





OPTIMUS: Software for Efficient Manipulation and Control of Industrial Robots

OPTIMUS: Software para la Manipulación y Control Eficiente de Robots Industriales
Michael Tovar Tovar ^{1*} , Andrés Eduardo Rivera Gómez ² , Pedro Daniel Hile Bustos ³ , Ruthber Rodríguez Serrezuela ⁴ 

Citación: Tovar, F.; Rivera, A.; Hile, P.; Rodríguez, R. OPTIMUS: Software para la Manipulación y Control Eficiente de Robots Industriales. I + T + C Investigación, Tecnología y Ciencia. Vol 1. Num. 17. 2023.



Derechos de autor: © 2023 por los autores. Presentado para posible publicación en acceso abierto bajo los términos y condiciones de la licencia Creative Commons Attribution (CC BY NC SA) (https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es_ES)

Nota del editor: El Sello editorial Unicomfauca se mantiene neutral con respecto a los reclamos que se generen en esta publicación.

- ¹ Corporación Universitaria del Huila – CORHUILA. mtovar-2020b@corhuila.edu.co
² Corporación Universitaria del Huila – CORHUILA. ae.rivera_2019-1@corhuila.edu.co
³ Corporación Universitaria del Huila – CORHUILA. pd.hile_2019-1@corhuila.edu.co
⁴ Corporación Universitaria del Huila – CORHUILA. ruthber.rodriguez@corhuila.edu.co
* Correspondencia: mtovar-2020b@corhuila.edu.co

Resumen: En el artículo se introduce un software de interacción offline que exhibe en su interfaz al robot MR5 desarrollado por la Corporación Universitaria del Huila - CORHUILA. Este robot es un sistema antropomórfico que posee 5 grados de libertad. El propósito principal de este software es posibilitar la interacción con el robot sin la necesidad de su presencia física. Esto se logra mediante la planificación de diversas trayectorias para el robot y la obtención de una simulación de su comportamiento físico. Esta simulación se realiza después de exportar e importar los datos al software de control en tiempo real llamado "Mach3". Durante esta interacción, se observan los movimientos previamente planificados del robot. En la interfaz del software se presenta un seguimiento detallado del comportamiento de cada una de las posiciones articulares del robot, incluyendo información sobre su velocidad y aceleración. La obtención del modelo cinemático, dinámico y sus análisis se presentan con el debido rigor matemático.

Palabras clave: Software; Modelo cinemático; Modelo dinámico; Robot MR5; Planificación de trayectorias

Abstract: The article introduces an offline interaction software that exhibits in its interface the MR5 robot developed by the Corporación Universitaria del Huila - CORHUILA. This robot is an anthropomorphic system with 5 degrees of freedom. The main purpose of this software is to enable interaction with the robot without the need of its physical presence. This is achieved by planning different trajectories for the robot and obtaining a simulation of its physical behavior. This simulation is performed after exporting and importing the data to the real-time control software called "Mach3". During this interaction, the previously planned movements of the robot are observed. A detailed tracking of the behavior of each of the robot's joint positions, including information about its velocity and acceleration, is presented in the software interface. The obtaining of the kinematic and dynamic model and its analysis are presented with due mathematical rigor.

Keywords: Software; Kinematic model; Dynamic model; MR5 Robot; Path planning

DOI: 10.57173/ritc.v1n17a5

1. Introducción

En el ámbito de la robótica, la resolución de problemas de alta complejidad ha sido un objetivo fundamental con el propósito de dotar a los robots articulados de la capacidad de alcanzar con precisión posiciones predefinidas dentro de su espacio de trabajo [1]. Esta búsqueda de precisión y eficiencia ha sido una prioridad en la formación de estudiantes en este campo.

El dinamismo y la interacción práctica son esenciales para la comprensión profunda de las teorías robóticas, permitiendo a los estudiantes validez de la aplicabilidad y precisión de los conceptos aprendidos en el aula [2]. Sin embargo, un desafío común radica en la falta de recursos para adquirir robots físicos costosos en muchas instituciones educativas.

En respuesta a esta limitación, se ha desarrollado una solución innovadora denominada "OPTIMUS". Este software ha sido diseñado para administrar trayectorias previamente definidas y controlar el robot MR5, presente en las instalaciones de la Corporación Universitaria del Huila – CORHUILA, ubicada en Neiva – Huila, Sede Prado Alto. Lo excepcional de esta solución es su capacidad para permitir la manipulación remota del robot, sin requerir la presencia física del estudiante junto al equipo.

La accesibilidad que ofrece "OPTIMUS" revoluciona la experiencia de aprendizaje en robótica al permitir a los estudiantes visualizar las coordenadas que definen la ubicación espacial del robot. Además, posibilita la planificación de trayectorias y la implementación de rutinas para proyectos específicos [3]. Una vez que se logran los resultados deseados, el software genera un archivo en formato gcode, que contiene todas las coordenadas espaciales necesarias. Este archivo se importa posteriormente al software "Mach3", que opera en tiempo real y controla el robot MR5, ejecutando cada una de las trayectorias previamente planificadas en "OPTIMUS".

Esta solución no solo hace que la robótica sea más accesible para los estudiantes, sino que también les brinda una experiencia de aprendizaje práctica y enriquecedora. Les permite explorar y aplicar sus conocimientos en un campo emocionante sin la necesidad de acceso directo a un costoso robot físico. En este artículo, se explica a detalle la implementación y beneficios de "Optimus" en el contexto de la enseñanza y la investigación en robótica.

2. Materiales y métodos

Los materiales implementados son los siguientes:



Figura 1. Robot MR5

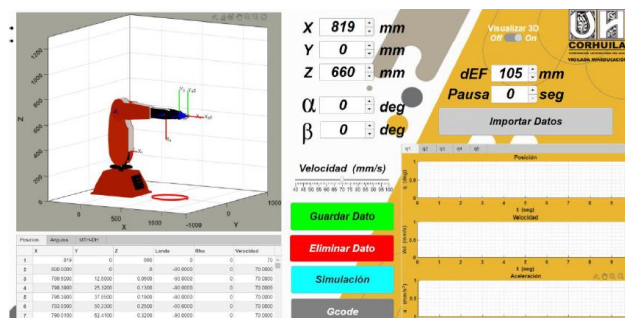


Figura 2. Software OPTIMUS

En este artículo, se presenta una detallada exposición de la metodología empleada para la implementación y evaluación del software "OPTIMUS" en el contexto de la enseñanza y la investigación en robótica. La metodología se ha diseñado meticulosamente para abordar aspectos claves.

Describiendo la configuración del entorno experimental, que incluye el robot MR5 y el software "OPTIMUS" Se detallan las especificaciones técnicas del robot y el software utilizado, así como la infraestructura disponible en la Corporación Universitaria del Huila – CORHUILA para llevar a cabo las mencionadas investigaciones.



Figura 3. Robot MR5 digitalizado

Uno de los puntos importantes fue digitalizar a escala real el robot MR5 con el fin de obtener resultados reales, posibles colisiones de este, interacción dinámica simulada en el software. Para interactuar con el robot es necesario realizar un análisis cinemático, encontrando así la posición angular que toma cada una de sus articulaciones y llevando al robot a una posición espacial en el espacio de trabajo permitido.

Análisis cinemático

La determinación precisa de la posición espacial del efector final de un robot se basa en la coordinación secuencial de todos los actuadores que lo conforman. La cinemática directa es la disciplina encargada de calcular la posición y orientación del efector final a partir de las mediciones angulares o distancias de sus articulaciones, utilizando ecuaciones que describen la geometría y estructura del robot. Por otro lado, la cinemática inversa aborda el problema contrario, es decir, calcular los valores requeridos para las articulaciones del robot a fin de lograr una posición y orientación específicas del efector final [4].

Cinemática directa

Se emplea la metodología de Denavit-Hartenberg para determinar los parámetros que describen los sistemas de coordenadas de cada eslabón de manera precisa. Estos parámetros se representan mediante matrices de transformación homogénea, que son fundamentales en la cinemática de los robots.

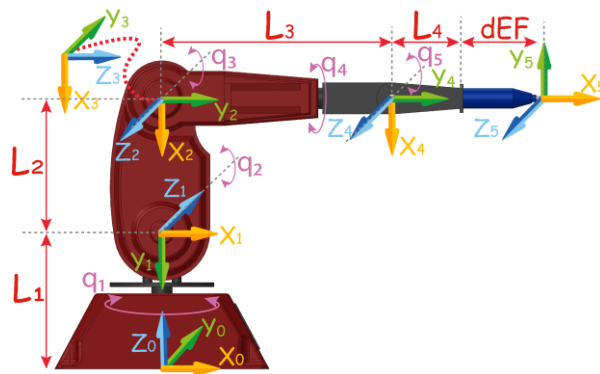


Figura 4. Dimensiones y disposiciones de los ejes de coordenadas para cada articulación.

Articulaciones	θ	d	a	α
1₀₋₁	q_1	L_1	0	$-\frac{\pi}{2}$
2₁₋₂	$q_2 + \frac{\pi}{2}$	0	$-L_2$	π
3₂₋₃	q_3	0	0	$-\frac{\pi}{2}$
4₃₋₄	q_4	L_3	0	$\frac{\pi}{2}$
5₄₋₅	q_5	0	$L_4 + dFE$	0

Tabla 1. Cinemática Directa con Denavit-Hartenberg

Cada fila de la tabla representa una matriz **i** con respecto al sistema **i-1**. Multiplicando cada una de las matrices obtenidas se obtiene la matriz de transformación homogénea donde se encuentran los datos de orientación y traslación del efector final con respecto a su sistema de origen.

La matriz de transformación homogénea tiene forma

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5$$

$$T = \begin{bmatrix} n_x & O_x & a_x & P_{5x} \\ n_y & O_y & a_y & P_{5y} \\ n_z & O_z & a_z & P_{5z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde “**n o a**” es la orientación del sistema **i** con respecto al sistema anterior **i-1**, creando una matriz de 3x3 llamada matriz de rotación y “**p**” representa la ubicación espacial de **i** con respecto a **i-1**, creando un vector de 3x1.

Cinemática inversa

Para realizar los cálculos relacionados con la cinemática inversa del robot, es imperativo emplear un enfoque geométrico que establezca relaciones precisas entre los movimientos de sus articulaciones. Esto conlleva una estratégica descomposición cinemática del robot, posicionando de manera específica en una configuración geométrica que simplifique el cálculo de las variables articulares correspondientes.

Para iniciar este proceso de cálculo geométrico, es de suma importancia comprender que la última articulación del robot mantiene una orientación constante,

independientemente de las variaciones en la posición del efector final del robot. Para preservar esta orientación constante, se vuelve esencial trabajar con el punto de la muñeca del robot (P_m), cuyas coordenadas pueden obtenerse siguiendo el procedimiento que se describe a continuación [5](Pag 4-5).

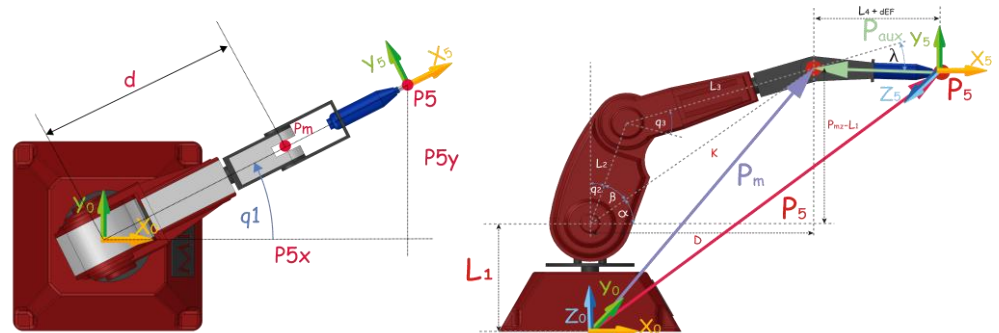


Figura 5. Robot MR5 – Cálculos geométricos

Para determinar la posición de la primera articulación del robot, q_1 , adoptamos una vista superior del robot. Una vez que hemos recibido las coordenadas del efector final P_5 para su orientación, podemos calcular q_1 con las coordenadas P_{5x}, P_{5y} de la siguiente manera:

$$q_1 = \text{atan2}(P_{4y}, P_{4x})$$

Una vez que hemos obtenido el valor de q_1 , es necesario indicar la orientación que deseamos que opte el robot en su efector final, para realizar este paso es necesario conocer las matrices de rotación, ya que para cada eje es una matriz rotación diferente, observando la figura 5 – vista lateral, se observa como su eje de origen al rotarlo, pasando primero por el sistema 4, hasta llegar al sistema 5 de su efector final, una vez orientado como su efector final se debe de ingresar la orientación deseada λ y φ :

$${}^0A_{d5} = \text{Rot}_z(q_1) * \text{Rot}_y\left(\frac{\pi}{2}\right) * \text{Rot}_x\left(\frac{\pi}{2}\right) * \text{Rot}_y(\varphi) * \text{Rot}_z\left(\frac{\pi}{2}\right) * \text{Rot}_z(\lambda)$$

La matriz deseada ${}^0A_{d5}$ tiene un tamaño de 3x3, donde extraemos los datos de vector columna X_5 , observando la figura 5 – vista lateral, el eje X_5 se encuentra colineal con el eslabón L_4 :

$$X_5 = {}^0A_{d5}(1:3,1)$$

Observando la figura 5 – vista lateral, para calcular las coordenadas del punto P_m , es necesario determinar el vector P_{aux} , trasladando la orientación dada en el efector final, al punto de la muñeca P_m

$$P_{aux} = X_5 * -L_4$$

Una vez que se tiene el vector P_{aux} , se suma al vector P_4 según lo ilustrado en la figura 5 – vista lateral, dando la posición de P_m :

$$P_m = P_4 + P_{aux}$$

Los siguientes cálculos se determinan de acuerdo con P_m . Para calcular la distancia D desde el origen hasta el punto P_m (figura 5 – superior), se aplica el teorema de Pitágoras considerando las coordenadas de P_m :

$$D = \sqrt{P_{mx}^2 + P_{my}^2}$$

Cuando las coordenadas varían, hacen que el brazo se estire y se encoja, cambiando su distancia D , desde una vista superior se observa como D , siempre se puede encontrar con las coordenadas en X y Y del punto de la muñeca.

En la *figura 5* – vista lateral, se presenta un triángulo compuesto por tres elementos (K , $P_{mz} - L_1$ y D). Hasta ahora, se tiene información sobre dos de estos elementos, lo que lleva a calcular la distancia K de la siguiente manera:

$$K = \sqrt{D^2 + (P_{mz} + L_1)^2}$$

La variable CODO se utiliza para indicar al robot si se debe posicionar con el codo hacia arriba o hacia abajo, donde:

$$CODO = 1 \quad \text{codo arriba}$$

$$CODO = -1 \quad \text{codo abajo}$$

Para calcular los valores de q_2 , primero, es necesario determinar los valores de α y β , que son cruciales para obtener C_β y S_β , que representan los catetos opuestos y adyacentes en un triángulo, ayudando a encontrar β :

$$C_\beta = \frac{L_3^2 - L_2^2 - K^2}{-2L_2K}$$

$$S_\beta = CODO \sqrt{1 - C_\beta^2}$$

$$\beta = \text{atan2}(S_\beta, C_\beta)$$

Los valores de P_{mz} y L_1 se obtienen desde el principio, junto con D . Al observar la *figura 5* – vista lateral, el ángulo α se calcula de la siguiente manera:

$$\alpha = \text{atan2}(P_{mz} - L_1, D)$$

Luego, se puede obtener el valor de q_2 con la siguiente expresión:

$$q_2 = \frac{\pi}{2} - (\beta + \alpha)$$

Estos pasos se repiten para calcular q_3 , considerando la configuración del triángulo como se muestra en la *figura 5* – vista lateral.

$$C_\delta = \frac{K^2 - L_3^2 - L_2^2}{-2L_2L_3}$$

$$S_\delta = CODO \sqrt{1 - C_\delta^2}$$

$$\delta = \text{atan2}(S_\delta, C_\delta)$$

$$q_3 = \delta - \frac{\pi}{2}$$

Para determinar los valores de q_4 y q_5 , se implementó el método de ángulos de Euler y ya obtenidos las primeras articulaciones (q_1, q_2, q_3) se integran a la tabla 1, obteniendo la matriz transformación homogénea 0A_3 , donde solo utilizamos su parte rotativa “noa”

$${}^0A_3 = {}^0A_1 {}^1A_2 {}^2A_3$$

En la interfaz gráfica del software los ángulos α y β , representan los ángulos λ y φ .

Previamente se obtuvo la matriz de rotación deseada ${}^0A_{d5}$, y junto a la matriz 0A_3 obtenemos la matriz 3A_5 de la siguiente manera:

$${}^3A_5 = ({}^0A_3)^T * {}^0A_{d5}$$

A partir de la matriz obtenida, se obtienen los valores de q_4 y q_5 con la siguiente expresión:

$$q_4 = \text{atan2}({}^3A_5(1, 3), -{}^3A_5(2, 3))$$

$$q_5 = \text{atan2}\left({}^3A_5(3, 1), {}^3A_5(2, 3) - \frac{\pi}{2}\right)$$

Construcción del Software

Para la elaboración del software "OPTIMUS", se utilizó la herramienta de MATLAB llamada Appdesigner. Esta aplicación simplifica la creación de programas y aplicaciones, capitalizando las ventajas que MATLAB brinda en cuanto a cálculos matemáticos, representaciones gráficas y otras funcionalidades. Al finalizar el proceso de desarrollo, es posible generar un archivo ejecutable a partir del software.



Figura 6. Interfaz inicial del software.

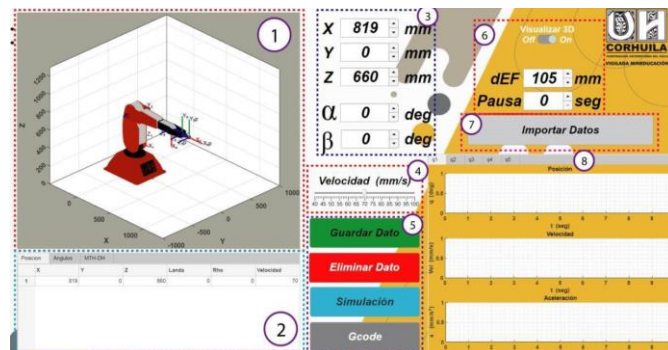


Figura 7. Interfaz interactiva del software.

1. El entorno de simulación muestra el robot en 3D y permite cambiar su apariencia a una forma más simple. También posibilita ajustar la vista según se desee.

2. La interfaz tiene una tabla con tres pestañas: "Posición" muestra coordenadas y orientaciones, "Ángulos" calcula ángulos y "MTH-DH" representa la posición del efector final.
3. Al iniciar, se muestra la posición inicial. Para ver otra, basta con modificar la coordenada deseada y guardarla.
4. La velocidad del robot en la simulación influye en el tiempo total de la rutina. La simulación no refleja necesariamente la velocidad real del robot físico.
5. Botones clave: "Guardar Dato" registra coordenadas, "Eliminar Dato" borra la última, "Simulación" muestra el movimiento, "Gcode" exporta datos.
6. Se puede alternar entre visualización 3D y una representación más simple del robot. También se ajusta la distancia del efector final y las pausas entre movimientos.
7. "Importar Datos" es un código adaptable para generar trayectorias que se ajusten al robot en mm como unidad de distancia.

Importación de datos

En esta sección se integra un ejemplo en el entorno de MATLAB, aclarando que el objetivo es como se deben de ingresar los datos, se puede utilizar cualquier entorno mientras se tenga claro que la primera columna estarían los valores en "X", la siguiente columna de datos estarían los "Y" y por ultimo los datos de "Z"

Creando vectores para las coordenadas en X, Y y Z, el objetivo es obtener los datos de una figura o trayectoria deseada y poder importar sus datos al entorno de trabajo del robot MR5, dentro de la interfaz gráfica del software OPTIMUS.

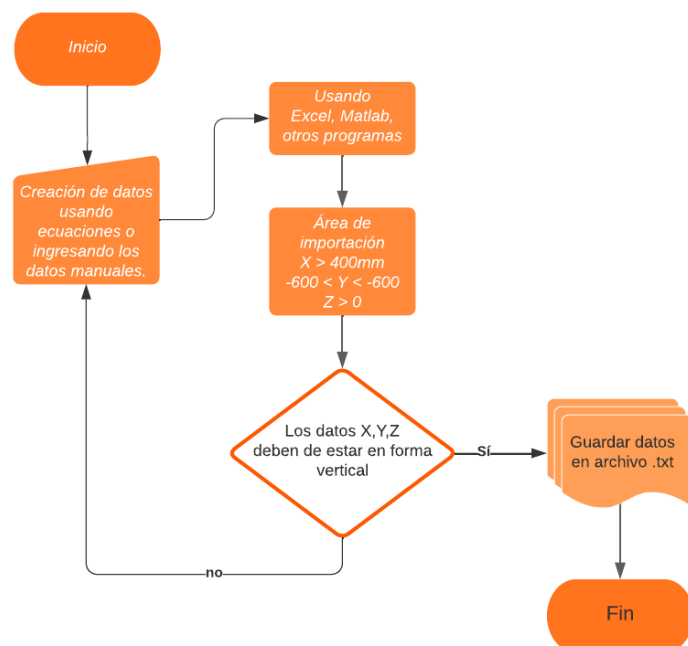


Figura 8. Adquisición de datos de una trayectoria deseada por medio de una ecuación.

En la figura 9, el robot sigue una trayectoria con su efector final en aplicaciones industriales como soldadura y pintura. Los puntos de la trayectoria son generados por código y se muestran en una tabla de posición, además de ser representados en el espacio de trabajo del robot MR5 en rojo.

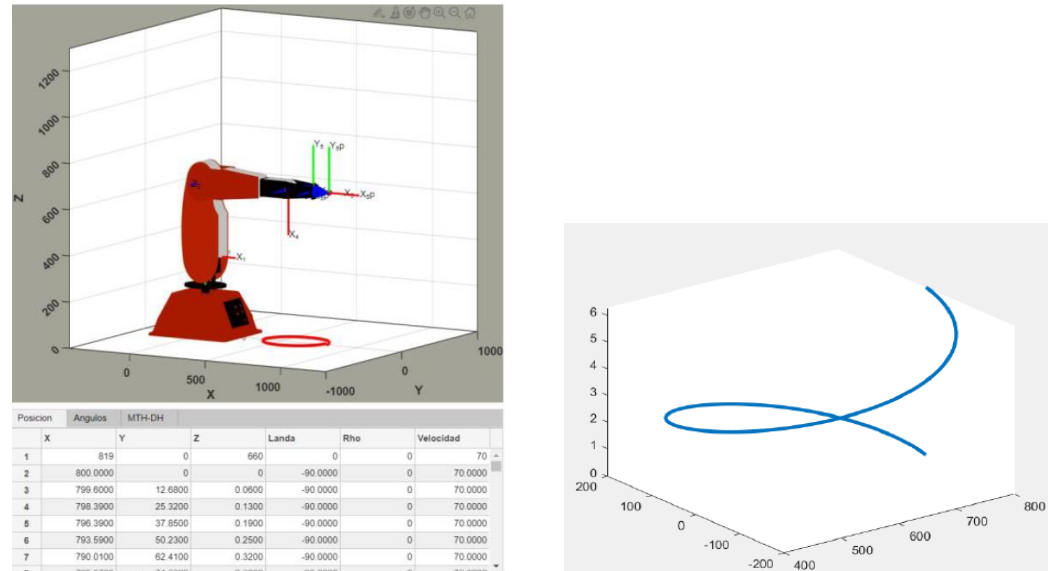


Figura 9. Puntos de trayectoria generados y representados en el espacio de trabajo del robot MR5

Interpoladores Spline Cúbico

Un spline cúbico es una técnica matemática ampliamente utilizada en robótica y en muchas otras disciplinas para describir y controlar trayectorias suaves. Utilizado principalmente como planificador de movimiento y control de trayectorias [6].

El robot debe moverse desde una posición inicial a una posición final de manera suave y continua, pasando a través de la función spline cúbico para generar trayectorias suaves que conectan punto a punto, sabiendo cual es el inicial y cuál es el final. A continuación, se observa las ecuaciones utilizadas para realizar el spline cúbico, indicando inicialmente el valor T_f siendo su tiempo de trayectoria final, con “n” cantidad de segmentos de trayectoria.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ T_f^3 & T_f^2 & T_f & 1 \\ 3T_f^2 & 2T_f & 1 & 0 \end{bmatrix}$$

Debe existir un B para cada coordenada y multiplicarla por A.

$$B = [P_{i5xyz} \quad 0 \quad P_{f5xyz} \quad 0]$$

Obteniendo un vector M_x , M_y y M_z , siendo t un vector tiempo que se segmenta n cantidad de veces

$$M_{xyz} = [A * B]^{-1}$$

$$t = \text{linspace}(0, T_f, n)$$

Dando como resultado el vector X, vector Y y vector Z, los cuales se grafican y se obtienen la posición con la siguiente ecuación:

$$xyz = (M_{xyz} * t^3) + (M_{xyz} * t^2) + (M_{xyz} * t) + M_{xyz}$$

Para obtener la velocidad y aceleración se deriva la ecuación de posición

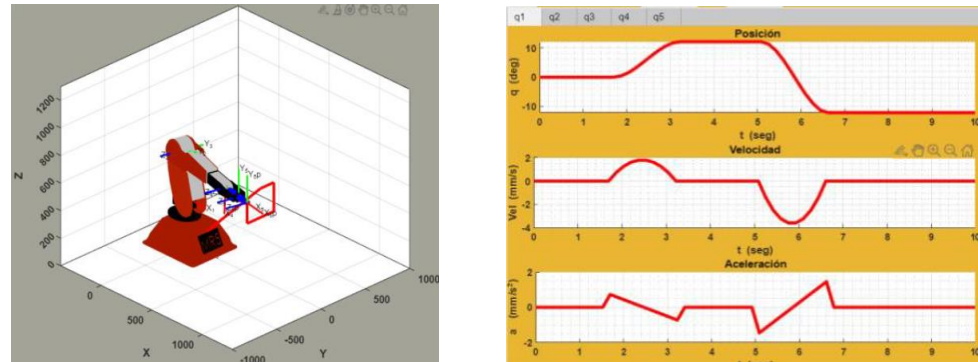


Figura 10. Resultado de trayectoria con spline cúbico, visualizando su posición, velocidad y aceleración para la articulación q_1

Análisis dinámico del robot MR5

El desplazamiento efectivo de las articulaciones de un robot es determinado por su dinámica, la cual considera la masa de cada eslabón y junta, la velocidad y aceleración de las articulaciones, y las fuerzas que actúan sobre cada junta. Para obtener el modelo dinámico del manipulador MR5, se empleó el método recursivo de Newton – Euler. Utilizado este método, se lograron calcular los torques en función de las porciones, velocidades y aceleraciones articulares [7]. El modelo dinámico del robot MR5 consiste en un conjunto de ecuaciones diferentes expresadas como sigue:

$$A(q) \cdot \ddot{q} + B(q) \cdot [\dot{q} \cdot \dot{q}] + C(q) \cdot [\dot{q}^2] + g(q) = \Gamma$$

Donde:

q es el n-vector de coordenadas espaciales generalizado que describe la posición del manipulador.

\dot{q} es el n-vector de velocidades angulares.

\ddot{q} es el n-vector de aceleraciones angulares

$A(q)$ es la matriz simétrica de inercia en el espacio articular de tamaño $n \times n$

$B(q)$ es la matriz de efectos de Coriolis de tamaño $n \times n (n-1) / 2$

$C(q)$ es la matriz de torques centrípetos de tamaño $n \times n$

$g(q)$ es el n-vector de torques gravitatorios

3. Resultados

En este capítulo, se presentan los resultados obtenidos a través de la investigación y experimentación llevada a cabo en el contexto del control de un robot.

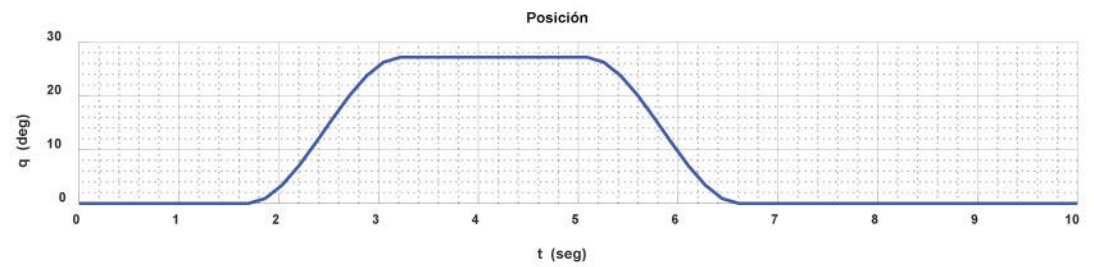


Figura 11. Posición de trayectoria de la articulación q_5

La *figura 11*, ilustra el movimiento articular de la última articulación del robot. Esta articulación se mueve por espacio de 4 segundos a 29 grados y luego se regresa a la posición de 0 grados.

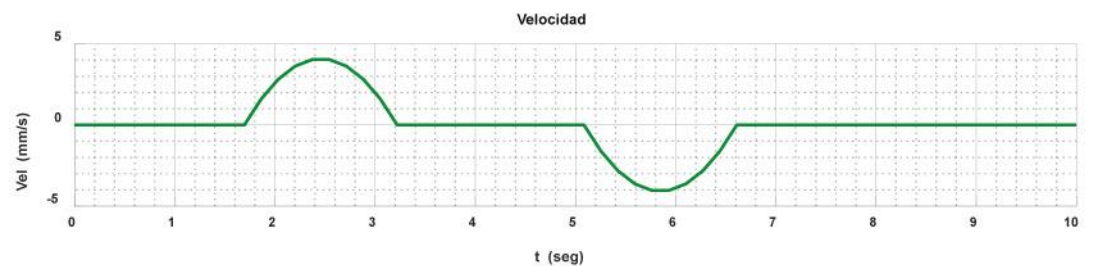


Figura 12. Velocidad de trayectoria de la articulación q_5

Los resultados de la velocidad de esa articulación se ilustran en la *figura 12*. La figura muestra una velocidad positiva de 1.5 mm/s en el primer segundo y a los siete segundos, respectivamente. De igual forma, presenta una velocidad negativa de -3 mm/s.

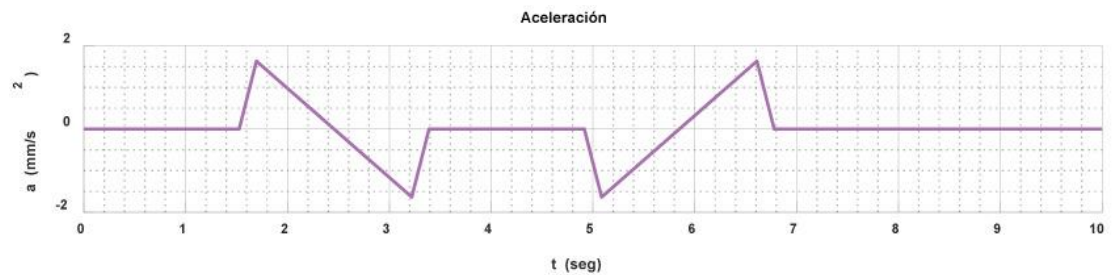


Figura 13. Aceleración de trayectoria de la articulación q_5

La *figura 13* muestra la presencia de dos aceleraciones una negativa en el intervalo de tiempo de los dos y tres segundos con un valor de -1.875 mm/s². Así mismo, una aceleración positiva en el intervalo del sexto y séptimo segundo con un valor de 1.875 mm/s². Las pendientes de ambas aceleraciones son lineales.

Este tipo de comportamiento demuestran que el control CTC no genera cambios brusco en las velocidades articulares del robot MR5



Figura 14. Interacción y planificación con software OPTIMUS y Robot MR5

4. Discusión

En este estudio, se presenta un software de interacción offline que permite controlar y supervisar el robot MR5 de manera remota, sin la necesidad de su presencia física. Los resultados obtenidos a través de la simulación de comportamiento físico son prometedores y tienen varias aplicaciones importantes en el campo de la robótica y la interacción humano-robot.

Los resultados de las pruebas muestran una concordancia significativa entre las trayectorias planificadas y la ejecución simulada por el robot MR5. La información detallada sobre la velocidad y aceleración de la articulación también se alinea adecuadamente con las expectativas del proyecto. Esto sugiere que la nuestra simulación es una representación precisa del comportamiento real del robot MR5 en diversas tareas.

Este enfoque de interacción offline tiene el potencial de tener un impacto significativo en una variedad de aplicaciones. Por ejemplo, en la industria manufacturera, podría utilizarse para el control y supervisión de robots en entornos peligrosos o de difícil acceso, mejorando la seguridad de los trabajadores. Además, en la educación y la investigación, facilitaría la experimentación y la enseñanza de conceptos de robótica de manera remota.

Este enfoque se diferencia de otros métodos de interacción humano – robot a distancia en su capacidad para simular el comportamiento físico del robot con un alto grado de precisión. Aunque existen soluciones similares, como el telecontrol de robots, la presente técnica ofrece ventajas adicionales en términos de capacidad de planificación y simulación detallada.

5. Conclusiones

- Avance en la Interacción Remota Robótica: El artículo presenta un logro significativo en la investigación robótica al introducir el software "OPTIMUS" para la interacción offline con el robot MR5. Este avance impulsa la capacidad de control y supervisión de robots.
- Precisión en la Simulación Dinámica: El enfoque se destaca por su capacidad para generar simulaciones altamente precisas del comportamiento físico del robot MR5. Los resultados experimentales confirman la capacidad de esta tecnología para replicar con precisión las trayectorias y el rendimiento de movimiento planificado.
- Diversificación de Aplicaciones Industriales: La tecnología "OPTIMUS" abre un abanico de posibilidades en aplicaciones industriales, como la automatización de procesos de fabricación, soldadura y pintura, donde la simulación precisa del movimiento del robot es esencial para garantizar la calidad del trabajo.
- Investigación y Educación: La herramienta "OPTIMUS" no solo tiene aplicaciones industriales, sino que también se presenta como una valiosa

herramienta en la investigación y la enseñanza de la robótica. Permite experimentación remota y el estudio de conceptos robóticos de manera efectiva.

- Integración Metodológica y Tecnológica: El artículo destaca la meticulosa integración de la metodología de desarrollo del software "OPTIMUS" con la infraestructura disponible en la Corporación Universitaria del Huila – CORHUILA, lo que respalda la viabilidad y el rigor técnico de la solución presentada

Referencias

1. S. Keshavdas, T. D. Murphey, and A. M. Okamura, "Robotic Teleoperation with Haptic Feedback: A Comprehensive Review," in *IEEE Transactions on Haptics*, vol. 13, no. 3, pp. 409-426, July-Sept. 2020.
2. S. M. A. A. Zaidi, A. N. Al-Omari, and A. Al-Smadi, "Teaching Robotics: A Review of Pedagogies, Assessment, and Curriculum," in *IEEE Access*, vol. 8, pp. 130965-130977, 2020.
3. F. P. Ferreira, R. C. Marques, and F. M. Martins, "A Low-Cost Educational Robotic Arm with an Open-Source Software Interface," in *IEEE Transactions on Education*, vol. 63, no. 2, pp. 215-223, May 2020.
4. Zaplana I, Arnau C, J, Basanez L "Análisis Cinemático de robots Manipuladores Redundantes: Aplicación a los Robots Kuka LWR 4+ y ABB Yumi", *RIAI: Revista Iberoamericana de Automática e Informática industrial* 15(2018) 192-202.
5. M. J. Mirzaei and F. C. Park, "A new geometric calibration method for Stewart platforms," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1534-1539. doi: 10.1109/IROS.2018.8594217.
6. A. K. Pandey and P. K. Biswas, "A novel approach to plan smooth trajectory using cubic spline for robotic arm," 2021 International Conference on Computing, Communication and Signal Processing (ICCCSP), Chennai, India, 2021, pp. 1-5. Doi: 10.1109/ICCCSP51409.2021.9471159.
7. J. Wang, X. Ma and Y. Liu, "Dynamic Modeling and Simulation of a 3-DOF Parallel Robot Based on the Newton-Euler Method," 2021 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 2021, pp. 1110-1115. doi: 10.1109/ICMA52212.2021.9558674.