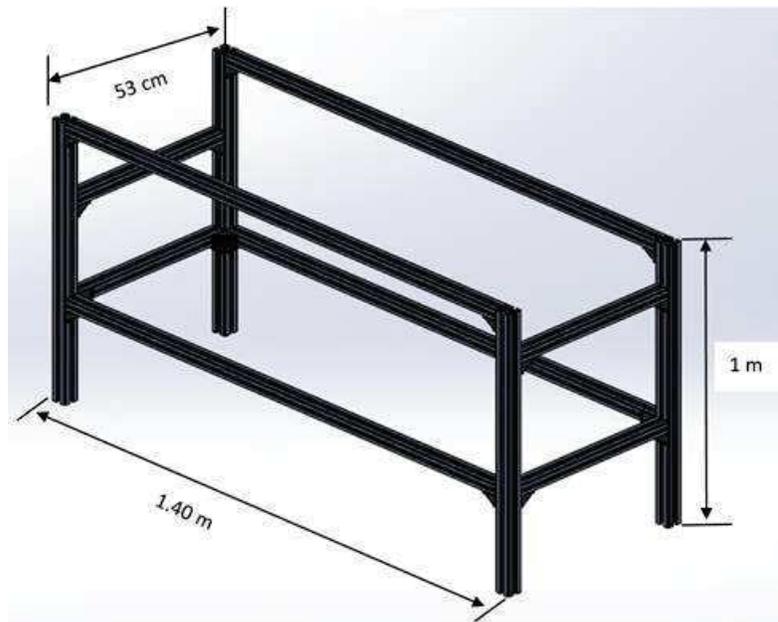


Figura 2 Estructura soporte del sistema de embotellado modelado en SolidWorks

La plataforma del mecanismo de transporte que se muestra en la figura 3; ésta consiste en una cinta transportadora movida por un motor eléctrico CA a 1.5 HP de 1725 RPM. Además se colocan brackets y guías que protegen al producto en movimiento a través de la banda.

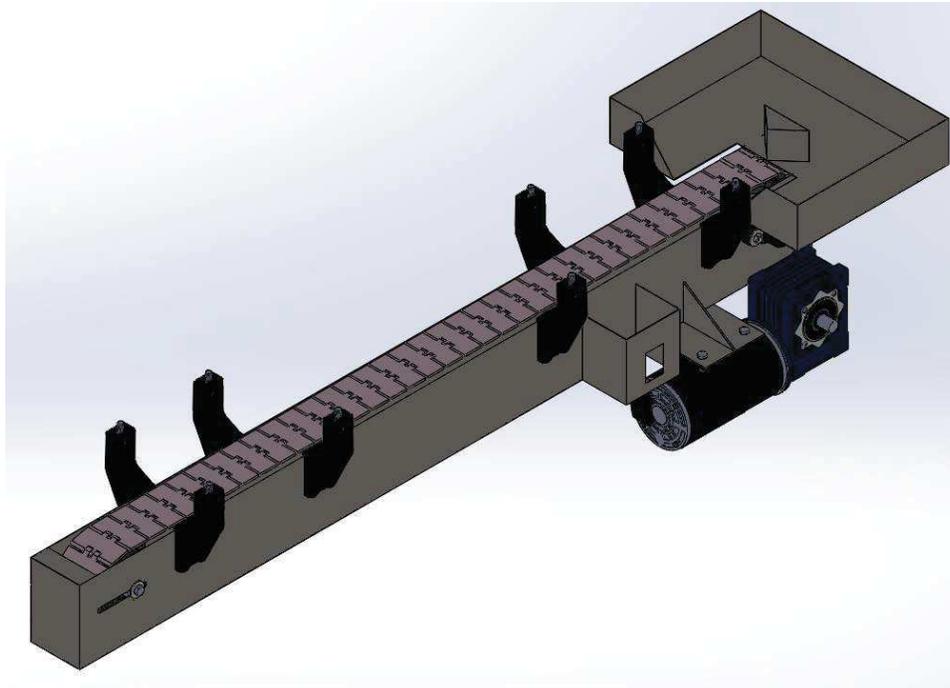


Figura 3 Modelado en SolidWorks del mecanismo de transporte de las botellas de 600 ml.

El carrusel de llenado cumple con el funcionamiento de posicionar la botella para realizar sus etapas de llenado y taponado mediante dos válvulas ubicadas en soportes sobre la cinta



transportadora, efectuando los movimientos para desplazar la botella en cada una de las válvulas que cumplen con las funciones antes mencionadas. El carrusel está conformado por dos ruedas iguales que se muestran en la figura 4.

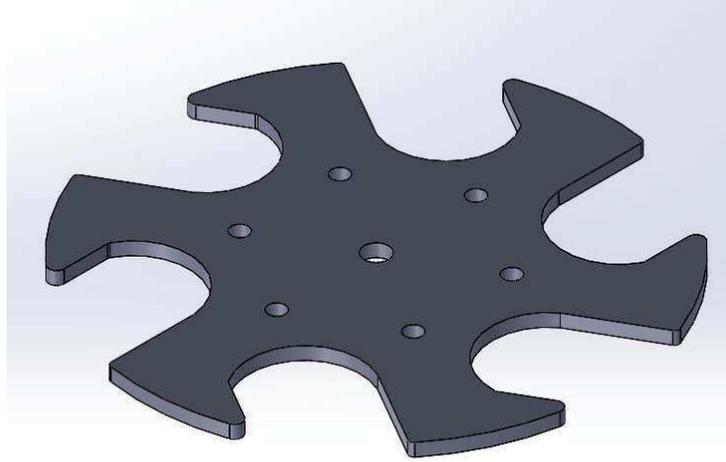


Figura 4 Rueda de carrusel de llenado para su manufactura

2.2.2 Proceso de manufactura del sistema de mecanismos

En general, la estructura de soporte del sistema de embotellado está compuesta por perfiles de aluminio serie 40 x 40 mm, esquinas para perfil de aluminio 4040, tuercas de deslizamiento y tornillos cabeza hueca botón Allen 5m, fijando sus elementos mediante soportes o torres estructurales apernadas o soldadas en una base sólida.

Algunos de los elementos que forman parte de los mecanismos fueron manufacturados mediante una impresora 3D, con el propósito de obtener flexibilidad para su fabricación, ahorrando tiempo y costo en el alcance de los objetivos planteados, la figura 5 muestra la impresión de una pieza que corresponde a un soporte del sistema.

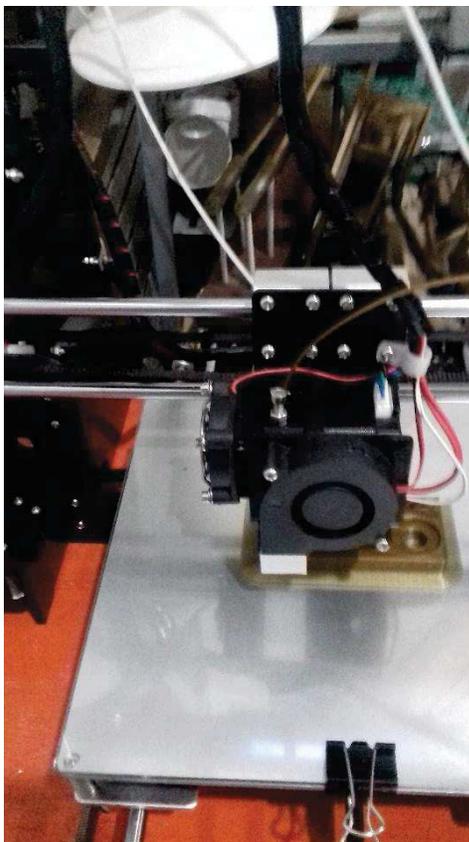


Figura 5 Impresión de una pieza del soporte.

2.2.3 Diseño de automatización al sistema de mecanismos

El sistema dispone de dos cilindros neumáticos de posición final ajustable neumáticamente o por amortiguadores, con una precisión de posición final de $< 0,05\text{mm}$, ajuste de la carrera de posición de avance y de retroceso, y dos cojinetes por barra de guía.

Por otra parte, contiene electroválvulas con apertura y cierre del caudal de aire mediante una señal eléctrica en una bobina; cuenta con 4 electroválvulas: dos de 5/2 vías y dos más tipo 3/2 vías, ver la figura 6.

El proceso de taponado de las botellas se lleva a cabo mediante una taponadora neumática con entrada 1/8 rosca npt, para tapa rosca de PET de 600ml de 300 x 35 mm de diámetro que requiere una fuente de alimentación de 110V / 220V – 50 / 60 Hz y una presión de Aire 0.4 -0 0.5 Mpa.

La función de la válvula taponadora es para enroscar la tapa a la botella sellando del producto evitando que se salga el líquido suministrado; se activa un sensor para desactivar el procedimiento al término de la colocación de la tapa en el envase.

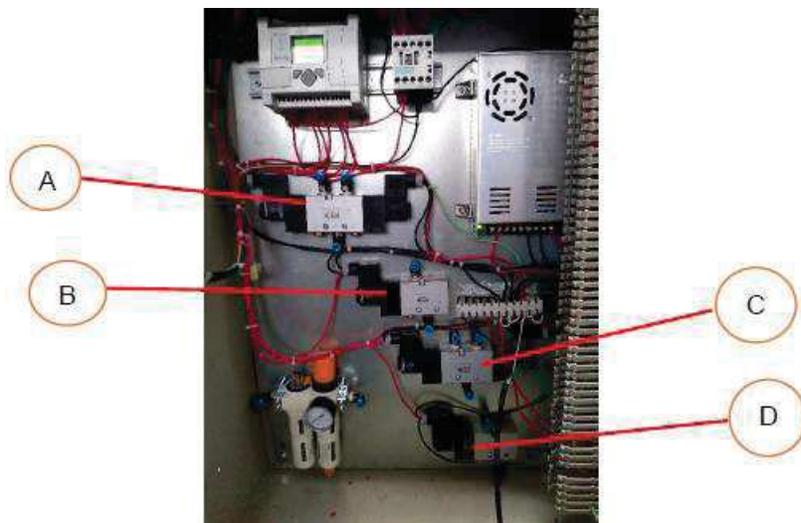


Figura 6 Electroválvulas: (A), (C): 5/2 vías; (B), (D): 3/2 vías.

La unidad de aire comprimido se encuentra conformada por un compresor, tanque de almacenamiento, un motor eléctrico, un manómetro, filtros de aire y accesorios de conexiones. Además, la alimentación de aire así como la interconexión con los elementos neumáticos se realizan mediante tubo flexible.

La banda transportadora traslada el envase PET a través de ella, hasta que se posiciona en el carrusel para realizar la primera etapa que es el llenado de la botella mediante una válvula que desciende por medio de un cilindro neumático, se suministra el líquido hasta llegar al tope de la botella; entonces se desactiva la válvula y asciende nuevamente. Posteriormente hace su segundo movimiento en el carrusel que vuelve a girar para volver a posicionar la botella en la segunda válvula para que haga el proceso de cerrado del producto mediante una taponadora neumática entrada 1/8 rosca npt, para tapa rosca de PET de 600ml; en donde de forma similar, un cilindro neumático asciende y desciende para realizar el proceso. Por último se coloca otra vez la botella en el carrusel que gira para colocar el envase lleno nuevamente en la banda transportadora que lo conduce al final del sistema, en donde de forma manual, una persona almacena el producto terminado.

El sistema de automatización consiste en un Controlador Lógico Programable (PLC) Allen Bradley MicroLogix 1100, el cual tendrá el control del funcionamiento de cada una de las etapas que comprende la estación de trabajo de embotelladora de PET. Así mismo se utilizará un sensor de proximidad D-M9P, a 24 Volts C.D para posicionar adecuadamente los envases, consecuentemente realizar la activación de cada cilindro controlando el nivel del líquido en la etapa de llenado y para el taponado de la botella.

Por otra parte, la programación del PLC se realiza mediante el software RSLogix Micro Starter Lite, que es propio del autómatas Allen Bradley MicroLogix 1100; por lo que no requiere licencia para su operación.

2.2.4 Implementación, puesta en marcha y operación del prototipo.

2.2.4.1 Ensamble de la estructura mecánica.



En esta etapa corresponde el ensamblado de los elementos que forman parte del prototipo, a continuación se muestra el armado de los componentes más relevantes de la embotelladora.

Se realizó el ensamble de la estructura principal, que tiene la función de soporte del sistema de embotellado, en la figura 7 se observa la colocación de los elementos de fijación en el perfil de aluminio 40x40.



Figura 7 Colocación de los elementos de fijación del perfil de aluminio.

En la figura 8 se muestra la estructura de aluminio terminada.



Figura 8 Estructura base terminada

En la estructura base se ensamblaron los componentes que conforman el mecanismo de transporte del producto envasado que consisten en una cinta transportadora, una polea de banda, cadena, ejes de transmisión manufacturados, soportes de ejes de transmisión, brackets de plástico, motor de CD 110V, a 1.5 HP de 1725 RPM y el motor reductor 1:50.

Terminados de colocar y fijar todos los brackets de plástico del lado izquierdo y derecho, se puede apreciar su montaje en la figura 9.

La base principal donde se montaron los componentes de la banda transportadora se fabricó en lámina de acero Inoxidable Gr. 304 Cal. 20 de 1 mm de espesor.

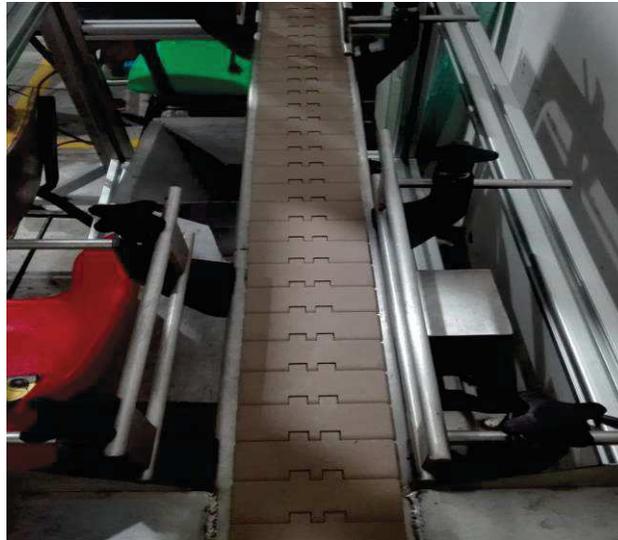


Figura 9 Cinta transportadora con brackets fijados y ajustados a la estructura.

El carrusel de llenado es el encargado de transformar el movimiento lineal de la banda transportadora en un movimiento rotativo que permite que los envases de PET puedan ser llenados con líquido, y posteriormente sellados.

Para el ensamble del carrusel de llenado son necesarias dos piezas manufacturadas de la rueda, 4 bases diseñadas e impresas con filamento ABS mediante una impresora 3D, son fijadas a una distancia de 11 cm en la base superior del carrusel y las otras 2 sobre la base inferior.



Figura 10 Carrusel fijado a la base.

La figura 10 muestra el carrusel fijado a la base de aluminio, que en conjunto con la cinta transportadora forman parte del mecanismo de posicionamiento de los envases PET en el sistema automatizado.



2.2.4.2 Ensamble del sistema de control.

El sistema de control neumático de la embotelladora de PET fue ensamblado dentro de un gabinete metálico, donde se colocaron dos electroválvulas 5/2 biestable, dos electroválvulas 3/2 monoestable, PLC Allen Bradley MicroLogix 1100, una fuente de alimentación y la unidad de mantenimiento neumática, como se puede observar en la figura 11.

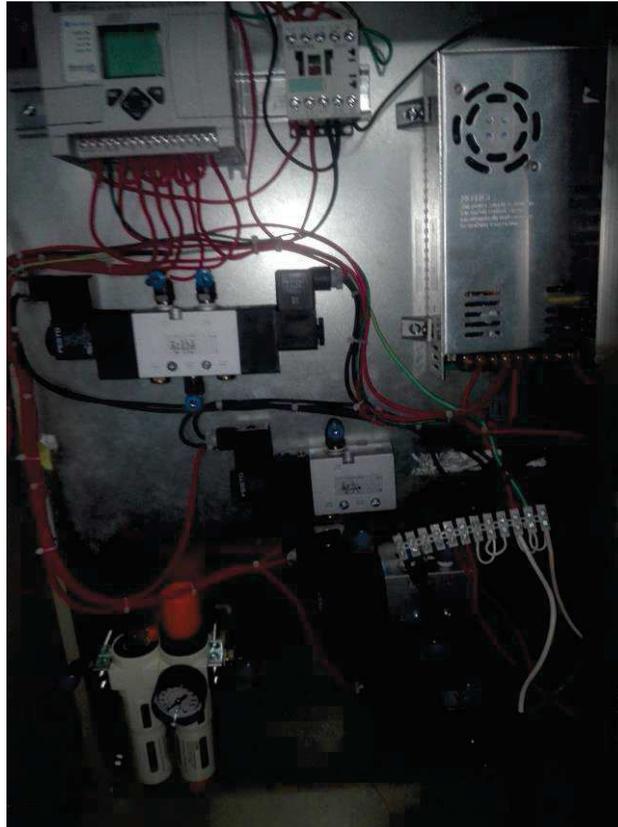


Figura 11 Sistema de control neumático.

2.2.4.3 Ensamble de los elementos neumáticos actuadores.

El sistema de control neumático convierte la energía del aire comprimido en energía mecánica, los cilindros neumáticos son los que reciben esta energía mecánica para actuar en la etapa de llenado y sellado de los envases. La figura 12 muestra como está acoplada la válvula de llenado al cilindro neumático y este a su vez al perfil de aluminio 40x80, con el que son fijados a la base del carrusel de llenado.



Figura 12 Válvula llenadora montada al soporte de aluminio.



La etapa de cerrado de las botellas es realizado por una taponadora neumática entrada 1/8 rosca npt, para tapa rosca de PET de 600ml; de forma similar que la válvula de llenado, la taponadora esta acoplada a un cilindro neumático y fijado a un perfil de aluminio 40 x 80 como se observa en la figura 13.



Figura 13 Cilindro neumático y taponadora neumática acoplados a la base del carrusel.

2 Resultados

Dentro de las limitaciones que se identifican en la implementación de maquinas embotelladoras actuales se considera principalmente que son diseñadas para manejar grandes volúmenes de producción, su adquisición solamente es factible para grandes empresas; para pequeñas microempresas resulta costosa una inversión para la adquisición de estas maquinarias que además requieren de un espacio considerable por su naturaleza robusta, otro factor es la instalación especializada de fuentes de alimentación suministrada por una subestación eléctrica.

Mediante la metodología descrita en la sección anterior se implementó el prototipo de una máquina embotelladora de arquitectura compacta que optimiza la producción de envasado de líquidos, haciéndola más eficiente.



Figura 13 Vista frontal del prototipo terminado.

La ventaja de la máquina embotelladora es concentrar las tareas que involucran el proceso de embotellado en PET en una máquina compacta, reduciendo la mano de obra; siendo multifuncional puesto que mientras un envase está en la fase de llenado, otro se encuentra en el colocado de taparrosca y otro más en el roscado, cubriendo ciclos de cinco envases en tres fases simultáneas.

Otra de las ventajas que ofrece el prototipo del sistema automatizado es que la potencia eléctrica que consume es mínima, puesto que sus componentes operan a 110 V; además no requiere una subestación eléctrica, tampoco es necesario hacer alguna instalación especial para su operación.

El sistema ha sido diseñado para que realice el proceso de forma continua, lo que lo hace eficiente de modo que proporciona los recursos necesarios para acelerar la producción de manera versátil.

El tiempo del proceso desde el llenado hasta el buen término del producto en el final de carrera del mecanismo de transmisión es de 10 segundos, por lo que se obtiene el llenado de un aproximado de 360 botellas de PET por hora.

Como etapa final, se elaboró un manual que integra la operación de máquina, así como los requerimientos para el óptimo mantenimiento de la misma, debido a que el diseño y fabricación es propio, por lo cual no se cuenta con manuales o guías de ningún tipo para su operación y mantenimiento en base en las normas oficiales Mexicanas NOM-201-SSA1-2002, productos y servicios. Agua y hielo para consumo humano, envasados y a granel. Especificaciones sanitarias, NOM-017-STPS-2008. Equipo de protección personal, selección, uso y manejo en los centros de trabajo, normas UNE-EN, ISO 9001 mantenimiento de infraestructura, en el ítem "6.3 Infraestructura", perteneciente al capítulo "6. Gestión de los recursos" de la norma ISO 9001:2008 y los manuales, recomendaciones y/o especificaciones de los componentes, mecánicos, hidráulicos, neumáticos y eléctricos utilizados en la fabricación de esta estación de trabajo de embotelladora de PET, para maximizar su eficiencia y durabilidad de los componentes, logrando su óptimo funcionamiento.

3 Conclusiones



En la actualidad los sistemas automatizados forman gran parte de la industria, donde la mayoría de ellas operan mediante sistemas de control, es importante el desarrollo tecnológico para optimizar el rendimiento de modo que su eficiencia aumente.

Debido al alto costo de los sistemas de control de embotellado, el micro y mediano empresario tiene limitaciones en la adquisición de sistemas actualizados, razón por la que éstos se inclinan por realizar el envasado en forma manual principalmente. Al proponer un sistema automatizado económico que realice dichos procesos representa un área de oportunidad para aumentar su capacidad de expandirse a un mercado más amplio, ofreciendo botellas de agua de 600 ml. al ser una opción que fácilmente se adopta para adquirir el producto en cualquier lugar, con una mayor demanda que los garrafones de 20 litros que habitualmente se distribuyen en la región de los Tuxtlas.

La importancia de garantizar la eficiencia y durabilidad de la máquina embotelladora es el factor que encausó a la elaboración de un manual de operatividad y mantenimiento, en base a las normas establecidas para maximizar su potencial.

La estructura metálica de lámina de acero esta acoplada a la estructura soporte de aluminio solo por medio de tornillería, el poner en marcha la banda transportadora genera vibraciones, por lo que es importante que a futuro se considere la implementación de componentes para reducirlas.

Para trabajos futuros a corto plazo, se contempla complementar el prototipo de embotelladora con el automatizado de un sistema que almacene el producto terminado en cajas.

Referencias

- [1] Castro J., Padilla J., Romero E. "*Metodología para realizar una automatización utilizando PLC*", Impulso, revista de electrónica, eléctrica y sistemas computacionales. Publicación semestral editada por el departamento de eléctrica y electrónica del Instituto Tecnológico de Sonora, Volumen 1, Número 1, pp.18-21, 2005.
- [2] E. V. Terzi, H. Regber, C. Löffler, F. "*Controles Lógicos Programables. Nivel Básico*". Festo Didactic. 2000.
- [3] Mendoza J., Cortés J., Muriel J. "*Control secuencial de un circuito electroneumático a través de un PLC*", Scientia Et Technica, vol. XVI, núm. 48, pp. 191-195, 2011.



Diseño de un Cicloconvertidor para un Sistema Mecatrónico de Rehabilitación Motriz de Rodilla

Navarrete Hernández Gabriel Antonio, Galván Chávez Jorge Moisés y Grande Olvera Daniel

Universidad Tecnológica del Centro de Veracruz
Jorge.galvan@utcv.edu.mx

Resumen

Este artículo muestra el diseño de un ciclo convertidor, el cual es una parte fundamental del control de un sistema mecatrónico para la rehabilitación de la rodilla. Muestra el desarrollo con detalles de los métodos para obtener los parámetros operativos óptimos con la ayuda de un Analizador / Fuente de Voltaje de Agilent Technologies modelo HP 6813B. Mediante una señal PWM programada en una tarjeta Arduino, es posible emular el comportamiento de la onda sinusoidal de la corriente alterna, logrando así una potencia óptima para el motor monofásico. A través de la electrónica de potencia, fue diseñar una fuente conmutada para los disparos de tiristores, que están conectados en una configuración de puente H. Se integra también una interfaz hombre-máquina que está diseñada para interactuar con la variación de velocidad en el motor, y así garantizar que el sistema se pueda automatizar y monitorear desde una aplicación de software.

Palabras clave: Sistema mecatrónico, rehabilitación, ciclo convertidor.

1. Introducción

Una parte fundamental dentro del desarrollo del Sistema Mecatrónico para la rehabilitación motriz de Rodilla es el diseño de la parte electrónica, la cual se encargara del accionamiento del motor, el control de la velocidad del sistema, del control de la fuerza aplicada por parte del dispositivo a la pierna de los pacientes, el control de posición para determinar el grado de flexión y extensión, de registrar la fuerza que cada paciente aplica y la comunicación entre la computadora y el prototipo.

Para el control de velocidad del prototipo se diseñó e implemento un ciclo convertidor, con el cual se controlará la velocidad de motor, al hacerlo se controlará la velocidad con la que el dispositivo flexiona la rodilla del paciente, ya que el motor eléctrico es el que proporciona la potencia necesaria para hacerlo.

El motor a controlar es un motor monofásico de inducción con arranque por condensador, su corriente nominal es de 1.5 A, su voltaje de alimentación es de 120 V a 60 Hz con una velocidad nominal es de 1750 rpm y una potencia de 1/20 HP. Este motor, dentro del sistema, este acoplado a una caja de engranes con la cual se reduce su velocidad y se aumenta su torque, la relación de reducción es 200:1.

Para el control de posición se utiliza un encoder incremental de cuadratura Hp modelo HEDS-5540, el cual tiene una resolución por canal de 1024 bits, cuenta con dos canales principales denominados A y B y un tercer canal que ayuda a encontrar la posición inicial denominado Índice; este encoder dentro del prototipo es instalado en el eje de la caja reductora de velocidad.

Para la parte de control se utilizará un Arduino Due el cual cuenta con un procesador de 32 bits a 84 MHz. ARM Cortex-M3 denominado SAM3X8E de Atmel. Este poderoso procesador le permite adquirir 1000 Kilo muestras por segundo, y una de sus peculiaridades por las cuales fue seleccionado es que permite trabajar varios procesos en paralelo.

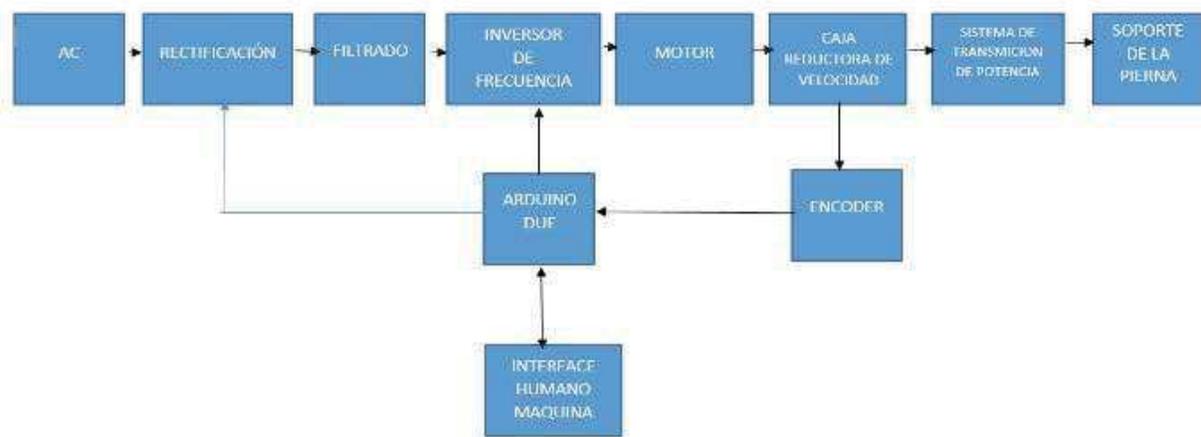


Figura 1. “Diagrama a bloques del sistema electrónico”

2. Ciclos convertidores

En la actualidad existen varios tipos de convertidores de potencia, están los que convierten corriente alterna en corriente directa (C.A-C.C), llamados rectificadores, también existe el convertidor de corriente directa a corriente directa (CC-CC), cuyo único fin es aumentar o disminuir el voltaje, también encontramos convertidores de corriente alterna a corriente alterna (CA-CA), los cuales a la salida proporcionan una tensión en corriente alterna controlada por una tensión de corriente alterna, estos no solo varían la intensidad de voltaje algunos también varían la frecuencia de trabajo y son conocidos como ciclo convertidores. Y por último tenemos convertidores de Corriente directa a corriente alterna, conocidos como inversores [1].

Este tipo de sistemas son ampliamente utilizados en una gran gama de aparatos electrónicos, ya que en estos días gracias al aumento del uso de estos aparatos se hace necesario transformar la corriente para ajustarla a las necesidades de cada dispositivo, ocasionando con esto cambiar de C.A a CD, comúnmente, de ahí su gran importancia; entre las aplicaciones que destacan, son su uso en control de motores.

En el caso del sistema mecatrónico desarrollado para poder controlar la velocidad del motor de inducción monofásico se debe convertir la corriente de la alimentación de CA en primer lugar a CD, el nivel de potencial de CD debe ser regulado, y después convertir de CD a CA, pero la frecuencia de CA debe ser modulada, lo que da como resultado un Cicloconvertidor CA a CA. El cual proporciona a la salida un potencial regulado y una frecuencia de trabajo modulada.

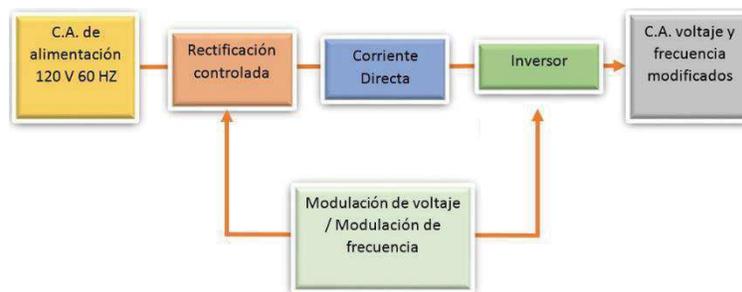


Figura 2. Diagrama a bloques de las etapas del Cicloconvertidor.



3. Obtención de parámetros de diseño del Cicloconvertidor

Un punto importante del este trabajo fue la obtención de los parámetros de diseño óptimos para el cicloconvertidor del sistema mecatrónico para la rehabilitación motriz de rodilla, por lo que se hizo uso de una fuente de alimentación CA / analizador de Agilent Technologies modelo HP 6813B, la cual tiene una capacidad de voltaje de salida de 1700 VA, permitiendo una variación de frecuencia de 0 a 1kHz. Dentro de sus funciones de analizador, se puede medir el factor de potencia, la corriente, la potencia en watts, los armónicos, entre otros. Se analizaron varias combinaciones de parámetros, con la ayuda de la fuente de voltaje/analizador. En la prueba 1 se alimentó al motor con un voltaje de 120 V y este se mantuvo constante, mientras se disminuía la frecuencia de trabajo. A continuación, se muestran los resultados.



Figura 3. Fuente de voltaje / Analizador Agilent Technologies modelo HP 6813B.

Tabla 1. Prueba 1 Voltaje constante a 120V.

PRUEBA 1: Voltaje constante a 120V	
Frecuencia	Corriente
60 Hz	1.08 A
55 Hz	1.28 A
50 Hz	1.78 A
45 Hz	2.28 A

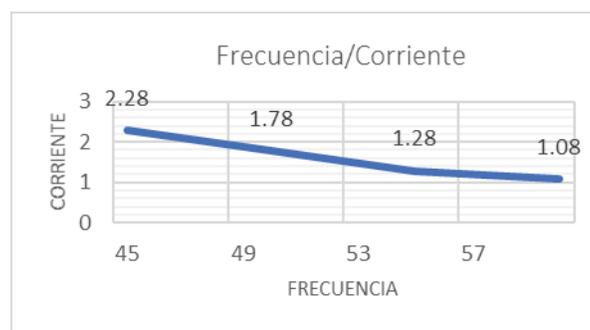


Figura 4. Gráfica relación frecuencia/ corriente

Dentro de los resultados de esta prueba se pudo ver claramente que al disminuir la frecuencia de trabajo y dejar constante el voltaje de alimentación, la corriente presentaba un aumento, por lo que no fue posible seguir disminuyendo la frecuencia más haya de 45Hz, ya que el motor se sobrecalentó a estos valores de frecuencia y corriente.

En la prueba 2 se mantuvo una relación V/F constante, es decir los parámetros iniciales fueron 120 V a 60 Hz lo que da una relación de 2, por lo que, al reducir la frecuencia a 55 Hz para mantener la relación constante, el voltaje se decremento a 110 V. como se muestra en la siguiente tabla.



Tabla 2. Prueba 2: relación V/F constantes.

Voltaje/Frecuencia	Tiempo en dar una vuelta	Corriente	RPM
120 V-60Hz	6.92s	1.08 A	1750 rpm
110 V-55Hz	7.56s	1.06 A	1600 rpm
100 V-50Hz	8.28s	1.13 A	1500 rpm
90 V-45Hz	9.28s	1.2 A	1330 rpm
80 V-40Hz	10.35s	1.4 A	1160 rpm
70 V-35Hz	12.1s	1.5 A	1000 rpm
60 V-30HZ	14.2s	1.6 A	870 rpm
50V-25HZ	17.5s	1.65 A	700 rpm
40 V-20HZ	23s	1.61 A	520 rpm

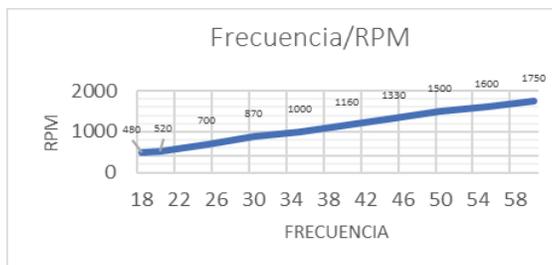
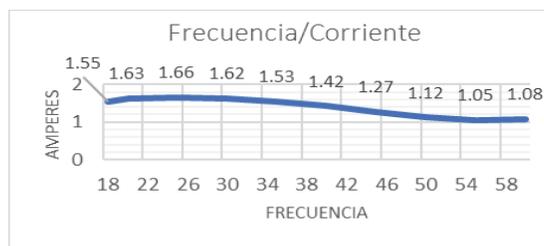


Figura 5. Graficas de resultados de la prueba 2.



En la prueba 2 se puede observar que al variar la relación V/F de manera uniforme, la corriente se mantiene relativamente estable, no sobrepasando el límite de 1.6 amperes. Lo anterior permite que la frecuencia se disminuya hasta los 20 Hz dando como resultado una baja velocidad, necesaria para la aplicación del sistema desarrollado, sin embargo, la respuesta de la variación de velocidad no es lineal como se puede observar en la gráfica.

En la prueba número 3, se buscó un valor de voltaje en el que el factor de potencia de la fuente fuera alto, de ésta manera a 55 v se observó que el factor de potencia era de 0.97 con una corriente de inicio de 0.78 A, y con una frecuencia inicial de 60 Hz, por lo que se procedió a reducir el valor de la frecuencia manteniendo el voltaje a 55 V constantes, a continuación, se muestran los resultados.

Tabla 2. Prueba 2: relación V/F constantes.

Frecuencia	RPM	Corriente	Tiempo	Potencia
60 Hz	1680	0.71 A	7.1s	38 w
	RPM			
55 HZ	1540	0.72 A	7.8s	38.5 w
	RPM			
50 HZ	1400	0.726 A	8.6s	36 w
	RPM			
45 Hz	1260	0.81 A	9.5s	36.5 w
	RPM			
40 Hz	1120	0.96 A	10.7s	38 w
	RPM			
35 HZ	980 RPM	1.19 A	12.3s	42 w
30 Hz	840 RPM	1.48 A	14.4s	50 w
25 Hz	700 RPM	1.8 A	17.4s	65 w
24 Hz	660 RPM	1.95 A	18.2s	70 w

De acuerdo a los resultados mostrados de la prueba 2, se observa que la respuesta del motor a la variación de frecuencia manteniendo un voltaje estable de 55 V, es más estable, ya que la corriente aumenta pero en ningún momento sobrepasa los 2 amperes, la respuesta de velocidad es lineal y proporcional a la variación de la frecuencia, por lo que se establece que los parámetros en los que trabajará el cicloconvertidor serán: 55 v, de 24 a 60 Hz, y una corriente máxima de trabajo de 1.9 A.

4. Diseño electrónico.

Para el diseño del cicloconvertidor, una parte fundamental es la regulación de la amplitud del voltaje de alimentación, así como la rectificación de la CA de entrada, esto con el fin de implementar un control V/F para la variación de velocidad, y con el fin de poder establecer un valor de voltaje menor al de voltaje de alimentación.

4.1. Etapa de rectificación y control de voltaje

Los diodos no son los únicos dispositivos electrónicos con los que se puede lograr rectificar la corriente alterna, otros semiconductores muy utilizados en la actualidad para lograr hacer esta tarea son los tiristores, los cuales tienen la característica de poder conducir la corriente en un solo sentido, al igual que un diodo, pero es necesario activarlo, esto se hace al pasar una corriente de activación a través de una de sus terminales llamada puerta y se desactiva por conducción natural o de línea.

Al ocupar este tipo de dispositivos semiconductores se logra hacer la denominada rectificación controlada; la rectificación a través de diodos solo proporciona un voltaje de salida fijo, al utilizar tiristores de control de



fase se pueden obtener a la salida voltajes controlados, es decir, se puede controlar el nivel de tensión de salida, controlando el Angulo de disparo de los tiristores.

El uso y las configuraciones para lograr la rectificación del voltaje de alimentación son muy parecidas a las hechas con los diodos, para el cicloconvertidor se optó por una configuración de rectificador semicontrolado como se muestra en la siguiente figura.

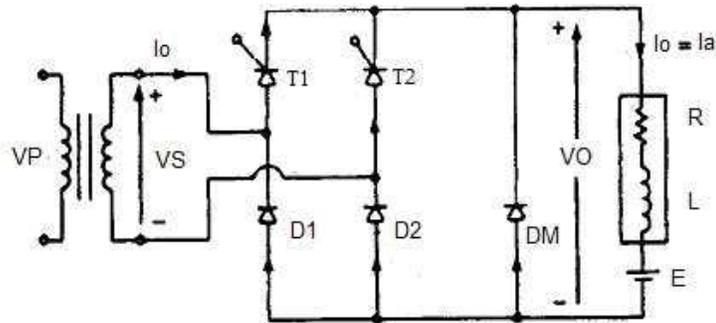


Figura 6. Circuito de rectificación controlada con tiristores.

Como se aprecia en la figura 6, se usa una configuración puente en la que se combinan dos tiristores denominados T1 y T2 con dos diodos D1 y D2 las formas, al disparar T1 con un ángulo α en el medio ciclo positivo, T1 y D2 estarán en polarización directa permitiendo pasar la corriente a la carga durante el ángulo de conducción, en el semiciclo negativo T2 y D1 serán los que hagan esa función. Las formas de ondas generadas se muestran en la figura 7.

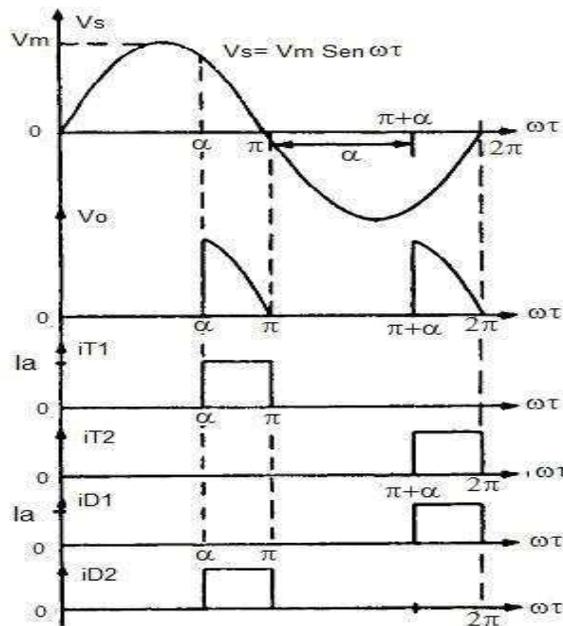


Figura 7. Señales de salida obtenidas del circuito de control de ángulo de conducción con tiristores.

Los tiristores seleccionados para el circuito a implementar fueron SCRs C106D, ya que soportan un voltaje de 200 a 600 V con una corriente de conducción de 4 Amperes, nuestro motor trabaja en un rango de 120 V con corrientes máximas de arranque de 2.5 Amperes, por lo que estos tiristores entran perfectamente



dentro de los parámetros de trabajo. Para poder disparar este SCR, de acuerdo a los datos proporcionados por el fabricante, es necesario hacer pasar una corriente de 0.1 amperes por su terminal de puerta.

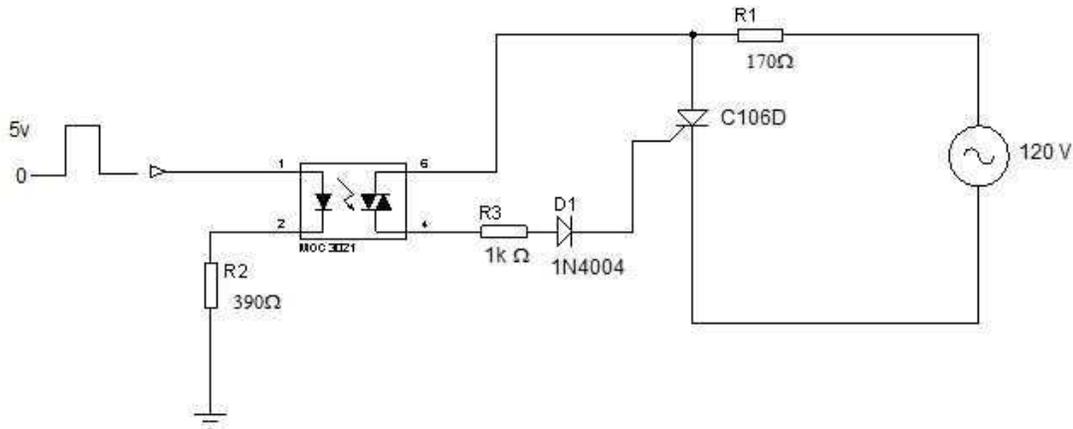


Figura 8. Circuito de disparo de un SCR por medio de un optoacoplador.

Para calcular el valor de resistencia se aplica la ley de ohm, la cual dice que la corriente es directamente proporcional al voltaje e inversamente proporcional a la resistencia.

$$I = V/R \tag{1}$$

Por lo tanto, la resistencia de puerta necesaria es de 1.2kΩ, la señal de disparo será enviada desde un Microcontrolador por lo cual es necesario utilizar un opto acoplador, en el diseño se utilizó un moc3011 que es salida a triac como se muestra en la figura 8. El Microcontrolador que se utilizará, como ya se había mencionado antes, será un Arduino Due, cuyas salidas y entradas digitales trabajan a 3.3 v, el moc3011 de acuerdo a las especificaciones del fabricante para poder disparar la puerta del optotriac es necesario hacer pasar por el fotodiodo una corriente de 12 mA, por lo que realizando el cálculo de la resistencia de entrada será de un valor de 270 kΩ.

El detectar el cruce por cero ayuda a sincronizar las señales de disparo de los SCR's con la onda sinusoidal de corriente alterna. El circuito de la figura 9 es un circuito con el cual se detecta cuando la señal pasa por cero, esta señal es enviada al Microcontrolador para que cada vez que registre un cruce mande la señal de disparo con su respectivo retardo en caso de querer regular el voltaje. Si se quiere tener la onda completa en el semiciclo positivos, en el momento que se detecta el cruce por cero se dispara el SCR a través de la señal de mando del Microcontrolador, para que toda la media onda pase por medio de él.

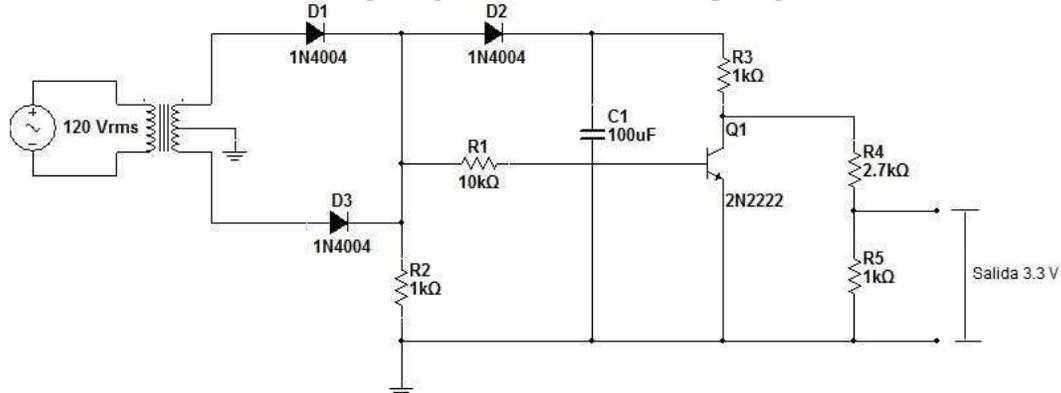


Figura 9. Circuito detector de cruce por cero

4.2. Inversor de frecuencia con puente h completo

Una vez obtenido un valor de voltaje de CD variable es necesario, construir un puente H con el cual será posible convertir la Corriente directa a corriente alterna, existen dos configuraciones para un puente H, una



es denominada medio puente y la otra como puente completo. La diferencia entre ambas radica en el número de dispositivos semiconductores ocupados para la conmutación, en el caso de un medio puente se ocupan 2 y en una configuración de puente completo se utilizan 4; los dispositivos de conmutación pueden ser transistores, Mosfets o IGBTs.

Una cuestión muy importante a la hora de diseñar un puente H son las señales de disparo para los mosfet, esos dispositivos son usados como interruptores controlados, este control se logra gracias a su terminal de puerta, la cual debe ser excitada, para hacerlo es necesario tener un voltaje de puerta-drenado entre un rango, dependiendo del fabricante, de 10-30 volts típicamente. Pero por la naturaleza del circuito las tierras deben tener referencias diferentes, ya que si se tuvieran en toda la misma referencia al conmutar los mosfet ser provocaría un cortocircuito dañando los dispositivos conectados a la alimentación.

Por ellos se hace necesario diseñar una fuente que proporcione los voltajes, pero con referencias distintas o como de manera común se conoce, que proporcione tierras flotantes, debido a esto fue necesario diseñar una fuente conmutada que tenga cuatro salidas de voltaje del 12V cada una, para alimentar las puertas de los mosfet y que estos entren en conmutación.

El diseño de la fuente es una Fli Back, su principio de funcionamiento se basa en conmutar a una alta frecuencia el primario de un transformador, al hacerlo en el secundario se induce una corriente, su valor estará determinado por la relación que exista entre las vueltas del primario y secundario y por el ciclo de trabajo del mosfet. Como se indica en la ecuación.

$$V_o = \left(\frac{N_s}{N_p} \right) \left(\frac{D}{1-D} \right) \quad (2)$$

Donde:

V_o = voltaje de salida.

N_s =Numero de vueltas del secundario.

N_p =Numero de vueltas del primario.

D = Ciclo de trabajo.

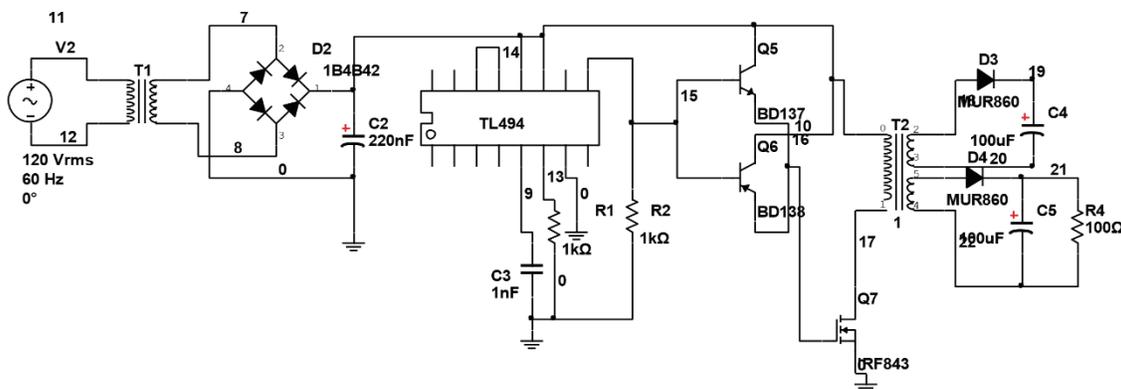


Figura 10. Diagrama del sistema electrónico de rectificación y señal PWM de control.

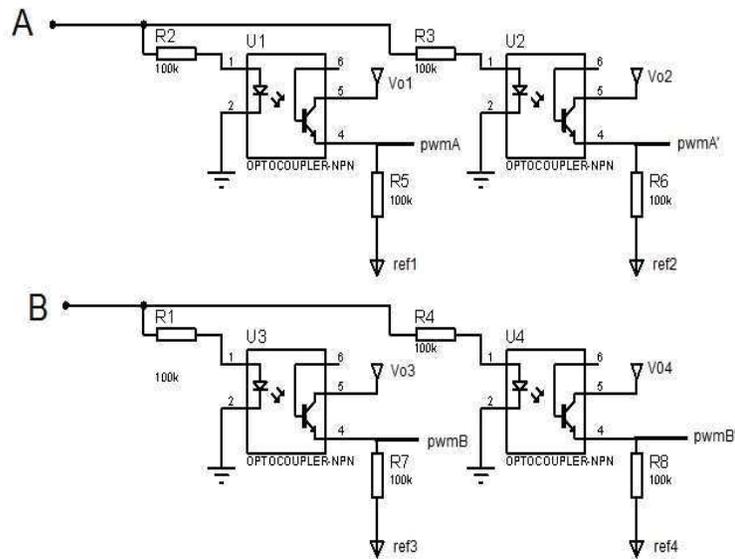
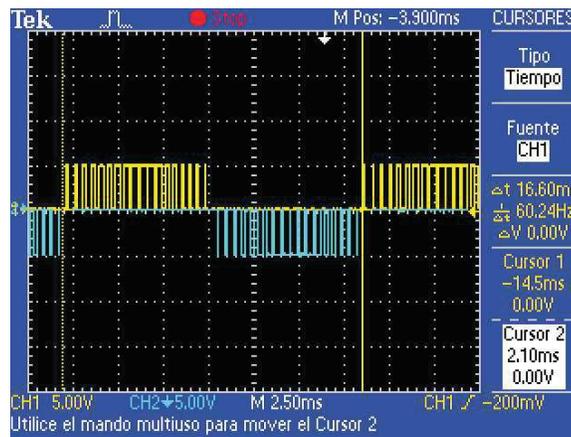
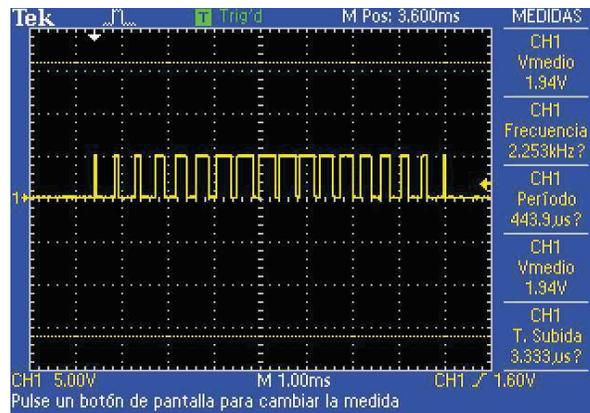


Figura 11. Circuito para el acoplamiento óptico

Las señales de salida que proporciona la fuente conmutada son cuatro, Vo1, Vo2, Vo3, Vo4, cada una con su respectiva referencia, ref1, ref2, ref3, ref4. Estas señales de voltaje van conectadas a un circuito optoacoplador, para poder acoplar las señales de PWM enviadas por el microcontrolador a las señales de la fuente conmutada y proporcionar un PWM con valores de voltaje de 12 v, ya que el potencial de salida del microcontrolador solo es de 3.3 v. el diagrama del circuito optoacoplador se muestra en la figura

Las salidas del tótem-pole son conectadas a las terminales de puerta y drenado de cada mosfet, la señal pwm A se conecta a la puerta del mosfet Q1 y su referencia ref1 a la terminal de drenado, la señal pwmA' se conecta al mosfet Q4, ya que estos dos serán conmutados al mismo tiempo. La señal pwmB y pwmB' son conectadas a los mosfet Q3 y Q2 respectivamente como se muestra en la figura.

Esta técnica consiste en emular el comportamiento de una onda senoidal a través del control de ancho de pulsos, en intervalos de tiempo, consiste en una señal de nueve pulsos por cuarto de ciclo y cada pulso debe ir aumentando su duración hasta llegar al cuarto de ciclo donde el siguiente cuarto de ciclo, los pulsos van en decremento. Para poder aplicar esta señal al inversos de puente completo es necesario generar dos señales, pero con un desfase. En el sistema se implementó esta técnica a través de un microcontrolador a continuación la simulación de sus salidas.



5. Resultados

A través de una interfaz diseñada en Visual Studio, y

Con la ayuda del microcontrolador, se hizo posible interactuar y controlar la velocidad de la motocicleta monofásica que controlaría la posición de la rodilla del paciente.

La Figura 15 muestra la interacción completa de los diferentes dispositivos involucrados en la construcción del cicloconvertidor. Es importante tener en cuenta que gracias a la comunicación en serie del microcontrolador es posible con una computadora monitorear el estado del sensor de velocidad, para poder implementar en un trabajo futuro un algoritmo de control más sofisticado.

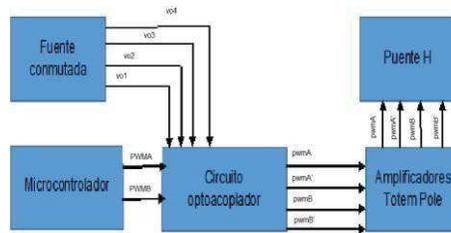


Figure 12. Diagrama a bloques de las etapas del sistema.

La figura 13 muestra la interfaz gráfica diseñada en visual studio, la cual registra los datos provenientes de una tarjeta arduino DUE. El Sistema de control mediante optoacopladores es posible con el uso de una señal PWM la cual activa a estos últimos, la magnitud de la señal se presenta por las barras indicadoras que se aprecian en la interfaz gráfica.

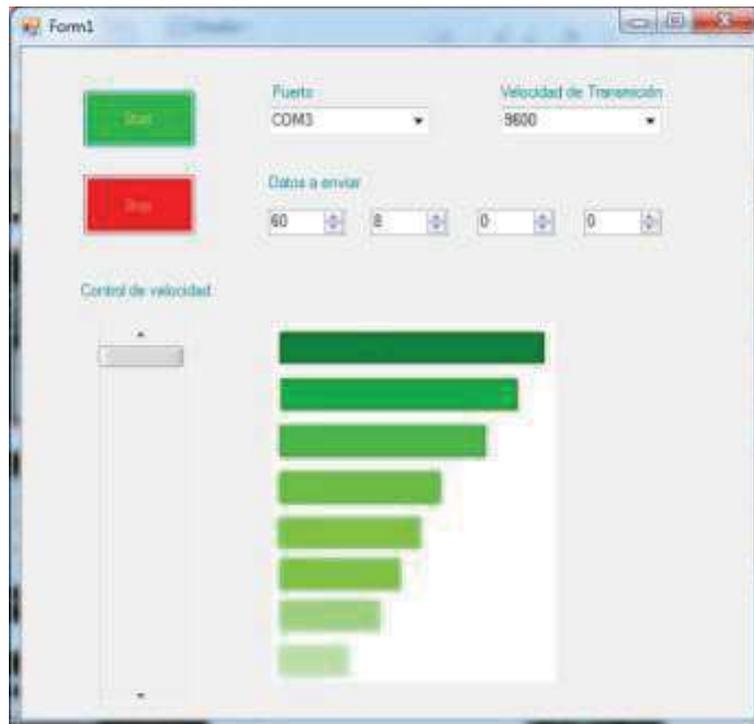


Figura 13. Interfaz gráfica del control mediante la señal PWM.

Se llevo a cabo el desarrollo de los circuitos impresos para cada etapa del sistema, en la figura 14 se aprecian las tarjetas electrónicas que se diseñaron y construyeron para el correcto funcionamiento del sistema.



Figura 14. Etapas del sistema con los componentes físicos.

6. Conclusiones

El diseño de un cicloconvertidor para un sistema de rehabilitación ha sido de gran aportación para controlar el motor que forma de parte fundamental de un sistema para la rehabilitación de rodilla. El trabajo alcanzado permitió el estudio y desarrollo de la implementación de un control de motor mediante una señal PWM haciendo uso de la tarjeta Arduino DUE, que se caracteriza por una mayor frecuencia de trabajo y mayor número de bits de resolución con respecto a las diferentes gamas de tarjetas Arduino. El trabajo desarrollado se integrará con una metodología de trabajo para realizar pruebas de operación con pacientes que requieran una rehabilitación de rodilla, y con ello validar el correcto funcionamiento del sistema y su eficiencia.

7. Referencias

- [1] Lünenburger L., Colombo G., Riener R.; “Biofeedback for robotic gait rehabilitation” *Journal of NeuroEngineering and Rehabilitation*; 23 January 2007, 4:1.p 1-11.
- [2] Organización Mundial de la Salud. (2001). *Clasificación Internacional del Funcionamiento de la Discapacidad y de la Salud (CIF)*.
- [3] Erhan Akdoğan • Ertuğrul Taçgın • M. Arif Adli, “Knee rehabilitation using an intelligent robotic system”, *Journal Intelligent Manufacturing* ; 8 January 2009, 20; p. 195–202.
- [4] Muhammad H. Rashid, *Electrónica de potencia, (circuitos, dispositivos y aplicaciones)*, Prentice Hall, México, 1993.
- [5] Ernesto Bárcenas; *Análisis y desarrollo de un inversor multinivel*, Tesis, Centro Nacional de Investigación y Desarrollo Tecnológico; Diciembre 2002, Cuernavaca, Morelos.
- [6] (2009). En T. M. Ned Mohan, *Electrónica de Potencia: convertidores, aplicaciones y diseño*. México: McGrawHill.
- [7] Jorge L. Díaz Rodríguez, L. D.-F. (2015). Novel Methodology for the calculation of transformers in Power multilevel converter. *Ingeniería y Competitividad*, 121-132.
- [8] Juan D. Gallego Gómez, J. B.-Q.-G. (2014). Analysis of Inefficient Power Flows Produced by Power Converters. *Información Tecnológica*, 93-102.



Automatización de un Invernadero para el Cultivo de Hongos

Ávila Mata Estela Nataly, Rentería Ramírez Luis Felipe, Aguilar Molina Yehoshua
y Brizuela Mendoza Jorge Aurelio

Universidad de Guadalajara, Centro Universitario de los Valles.

Contacto: pipeh1@hotmail.com

Resumen

Este documento presenta el protocolo planteado para el desarrollo del proyecto “automatización de un invernadero para el cultivo de hongos”, describe de manera general cada una de las etapas necesarias, y que se realizaran a lo largo del curso de diseño mecatrónica, para el diseño e implementación de una opción de automatización para un invernadero, por medio del diseño de un sistema de control y monitoreo de las variables que intervienen en esta actividad. Este proyecto es el resultado de la observación de algunas de las actividades que se utilizan actualmente para el cultivo de hongos, y tiene como principal objetivo; ofrecer a los productores una demostración de los recursos disponibles para mejorar la eficacia de la producción de hongos comestibles y obtener el mejor rendimiento posible con la implementación un invernadero inteligente.

Palabras Clave: Automatización, invernadero, hongos, cultivos, diseño, electrónica

1. Introducción

Los sistemas automáticos para la generación de microclimas en invernaderos, se han venido utilizando cada vez durante la última década, con el fin de implementar las mejores condiciones posibles para el cultivo y aumentar la producción. Estos sistemas permiten a los productores poder cultivar en distintas regiones con climas y condiciones ambientales distintos.

Uno de los cultivos dentro de invernadero que ha tomado auge en los últimos años es el de hongos comestibles, debido a la creciente demanda por la tendencia de consumir productos orgánicos. El cultivo de estos productos dentro de un invernadero, suele ser muy diferente a lo que se acostumbra en cultivos de frutas y vegetales, debido a las características específicas de los hongos y setas, así como a sus necesidades de reproducción

El presente proyecto está enfocado al desarrollo de un invernadero inteligente por medio de la automatización de las condiciones de un invernadero dedicado al cultivo de hongos comestibles. La propuesta de automatización tiene como objetivo mejorar el rendimiento en la producción de hongos y controlar mayoritariamente las variables involucradas en este proceso, por medio del diseño de un sistema de control y la implementación de una simulación a escala de una sala de cultivo. Este trabajo busca demostrar las posibilidades de optimización de un cultivo, poniendo en práctica algunos de los recursos estudiados a lo largo de la carrera de Ing. Mecatrónica.



2. Planteamiento del Proyecto

En la actualidad el consumo de productos orgánicos como los hongos y setas ha venido en aumento, y para abastecer la demanda de estos productos, se han establecidos sistemas de cultivo y producción de hongos, por medio de invernaderos. El cultivo de hongos en invernaderos suele necesitar de la intervención humana en todo momento para cubrir cada una de las necesidades del producto durante la etapa de cultivo. Los invernaderos que no cuentan con sistemas de monitoreo y automatización, tienen una producción limitada y parcialmente dependiente de agentes externos; además de que son más susceptibles a producir solo en periodos cortos, dependiendo de las condiciones climáticas del entorno, generando pérdidas temporales e incluso el abandono. Así, la implementación de un microclima adecuado para el cultivo de hongos comestibles se convierte en un reto particular. La producción de hongos para su venta inmediata también se ve limitada y dependiente, debido a que el monitoreo del estado de las plantas solo se hace en pocos periodos y por medio del personal de los invernaderos.

Este proyecto tiene como objetivo la optimización de un invernadero para el cultivo de hongos comestibles, por medio de la implementación del mejor ambiente posible para esta actividad, desarrollando un sistema de monitoreo de las condiciones dentro de la sala de cultivo y el control de las variables que intervienen dentro del proceso. También, busca automatizar el proceso riego, siendo este, uno de los factores principales para obtener plantas sanas y por consecuencia mejorar la productividad. En general, se pretende presentar una opción más confiable y eficaz, para aumentar la productividad de un invernadero, reduciendo costos y personal y ofreciendo la posibilidad de un sistema con capacidad de trabajo 24/7.

2.1 Objetivo

Objetivo general:

Diseñar e implementar un prototipo para el monitoreo y automatización de un invernadero dedicado al cultivo de hongos comestibles.

Objetivos específicos:

- Implementar una simulación del microclima ideal para el cultivo de hongos comestibles.
- Diseñar un sistema de control, considerando las variables de temperatura, humedad e iluminación.
- Diseñar e implementar una interfaz de monitoreo de las variables que intervienen en el proceso.
- Proponer un sistema de automatización de riego, para el cultivo de hongos.



2.2 Antecedentes de la automatización y control en los invernaderos.

Un **invernadero inteligente** tiene la capacidad de controlar las variables ambientales que afectan al cultivo. Enumeramos los principales sistemas controlados en un invernadero inteligente:

Control climático: Se instalan dos estaciones meteorológicas, una en el interior para controlar los parámetros climáticos del cultivo, y otro en el exterior para el control del clima externo con objeto de realizar las operaciones pertinentes como el cierre de la ventilación en caso de lluvia, o fuertes vientos.

Control del riego y la aplicación de nutrientes: Controla la **frecuencia del riego** y la **aplicación de nutrientes** mediante una **programación** impuesta por el agricultor o el técnico de la explotación. La programación de la aplicación de nutrientes se realiza a partir de la **programación del riego**, programando un equilibrio nutricional concreto para cada estado fisiológico del cultivo.

Control de la temperatura: El control de la temperatura se realiza mediante **sondas de temperatura** instaladas en una **estación climática** en el interior del invernadero. A partir de la medida de la temperatura se programa una serie de actuadores dependiendo de la misma. Así podemos encontrar entre los automatismos mecanismos de apertura y cierre de **ventanas cenitales y laterales** y **ventiladores** para provocar la bajada de temperatura en el interior del invernadero y **sistemas de calefacción** para aumentar la temperatura.

Control de la humedad: La **humedad relativa** es monitoreada en la **estación climática del interior del invernadero** y actúa sobre el funcionamiento de sistemas de nebulización para aumentar la humedad o sobre los sistemas de ventilación forzada para evacuar el aire demasiado húmedo del invernadero.

Control de la iluminación: La iluminación se controla mediante el accionamiento de mecanismos que extienden **pantallas de sombreo** instaladas normalmente en el interior del invernadero que **reducen la radiación incidente** sobre el cultivo cuando esta es demasiado elevada, lo que evita que se produzcan quemaduras en las hojas de las plantas.

2.2.1.- Diseño de un sistema de monitoreo y control de cámaras de incubación y fructificación para la producción de hongos comestibles y medicinales (Bolivia, 2017)

Resumen:

La producción agrícola para consumo humano es el punto de inicio para este proyecto. Actualmente la automatización está presente en muchos rubros, ya que ello conlleva a tener datos cuantificados más confiables y de mayor exactitud que nos ayudan enormemente. Este proyecto se basa fundamentalmente en la adquisición, visualización, almacenamiento de datos y control para una óptima producción.

El presente proyecto pretende aportar a la sociedad una nueva forma de producción agrícola -controlada y monitoreada electrónicamente- y hacer posible que los hongos comestibles sean parte de la dieta alimentaria de la población.

Tradicionalmente, Cochabamba ha sido una región agrícola variada de cereales, tubérculos y hortalizas, entre otros. Sin embargo, la producción de hongos no es parte de la producción agrícola común, dado que los hongos requieren -principalmente en su etapa micelial- condiciones especiales para su fructificación. Es así que es necesario un descenso de temperatura y aumento de humedad relativa elevada e iluminación, pues esto hace que su producción cumpla con ciertas especificaciones. En la actualidad, y gracias a la tecnología, se pueden recrear



condiciones atmosféricas de manera artificial en ambientes controlados para poder así cultivar hongos comestibles y medicinales.

2.2.2.- Automatización de Clima Para un Invernadero (México, 2014)

Resumen

Este trabajo se realizó con el fin de proponer un diseño de automatización para los principales sistemas de un invernadero ubicado en Cuajimalpa D.F. Anteriormente el invernadero estuvo en funcionamiento sin equipos automáticos para la producción de verduras, en el se cultivó aproximadamente solo por un año desde su construcción, para poder desarrollar la propuesta de diseño fue necesario hacer el levantamiento general del invernadero, analizar la red eléctrica para la adaptación de equipos digitales y mecánicos, así mismo se diseñó la propuesta de la red de comunicación Devicenet para la interacción entre mando de control y actuadores, el combustible propuesto para la generación de energía calorífica es el biogas que de acuerdo a investigación es menos contaminante que otros gases y ya que el horticultor tiene su propio biodigestor donde producirá biogás; se implementó el diseño del sistema de calefacción para el uso de este gas, esto da una ventaja de no tener que comprar otro combustible para la generación de calor. La propuesta se desarrolló con el objetivo de reacondicionar y utilizar el invernadero el cual tiene un área de 495 m² que actualmente está sin ocuparse y por lo tanto sin producción, lo que generara una pérdida de material y construcción del bien si no se readapta para producción. Se ha propuesto el sistema Automático de calefacción de acuerdo a las condiciones ambientales de la zona, que mayor parte del tiempo es fría y esto propicio principalmente a dejar de cultivar por afectaciones en los vegetales, como ejemplo se ha diseñad o el sistema para la producción de jitomate tomando en cuenta las variables que conllevan a su desarrollo. Los resultados que se esperan obtener es aumentar la producción de verduras en el invernadero en comparación con las que se obtuvieron anteriormente sin sistema automático, ya que el sistema operaria todo el año sin afectaciones del tiempo invernal.

2.2.3.- Control de temperatura y humedad para cultivo de hongos comestibles SHITAKE (2011)

Resumen

La seguridad alimentaria requiere investigación y desarrollo para transferir y adoptar tecnología que haga eficiente el sistema productivo agrícola rural y urbano. En particular, se ha documentado que los hongos comestibles shiitake (HCS) son una fuente de aminoácidos, vitaminas, minerales, ácidos grasos insaturados, favorables al consumo humano por sus bajas concentraciones de colesterol; y actúan además como degradadores de materia orgánica en los ciclos biológicos, lo que preserva cualquier ecosistema. El presente documento describe la construcción de una sala para cultivo de HCS, controlada por medio de sensores de temperatura y humedad, de manera inalámbrica. El envío y recepción de datos desde los sensores se enrutan a un microcontrolador para que este, por vía USB, se conecte con el computador y se visualicen los datos en pantalla a través de una interface de usuario, programada en Java, desde donde se monitorean las condiciones de temperatura y humedad ideales para el proceso reproductivo.



2.3 Requerimientos Técnicos

A continuación, enlistamos algunos de los materiales necesarios para la implementación del proyecto, estos complementaran la construcción de la estructura basada en un diseño inicial (Figura 1).

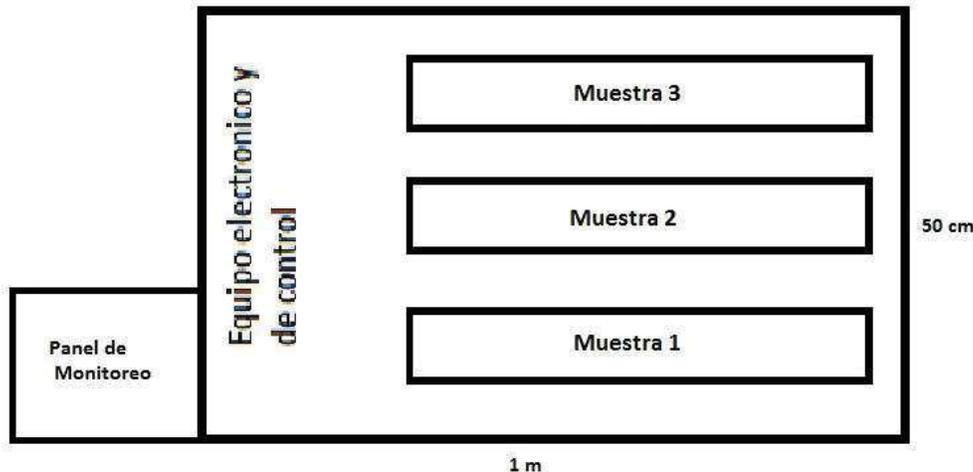


Figura 1. Diseño inicial propuesto de la maqueta

Los componentes siguientes, son principalmente electrónicos y fueron seleccionados de acuerdo a nuestro primer acercamiento al diseño del proyecto, con la posibilidad de aumentar el número de componentes e incluso sustituir algunos en el futuro si es necesario. Se describen las características generales de los materiales seleccionados hasta el momento.

Tabla 1. Materiales Necesarios

Componente	Características
Arduino Mega	El Arduino Mega es una placa microcontrolador basada ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset.
Mini bomba	Mini Bomba de agua sumergible sin escobillas, cabezal de elevación de hasta 2.5 metros, 12V DC.
Sensor de temperatura y humedad	El módulo DHT11 dispone de una salida digital exclusiva para la adquisición de señales digitales, mediante este pin se realiza el muestreo sincronizado de los datos. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante del medio ambiente
Fuente de alimentación	Fuente de poder vorago atx 500w psu-101
Ventiladores	Ventilador de computadora que opera a 12v.
Herramientas	Cables, pinzas, caudín, soldadura, etc.
Materiales para la construcción de la estructura	Madera, cable, alambre, maya-sombra, plástico, varillas, etc.



3. Implementación y Resultados

3.1 Operación del sistema:

A continuación, se presenta el funcionamiento esperado del sistema de control por medio de un diagrama de flujo:

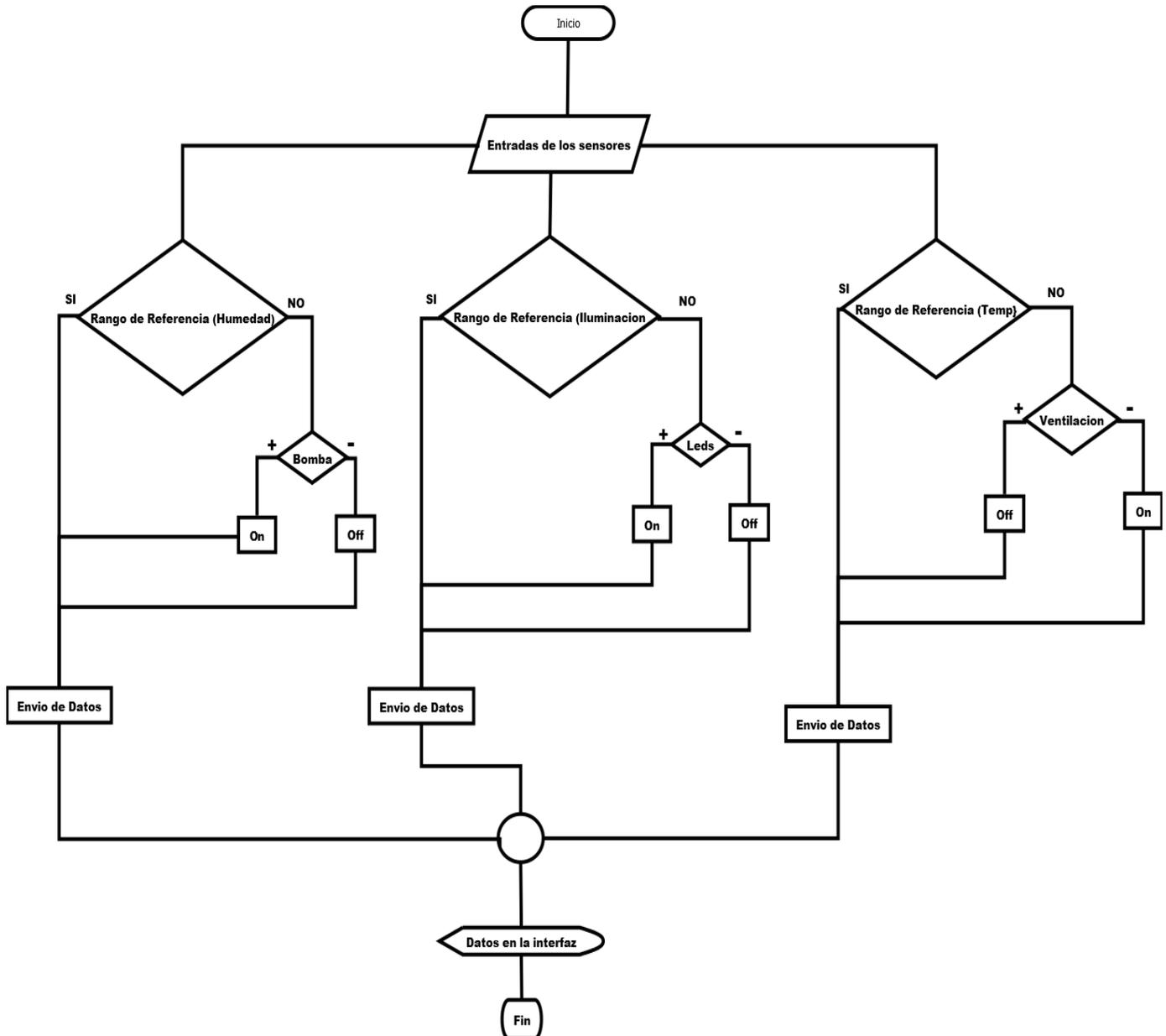


Figura 2. Diagrama de funcionamiento

3.2 Etapas del Proyecto

Etapa de Diseño: Durante esta etapa se plantearon algunos bocetos para la distribución de los componentes de la maqueta que se construyó. En la siguiente figura se observa el diseño final, así como la ubicación de cada una de las partes.

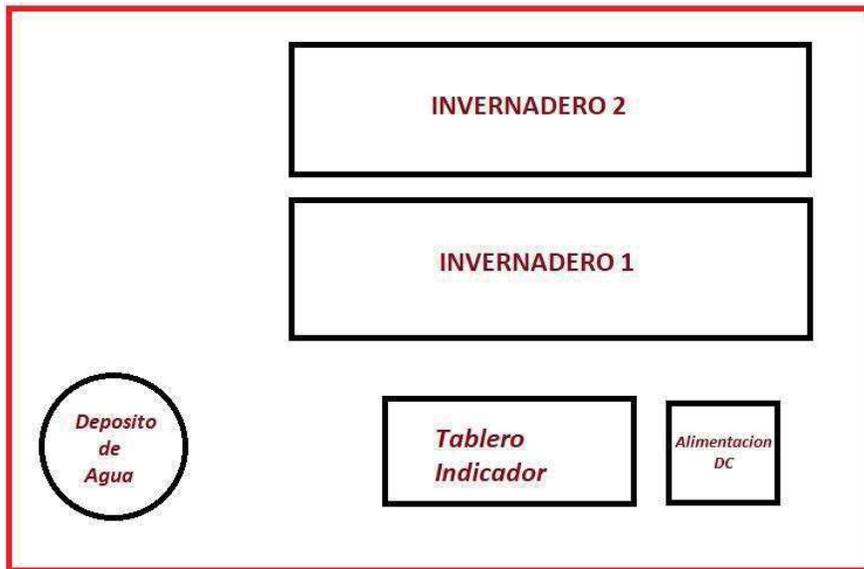


Figura 3. Distribución final de la maqueta

Dependiendo la distribución de la estructura, también se realizó un esquema del circuito de riego donde se definió la ubicación de las electroválvulas.

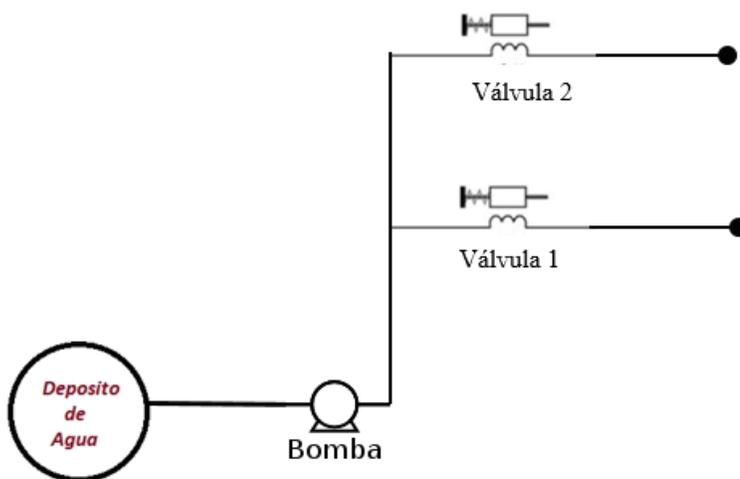


Figura 4. Circuito de Riego



Etapa de Construcción: La construcción de la maqueta, comenzó con la elección del material para la base y la estructura.



Figura 5. Base de madera

Se construyó la estructura de cada invernadero, de acuerdo a las especificaciones previas. Una vez terminada la estructura, se colocaron las mangueras correspondientes al circuito de riego, y se realizaron algunas pruebas con la presión del agua generada por la bomba elegida.



Figura 6. Estructura base

Se construyó el tablero con las tapas indicadoras, donde se alojarían los circuitos y conexiones necesarios para la etapa electrónica.



Figura 7. Tablero construido

A la estructura se le agregaron las electroválvulas, los sensores dht11 y los ventiladores dentro de cada invernadero. También se colocaron recubrimientos plásticos en los invernaderos, para aislar el cultivo de las condiciones externas.



Figura 8. Estructura con recubrimiento plástico



Figura 9. Estructura terminada

Etapa electrónica: Durante esta etapa, se diseñaron los circuitos electrónicos necesarios para las conexiones de los distintos dispositivos con la tarjeta arduino y se definieron los voltajes de operación de cada uno de acuerdo a sus características.

A continuación, se presentan individualmente los diagramas electrónicos utilizados:

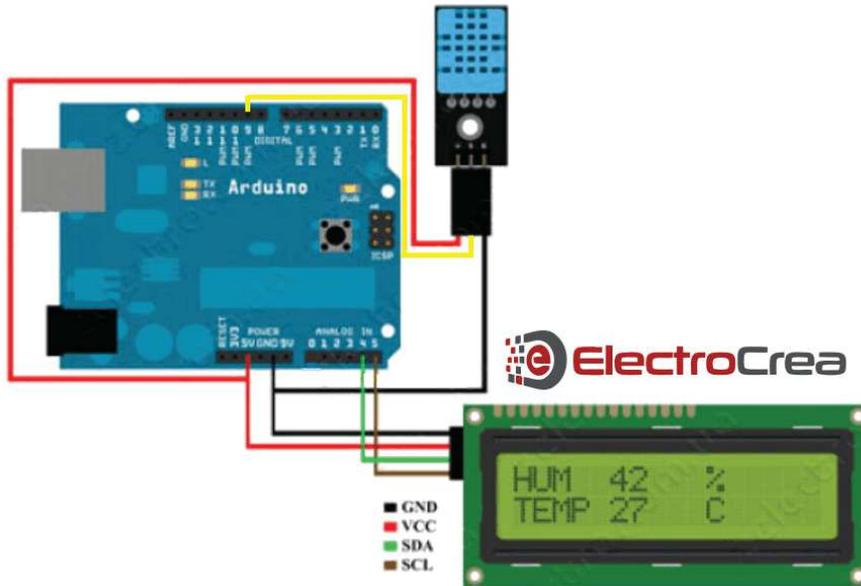


Figura 10. Diagrama del sensor DHT11 Y pantalla lcd con módulo i2c

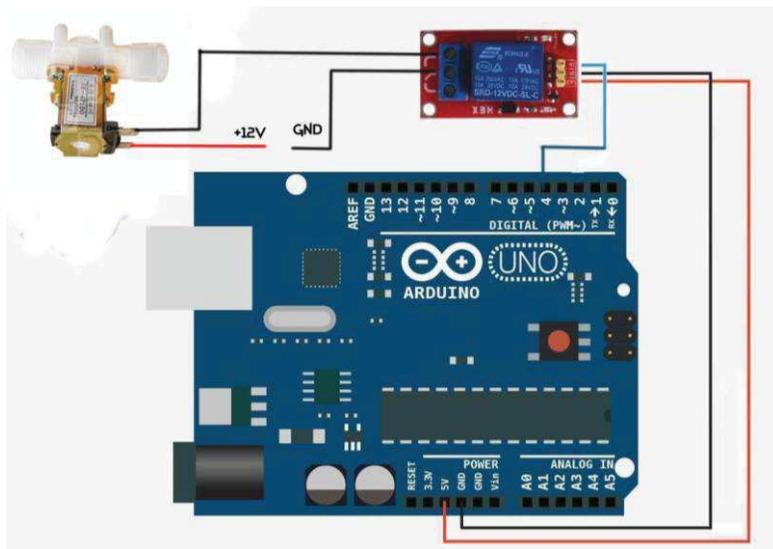


Figura 11. Diagrama para el control de la electroválvula

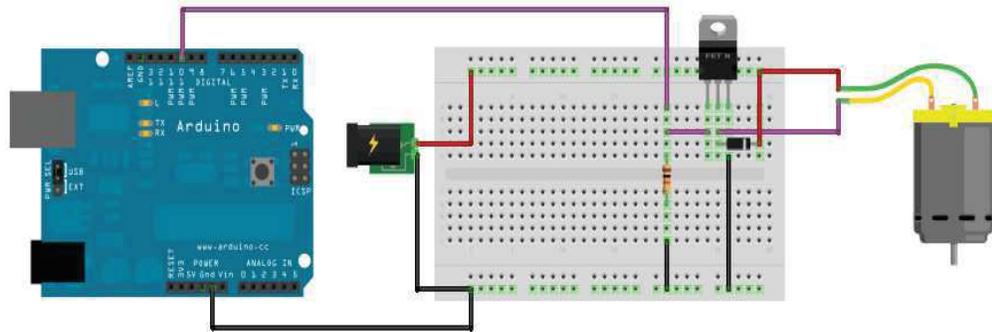


Figura 12. Diagrama para el control de la bomba de agua de dc.

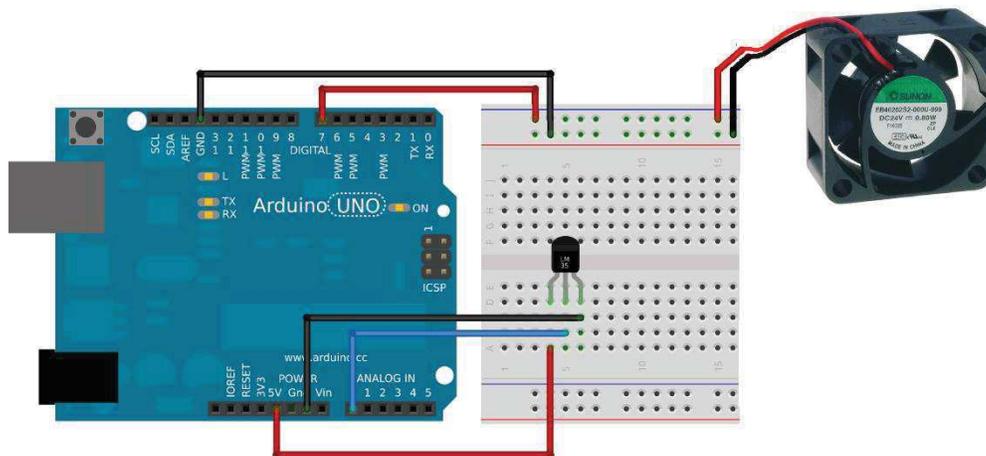


Figura 13. Diagrama para el control del ventilador

De acuerdo a la investigación previa realizada y a la necesidad de la calibración de los componentes, en esta etapa se definieron los valores de referencia para las variables de temperatura y humedad, las cuales fueron:

Humedad = 560%

Temperatura = 24°C

3.3 Resultados

Una vez terminada completamente la estructura, se agregaron las semillas para el cultivo de hongos dentro de cada invernadero. A continuación se presentan imágenes del aspecto final del proyecto.



Figura 14. Semillas dentro de los invernaderos



Figura 15. Funcionamiento del sistema completo.

Después de un periodo de observación se logró comprobar el correcto funcionamiento del proyecto, por medio del control de las variables de temperatura y humedad dentro de cada invernadero.



Figura 16. Prototipo terminado

4. Conclusiones

La realización de este proyecto nos permitió integrar los conocimientos adquiridos, hasta ahora, en la carrera. Donde aplicamos conceptos como el control de variables físicas, diseño de circuitos electrónicos y automatización y monitoreo de procesos. La aplicación de estos conceptos a un proyecto de agricultura como lo es el cultivo de hongos nos acerca a nuevos retos para diseñar e implementar proyectos eficaces para optimizar las tareas en este campo. También, el acercamiento logrado a esta actividad de cultivo, nos permitió conocer las actividades que se llevan a cabo para lograr obtener un buen producto y colocarlo en el mercado, donde nos pudimos dar cuenta que algunas de estas pueden ser mejoradas e incluso automatizadas con la aplicación de sistemas de control electrónicos. Así, nos abrimos las puertas al campo del cultivo de hongos como una nueva aplicación de la ingeniería mecatrónica.

El cultivo de hongos es una técnica empleada desde hace tiempo atrás, pero la implementación del uso de los invernaderos es una técnica más reciente. La cual consiste en crear o imitar el ambiente necesario para su crianza y cultivo de la misma. El hongo o seta es una planta sumamente delicada que necesita de varios factores importantes para la crianza ya que este espécimen es delicado; dichos factores son la humedad, la temperatura, la luminosidad. El invernadero fue adecuado para cubrir la mayoría de estas necesidades, las cuales por medio de dos naves o granjitas que en su interior contienen un ventilador y un sensor de humedad-temperatura, logramos controlar las especificaciones que el hongo necesita para su buen crecimiento.

Referencias

- [1]O. Guadarrama Leon, "Automatización de Clima Para un Invernadero", Ingeniero en Control y automatización, Instituto Politecnico Nacional, 2014.
- [2]"¿En qué consiste un invernadero inteligente?", *Novagric.com*, 2015. [Online]. Available: <http://www.novagric.com/es/blog/articulos/que-es-un-invernadero-inteligente>. [Accessed: 28-Jan- 2019].



- [3]H. Perez Rojas and M. Cortes, "SIMULACION Y CONTROL DE LA TEMPERATURA DENTRO DE UN INVERNADERO", Ingeniero de Diseño y Automatización Electrónica, UNIVERSIDAD DE LA SALLE, 2007.
- [4]J. Rodriguez, N. Sanabria and A. Chacon G., "Control de temperatura y humedad para e cultivo de hongos SHITAKE", *Visión Electrónica*, no. 1, pp. 104-114, 2011.
- [5]"Artículo | Omniblug", *Omniblug.com*. [Online]. Available: <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>. [Accessed: 22- May- 2019].
- [6]"Control de la Humedad Relativa en los cultivos en Invernadero", *Novagric.com*, 2014. [Online]. Available: <http://www.novagric.com/es/blog/articulos/control-de-la-humedad-relativa-en-los-cultivos-en-invernadero>. [Accessed: 22- May- 2019].
- [7]"Control de la Temperatura en un Invernadero", *Novagric.com*, 2014. [Online]. Available: <http://www.novagric.com/es/blog/articulos/control-de-la-temperatura-en-un-invernadero>. [Accessed: 22- May- 2019].
- [8]Y. Mattarollo, "¿Cómo funcionan las electroválvulas o válvulas solenoides de uso general?", *Altecdust.com*, 2014. [Online]. Available: <https://www.altecdust.com/blog/item/32-como-funcionan-las-electrovalvulas-o-valvulas-solenoides-de-uso-general>. [Accessed: 22- May- 2019].
- [9]S. Olguin, "Que es un invernadero", *Innatia.com*. [Online]. Available: <http://www.innatia.com/s/c-huerta-organica/a-que-es-un-invernadero.html>. [Accessed: 22- May- 2019].



Automatización del Lago Artificial e Implementación de Fuentes Danzantes

Aceves Becerra Octavio Martin, Morales Medina José Luís, Tostado Estrada Álvaro Emanuel, Aguilar Molina Yehoshua y Domínguez García Rodolfo Omar

Universidad de Guadalajara Centro Universitario de los Valles

Resumen

En la actualidad es común encontrarse con lagos artificiales y en su defecto fuentes convencionales aunque las fuentes danzantes están teniendo un mayor peso ya que brindan entretenimiento y mayor belleza y atractivo visual al área donde se encuentra gracias a que se le puede agregar una sincronización con música y luces de colores, las cuales, las hacen lucir más llamativas que una simple fuente o una simple secuencia en su caso, más sin embargo la implementación de estas suele ser muy costosa y algo complicado.

En este proyecto se desarrollara un prototipo de control de bajo costo para las bombas que suministran el agua para controlar los encendidos y apagados de las mismas como también encendido por temporizador digital así como la manipulación de las secuencias de las fuentes para así hacerlas danzantes mediante un dispositivo móvil ya sea celular o Tablet también se podrá manipular manualmente para encender y manejar un temporizador, además, este trabajo contara con la implementación del Internet Of Things, por lo que tendrá acceso a una nube de datos para guardar las secuencias y así en un futuro si se desea usar nuevamente estas secuencias ,además de poder observar cómo ha ido funcionando el sistema por medio de las bases de datos y así saber que se encuentra haciendo en tiempo real.

Palabras clave: Palabras que por si mismas dan idea del tema o área del artículo. Usar letras de tipo Arial, normal de 10 puntos.

1. Introducción

A lo largo de la historia las plazas públicas de todo el mundo han formado una parte fundamental para su población ya que representan un área importante, un ejemplo de esto fueron las ágoras que eran consideradas el motor de las polis de la antigua Grecia debido a las actividades que se desarrollaban en dicho lugar, tales como Comercio, desarrollo de la democracia y el lugar para el ocio que era la principal [1].

En la actualidad se fueron implementando diferentes cosas tanto en plazas públicas como privadas como jareas de juegos para niños, fuentes de sodas para proveer de alimentos a los visitantes, lagos artificiales y fuentes danzantes, en estas dos últimas nos enfocaremos para el desarrollo de nuestro trabajo. Debido a la espectacularidad de las fuentes danzantes y el atractivo visual que estas ofrecen estas están siendo implementadas a lo largo del mundo y no solo en plazas públicas sino también en hoteles y restaurantes lujosos, un ejemplo de esto son los diversos establecimientos en La Vegas, Nevada, que cuentan con una gran cantidad de fuentes danzantes.

Habitualmente el control de las bombas para el suministro de agua es manual más sin embargo con el avance de la tecnología buscamos hacer esto automatizado para que de esta manera tanto los lagos artificiales como las fuentes danzantes no solo sean ornamentales, sino que también tengan una interacción con las personas que se encuentran en dicha localización. La mayoría de estas son



solamente para contemplar ya que tienen una programación predeterminada para que sigan ciertos patrones lo cual podría cambiar ya que el usuario tendrá la posibilidad de programar las secuencias.

Habitualmente se incorpora la tecnología PLC (Controlador Lógico Programable) en todos aquellos proyectos con fuentes danzantes de complejidad baja o media y que no incorporan coreografía musical.

La implementación de esta tecnología es costosa en la mayoría de los casos debido a que necesita de hardware y software especializado, sin embargo, la inversión de este tipo de tecnologías representa ganancias debido a la mayor afluencia de personas a los lugares donde se utiliza.

En este proyecto se pretende agregar un control tanto para las bombas como para las fuentes danzantes además usando la ayuda de los dispositivos móviles ya que facilitan las tareas diarias, además de aumentar el interés en el espectador, lo cual llevaría a tener un mayor impacto social.

2. Planteamiento del proyecto

En la actualidad la mayoría de lagos artificiales y fuentes danzantes están conformados para destacar principalmente en un ámbito estético para que sea llamativo visualmente más, sin embargo, se busca una personalización que vaya más allá de la interacción visual, principalmente, para la entretención de la sociedad ya que con el paso del tiempo se necesita ir innovando para poder lograrlo. Con la llegada de los dispositivos móviles nos damos cuenta de que las personas prefieren interactuar con esta tecnología ya que se nos facilita la realización de varias tareas además de que pone a nuestra disposición diferentes herramientas.

Aprovechando las tecnologías móviles buscamos realizar el control de bombas trifásicas para expulsar el agua de diferentes maneras, realizando la conexión con la tarjeta por medio de wifi para así poder almacenar datos en la nube y entrar en el campo del internet of the things, ahorrando tiempo y dinero ya que no habrá necesidad de dispositivos/objetos adicionales.

2.1 Estado del arte

A diferencia del pasado, hoy en día las los lagos artificiales y fuentes danzantes son un medio de entretenimiento debido a los espectáculos que dan con distintas rutinas pre programadas sincronizándolas con música y luces de diversos colores para así poder atraer a un mayor número de gente e interactuando con los mismos. Trabajos previos o similares al proyecto a desarrollar.

En Zapopán se construyeron varias fuentes danzantes entre las cuales destaca la fuente interactiva construida en la explanada de la plaza de las Américas Juan Pablo II, en la cual se intervino una superficie de 72 metros cuadrados, con una inversión de 9 millones de pesos. Dicha fuente interactiva fue construida a nivel de piso y consta de 82 chorros de agua, de los cuales 12 son dobles [2].

Otra implementación de fuentes interactivas son los llamados Dancing Water Speakers o bocinas danzantes, que están constituidos por una bocina como base que cuenta con 4 chorros de agua que cuentan con iluminación LED, así, al introducir música mediante USB dichos chorros de agua responderán al ritmo de la música, aunque este derivado de las fuentes danzantes no es interactivo, ya que no se pueden hacer modificaciones manuales.

2.2 Justificación

En la antigüedad, las fuentes danzantes se instalaban en plazas, centros comerciales, lugares públicos o privados, meramente con el fin decorativo del lugar para hacerlo más estético, en la actualidad el propósito de este cambio ya que ahora además de ser decorativos son un medio de entretenimiento y atracción para el público gracias a lo llamativas que son por la sincronización con la iluminación y en ocasiones con música.



2.3 Objetivos

Objetivo general

Como objetivo general se buscó implementar una automatización mediante una aplicación por conexión wifi para así poder controlar las bombas que suministran el agua al lago artificial y ayudar a los peces del mismo con oxigenación, esta cuenta con un control ON/OFF tanto en la app como manual al igual que un temporizador, de igual manera manual o mediante la app como el operario prefiera o se requiera, además de implementar secuencias o rutinas para los chorros de agua y así automatizar el proceso de una manera eficiente.

Objetivo particular

El objetivo específico es obteniendo el objetivo general, implementar una sincronización en iluminación y en la música para un mayor atractivo visual, además de hacerlo interactivo mediante el concepto de Internet Of Things para guardar los datos en la nube y así el usuario pueda crear sus propias rutinas.

2.4 Requerimientos técnicos

Se necesitaron bombas trifásicas para el flujo del agua por chorro hacia del aire para su control sabiendo que en México el voltaje común es de 127 V con una frecuencia de 60 Hz, el prototipo a realizar para el control del lago artificial será instalado en el lago artificial del Centro Universitario De Los Valles para el control de las bombas de agua mientras que el prototipo de fuentes danzantes se llevará a cabo en una pecera.

Para el desarrollo de la aplicación debimos recurrir al uso del programa AndroidStudio, que es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA el cual se puede programar en java o en kotlin.

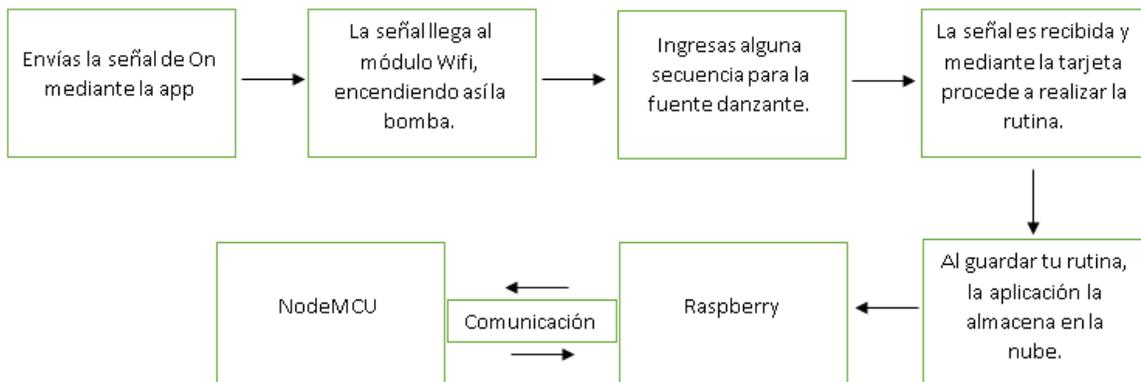
Además, se necesitaron algunos componentes como la pecera donde se desarrolló el prototipo de fuentes danzantes, algunas mini bombas de agua, iluminación led, un módulo wifi NodeMCU con una Raspberry para poder guardar los datos en una nube, usando así lenguajes de programación C++, Java, SQL y CSS que este último sirvió para poder crear documentos HTML, coloquialmente conocido como "Página Web" para mayor comodidad del usuario.

Al implementarlas con la opción de que el usuario pueda interactuar con esta, lo hará más atractivas hacia lo sociedad, fomentando su curiosidad y así tener mayores beneficios, principalmente en el sector de turismo.

2.5 Operación del sistema

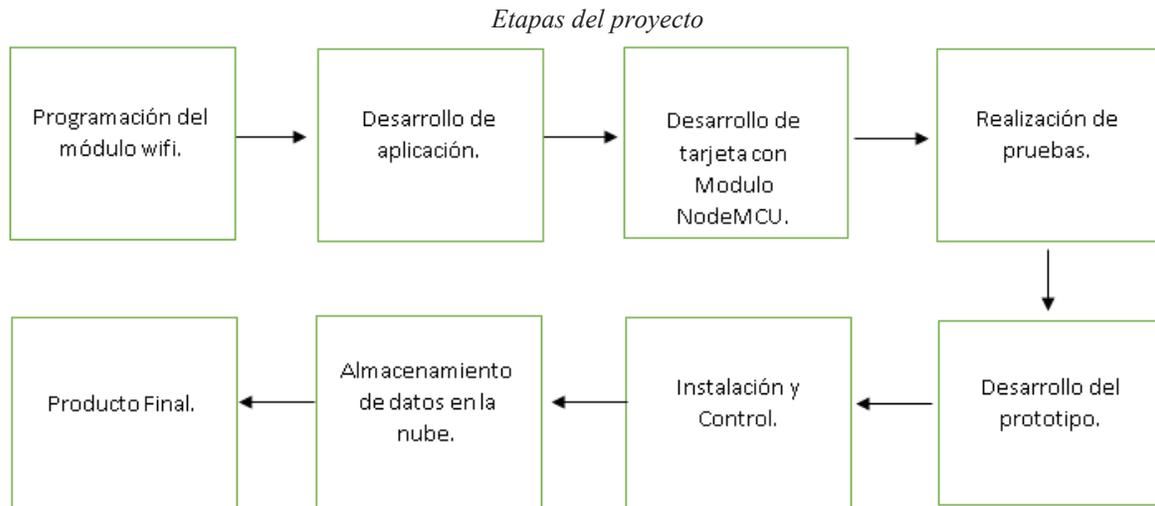
Mediante la aplicación que se realizó se activan las bombas trifásicas para suministro de agua, la aplicación cuenta con un apartado On/Off para las bombas al igual que un apartado de timer para programar el encendido y apagado automático, así como un panel de control donde es posible realizar las secuencias que desees de las fuentes danzantes, además de poder guardarlas en la nube para un posterior uso.

Funcionamiento del equipo





2.6 Etapas del proyecto



3. Resultados

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
}
```

Figura 1: Código antes de la aplicación de formato

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
    // Get reference to the drawer layout and set event listener
```

Figura 2: Código después de la aplicación de formato



INICIO

Esta visualización muestra la pantalla de inicio de la aplicación en la cual se podrá elegir entre tres opciones, crear secuencia, mis secuencias y secuencia en tiempo real.

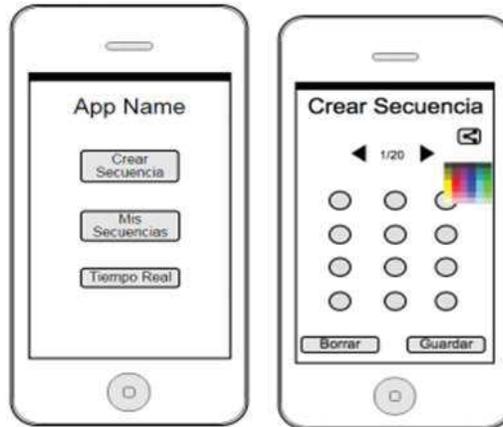


Figura 3: Interfaces de usuario

TIEMPO REAL

Finalmente mostramos la pantalla de tiempo real en la cual se dibuja el patrón que se desea realizar.



Figura 4: Interface en tiempo real



Código del microcontrolador

```
#include <ESP8266WiFi.h>

#ifndef STASSID
#define STASSID "WIN-JBC98VQA09R 5033"//"INFINITUM80672D"
#define STAPSK  "porganar"//"8E1D79D7B7"
#endif
#define led1 2
#define led2 5
const char* ssid = STASSID;
const char* password = STAPSK;

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);

  // prepare LED
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print(F("Connecting to "));
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(F("."));
  }
  Serial.println();
  Serial.println(F("WiFi connected"));

  // Start the server
  server.begin();
  Serial.println(F("Server started"));

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  Serial.println(F("new client"));

  client.setTimeout(5000); // default is 1000

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(F("request: "));
  Serial.println(req);
```



```
// Match the request
int val;
if (req.indexOf(F("/gpio/0")) != -1) {
  val = 0;
} else if (req.indexOf(F("/gpio/1")) != -1) {
  val = 1;
} else {
  Serial.println(F("invalid request"));
  val = digitalRead(led1);
}
int val2;
if (req.indexOf(F("/gpio/2")) != -1) {
  val2 = 0;
} else if (req.indexOf(F("/gpio/3")) != -1) {
  val2 = 1;
} else {
  Serial.println(F("invalid request"));
  val2 = digitalRead(led2);
}
// Set LED according to the request
digitalWrite(led1, val);

// read/ignore the rest of the request
// do not client.flush(): it is for output only, see below
while (client.available()) {
  // byte by byte is not very efficient
  client.read();
}

// Send the response to the client
// it is OK for multiple small client.print/write,
// because nagle algorithm will group them into one single packet
client.print(F("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\nGPIO is now "));
client.print((val ? F("high") : F("low")));
client.print(F("<br><br>Click <a href='http://'>"));
client.print(WiFi.localIP());
client.print(F("/gpio/1'>here</a> to switch LED GPIO on, or <a href='http://'>"));
client.print(WiFi.localIP());
client.print(F("/gpio/0'>here</a> to switch LED GPIO off.</html>"));

// The client will actually be 'flushed' then disconnected
// when the function returns and 'client' object is destroyed (out-of-scope)
// flush = ensure written data are received by the other side
Serial.println(F("Disconnecting from client"));
}
```

Figura 5: Código empleado

Tarjeta NodeMCU

Utilizada para el control de entradas y salidas y procesamiento de señales.



Figura 6: Tarjeta NodeMCU

Arduino conexión a contactor y motor trifásico

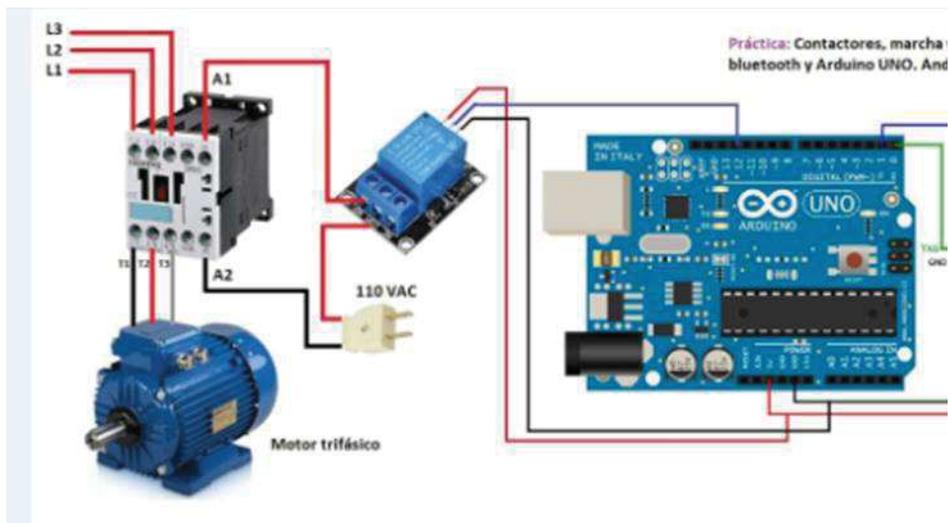


Figura 6: Diagrama de conexión Arduino, contactor y motor trifásico

4. Conclusiones

Para la elaboración del proyecto fueron necesarios conocimientos previos de programación de microcontroladores en C++ y programación de aplicaciones móviles como es el caso de Android en el cual se usó java y conocimientos en electrónica digital y analógica, así como el conocer la electrónica de potencia.

Se buscó una conectividad estable entre la placa y dispositivo móvil, para así lograr el efecto vómetro arrojando un resultado satisfactorio haciendo más intuitiva la aplicación y amigable con el usuario. La placa funcionó como se requería dando los resultados deseados

Referencias

- [1]"Ágora Griega - EcuRed", *Ecured.cu*, 2019. [Online]. Available: https://www.ecured.cu/%C3%81gora_Griega.
- [2]"Gobierno Municipal de Zapopan | Embellece Zapopan el Centro Histórico para impulsar el comercio local y turismo", *Zapopan.gob.mx*, 2018. [Online]. Available: <https://www.zapopan.gob.mx/embellece-zapopan-el-centro-historico-para-impulsar-el-comercio-local-y-turismo/>.
- [3]"Lenguaje CSS - Documentación sobre diseño web", *Lenguajecss.com*, 2019. [Online]. Available: <https://lenguajecss.com/>.
- [4]"Definición de vómetro", *Definicion.org*, 2019. [Online]. Available: <https://definicion.org/vumetro>.



Desarrollo de un Simulador Robótico Utilizando el Método Evolutivo de las Entregas Incrementales

Jiménez López Eusebio¹, Zarza Herreros Melquisedec², Rivera Nieblas Jorge Oswaldo³, Urbalejo Contreras Arturo⁴, Amavizca Valdez Laura Olivia⁴ y Vázquez Brindis Juan Carlos⁴

¹Universidad La Salle Noroeste - CIAAM de la UTS-IIMM, ejimenezl@msn.com

²Universidad La Salle Noroeste

³Universidad Tecnológica de Hermosillo

⁴Universidad Tecnológica del Sur de Sonora

Resumen

En este artículo se presenta la aplicación del método evolutivo de las entregas incrementales para desarrollar un simulador robótico. El modelo matemático del robot fue desarrollado utilizando el álgebra de Cuaterniones. Para resolver el problema cinemático inverso, el cual originó un sistema de ecuaciones no lineales, se utilizó el método de Newton-Rapshon. El código y la visualización de los movimientos del robot fueron programados en el software Mathematica v8. La documentación del programa se llevó a cabo mediante el desarrollo de tres incrementos y fue utilizada la herramienta de diseño UML (Unified Modeling Language). Los resultados obtenidos muestran que la aplicación de la ingeniería de software en el desarrollo y la documentación del programa de un simulador, permite tener claridad, sistematización y mantenimiento en un producto de software.

Palabras clave: Simulador, Robótica, Ingeniería de Software.

1. Introducción

En el contexto técnico de la enseñanza de la Mecatrónica, el desarrollo de simuladores es una tarea difícil para los alumnos, pues por lo general no utilizan métodos sistemáticos que hagan posible que el producto obtenido (simulador) tenga características de más valor. En general, los programas que desarrollan, se basan en su propia lógica, no son óptimos y se dificulta usarlos para la enseñanza [1]. Durante la formación del ingeniero mecatrónico se desarrollan diversos programas de computación debido a las materias que llevan o los proyectos que trabajan. El material generado solo es útil para tener una calificación de un proyecto y no es posible interpretarlo debido a que el único orden en la programación lo conoce el alumno que lo desarrolló. Es necesario aplicar métodos que permitan documentar el material de programación que desarrollan los alumnos para poder darle uso en otras materias o para generar productos mecatrónicos con alto valor agregado.

Los Modelos de Ingeniería de Software, como son los casos del Modelo Evolutivo con Entregas Incrementales o el Modelo en Espiral [2], ofrecen metodologías capaces de generar productos de software eficientes, de alto valor agregado, adecuados a las necesidades del cliente en particular, o genéricos para usuarios de todo tipo. Estos modelos de Ingeniería de Software pueden ser útiles para la generación de simuladores mecatrónicos, desarrollo de software y



programación, tanto didácticos, como industriales, que requieran ser llevados más allá de una fase experimental (producto comercial).

El uso de la Ingeniería de Software daría las pautas para que los problemas de la Ingeniería en general que requieran de un simulador, se realicen de forma ordenada, clara y precisa, contribuyendo con esto al uso de los códigos de los programas en la enseñanza. Además, el producto generado al aplicar algunos de los modelos de Ingeniería de Software, se asemeja más a las características de productos comerciales que a las características de productos didácticos [1].

Existen diversas aplicaciones de la ingeniería de software y sus herramientas al diseño y documentación de un producto software. Por ejemplo en [1] se aplicó el método de las entregas incrementales [3] para desarrollar un simulador de trayectorias rectilíneas. El simulador fue diseñado por medio de 5 incrementos y la plataforma de programación fue el software Matlab. En otro trabajo similar se diseñó el software de un péndulo en la plataforma Labview. Fueron utilizadas herramientas de UML [4, 5] y el método de las entregas incrementales [2] para desarrollar dicho simulador. En [6] se utilizó también el UML en una aplicación de robótica de servicios en la que un robot manipulador antropomórfico realiza la tarea de seguimiento de un objeto en movimiento. Este caso de estudio se ejecuta sobre ROS (Robotic Operation System). En [7] se utilizó UML para fundamentar a través de diagramas de procesos y objetivos el desarrollo de un robot interactivo utilizado para la distribución de información.

En este artículo se presenta el desarrollo de un simulador robótico utilizando el Método de las Entregas Incrementales y las herramientas de diseño UML. La plataforma de programación utilizada será Mathematica v8 [8].

2. Métodos

En esta sección se presentan los principales métodos que fueron utilizados para el desarrollo del simulador.

2.1 El método de las entregas incrementales

De acuerdo con [1, 4] son 4 etapas básicas que deben aplicarse para utilizar el modelo de la entrega incremental [2]: Análisis, Diseño, Codificación y Pruebas. Cada etapa contiene pasos, los cuales se mencionan a continuación:

- 1) Aplicación del paradigma evolutivo para un incremento:
 - Análisis.
 - Detección de la necesidad.
 - Planteamiento del problema.
 - Análisis de requerimientos.
 - Especificación de la interfaz.
 - Diseño.
 - Diseño de la interfaz del usuario.
 - Diseño de la estructura del sistema.
 - Codificación.
 - Pruebas.
 - Implementación y prueba de unidades (codificación).
 - Integración y prueba del sistema.
- 2) Iteración y evaluación para el incremento siguiente.



2.2 Pasos del UML

Para aplicar el UML en el desarrollo de los casos de uso y diagramas de secuencias se utilizaron los pasos siguientes:

- 1) Establecer el problema.
 - 1.1) Clasificar los requisitos para el problema.
 - 1.2) Modelado de los diagramas de casos de uso.
 - 1.3) Establecer los alcances del diagrama de caso de uso y si fuera necesario realizar una iteración para redefinir el problema.
- 2) Identificar la secuencia de los eventos entre los casos de uso.
- 3) Relacionar los casos de uso.
- 4) Con la secuencia de eventos, establecer los objetos de que influyen en el modelo.
- 5) Realizar el diagrama de secuencia en el tiempo.
- 6) Si fuera necesario realizar una iteración para redefinir el problema o para restablecer la secuencia de eventos.

2.3 Modelado del robot

Los pasos generales que se utilizan para modelar la cinemática de posición y desplazamiento de un robot son:

- 1) Se estudia la configuración del robot y se vectoriza.
- 2) Se generan las ecuaciones de posición y de desplazamiento del robot.
- 3) Se formulan los problemas directos e inversos, respectivamente.
- 4) Si los modelos matemáticos resultan no lineales, entonces se deben utilizar algún método de solución numérica, como por ejemplo el Método del Newton-Raphson [9].
- 5) Se modela una trayectoria.
- 6) Se resuelve el problema inverso utilizando datos de la trayectoria.
- 7) Se construye un simulador de los movimientos.

3. Caso de estudio

En esta sección se presenta un resumen de los incrementos que se utilizaron para el desarrollo del simulador del robot. Se explicará con más detalle el primer incremento y solo se pondrán los resultados de los dos restantes.

3.1 Configuración del robot

El robot motivo de estudio en este artículo se muestra en la Figura 1. El robot es antropomorfo (de cadena abierta) de 3 GDL (Grados de Libertad) y está compuesto por tres eslabones rígidos y en el extremo final del último eslabón se localiza el punto final en donde se coloca una herramienta. El robot tiene ejes de rotación en (Z,Y,Y) y tres desplazamiento angulares, como se muestra en la Figura 1.

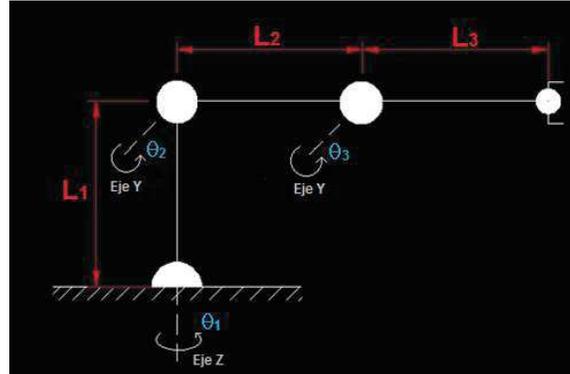


Figura 1. Configuración del robot.

3.1 Análisis previo

El objetivo del simulador es imitar los movimientos del robot en una computadora. Para lograrlo se requiere construir las ecuaciones matemáticas que modelan el comportamiento mecánico de posición y de desplazamiento del robot utilizando estructuras algebraicas, como es el caso del álgebra de Cuaterniones [10]. El proceso del modelado del robot requiere de la formulación de dos problemas: 1) El problema directo y 2) El Problema inverso [11]. El problema directo consiste en determinar por medio de un modelo matemático las coordenadas (X,Y,Z) del órgano terminal si se conocen los desplazamientos angulares. Por el contrario, el problema cinemático inverso consiste en determinar los ángulos de rotación (mostrados en la Figura 1) mediante un modelo matemático si se conoce la posición del punto terminal.

El problema cinemático directo solo requiere de la sustitución de datos para resolverse para el caso de robots antropomorfos; en cambio el problema inverso generalmente implica un modelo matemático no lineal, para lo cual se deben utilizar métodos numéricos [11] o bien redes neuronales [12]. Una vez que los modelos matemáticos han sido resueltos es necesario generar: 1) Una simulación numérica de los resultados y 2) Una simulación gráfica de los mismos.

Este análisis previo permite identificar unos de los pasos más complicados del Método Evolutivo Incremental, esto es, definir el número de incrementos que deben ser aplicados para el desarrollo del simulador. Un número grande de incrementos no es lo más recomendable por el hecho de que el desarrollo requerirá demasiada información y procesamiento, y un número de incrementos muy mínimo requerirá un diseño muy amplio. Por ello, solo un conocedor del tema de robótica podrá identificar y definir el número de incrementos que para este caso fueron los siguientes:

- 1) Primer incremento: cinemática directa.
- 2) Segundo incremento: cinemática inversa.
- 3) Tercer incremento: unión de interfaces.

3.2 Explicación de los incrementos

En esta sección se describe un resumen de cada uno de los tres incrementos mediante los cuales se generó el simulador.

3.2.1 Primer incremento: cinemática directa

Para el primer incremento del simulador se requiere que se cumplan los siguientes aspectos:



- Utilizar las fórmulas teóricas del análisis previo al modelo, para poder graficar cada elemento que conforma al robot.
- Utilizar la ecuación cinemática dada por el análisis previo al modelo, para poder visualizar en pantalla las coordenadas de la posición final del efector del robot.

- **Planteamiento del Problema**

Se desea programar y simular el modelo matemático asociado al problema cinemático directo de un robot manipulador de 3 GDL. El modelo matemático debe ser modelado con el álgebra de Cuaterniones [10]. Para darle solución al problema cinemático directo es necesario conocer los valores de las articulaciones ($\theta_1, \theta_2, \theta_3$) y los valores de las longitudes de cada eslabón del robot (L1, L2, L3), representados en la Figura 1.

- **Análisis de Requerimientos**

Se requiere definir los requerimientos para el primer incremento; estos son:

- 1) Requerimientos Funcionales: Datos requeridos, datos de salida y fórmulas requeridas).
- 2) Requerimientos no funcionales: a) Mostrar en pantalla los datos numéricos de la posición final del robot, b) Mostrar los datos numéricos de los ángulos y c) dibujado de los elementos de trabajo.
- 3) Requerimientos del Usuario: A) Manipulación sencilla de los controles, B) No se podrán modificar los rangos de movimientos preestablecidos del robot y C) El usuario podrá visualizar los ejes coordenados.
- 4) Requerimientos del sistema

Para definir los requerimientos del sistema se utilizó una herramienta de UML llamada "La vista de casos de usos". El caso de uso que se desarrolló para primer incremento del simulador se llamó "Caso de uso: Cinemática Directa" y se puede observar en la Figura 2. Este caso de uso muestra de manera general lo que el usuario puede realizar, y la interacción que existe entre las funciones del software. Cada caso de uso se describe de forma explícita, pero para efectos de este trabajo solo se explica uno de ellos.

El caso de uso "Primer ángulo" permite al usuario darle un movimiento de giro al robot sobre el eje "Z" en el primer eslabón. Donde a su vez le proporciona un valor numérico expresado en ángulo a los casos de uso "Eslabones CD" y "Articulaciones CD"; los cuales al recibir los valores realizan los cálculos correspondientes para generar las coordenadas y el tamaño de cada eslabón y articulación del brazo robótico.

Para que los casos de uso "Eslabones CD" y "Articulaciones CD" puedan realizar dichos cálculos necesitan el auxilio de los casos de uso "Primera rotación", "Segunda rotación", "Tercera rotación" y "Transformación Vectorial", quiénes son los encargados de efectuar los cálculos de las rotaciones, para cada eslabón y articulación del robot.

Pero cuando el usuario hace acción del caso de uso "Primer ángulo" es el caso de uso "Primera rotación" quien realiza el cálculo de la rotación del robot sobre el eje "Z" y el caso de uso "Transformación Vectorial" es quien realiza el cálculo de pasar las coordenadas generadas por la rotación al espacio tridimensional.

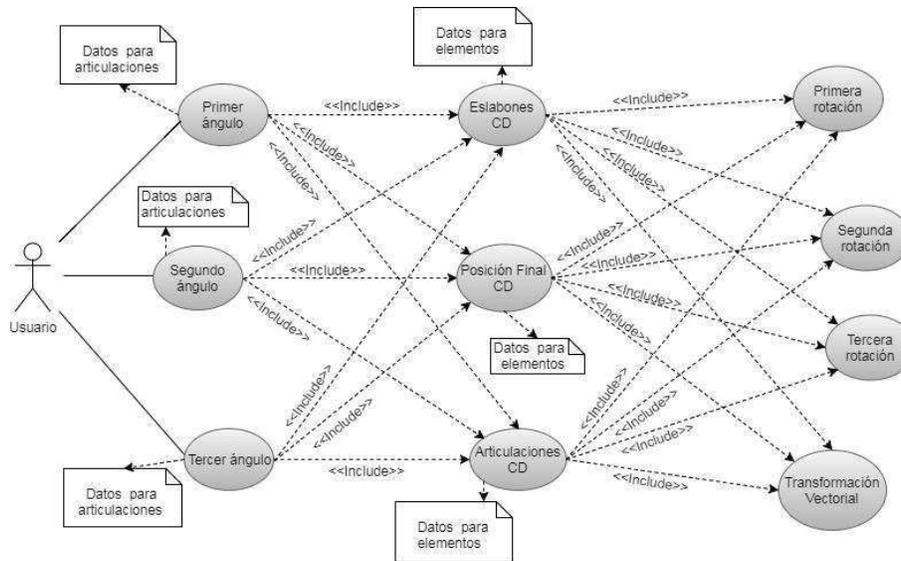


Figura 2. Diagrama de caso de uso: Cinemática directa.

- Especificaciones de la interfaz

Las especificaciones de la interfaz son: 1) Mostrar de manera clara y sencilla los datos de salida, 2) Visualización numérica de la posición final del efector del robot y 3) La graficación de cada elemento que conforman al brazo robótico) y los datos requeridos. Los rangos de movimiento de cada articulación son: 1) 0 a 180 grados, 2) -180 a 0 y 3) 90 a -90, para la primera, segunda y tercera articulación, respectivamente.

- Diseño del primer incremento

En esta sección se dará a conocer la etapa de diseño del modelo evolutivo. La etapa de diseño está dividida en dos partes: el diseño de la interfaz de usuario y el diseño de la estructura del sistema.

-Diseño de la interfaz de usuario

Esta parte del diseño se encarga de mostrar al usuario el aspecto visual que tendrá la interfaz en pantalla. Al usuario no se le permite introducir datos de entrada al sistema de manera textual, sino que solamente puede manipular un rango de valores preestablecidos por el desarrollador, de esta manera irá introduciendo esos valores al sistema para realizar los debidos cálculos. Para este rango de valores se asignará un panel con el nombre de "Controles de Cinemática Directa". En este panel se encuentran los controles que mueven los eslabones del brazo robótico.

Existe otro panel etiquetado con el nombre de "Brazo robótico 3DGL - Cinemática Directa" donde se encuentra graficado cada componente del robot y la visualización de los ejes coordenados (X, Y, Z). Dentro de este mismo panel existe un sub panel con la etiqueta de "Posición Final" que muestra la posición final en la que se encuentra el efector del robot en el espacio. En la Figura 3 se puede ver expresado cada uno de los paneles y sub panel mencionado anteriormente.

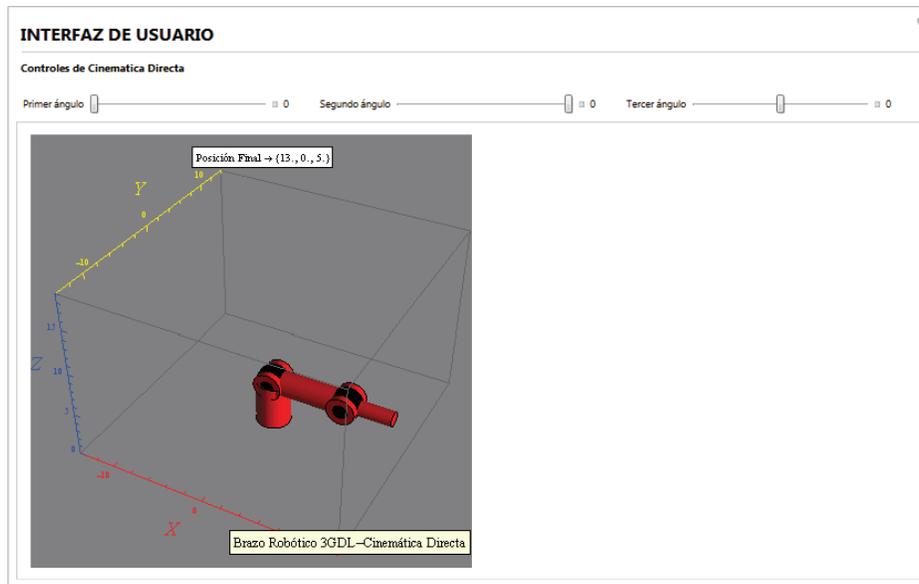


Figura 3. Interfaz de usuario del simulador en su primer incremento.

-Diseño de la estructura del sistema

Esta parte del diseño se encarga de mostrar la estructura del código del sistema utilizando una herramienta del UML llamada “Diagrama de secuencia”. El diagrama de secuencia muestra la interacción que existe entre las funciones que se desarrollaron en el caso de uso “Cinemática Directa”. El Diagrama de secuencia muestra solamente un bosquejo general de cómo interactúan las funciones en el tiempo.

En el diagrama de la Figura 4, el usuario pulsa y mueve el control llamado “Primer ángulo”, el cual va proporcionando los “datos para articulaciones” para las funciones “Eslabones CD” y “Articulaciones CD” y “Posición Final CD”. Las primeras dos funciones “Eslabones CD” y “Articulaciones CD” realizan dos tareas: la primera es la de visualizar por la pantalla los eslabones y articulaciones del brazo robótico, y la segunda es la de activar a la función “Primera rotación” y “Transformación Vectorial” para que realice el cálculo correspondiente y hacer girar el primer eslabón del brazo robótico junto con la demás estructura. La función “Posición Final” al recibir los datos proporcionados por el control “Primer ángulo”, realiza el cálculo necesario para determinar la posición final en la que se encuentra el efector del robot, mostrando por pantalla la visualización numérica de la posición en sus componentes (X, Y, Z).

De igual manera cuando el usuario ahora pulsa y mueve el control llamado “Segundo ángulo” y “Tercer ángulo”, le proporciona los “Datos para articulaciones” a las funciones “Eslabones CD” y “Articulaciones CD” y realiza las mismas tareas mencionado antes, pero con la diferencia que para el caso del control “Segundo ángulo”, las funciones “Eslabones CD” y “Articulaciones CD” activan la función “Segunda rotación” y “Transformación Vectorial” la cual hace girar al segundo eslabón del brazo robótico junto con la demás estructura. Y al activar el control “Tercer ángulo”, las funciones “Eslabones CD” y “Articulaciones CD” activan a la función “Tercera rotación” y “Transformación Vectorial”, donde realiza la tarea de girar el tercer eslabón del brazo robótico junto con la demás estructura.

- Codificación del primer incremento

Una parte de la codificación del algoritmo para este primer incremento se muestra en la Tabla 1, la cual describe cada bucle y su respectiva fórmula programada en el lenguaje formal de



cálculo simbólico Mathematica 8.0. Cada ecuación está en función de Cuateriones [10]. La codificación se realiza de forma ordenada y documentada, con la finalidad de que pueda ser leída e interpretada. Algunas de las variables estáticas utilizadas fueron las siguientes:

1. "Brazo1CD": variable donde se guarda el valor de la longitud del primer eslabón del brazo robótico.
2. "e0": variable donde se almacena el vector unitario {0, 0, 0, 0}.
3. "e1": variable donde se almacena el vector unitario {0, 1, 0, 0}, que representa al vector x.
4. Los mensajes entre las variables para establecer intercomunicaciones derivadas.

Tabla 1. Codificación del primer incremento.

Código en Mathematica	Comentario
Las funciones llamadas: "Multiplicación", "Conjugado", "Rotación" y "Vectores de rotación", son funciones básicas que hacen posible el cálculo para las funciones llamadas: "Vectores de rotación rotados 1", "Vectores de rotación rotados 2", "Primera rotación", "Segunda rotación" y "Tercera rotación".	
<pre> Multiplicacionpq[p_, q_] := { p[[1]] q[[1]] - p[[2]] q[[2]] - p[[3]] q[[3]] - p[[4]] q[[4]], p[[1]] q[[2]] + p[[2]] q[[1]] + p[[3]] q[[4]] - p[[4]] q[[3]], p[[1]] q[[3]] + p[[3]] q[[1]] - p[[2]] q[[4]] + p[[4]] q[[2]], p[[1]] q[[4]] + p[[4]] q[[1]] + p[[2]] q[[3]] - p[[3]] q[[2]] }; </pre>	<i>Multiplicación:</i> Función que realiza la multiplicación de dos cuaterniones y da su correspondiente resultado.
<pre> Conjugado[p_] := {p[[1]], -p[[2]], -p[[3]], -p[[4]]}; </pre>	<i>Conjugado:</i> Función que realiza la operación de invertir a signo negativo la parte imaginaria del cuaternion.
<pre> Rotacionpq[p_, q_] := Multiplicacionpq[Multiplicacionpq[p, q], Conjugado[p]]; </pre>	<i>Rotación:</i> Función que realiza el cálculo matemático entre el eje de rotación y el elemento a rotar. El cual dará como resultado las coordenadas finales para posicionar el elemento rotado.

- Pruebas

El objetivo de esta etapa de pruebas del primer incremento es verificar y comprobar que el programa da como resultado los datos de salida deseados de manera correcta y funcional. Esto se logrará mediante la comparación de los resultados teóricos obtenidos del modelo matemático del robot, en este caso la posición final, contra los resultados obtenidos por el programa de la codificación del modelo matemático del robot e incluso realizando el cálculo de manera analítica.

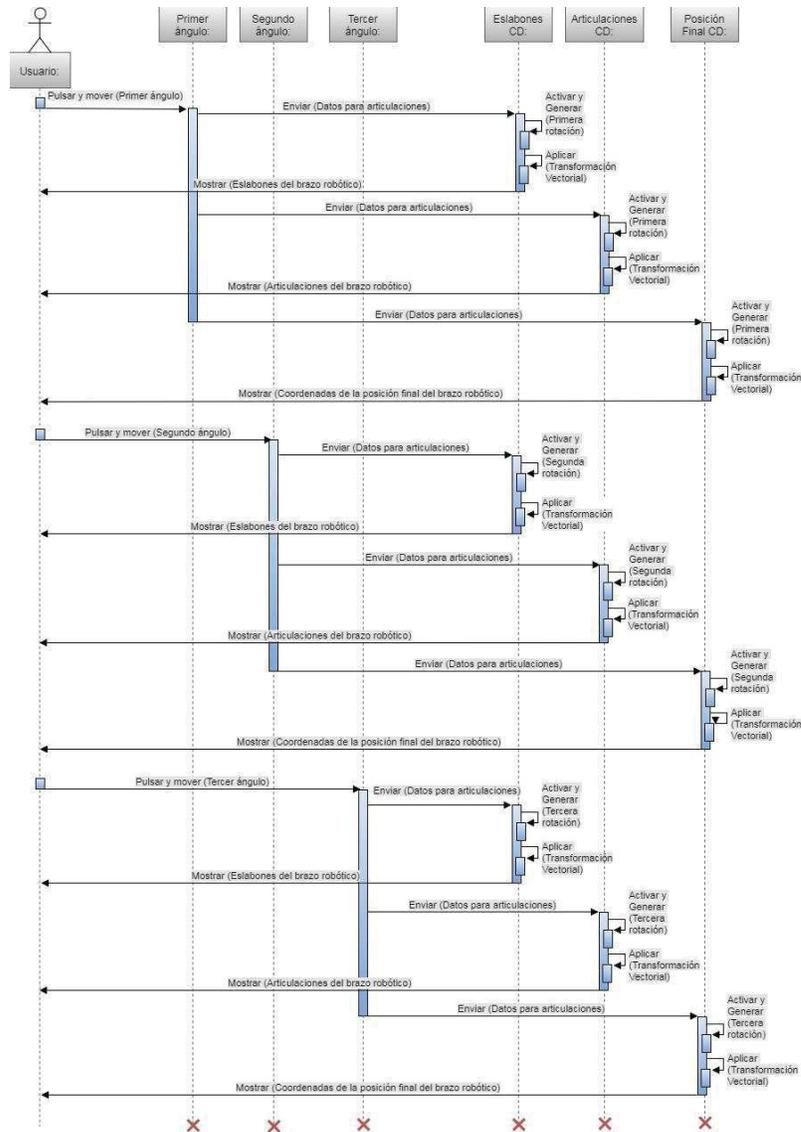


Figura 4. Diagrama de secuencia para el primer incremento.

3.2.2 Segundo incremento: cinemática inversa.

- Detección de la necesidad

Para sistematizar el problema de la cinemática inversa se requiere de:

- Utilizar las fórmulas del análisis previo al modelo de la cinemática inversa para poder graficar la estructura del robot.
- Utilizar las fórmulas del análisis previo al modelo de la cinemática inversa para poder visualizar en pantalla las coordenadas de la posición final del robot y los ángulos en los que se encuentra cada eslabón del robot manipulador conforme a la trayectoria.



- Planteamiento del problema

Se requiere conocer, analizar y simular el modelo cinemático inverso de un robot manipulador de 3 GDL, que utilice el álgebra de cuaterniones y un método numérico iterativo para resolver el problema cinemático inverso con respecto a una trayectoria preestablecida. El método numérico a utilizar es el método Newton-Rhapson.

El diagrama de caso de uso relacionado con el segundo incremento se muestra en la Figura 5, el diagrama de secuencia se muestra en la Figura 6 y la Figura 7 muestra la interfaz gráfica del problema inverso.

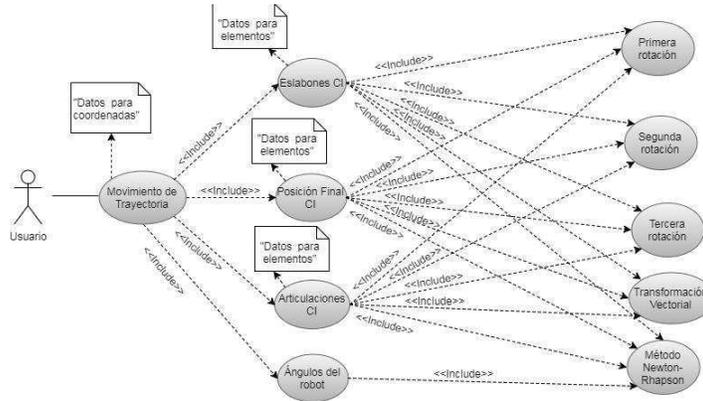


Figura 5. Diagrama de caso de uso del segundo incremento.

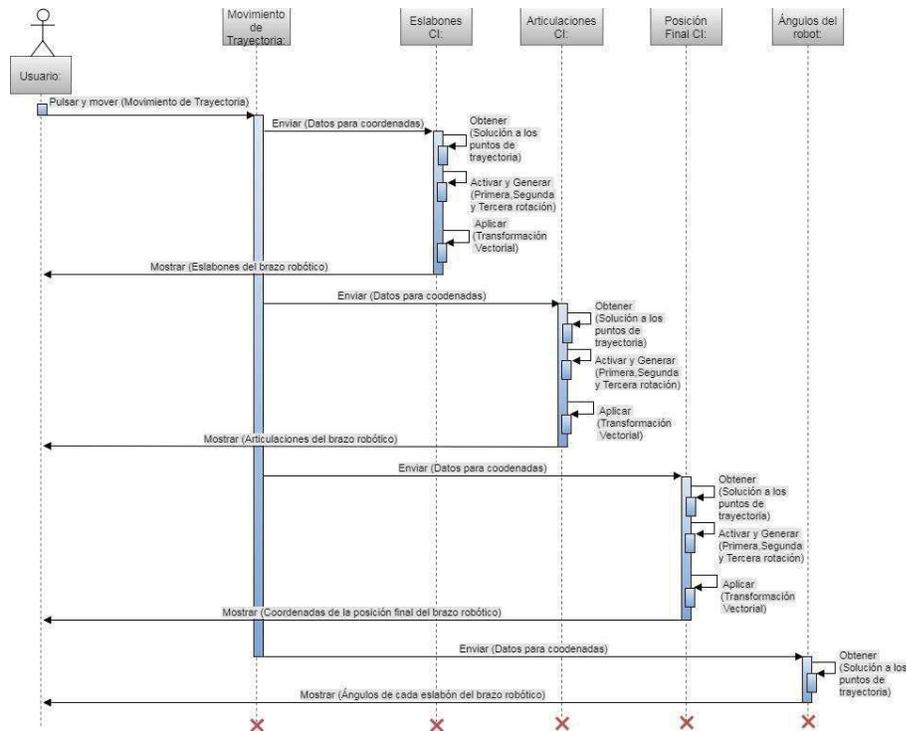


Figura 6. Diagrama de secuencia del segundo incremento.

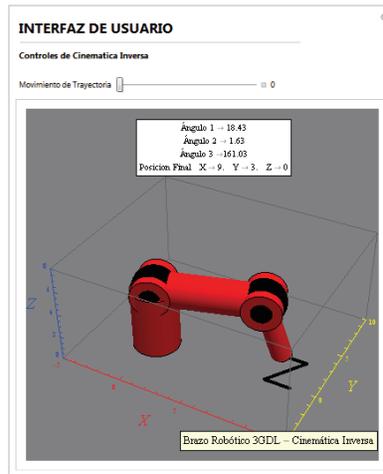


Figura 7. Interfaz del segundo incremento.

3.2.2 Tercer incremento: Unión de interfaces.

- Detección de la necesidad

En los incrementos anteriores se había generado una interfaz para la cinemática directa y otra para la cinemática inversa independientes una de otra. Se requiere ahora unir estas dos interfaces y generar una interfaz única.

- Planteamiento del problema

Se desea generar una interfaz única del simulador de la cinemática del robot manipulador de 3 GDL mediante la unión de la interfaz de la cinemática directa e inversa de los incrementos anteriores con el uso de la plataforma de programación Mathematica v8.

El diagrama de caso de uso para el tercer incremento se muestra en la Figura 8 y la interfaz gráfica del tercer incremento se observan en la Figura 9.

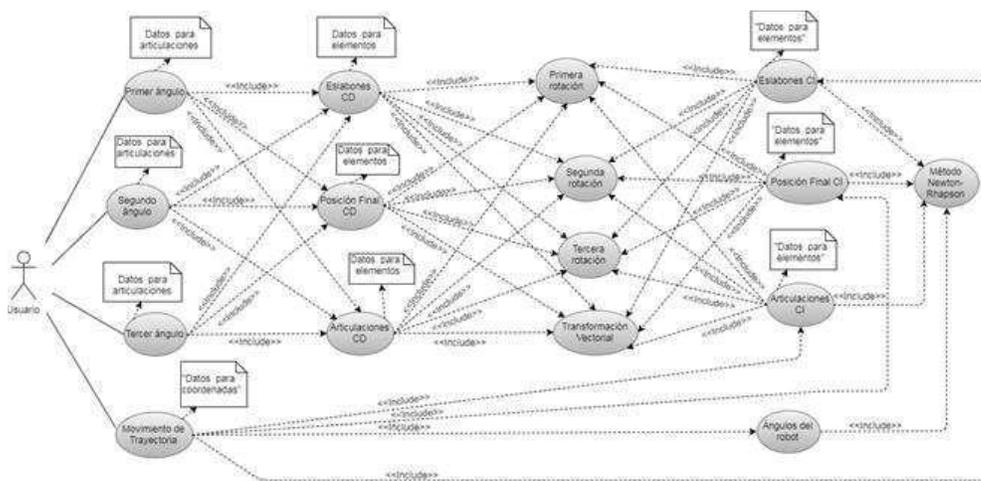


Figura 8. Diagrama de casos de uso del tercer incremento.

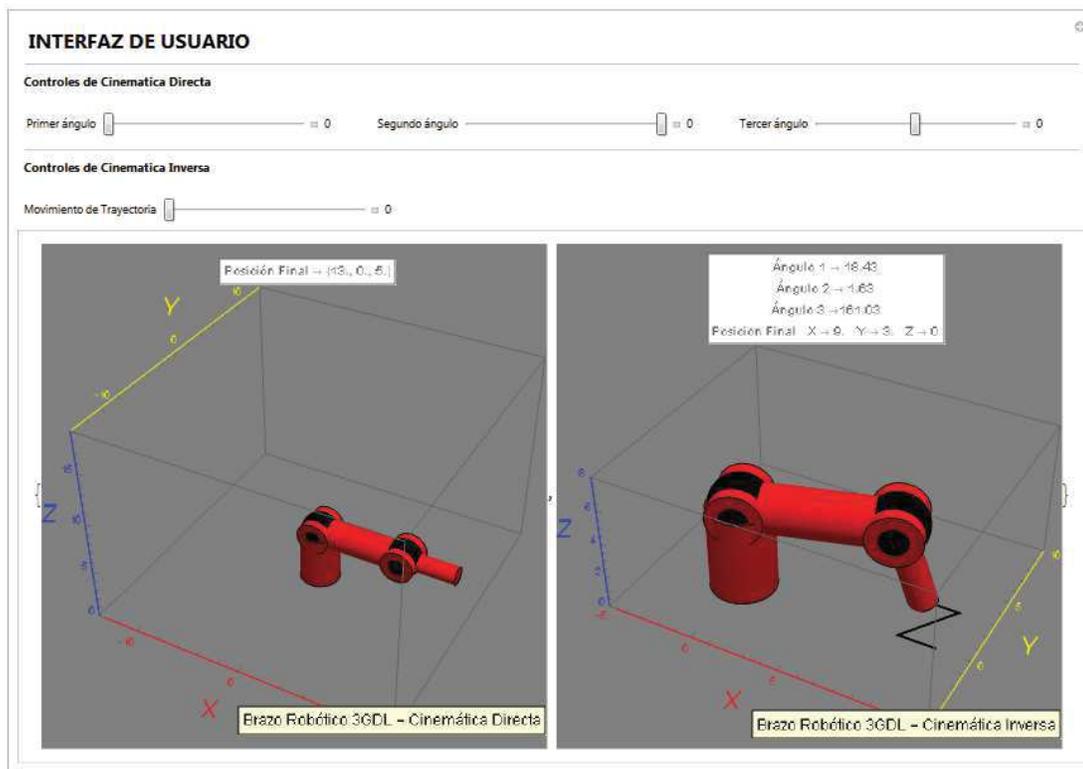


Figura 9. Interfaz del tercer incremento.

4. Conclusiones

En este artículo se ha presentado el desarrollo de un simulador robótico utilizando el método de ingeniería de software: modelo evolutivo de las entregas incrementales. Las principales conclusiones se resumen en los puntos siguientes:

- El método de las entregas incrementales permitió sistematizar el procedimiento para desarrollar un simulador cuyas características iniciales fueron de un programa realizado en clases, a un producto de software funcional y con valor agregado.
- La fundamentación del programa utilizando UML permitió tener una explicación del diseño a detalle, ordenada, sistemática, esquemáticamente visible y entendible.
- De los tres incrementos propuestos el más complicado fue el relacionado con la solución del problema cinemático inverso, por el hecho de que las pruebas requirieron la justificación del Newton-Rapshon ya que los modelos del robot generaron sistemas de ecuaciones no lineales.
- El método de las entregas incrementales se aplica en el desarrollo de programas independientemente de la plataforma o lenguaje de programación o modelo matemático. En este estudio se utilizó el software Mathematica v8 para el desarrollo del simulador robótico.

Agradecimientos

Los autores de este trabajo agradecen a las Universidades que conforman la Red ALFA (Universidad La Salle Noroeste, Universidad Tecnológica del Sur de Sonora, Instituto Tecnológico Superior de Cajeme) y a la empresa Innovación en Ingeniería de Manufactura y Mantenimiento S de RL MI., así como a la Universidad Tecnológica de Hermosillo, por el apoyo brindado a la presente investigación.



Referencias

- [1] Longorio, R., Jiménez, E., Rodríguez, L., Orduña, F., Reyes, L., López, J., Ochoa, F., Sánchez, F. (2009). “*Aplicación del método evolutivo incremental en el desarrollo de un simulador didáctico de trayectorias planas.*” Memorias del XV Congreso Internacional anual de la SOMIM 23 al 25 de septiembre, 2009 CD. Obregón, Sonora. México. pp, 513-524.
- [2] Sommerville, I. “*Software Engineering*”, Addison-Wesley Inter. Comp. Science series, 1993.
- [3] Pressman, R. “*Software Engineering*”. A Practitioners Approach. Fifth Edition. MC Graw Hill. 2001.
- [4] Rodríguez, L. “*Propuesta de un método de ingeniería de software para sistemas mecatrónicos: simulador de un péndulo balístico*”. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica.” Instituto Tecnológico Superior de Cajeme. 2009.
- [5] Booch, G., Rumbaugh, J., y Jacobson, I. “*The unified modelling language user guide*”. 2nd Edition, Addison-Wesley Professional. 2005
- [6] Estévez, E., Sánchez, A., Gámez, J., Gómez, J. “*Aproximación Basada en UML para el Diseño y Codificación Automática de Plataformas Robóticas Manipuladoras*”, Revista Iberoamericana de Automática e Informática industrial, 14, 82–93. 2017.
- [7] Serna, S. “*Desarrollo de un robot interactivo para la distribución de la información*”. Revista Tecnológicas, 18, pp. 137-169. 2007.
- [8] Diogo, C., Thiago, S. “*A finite element elasticity programming in Mathematica software*”, Computer Applications in Engineering Education, 2017.
- [9] Jiménez, E., Reyes, A., Islas, M., Ávila, F., Ochoa, F., Barceló, B. “*A 2-DOF Robot Modelation using Quaternions and Labview Simulations*”. VI International Conference on Engineering and Computer Education, pp. 1252-1256, Buenos Aires, Argentina. 2009.
- [10] Palomares, J. “*Modelación cinemática del robot CRS A465, utilizando el álgebra de Quaterniones*”. Tesis de Maestría en Ingeniería Mecánica. Universidad Nacional Autónoma de México. México, D.F. 2006.
- [11] Jiménez, E., Vázquez, I., Reyes, L., Chávez E., Rivera J., López F., Delfín, J., Urbalejo, A. “*Modelación y Simulación de un Robot Paralelo Plano Usando Quaterniones y Labview*”. 10º Congreso Nacional de Mecatrónica Noviembre 3 y 4. Puerto Vallarta, Jalisco. Asociación Mexicana de Mecatrónica A.C. Instituto Tecnológico Superior de Puerto Vallarta. pp. 269-274. 2011.
- [12] Jang, W., Jang, S., Hong, M. “*A Study on the Development of a Robot Vision Control Scheme Based on the Newton-Raphson Method for the Uncertainty of Circumstance*”. Transactions of the Korean Society of Mechanical Engineers A, 40, pp. 305-315. 2016.
- [13] Fadhil, H. “*Proposed Algorithm to Solve Inverse Kinematics Problem of the Robot*”. European Journal of Scientific Research, 49, pp. 376-384, 2018.



Diseño e Implementación de un Tablero Educatrónico para Apoyo de la Enseñanza de los Microcontroladores

¹Montiel Jiménez Elmar, ²Guzmán Rodríguez Alejandro,
³Córdova Morales Guillermo y ³Bonilla Reyes Susana.

¹²³Instituto Tecnológico Superior de Libres
pumandrin2@hotmail.com

Resumen

El presente artículo describe los resultados obtenidos de la metodología desarrollada para la construcción de un Tablero Educatrónico, el cual tiene como objetivo generar una herramienta amigable para la implementación de prácticas de electrónica y programación con microcontroladores para alumnos de nivel medio y superior, usando un microcontrolador ATMEGA328P como sistema principal, con 10 elementos electrónicos y electromecánicos seleccionados para adentrarse en el mundo de la programación de microcontroladores, además el tablero cuenta con una Web APP de apoyo, donde los estudiantes pueden consultar información relacionada al uso del tablero y varios ejemplos de prácticas que se pueden generar con el mismo. El Tablero Educatrónico fue diseñado por medio del software de Sketchup para su posterior impresión 3D y pueda ser fácil transportarlo y utilizarlo de una forma más ergonómica.

Palabras clave: Tablero Educatrónico, ATMEGA328P, Impresión 3D, Web APP.

1. Introducción

La Educatrónica desarrolla escenarios de aprendizajes científico que resultan muy atractivos para jóvenes, que permite desarrollar habilidades del pensamiento científico, creatividad y construcción orientada a una sociedad moderna con la ayuda de conceptos básicos de mecánica, electricidad, cinemática, informática, matemática y cuidado del medio ambiente, mientras construyen sus propias prácticas.

El término Educatrónica lo utilizó inicialmente el profesor mexicano Enrique Ruiz-Velasco Sánchez con una connotación orientada al uso de la robótica en la educación, especialmente educación secundaria. Luego el ingeniero colombiano Eduardo Cote construyó un modelo de pensamiento y plataformas especializadas en la convergencia de la Informática, la Mecatrónica, y la Educación.

Actualmente en el Instituto Tecnológico Superior de Libres se están desarrollando investigaciones enfocadas a la implementación y desarrollo de nuevas tecnologías aplicadas a la enseñanza de las ingenierías, por lo anterior descrito, surge como respuesta la necesidad del diseño e implementación de un Tablero Educatrónico a base de microcontroladores de uso general, que son herramientas conocidas a nivel mundial en el ámbito de la electrónica y programación, además cuenta con información de respaldo en la red. El Tablero Educatrónico se compone de dispositivos electrónicos (sensores, drivers), dispositivos electromecánicos (botones, relevadores y motores de CD), los cuales fueron seleccionados adecuadamente, de acuerdo a la popularidad, en los tutoriales analizados. También el tablero Educatrónico cuenta con la ayuda de una aplicación web móvil desarrollada con la ayuda de HTLM, BootSTRAP, Java Script y CCS, la que sirve como guía para la elaboración de las prácticas en sentido de las conexiones eléctricas y ejemplos de programación. Para el diseño del Tablero se ocupó el software de Sketchup, y para la impresión el software CURA y una impresora 3D marca Creality X.

El artículo está estructurado de la siguiente forma; en el apartado dos se encuentra el marco teórico de los principales elementos que se utilizaron en el desarrollo de la metodología, en el apartado tres, se puede observar la metodología utilizada para el desarrollo del prototipo, en el apartado cuatro se especifica un análisis de resultados que se realizaron sobre el tablero y por último en el apartado seis se muestran las conclusiones de la metodología desarrollada.



2. Marco de Referencia

2.1 *Microcontrolador Atmega328P*

Microcontrolador de alto desempeño Atmega328P de 8 bits, con arquitectura RISC. El Atmega328P en formato DIP es utilizado en el Arduino Uno R3.

Posee características como: 32 KB de memoria flash; con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interface serial orientada a byte de 2 cables, SPI e I2C, 6 canales 10-bit Conversor A/D, "WATCHDOG TIMER" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Su arquitectura le permite ejecutar instrucciones en un solo ciclo de reloj, alcanzando una potencia de 1 MIPS. En la figura 2.1 se puede apreciar la vista física del microcontrolador. [3]

Microcontrolador de alto desempeño Atmega328P de 8 bits, con arquitectura RISC. El Atmega328P en formato DIP es utilizado en el Arduino Uno R3. Las especificaciones técnicas:

- Chip: ATMEGA328P-PU
- Formato DIP
- Pines: 28
- Memoria FLASH: 32KB
- Memoria RAM: 2KB
- EEPROM: 1KB
- Máxima frecuencia de funcionamiento: 20Mhz
- CPU: 8-bit AVR
- Pines de entrada/salida: 23
- Entradas Analógicas (ADC): 6

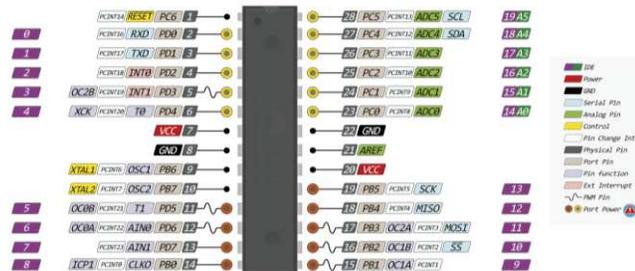


Figura 2.1 PINOUT del Microcontrolador Atmega328P.

2.2 *Impresión 3D*

La impresión 3D es un grupo de tecnologías de fabricación por adición donde un objeto tridimensional es creado mediante la superposición de capas sucesivas de material.

Las impresoras 3D son por lo general más rápidas, más baratas y más fáciles de usar que otras tecnologías de fabricación por adición, aunque como cualquier proceso industrial, estarán sometidas a un compromiso entre su precio de adquisición y la tolerancia en las medidas de los objetos producidos. Las impresoras 3D ofrecen a los desarrolladores del producto la capacidad para imprimir partes y montajes hechos de diferentes materiales con diferentes propiedades físicas y mecánicas, a menudo con un simple proceso de ensamble. Las tecnologías avanzadas de impresión 3D pueden incluso ofrecer modelos que pueden servir como prototipos de producto.



Desde 2003 ha habido un gran crecimiento en la venta de impresoras 3D. De manera inversa, el costo de las mismas se ha reducido. Esta tecnología también encuentra uso en campos tales como joyería, calzado, diseño industrial, arquitectura, ingeniería y construcción, automoción y sector aeroespacial, industrias médicas, educación, sistemas de información geográfica, ingeniería civil y muchos otros.

Un gran número de tecnologías en competencia están disponibles para la impresión 3D; sus principales diferencias se encuentran en la forma en la que las diferentes capas son usadas para crear piezas. Algunos métodos usan fundido o ablandando el material para producir las capas, por ejemplo sinterizado de láser selectivo (SLS) y modelado por deposición fundida (FDM), mientras que otros depositan materiales líquidos que son solidificados con diferentes tecnologías. En el caso de manufactura de objetos laminados, se cortan capas delgadas para ser moldeadas y unidas juntas, en la tabla 2.1 se muestran de forma resumida las principales tecnologías de impresión. [3]

Tabla 1.1 Principales tecnologías de impresión 3D.

Tipo	Tecnologías	Materiales
Extrusión	Modelado por deposición fundida (FDM)	Termoplásticos, por ejemplo PLA, ABS, HDPE, poliuretano termoplástico (TPU), metales eutécticos, materiales comestibles
Hilado	Fabricación por haz de electrones (EBF3)	Casi cualquier aleación
Granulado	Sinterizado directo de metal por láser(DMLS)	Casi cualquier aleación
Laminado	Fusión por haz de electrones (EBM)	Aleaciones de titanio
Fotoquímicos	Sinterizado selectivo por calor (SHS)	Polvo termoplástico

2.3 Web App

Durante los últimos años hemos visto como la forma de navegar por internet ha cambiado de forma radical. Atrás quedan esos tiempos de tener que esperar a llegar a casa para utilizar la computadora y ponernos al día de las noticias. Ahora, y con los avances tecnológicos, tenemos toda la actualidad al alcance de nuestra mano. La aparición de los terminales inteligentes ha supuesto todo un cambio de era pero también un cambio en nuestros hábitos de vida. Porque... ¿Quién de nosotros no va en el metro o en el bus con el smartphone en la mano informándose de todo lo está pasando en el mundo? Esta es la principal razón por la que muchos webmasters han visto el desarrollo de web apps para sus páginas como una clara necesidad.

Una Web App es una versión de la página web optimizada y adaptable a cualquier dispositivo móvil. Dicho de otra manera, es una página que se puede abrir desde el navegador de cualquier terminal independientemente del sistema operativo que utilice. Esta optimización es posible gracias a HTML5 y CSS3.

La principal ventaja que tienen es su capacidad de adaptación a cualquier dispositivo móvil, como hemos dicho antes. Es suficiente con que el dispositivo cuente con un navegador móvil actualizado, como Chrome o Safari. Eso sí, para poder visualizar la página correctamente en las diferentes plataformas es necesario incluir algún fragmento de códigos especiales para cada una. Pero hay más, no ocupa memoria en los dispositivos ya que no es una app que te puedas descargar desde los Market Places, no hacen falta actualizaciones ya que siempre se accederá a la última versión de la página como cualquier página web, al no ser una aplicación nativa, el consumo de recursos del dispositivo es mínimo. Por lo tanto, no afecta al rendimiento del mismo. [6]



3. Metodología

Una vez realizado el análisis de las herramientas propuestas, se procedió a diseñar una metodología con la cual se pudiera dar solución a la problemática planteada en el apartado uno, en la figura 3.1 se puede apreciar la metodología desarrollada.



Figura 3.1. Metodología desarrollada.

Como segunda etapa se procedió a realizar distintos bocetos del diseño del Tablero Educatrónico por medio del software de Sketchup, en la figura 3.2 se muestra un ejemplo de un boceto.

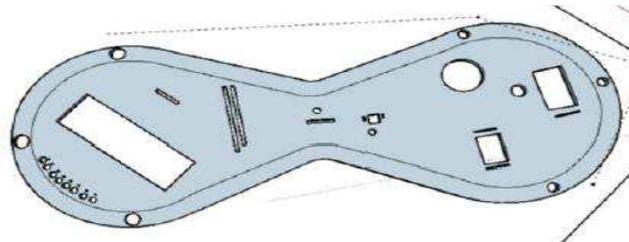


Figura 3.2 Diseño en Sketchup del Tablero.

Una vez concluido el diseño en el software de Sketchup, se continuó con la impresión en 3D del objeto diseñado, el software que se ocupó para administrar la impresora 3D es CURA y el modelo de la impresora es Creality X, en la siguiente figura 3.3 se puede mostrar un ejemplo de la configuración del software para imprimir una parte del tablero.

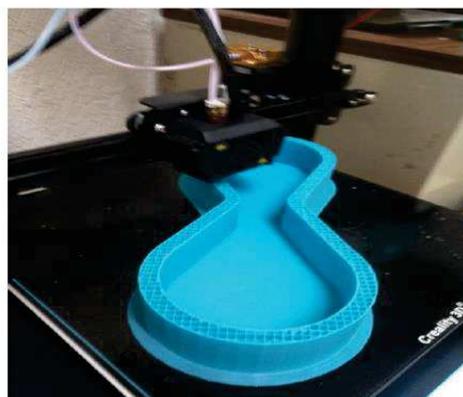


Figura 3.3 Proceso de impresión 3D del prototipo.

Concluida la etapa de impresión se procedió al montaje de los dispositivos electrónicos seleccionados, los cuales son; Motor de CD, Display de 7 segmentos, pantalla de LCD, potenciómetro, leds, push botton, bluetooth, sensor se temperatura y humedad HC11, en la figura 3.4 se puede observar el proceso de montaje.

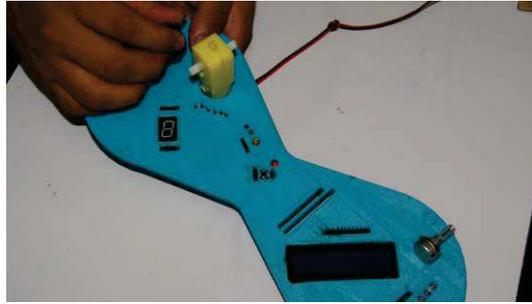


Figura 3.4 Conexión de dispositivos electrónicos.

Concluida la etapa dos, se propusieron y desarrollaron las prácticas, programación y simulación, con los dispositivos electrónicos antes mencionados, se incluirán en el tablero, para este caso se realizó en una versión demo el diseño de las prácticas en un demo de Proteus, como se puede apreciar un ejemplo en la figura 3.5, un ejemplo de barrido de leds.

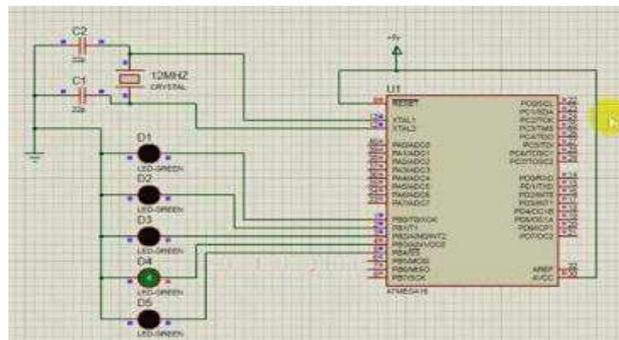


Figura 3.5 Simulación de un barrido de leds con Atmega 328P.

Para finalizar el proceso de desarrollo de la metodología, se realizó el boceto de la Web App, utilizando herramientas como HTML 5, CCS, Java Script y BootStrap, en la figura 3.6 se muestra el ejemplo del diseño propuesto.



Figura 3.6 Boceto del diseño de la página web.

4. Resultados

En el proceso de implementación de cualquier metodología siempre se encuentran errores, lo cual provoca cambiar algunas etapas de la planeación inicial y con el objetivo de tener mejores resultados. En la siguiente figura 4.1 se puede observar el resultado obtenido de la etapa de impresión 3D, y del montaje del tablero dando como resultado el Tablero Educatrónico.



Figura 4.1 Prototipo de Tablero Educatrónico.

Al final se desarrollaron siete prácticas que fueron simuladas, con el fin de eliminar la mayor cantidad de errores, dichas prácticas se agregaron en la página web, para que puedan ser desarrolladas por los alumnos, en la figura 4.3 se muestra la página web con los ejemplos desarrollados (<https://iscalejandro.000webhostapp.com/bootplus/index.html>).



Figura 4.3 Pagina Web.

Para comprobar la funcionalidad del prototipo Educatrónico se realizaron pequeños talleres a los alumnos de la carrera de Ingeniería en Sistemas Computacionales, (30 alumnos de diferentes semestres), donde se les explico el uso de la tecnología y como realizar conexiones físicas utilizando la página web desarrollada, al finalizar la sesión se les realizo un cuestionario donde los resultados se observan en la tabla 4.1.

Tabla 4.1 Preguntas sobre tablero Educatrónico.

No.	Pregunta	SI	NO
1	¿Tiene buen aspecto el Tablero?	90%	10%
2	¿Es fácil de conectar el Tablero?	95%	5%
3	¿Te gustaría contar con esta herramienta en laboratorios?	100%	0%
4	¿Tiene buena imagen la página web de apoyo?	100%	0%
5	¿Te sirvió la página web como apoyo?	90%	10%

Como se puede apreciar en la tabla 4.1, al realizar las preguntas se observa que el tablero tiene una buena aceptación y funcionalidad, ya que las primeras tres preguntas referentes al tablero fueran contestadas con un rango entre 90% a 100%, en cuanto a la página web también se puede claramente que es funcional y da muchas ventajas el utilizar esta tecnología web.



5. Conclusiones

Es de gran utilidad el uso de la impresión 3D ya que te permite un prototipado rápido de proyectos, el único defecto son las dificultades que presenta al moverse las impresiones o la mala configuración de la impresión, que en este caso fue con el software de Cura. El utilizar herramientas CAD, como lo es el software de Sketchup, el cual se utilizó para realizar los bocetos del tablero Educatrónico, ya que es una herramienta amigable y muy intuitiva para su uso, brinda una alta posibilidad de evitar errores al realizar proyectos físicos.

Siempre la realización de herramientas educativas que apoyen a los docentes, serán bienvenidas, ya que facilitan el proceso de enseñanza aprendizaje, y ahorro de tiempo a la hora de implementar prácticas, y que los estudiantes solo utilicen sus placas de desarrollo para los proyectos integradores.

La realización de una página web de apoyo, permite tanto a docentes como alumnos realizar las prácticas sin la necesidad de ser expertos en estas áreas, ya que en el contenido se puede encontrar desde la forma de realizar la conexión hasta el código fuente del programa que se monta al microcontrolador.

6. Referencias

- [1] Carrión, P., Castro Gil, M., Díaz Orueta, G., & García-Sevilla. (12 de 07 de 2006). Nuevos materiales educativos en la enseñanza teórica y práctica de la electrónica. . Recuperado el 2016, de <http://e-spacio.uned.es/fez/view/taee:congreso-2006-1020>.
- [2] Compañ, P., Satorre, R., Llorens, F., & Molina, R. (15 de 09 de 2015). redalyc. Recuperado el 10 de 05 de 2016, de Enseñando a programar: un camino directo para desarrollar el pensamiento computacional: <http://www.redalyc.org/articulo.oa?id=54741184011>.
- [3] Equipo_Educacontic. (2015). Educa con TIC. Obtenido de RÓBOTICA EDUCATIVA CON ARDUINO EN EL AULA DE ESO: <http://www.educacontic.es/blog/robotica-educativa-con-arduino-en-el-aulade-eso-incubegg-kubo-e-izar-galaktik-mertxe-j-badiola>.
- [4] Marotias, L. (2016). Educar. Obtenido de Conectando el mundo digital y el analógico con Arduino: <https://www.educ.ar/sitios/educar/recursos/ver?id=125792>
- [5] Méndez, A., & UNIR. (04 de 2015). Universidad Internacional de la Rioja. Recuperado el 21 de 05 de 2016, de <http://www.unir.net/ingenieria/curso-electronica-robotica-impresion3d/549200191864/>.
- [6] Velandía, M. A., Fernández Morales, F. H., & Duart, J. E. (2007). Material educativo computarizado para enseñanza de la instrumentación básica en electrónica. Recuperado el 2016, de <http://www.redalyc.org:9081/articulo.oa?id=25702100801>

Tecnificación Mecatrónica

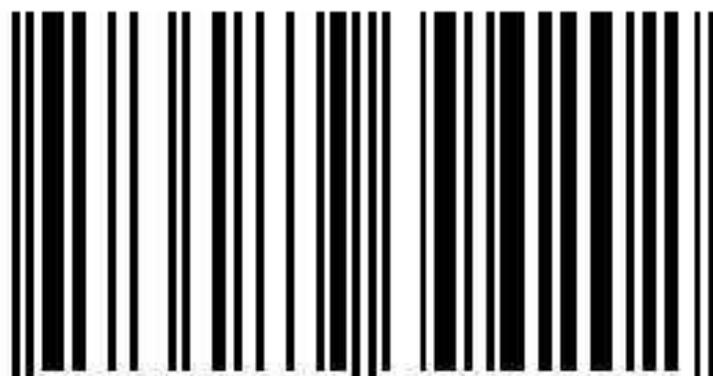
Asociación Mexicana de Mecatrónica A.C.
Prolongación Corregidora Norte No. 168, interior E
Col. El Cerrito, C.P. 76160, Querétaro, Qro.
Distribución nacional e internacional en formato
tipo internet, pp. 397, Tamaño: 40 Mb.

ISBN: 978-607-9394-20-2

Tecnificación Mecatrónica

Vargas Soto José Emilio, Obil Martínez Octavio
y Orozco Ramírez Jorge Enrique

ISBN: 978-607-9394-20-2



9 786079 394202

