

Diseño de un Simulador 3D para Robots Manipuladores de 3 Grados de Libertad

Bravo Hernández Paola, Reyes Cortes Fernando, Vergara Limón Sergio, Ríos Silva Eduardo

Facultad de Ciencias de la Electrónica, Posgrado en Ciencias de la Electrónica, División de Robótica.
Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 18 Sur Edificio 182-CU, Col. Jardines de San Manuel, Puebla, Pue. 72570, México
Tel. 229-55-00 ext. 7411.

Resumen

Este artículo expone el desarrollo de un simulador con ambiente gráfico tridimensional para Robots Manipuladores de 3 grados de libertad llamado SimRoMa, con programación en arquitectura abierta facilitando el mantenimiento y modificación del programa. SimRoMa es capaz de simular el movimiento real de un robot de transmisión directa de tres grados de libertad (Rotradi) dependiendo del algoritmo de control a evaluar y de la posición deseada. Este robot es usado dentro del Laboratorio de de Robótica en la Facultad de Ciencias de la Electrónica de la BUAP. El software del simulador fue programado en Microsoft Visual C++ 2008 con formato Win32, que incluye programación orientada a objetos, y con la librería de DirectX se implemento el motor gráfico 3D, logrando mejorar la visión espacial del comportamiento dinámico del robot, se obtiene el modelo dinámico y las ecuaciones de movimiento de Lagrange para evaluar algoritmos experimentales para la manipulación del prototipo real, y mejorando la capacidad de anticipar colisiones y daños mecánicos al robot o al entorno dentro de su espacio de trabajo. Finalmente se muestran los resultados obtenidos con el simulador al evaluar algoritmos de control experimentales para la definición de su trayectoria y el par de fuerzas aplicados..

Palabras clave: Gráficos 3D, Robot 3GDL, Lagrange, Modelo dinámico.

1. Introducción

Este trabajo muestra el desarrollo de un software de simulación de un robot manipulador de tres grados de

libertad donde la característica del robot principalmente es de motores de transmisión directa. Un simulador 3D nace de la necesidad de poseer una visión espacial, anticipar sucesos, predecir hallazgos o analizar fenómenos sin la necesidad de la plataforma física. Con la codificación en el compilador de C++ se pueden presentar los datos en un entorno 3D a través de librerías gráficas como lo es DirectX. EL simulador con ambiente gráfico 3D tiene la ventaja de ser amigable al usuario y fundamental elemento para tareas de diseño, evaluación de algoritmos experimentales en robots manipuladores.

El simulador resuelve el problema de control de posición que consiste en llevar el efector final del extremo del robot de una posición inicial hacia una posición deseada, de acuerdo a la solución de su ecuación de movimiento de Lagrange, se muestran las ecuaciones deducidas y que fueron aplicadas en el programa con el método de Runge Kutta de cuarto orden para su solución [1], para obtener resultados reales con el simulador, y que facilite el análisis de su comportamiento al visualizarlo en el ambiente tridimensional y lograr la definición de trayectorias libre de colisiones y un análisis de sus características dinámicas.

2. Sistema físico

En general un robot manipulador es una estructura mecánica que consiste en una secuencia de cuerpos rígidos (links) conectados por medio de articulaciones (joints), un manipulador se caracteriza por un brazo que asegura movilidad, una muñeca que confiere destreza y un efector final que se encarga de realizar las tareas requeridas por el robot [2].

2.1 Rotradi

Es un robot experimental de *arquitectura abierta*. Este brazo robot compuesto por eslabones rígidos de aluminio cuenta con tres ejes accionados con servomotores, lo que suministra tres grados de libertad, además de contar con la característica de “transmisión directa”, es decir, que los motores se encuentran directamente acoplados a los ejes, evitando con esto la utilización de transmisores o reductores de par o fuerza (figura 1).



Fig. 1 Robot manipulador de transmisión directa de 3 grados de libertad.

El simulador gráfico permite visualizar la ejecución del robot conforme el tiempo transcurre evitando dañar al robot físicamente con colisiones o pares elevados de fuerza e incluso elevando la seguridad del usuario, porque de presentarse una anomalía del sistema solo se necesita reiniciar la sesión del simulador y empezar de nuevo el análisis.

2.2 Ecuación Dinámica del Robot.

En esta sección se procede a determinar la regla matemática que vincula las variables de entrada y salida del sistema. Esta caracterización matemática se describe por medio de ecuaciones diferenciales. El modelo matemático del sistema a controlar se realiza de forma analítica es decir basado en las leyes de la física que rige el comportamiento del sistema. Las ecuaciones dinámicas de un robot manipulador pueden obtenerse a partir de las ecuaciones de movimiento de Newton. El inconveniente que presenta este método es que el análisis se complica notablemente cuando aumenta el número de articulaciones del robot. En estos casos es conveniente emplear las ecuaciones de movimiento de Lagrange.

En este planteamiento, la problemática de control de posición consiste en llevar el extremo del robot a un punto en el espacio independientemente de su posición inicial, generando una trayectoria, es por esa razón que necesitamos el modelo dinámico y uno

de los procedimientos más empleados para la obtención del modelo dinámico del robot manipulador de tres grados de libertad, en forma cerrada, está basado en las ecuaciones de movimiento de Lagrange.

El uso de las ecuaciones de Lagrange para el modelado dinámico del robot manipulador de tres grados de libertad se reduce a cuatro etapas:

1. Cálculo de la energía cinética.
2. Cálculo de la energía potencial.
3. Cálculo del lagrangiano.
4. Desarrollo de las ecuaciones de Lagrange.

Para el estudio y simulación del Robot (Robot de transmisión directa) de 3 grados de libertad se realiza el análisis para la obtención de las ecuaciones dinámicas de Lagrange para la base (τ_1), hombro (τ_2) y codo (τ_3), logrando como resultado respectivamente las siguientes ecuaciones de movimiento.

$$\begin{aligned} \tau_1 = & [I_1 + m_2 l_{c2}^2 \text{sen}^2(q_2) + m_3 [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)]^2 + I_{22} \cos^2(q_2) + I_{y2} \text{sen}^2(q_2) \\ & + I_{z3} \cos^2(q_2 + q_3) + I_{y3} \text{sen}^2(q_2 + q_3)] \ddot{q}_1 \\ & + 2 [m_2 l_{c2}^2 \text{sen}(q_2) \cos(q_2) + (I_{y2} - I_{z2}) \text{sen}(q_2) \cos(q_2) \\ & + (I_{y3} - I_{z3}) \text{sen}(q_2 + q_3) \cos(q_2 + q_3) \\ & + m_3 [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)] [l_2 \cos(q_2) + l_{c3} \cos(q_2 + q_3)]] \dot{q}_1 \dot{q}_2 \\ & + 2 [(I_{y3} - I_{z3}) \cos(q_2 + q_3) \text{sen}(q_2 + q_3) \\ & + m_3 l_{c3} \cos(q_2 + q_3) [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)]] \dot{q}_1 \dot{q}_3 \end{aligned} \quad (1)$$

$$\begin{aligned} \tau_2 = & [I_2 + I_3 + m_3 l_2^2 + m_2 l_{c2}^2 + m_3 l_{c3}^2 + 2m_3 l_2 l_{c3} \cos(q_3)] \ddot{q}_2 \\ & + [I_3 + m_3 l_{c3}^2 + m_3 l_2 l_{c3} \cos(q_3)] \ddot{q}_3 \\ & - [m_2 l_{c2}^2 \text{sen}(q_2) \cos(q_2) \\ & + m_3 [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)] [l_2 \cos(q_2) + l_{c3} \cos(q_2 + q_3)] \\ & + [I_{y2} - I_{z2}] \cos(q_2) \text{sen}(q_2) + [I_{y3} - I_{z3}] \cos(q_2 + q_3) \text{sen}(q_2 + q_3)] \dot{q}_1^2 \\ & - m_3 l_2 l_{c3} \text{sen}(q_3) \dot{q}_3^2 - 2m_3 l_2 l_{c3} \text{sen}(q_3) \dot{q}_2 \dot{q}_3 + [m_2 l_{c2} + m_3 l_2] g \text{sen}(q_2) \\ & + m_3 g l_{c3} \text{sen}(q_2 + q_3) \end{aligned} \quad (2)$$

$$\begin{aligned} \tau_3 = & [I_3 + m_3 l_{c3}^2] \ddot{q}_3 + [I_3 + m_3 l_{c3}^2 + m_3 l_2 l_{c3} \cos(q_3)] \ddot{q}_2 \\ & - [m_3 [l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)] l_{c3} \cos(q_2 + q_3) \\ & + [I_{y3} - I_{z3}] \cos(q_2 + q_3) \text{sen}(q_2 + q_3)] \dot{q}_1^2 + m_3 l_2 l_{c3} \dot{q}_2^2 \text{sen}(q_3) \\ & + m_3 l_{c3} g \text{sen}(q_2 + q_3) \end{aligned} \quad (3)$$

Variable	Descripción
q_1	Posición angular del primer eslabón.
q_2	Posición angular del segundo eslabón.
q_3	Posición angular del tercer eslabón.
l_1	Longitud del primer eslabón.
l_2	Longitud del segundo eslabón.
l_3	Longitud del tercer eslabón.
lc_1	Distancia al centro de masa del primer eslabón.
lc_2	Distancia al centro de masa segundo eslabón.
lc_3	Distancia al centro de masa del tercer eslabón.
m_1	Masa del primer eslabón.
m_2	Masa del segundo eslabón.
m_3	Masa del tercer eslabón.
I_1	Inercia del primer eslabón respecto al centro de masa.
I_2	Inercia del segundo eslabón respecto al centro de masa.
I_3	Inercia del tercer eslabón respecto al centro de masa.
g	Aceleración de la gravedad.

Las expresiones anteriores representan las ecuaciones dinámicas del robot manipulador las cuales pueden expresarse de la forma:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) \quad (4)$$

Donde $\tau \in \mathbb{R}^{3 \times 1}$ llamado vector de fuerzas externas, siendo generalmente los pares y fuerzas aplicados por los accionadores en las articulaciones; $M(q) \in \mathbb{R}^{3 \times 3}$ es una matriz simétrica definida positiva denominada matriz de inercia, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 1}$ es un vector llamado vector de fuerzas centrífugas y de Coriolis, y $g(q) \in \mathbb{R}^{3 \times 1}$ es el vector de fuerzas o pares gravitacionales y la fricción es modelada por un vector $f(\dot{q}) \in \mathbb{R}^n$ dependiente solamente de la velocidad articular \dot{q} , y que combina los denominados fenómenos de fricción viscosa y de fricción de Coulomb:

$$f(\dot{q}) = F_{m1}(\dot{q}) + F_{m2} \text{sgn}(\dot{q}) \quad (5)$$

donde F_{m1} y F_{m2} son matrices de $n \times n$ diagonales definidas positivas. Los elementos de la diagonal de F_{m1} corresponden a los parámetros de fricción viscosa (b) y los de F_{m2} corresponden a los

parámetros de fricción de Coulomb (f_c) en donde se encuentra que:

$$f(\dot{q}) = b\dot{q} + f_c \text{sgn}(\dot{q}) \quad (6)$$

3. Desarrollo de Software

3.1 Simulador con robot modelado en 3D

Los gráficos por computadora siempre han sido considerados un asunto de gran complejidad, principalmente debido a la necesidad de conocimientos matemáticos y de programación, con el surgimiento de las librerías gráficas se pueden desarrollar gráficos interactivos en tres dimensiones de una forma sencilla así como la facilidad de su programación [3].

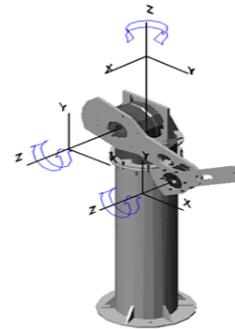


Fig. 2 Robot en el espacio tridimensional.

El modelado en 3D consiste en el desarrollo de una representación visual de un objeto o conjunto de objetos mediante un ordenador para observar el modelo final desde cualquier ángulo (figura 2).

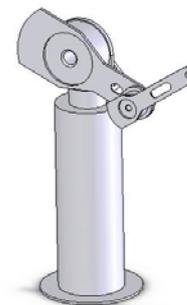


Fig. 3 Robot manipulador diseñado en Solidworks.

La primera necesidad para el desarrollo del simulador es el dibujar el robot manipulador de manera gráfica y en tres dimensiones por lo que se utilizó el software de Solidworks para el diseño del

robot manipulador y posteriormente ser exportado a DirectX. La figura 3 muestra la reproducción del brazo, con la ayuda de Solidworks.

La importancia del modelado en 3D se encuentra en la facilidad que da el programador al usuario para ver el comportamiento dinámico del robot a estudiar antes de llevarlo a la práctica, logrando detectar anomalías o mejoras que serían difíciles de observar con solo datos numéricos o gráficas.

La necesidad que debe cumplir la interfaz es el requerimiento de ser amigable y de fácil manipulación para el usuario, como lo es la manipulación de ventanas en Windows, siendo la mejor parte que la programación se hace en el entorno de Win32 y de esta forma obtener un software de arquitectura abierta sin la necesidad de casarse con una librería dinámica comercial como lo es la MFC de Visual C, dando la ventaja de poder acceder al código para mantenimiento o manipulación del mismo para los que el usuario convenga.

3.2 Aplicación Windows y Librería grafica Directx 3D Ecuación Dinámica del Robot.

Una ventana es la forma interactiva ante el usuario en que se presentan las aplicaciones de Windows. Cada aplicación tiene al menos una ventana principal, y todas las comunicaciones entre usuario y aplicación se canalizan a través de una ventana. Siempre se ha dicho que Windows tiene la ventaja de ser fácil para el usuario y complicado para el desarrollador. El motivo es porque esta manera de programar y codificar es bastante peculiar. Otra de las ventajas que se encuentra al programar bajo la API Win32 es desechar código innecesario que no sabemos el porqué de su existencia, facilitando su mantenimiento y accesibilidad al código para un mejor entendimiento.

El entorno de Microsoft Windows está orientado a los gráficos, lo que significa que proporciona una Interfaz Gráfica de Usuario (GUI) con funciones para la gestión de ventanas y manejo de mensajes [4], siendo responsable de implementar y facilitar a DirectX la facultad de dibujar en una ventana interactiva para el usuario a través de controles para el intercambio de datos. Dado que los elementos de interfaz gráfico se pueden describir y utilizar mucho más fácilmente como objetos, se utiliza la programación orientada a objetos (POO) para la programación del simulador de robots manipuladores.

DirectX surgió en septiembre de 1995 con la aparición del Sistema Operativo Microsoft Windows 95, en un principio fue creada para mejorar los gráficos y sonidos en la plataforma de Microsoft Windows, en la actualidad es un conjunto de interfaces de programación de aplicaciones (API) que permite construir tanto videojuegos como tipos de aplicaciones multimedia de alto rendimiento para un modelado portátil, rápido y elegante cuyos componentes principales son:

DirectX Graphics, que incluye dos librerías:

1. Direct3D para dibujar gráficos en 3D permitiendo optimizar el resultado al mostrarlo en pantalla tomando en cuenta las ventajas del hardware que existe en la computadora, y si no se encuentra hardware lo hace a nivel software.
2. DirectDraw para manipular la memoria de video.
3. DirectInput, es la librería que facilita el acceso a los dispositivos de entrada (teclado, ratón, etc.).

4. Análisis de resultados

En la figura 4 se aprecia una de las funciones necesarias para la interacción con el robot manipulador, basándose en una librería que se crea para manipular variables en la ejecución de la simulación y en la fase de la evaluación del modelo dinámico para el movimiento y posición deseada.

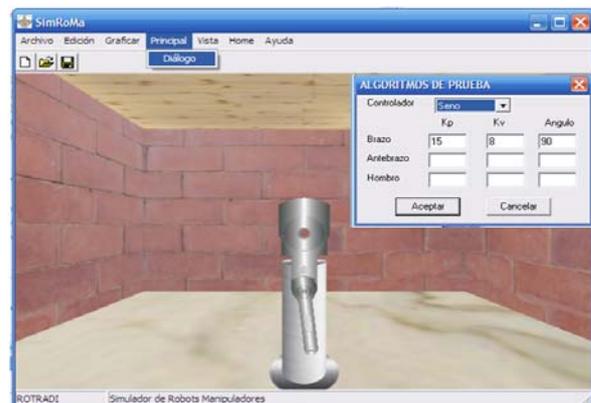


Fig. 4 Ventana de interacción.

La figura 5 ilustra la interfaz gráfica del simulador de robots manipuladores, una vez programado con DirectX y Visual C++.

Se puede agregar que el programa consta de una función que permite el movimiento del punto de visión como si fuera una cámara de video que permite crear un panorama más real del espacio en donde el robot desarrolla su ejecución.

Después de haber obtenido las ecuaciones dinámicas de movimiento de Lagrange, del robot de tres grados de libertad, se codificó el algoritmo en el simulador para comprobar su funcionamiento y evaluación de algoritmos para la ejecución de los movimientos del robot.

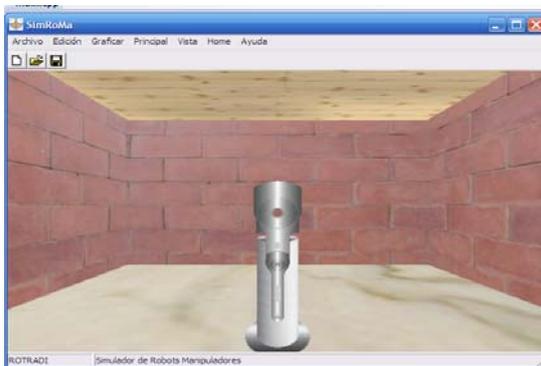


Fig. 5 Interfaz Gráfica del simulador.

En la figura 6 se ilustra el resultado cuando se ejecuta el simulador con la evaluación de un controlador de tipo Tangente hiperbólico y una posición deseada de 90° tanto para el codo como el hombro.

Parámetros	Notación	Eslabón 1	Eslabón 2
Masa	m	24.25	3.87
Longitud	l	0.45	0.45
Longitud centro de masa	l_{cm}	0.9028	0.045
Inercia respecto al centro de masa	I	1.267	0.0925
Coefficiente de fricción viscosa	F_v	2.288	0.175
Coefficiente de fricción de Coulomb	F_c	7.17	1.734
Ganancia proporcional	k_p	55	35
Ganancia derivativa	k_v	40	25

Posición deseada	q_d	90	90
------------------	-------	----	----

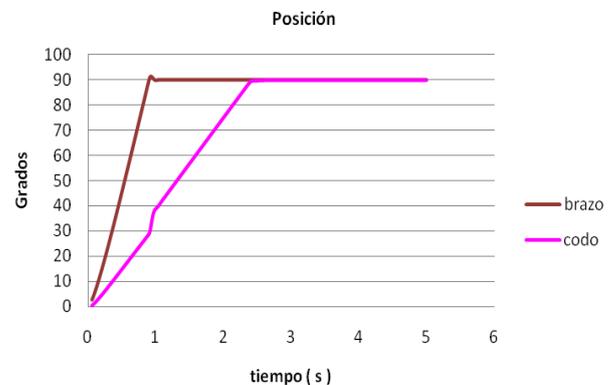


Fig. 6 Ejemplo de simulación del codo-brazo.

Finalmente en el simulador se puede escoger la modalidad de 3 vistas es decir, como si se tuviera 3 cámaras con diferentes ángulos seleccionando diferentes colores y espacios donde tendrá la manipulación el robot, visualizando la creación de la gráfica en tiempo real mientras se ejecuta la trayectoria de movimiento ver figura 7.

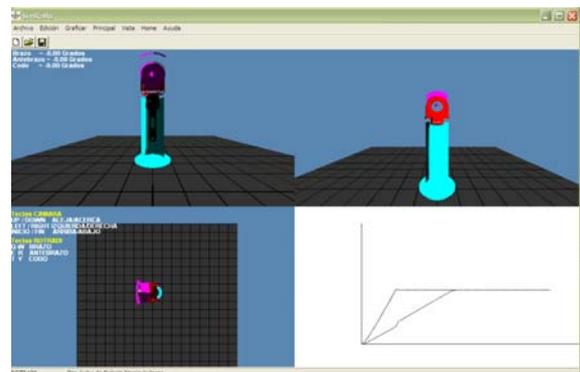


Fig. 7 Simulación con diferentes vistas.

5. Conclusiones

En el presente artículo se ha descrito el desarrollo del software SimRoMa. Simulador de Robots Manipuladores bajo el entorno de Windows realizado el programa con el compilador de Visual C++, lenguaje ampliamente empleado para la programación debido a su gestión automatizada de memoria, además de ser un lenguaje estructurado y orientado a objetos.

Dentro de las aportaciones de la investigación se proporciona el desarrollo del software para la creación del simulador gráfico en 3D, destacando que no solo es una animación sino la simulación de los fenómenos físicos. Con la simulación del comportamiento del robot, se satisface la necesidad de controlar las variables y analizar diversos controladores con el fin de calcular cual debe ser la señal de salida para llegar a la posición deseada.

En la facultad de ciencias de la Electrónica cuenta con 5 prototipos del robot manipulador por ese motivo se necesito en desarrollar un simulador pero con las ventajas de poseer visión espacial y ofrecer un medio de enseñanza para los estudiantes de la misma como material didáctico y de investigación al probar con el simulador nuevos controladores de tipo experimental y así observar el comportamiento dinámico al implementarlos.

El simulador a futuro implementara la manipulación del robot real y con ello se obtendrá seguridad, cuidando la integridad del usuario y del mecanismo físico del robot, anticipando comportamientos no deseados o peligrosos con lo cual se logra un mando de control a distancia.

Este proyecto se puede continuar, para mejorarlo y lograr un sistema lo bastante robusto para aplicación a otros proyectos.

En esta sección se describen los avances del trabajo presentado. Por lo general, también se muestran reflexiones hacia trabajos futuros o aportaciones logradas. Así mismo, también puede mostrarse el estado actual del trabajo que se presenta y los trabajos a realizar a corto plazo.

Referencias

- [1] Teukolsky, P., Vetterling. 2002. *Numerical Recipes in C++*. Cambridge University Press, USA.
- [2] Kelly R., Santibañez V., 2003. *Control de Movimiento de Robots Manipuladores*. Prentice Hall, Madrid.
- [3] Landa, A. 2006 *DirectX Programación de Gráficos 3D*. Users .code. Banfield.
- [4] Schildt H. 1995. Programacion en C y C++ en Windows 95. McGraw-Hill, Mexico.