

# Desarrollo de VI (Virtual Instruments) en LabVIEW para la simulación y el control de un robot paralelo didáctico

Velderrain Ruiz Sergio<sup>1</sup>, Jiménez López Eusebio<sup>2</sup>, Urbalejo Contreras Arturo<sup>3</sup>, Núñez Pérez Eduardo<sup>4</sup>, Reyes Ávila Luis<sup>5</sup>, Torres Campoy José Mario<sup>6</sup>, Luna Acosta Noé<sup>7</sup>, Vázquez Cuevas Ignacio<sup>8</sup>.

<sup>1</sup>Universidad La Salle Noroeste – Pinnacle Aerospace. [svelderrain@hotmail.com](mailto:svelderrain@hotmail.com)

<sup>2</sup>CINNTRA-CIAAM de la Universidad Tecnológica del Sur de Sonora-ULSA Noroeste-IIMM

<sup>3,7</sup>CIAAM de la Universidad Tecnológica del Sur de Sonora

<sup>4,6</sup>Universidad La Salle Noroeste

<sup>5</sup>Instituto Mexicano del Transporte

<sup>8</sup>Universidad Tecnológica de Nogales, Sonora

## Resumen

*En este artículo se utiliza el software LabVIEW y la tarjeta Arduino para construir un simulador y un control de posición de un robot paralelo planar didáctico. Para operar el control con LabVIEW fue necesario importar una función hecha en MatLAB la cual integra el modelo cinemático inverso del robot modelado con Cuaterniones. Se usa un archivo en Excel para ingresar datos sobre las trayectorias que el robot puede realizar para posteriormente evaluar dichos datos en la función de la cinemática del robot. El sistema generado en LabVIEW consta de cinco módulos: 1) de inicialización, 2) interfaz de usuario, 3) procesamiento de datos, 4) simulación en 3D y 5) salidas digitales para los servomotores. Los resultados obtenidos en este trabajo pueden ser usados para la enseñanza de la Mecatrónica.*

**Palabras clave:** LabVIEW, Cuaterniones, Robots Paralelos.

## 1. Introducción

En las aplicaciones de la Mecatrónica, la simulación computacional es una herramienta muy poderosa que utilizan las industrias para el diseño y la modificación de los sistemas productivos. Simular es sinónimo de imitar el comportamiento de un sistema con algún propósito específico [1]. La simulación computacional requiere no sólo de modelos

matemáticos, sino también de plataformas computacionales que permitan visualizar los movimientos de la maquinaria o de los elementos de trabajo.

Una de las tareas más importantes en el estudio y las aplicaciones de la Mecatrónica es la de seleccionar el software más adecuado para simular los sistemas mecánicos, así como las herramientas matemáticas que soportan la modelación [2]. Sin embargo, no existe una metodología única que indique o señale qué o cuál software usar para una determinada aplicación, más bien los diseñadores seleccionan los paquetes computacionales de acuerdo a su experiencia, a la exigencia de un proyecto o al tipo de aplicación, entre otras características y criterios. Por ejemplo, hay paquetes de cálculo simbólico como Mathematica, MatLAB o Maple, los cuales son usados para programar modelos matemáticos relacionados con el fenómeno del movimiento. En [3] se puede observar el uso de Mathematica para simular los movimientos de un robot delta Hércules. Otros paquetes como el LabVIEW, se usan para generar el control o para el desarrollo de interfaces entre un modelo de simulación y los actuadores de una máquina. Por ejemplo en [4], se usó LabVIEW para controlar dos servomotores de aeromodelismo para un robot delta plano. Otros paquetes que se usan en la Mecatrónica son los de dibujo y análisis de ingeniería por computadora. En [5] se usa *Edge* para diseñar un robot delta.

Otra opción es usar la co-simulación como alternativa para las aplicaciones en Mecatrónica. Por ejemplo en [6] se usa de forma integrada MatLAB y LabVIEW para evaluar y monitorear señales en telecomunicaciones. En [7] se ha usado co-simulación para modelar, programar y simular los movimientos de un robot cartesiano industrial. En este caso se usó Mathematica, LabVIEW y SolidWorks. Cabe mencionar que la co-simulación puede desarrollarse de dos maneras: 1) usando primero un software para obtener funciones o datos, y posteriormente usar otro paquete y/o 2) usar dos paquetes en tiempo real. Para mostrar los beneficios de la co-simulación, en este artículo se presenta la simulación y el control de un robot paralelo delta planar didáctico [8] usando MatLAB y LabVIEW. Para el desarrollo de la simulación se hará uso del modelo del robot desarrollado en [9] y modelado con Cuaterniones [10]. La estructura del artículo se describe a continuación: en la parte 1 se presenta la introducción, en la parte 2 se describe el prototipo del robot paralelo, en la parte 3 se describen los resultados del modelo y la programación y control en LabVIEW y, finalmente, se presentan las conclusiones.

## 2. Prototipo del robot paralelo

El robot motivo de estudio en este artículo se muestra en la figura 1. Dicho robot es usado para propósitos didácticos en la Universidad La Salle Noroeste [8,11].



Fig. 1. Prototipo robot paralelo didáctico

El prototipo de robot mostrado en la figura anterior, está construido con eslabones de aluminio y plataformas con postes de acero inoxidable. El movimiento del robot es producido por tres servomotores de aeromodelismo Power HD

hd1501mg. La transmisión entre los ejes del robot y los eslabones es por medio de un sistema de engranes rectos. El punto de análisis es en el centroide de la plataforma móvil (eslabón circunferencial) y consta de tres cadenas cinemáticas independientes y en cada cadena está localizado un servomotor. El robot en estudio es de 3GDL y consta de 8 eslabones rígidos incluyendo la tierra del sistema (plataforma fija triangular).

## 3. Análisis de resultados

En esta sección se presenta el modelo del robot y la programación hecha en LabVIEW para el control y la simulación de sus movimientos.

### 3.1 Modelo del robot paralelo.

El modelo matemático del robot mostrado en la figura 2 fue desarrollado en [9]. Dicho modelo fue modelado usando el álgebra de Cuaterniones [10].

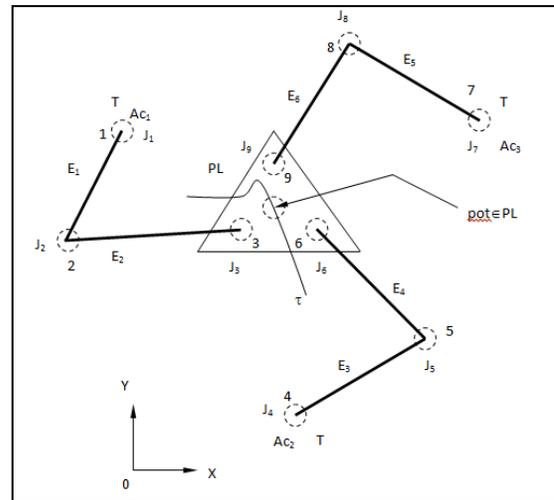


Fig. 2. Configuración del robot paralelo

Las ecuaciones cinemáticas relacionadas con el robot motivo de estudio son las siguientes:

$$1) \Gamma_{3,0}' = \Gamma_{1,0} \oplus I_{2,1} \cdot \{ P \otimes p \otimes \underline{e}_1 \otimes \bar{p} \otimes \bar{P} \} \oplus I_{3,2} \cdot \{ Q \otimes q \otimes \underline{e}_1 \otimes \bar{q} \otimes \bar{Q} \} \quad (1)$$

$$2) \Gamma_{6,0}' = \Gamma_{4,0} \oplus I_{5,4} \cdot \{ R \otimes r \otimes \underline{e}_1 \otimes \bar{r} \otimes \bar{R} \} \oplus I_{6,5} \cdot \{ S \otimes s \otimes \underline{e}_1 \otimes \bar{s} \otimes \bar{S} \}$$

$$3) \Gamma_{9,0}' = \Gamma_{7,0} \oplus I_{8,7} \cdot \{ T \otimes t \otimes \underline{e}_1 \otimes \bar{t} \otimes \bar{T} \} \oplus I_{9,8}$$

$$\cdot \{ U \otimes u \otimes \underline{e}_1 \otimes \bar{u} \otimes \bar{U} \}$$

Aquí,  $p, P, q, Q, r, R, s, S, t, T, u, U \in \mathfrak{R}^4$  son Cuaterniones de norma unitaria,  $\Gamma_{3,0} \in \mathfrak{R}^4, \Gamma_{6,0} \in \mathfrak{R}^4, \Gamma_{9,0} \in \mathfrak{R}^4, \Gamma_{1,0} \in \mathfrak{R}^4, \Gamma_{4,0} \in \mathfrak{R}^4, \Gamma_{7,0} \in \mathfrak{R}^4$ , son vectores de posición y  $l_{2,1}, l_{3,2}, l_{5,4}, l_{6,5}, l_{8,7}, l_{9,8}, l_{3,6}, l_{3,9}, l_{pot,3} \in \mathfrak{R}^+$  son las dimensiones de los eslabones; además, las ecuaciones de norma unitaria son las siguientes:

$$\begin{aligned} 1) P_0^2 + P_3^2 &= 1; Q_0^2 + Q_3^2 = 1; R_0^2 + R_3^2 = 1 \\ 2) S_0^2 + S_3^2 &= 1; T_0^2 + T_3^2 = 1; U_0^2 + U_3^2 = 1 \end{aligned} \quad (2)$$

Por otro lado, las relaciones entre los parámetros de los Cuaterniones y los desplazamientos angulares de los eslabones medidos con respecto al eje X son las siguientes:

$$\begin{aligned} P_o &= \pm \cos \frac{\alpha_1}{2}, \underline{P}_V = \pm \sin \frac{\alpha_1}{2} \underline{a}_3 \\ Q_o &= \pm \cos \frac{\alpha_2}{2}, \underline{Q}_V = \pm \sin \frac{\alpha_2}{2} \underline{a}_3 \\ R_o &= \pm \cos \frac{\alpha_3}{2}, \underline{R}_V = \pm \sin \frac{\alpha_3}{2} \underline{a}_3 \\ S_o &= \pm \cos \frac{\alpha_4}{2}, \underline{S}_V = \pm \sin \frac{\alpha_4}{2} \underline{a}_3 \\ T_o &= \pm \cos \frac{\alpha_5}{2}, \underline{T}_V = \pm \sin \frac{\alpha_5}{2} \underline{a}_3 \\ U_o &= \pm \cos \frac{\alpha_6}{2}, \underline{U}_V = \pm \sin \frac{\alpha_6}{2} \underline{a}_3 \end{aligned} \quad (3)$$

Cabe mencionar que el modelo completo de este robot puede consultarse en [9]. Hay que considerar que las ecuaciones de lazo permiten conocer las coordenadas del punto *pot* (ver figura.2) y la orientación de la plataforma PL medida sobre el eje Z desde el sistema de referencia XY. Por otro lado, las ecuaciones de norma unitaria se usan para completar el sistema de ecuaciones. Finalmente, las relaciones entre los parámetros de los Cuaterniones y las rotaciones son importantes en el sentido de que con dichas relaciones se conocen los ángulos de rotación de los eslabones y los ejes de rotación que, para este caso, están en el eje Z.

### 3.1.1 Problema inverso

“ Dados  $\Gamma_{3,0} \in \mathfrak{R}^4, \Gamma_{6,0} \in \mathfrak{R}^4, \Gamma_{9,0} \in \mathfrak{R}^4, \Gamma_{1,0} \in \mathfrak{R}^4, \Gamma_{4,0} \in \mathfrak{R}^4, \Gamma_{7,0} \in \mathfrak{R}^4, l_{2,1}, l_{3,2}, l_{5,4}, l_{6,5}, l_{8,7}, l_{9,8}, l_{3,6}, l_{3,9}, l_{pot,3} \in \mathfrak{R}^+, p = \{p_0, 0, 0, p_3\}, q = \{q_0, 0, 0, q_3\}, r = \{r_0, 0, 0, r_3\}, s = \{s_0, 0, 0, s_3\}, t = \{t_0, 0, 0, t_3\}, u = \{u_0, 0, 0, u_3\};$  con  $\|p\| = \|q\| = \|r\| = \|s\| = \|t\| = \|u\| = 1$ , encuentre:  $P = \{P_0, 0, 0, P_3\}, Q = \{Q_0, 0, 0, Q_3\}, R = \{R_0, 0, 0, R_3\}, S$

$= \{S_0, 0, 0, S_3\}, T = \{T_0, 0, 0, T_3\}, U = \{U_0, 0, 0, U_3\}$  tal que las expresiones (1) y (2) sean satisfechas.

El problema inverso del robot motivo de estudio genera un sistema de 12 ecuaciones no lineales con 12 incógnitas del tipo polinomial, por lo que es necesario usar un método numérico. En este artículo se usó el Newton – Raphson [9].

## 3.2 Simulación y control en LabVIEW.

En esta sección se presenta el desarrollo de la simulación y el control del prototipo de robot paralelo mostrado en la figura 1. La modularización en LabVIEW se muestra en la figura 3.

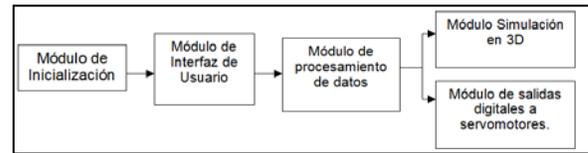


Fig. 3. Módulos del sistema

### 3.2.1 Simulación y control en LabVIEW.

La figura 4 muestra el VI (Virtual Instruments) o instrumento virtual relacionado con el módulo de inicialización.

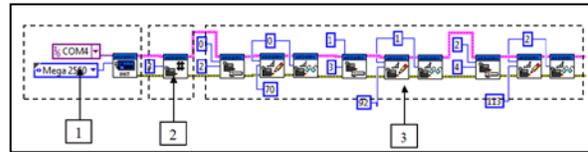


Fig. 4. Inicialización con Arduino

A continuación se describen las partes mostradas en la figura anterior.

- 1.- Esta parte de la inicialización recibe como argumentos el puerto en el cual está el Arduino y el tipo de Arduino que se está utilizando para realizar la conexión. Sin estos argumentos puestos de manera correcta, el programa no logrará una comunicación.
- 2.- Esta parte de la inicialización se describe cuántos servomotores habrá en la sesión de Arduino.
- 3.- Esta parte le da una dirección a la Arduino y también le proporciona una posición inicial (al robot) que, en este caso, se ajustó a prueba y error para que empiece en una posición tal que el robot no se estuviera esforzando.

Por otro lado, la figura 5 muestra el VI asociado con la interfaz de usuario. A continuación se describen las partes mostradas en la figura 5.

- 1.- Esta parte del programa selecciona de dónde van a venir los datos (X, Y y Theta) en el programa. Si el selector está en la posición de TRUE, selecciona la interfaz automatizada por Excel; y si está en la posición de FALSE, le da al usuario la opción de escoger entre el control por medio de sliders o de un joystick.
- 2.- Esta parte del programa selecciona ya sea control por medio de sliders o por medio de joysticks dependiendo del estado en el que esté el switch (TRUE/FALSE).

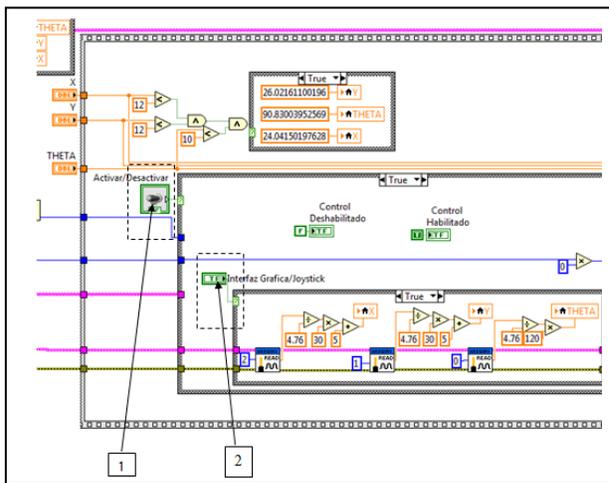


Fig. 5. Interfaz de usuario

La figura 6 muestra el VI relacionado con el procesamiento de datos. A continuación se describen las partes mostradas en dicha figura.

- 1.- Aquí se muestran las variables de entrada de la estructura MathScript, las cuales representan las salidas X, Y y Theta de la interfaz de usuario.
- 2.- Este es el modelo matemático dentro del ambiente de desarrollo de la estructura de MathScript (el modelo cinemático fue programado en MatLAB).
- 3.- Estas son las 6 variables de salida, las cuales representan los ángulos de los 6 codos del robot, ángulos que se utilizarán para el dibujado en 3D y como argumentos para los subVI de manejo de servomotores.
- 4.- Está *case structure*, cuando está en su estado falso, hace que se imprima en la interfaz de usuario la leyenda de "Área de trabajo fuera de rango" y activa un Buzzer para alertar al usuario que el robot no puede operar en esa área. Además, al momento de entrar a su estado de FALSE, no va a alimentar ningún ángulo, ni al dibujado en 3D ni a la impresión de servomotores, debido a que serían datos

incorrectos, así que guarda la última solución a la que se llegó antes de entrar en la zona que no opera robot para mantenerlo en esa posición.

5.- Esta es la variable booleana que indica que el modelo matemático no puede encontrar una solución y activa la *case structure* para que alerte al usuario y proteja al robot como se indica en el punto 4.

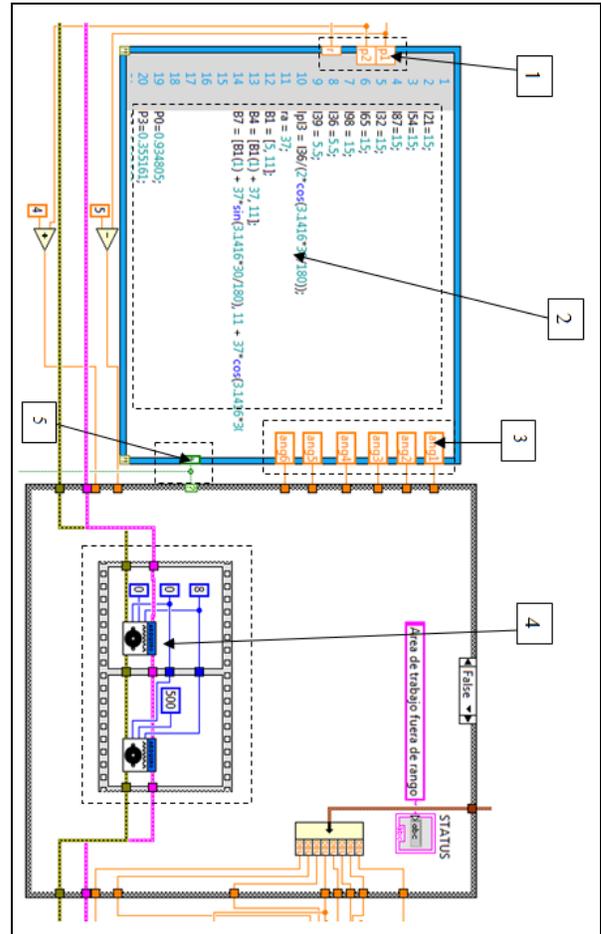


Fig. 6. VI de procesamiento de datos

Por otro lado, la figura 7 muestra el VI de simulación.

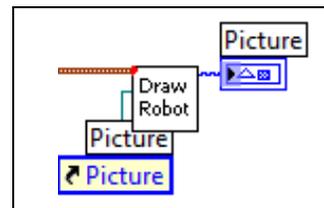
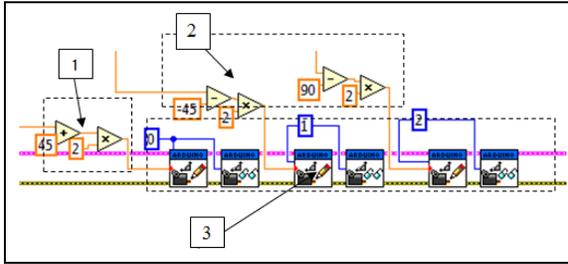


Fig. 7. VI de simulación

Esta parte del programa recibe el conjunto de elementos obtenidos mediante el modelo matemático y se encarga de dibujar en 3D la estructura del robot.

La figura 8 muestra el VI de salidas digitales para los servomotores.

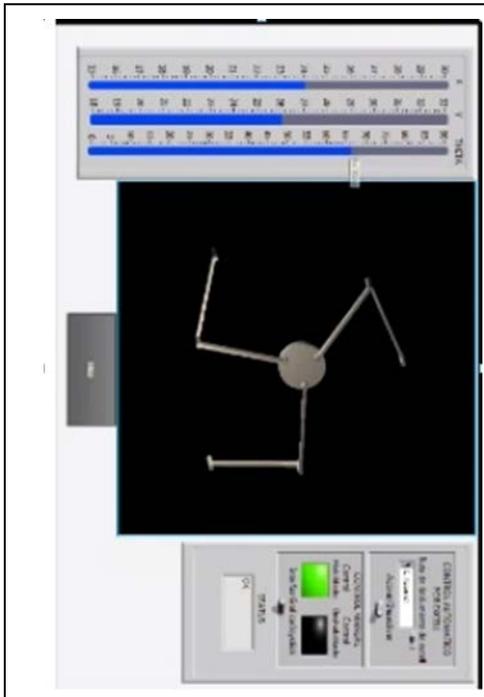


**Fig. 8. VI de salidas digitales a los servomotores.**

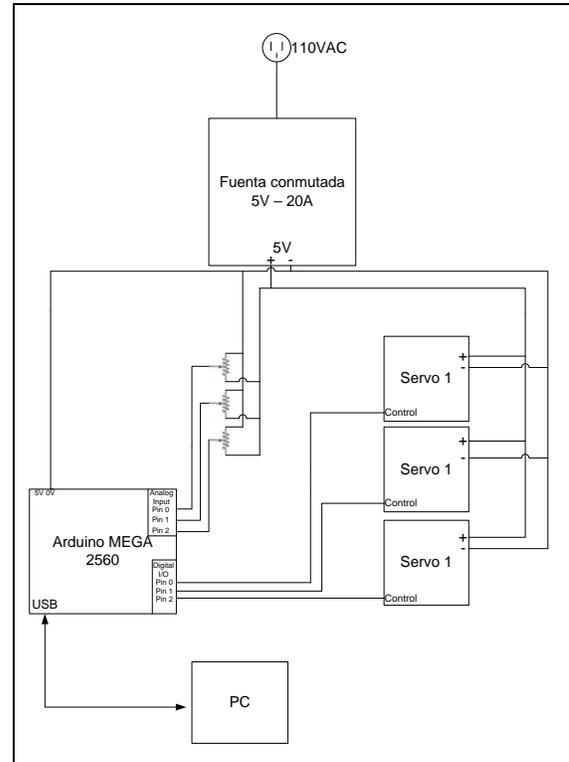
A continuación se describen las partes mostradas en la figura anterior:

- 1.2.- Estas partes del programa se encargan de tomar los ángulos proporcionados por el modelo matemático y darles el ajuste final para que se mueva el servomotor de manera correcta.
- 3.- Estos subVI se encargan de darle el ángulo correcto al servomotor, dependiendo del número o dirección que se les asignó en la inicialización.

Finalmente, las figuras 9, 10 y 11 muestran la salida gráfica del robot en LabVIEW, el diagrama de conexiones y el robot con su sistema de control.



**Fig. 9. Salida gráfica del simulador en LabVIEW.**



**Fig. 10. Diagrama de conexiones.**



**Fig. 11. Robot y caja de control.**

#### 4. Conclusiones

En este artículo se ha usado LabVIEW, MatLAB y la tarjeta Arduino para simular y controlar un robot paralelo planar didáctico. Las principales conclusiones se resumen en los puntos siguientes:

- LabVIEW, en cuanto al desarrollo del control para el robot paralelo, presenta ventajas con respecto al tiempo de desarrollo contra sus competidores (C++, C#) debido a

su naturaleza de poder integrar muchas disciplinas y módulos en un solo ambiente de desarrollo.

- La manera en la que se programó el robot, través del seguimiento de una secuencia ordenada y encapsulando rutinas dentro de Sub VI, hace al control del mismo fácil de entender para futuros desarrolladores que se basen en este trabajo
- Arduino, por ser Open Source y de bajo costo, tiene la misma funcionalidad que una Tarjeta DIO de National Instruments de 700 dólares, aparte de que por ser una tarjeta muy usada existe ya una biblioteca para poder manipular al arduino como una DIO.
- El control y la simulación se facilitaron por el hecho de usar una función única de la cinemática inversa del robot paralelo.

## Referencias

- [1] Jiménez E., Navarro J., Reyes L, Luna N., Urbalejo A., Islas M., Castro J. *Simulación de una celda de manufactura usando Quaterniones y Mathematica*. 11° Congreso Nacional de Mecatrónica, Octubre 4 y 5. Villahermosa, Tabasco. 2012.
- [2] Jiménez E., Islas M., Reyes L., Vázquez I., Urbalejo A., Castro J., Velderrain S. *Uso del software Labview y C# para la programación y simulación de los movimientos de multicuerpos rígidos en el plano. Entorno Académico*. Año 9, Número 10, 2012.
- [3] Meneses X., Méndez M., Cortez E. *Diseño y Control de un Robot Paralelo*. 6to. Congreso Nacional de Mecatrónica, Noviembre 8-10, Instituto Tecnológico de San Luis Potosí, S.L.P. Asociación Mexicana de Mecatrónica A.C. 2007
- [4] Chávez M., Rivera J. *Software e interfaces de control de servomotores del tipo Brushless, con aplicación en robot paralelo*. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica. Instituto Tecnológico Superior de Cajeme. 2008.
- [5] Rueda J. *Metodología para el diseño de un robot paralelo industrial tipo delta*. Monografía. Universidad Pontificia Bolivariana. Escuela de Ingeniería y Administración. 2008.
- [6] Armenta E. *Estudio de un radio enlace satelital utilizando modelos basados en series de Volterra para el amplificador de potencia*. Tesis de Licenciatura en Ingeniería Mecatrónica. Universidad La Salle Noroeste. Cd. Obregón, Sonora, México. 2013.
- [7] Sobarso J. *Modelo y simulación de las trayectorias de un robot cartesiano usando la integracion LabVIEW-SolidWorks para aplicaciones industriales*. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica. Instituto Tecnológico Superior de Cajeme. 2012.
- [8] Jiménez E., Bojórquez G., García L., Amavizca L., Martínez V., López L., Luna G., Hernández E. *Experiencias de la aplicación del ABP en una materia de mecanismos en la Universidad La Salle Noroeste*. VIII International Conference on Engineering and Computer Education. March 03 - 06, Luanda, ANGOLA. Pp. 421-425, 2013
- [9] Reyes L., Jiménez E., Olvera E., Vázquez I., Mendiola E., Rivera J., López F., Vázquez D., Urbalejo A. (2012). *Aplicaciones del álgebra hipercompleja a la modelación y simulación de un robot paralelo planar de 3GDL, tipo RRR. La Mecatrónica en México*, Vol. 1, No. 1, páginas 1 – 11, 2012.
- [10] Reyes L. *Quaternions: Une Representation Parametrique Systematique Des Rotations Finies. Partie 1: Le Cadre Theorique*. Rapport de Recherche INRIA Rocquencourt. France. 1990
- [11] Velderrain S. *Control de un robot paralelo delta planar usando LabVIEW y Arduino*. Tesis de Licenciatura en Ingeniería Mecatrónica. Universidad La Salle Noroeste. Cd. Obregón, Sonora, México. 2013.