

Simulación y control de manipuladores robóticos mediante realidad virtual en la web

Simulation and control of robotic manipulators using virtual reality on the web

Michel A. Cruz Martínez ^a, Alejandro Rodríguez Molina ^b, Oscar A. Gómez Vargas ^c, Allan B. Rueda Gutiérrez ^d, José Solís Romero ^e, Miguel Á. Paredes Rueda ^f

Abstract:

Robotic manipulators are electromechanical systems used in various applications that require movements similar to those of a human limb, but with greater agility, strength, and precision. To execute any task with these systems, it is essential to have a controller that efficiently manages and coordinates their movements. Tuning the controller parameters is crucial to optimize the execution of a specific task, and this tuning is usually done through controlled simulations before implementing it in the real system. Currently, these simulations are carried out in specialized environments that require proprietary software or expensive hardware, which limits their accessibility and scalability. Furthermore, these platforms do not always offer an immersive and interactive experience for real-time visualization and control. This work proposes an approach to develop web-based virtual reality simulations, which allow tuning robotic manipulator controllers. The developed virtual simulations seek to overcome the limitations of current simulation technologies. The approach has been successfully tested in the virtual simulation of a two-degree-of-freedom manipulator robot.

Keywords:

Robotic manipulator, controller tuning, virtual reality, simulation, web technology.

Resumen:

Los manipuladores robóticos son sistemas electromecánicos utilizados en diversas aplicaciones que requieren movimientos similares a los de una extremidad humana, pero con mayor agilidad, fuerza y precisión. Para ejecutar cualquier tarea con estos sistemas, es fundamental contar con un controlador que gestione y coordine eficientemente sus movimientos. El ajuste de los parámetros del controlador es crucial para optimizar la ejecución de una tarea específica, y este ajuste suele realizarse mediante simulaciones controladas antes de implementarlo en el sistema real. Actualmente, estas simulaciones se llevan a cabo en entornos especializados que requieren software propietario o hardware costoso, lo que limita su accesibilidad y escalabilidad. Además, estas plataformas no siempre ofrecen una experiencia inmersiva e interactiva para la visualización y el control en tiempo real. Este trabajo propone un enfoque para desarrollar simulaciones de realidad virtual basadas en la web, que permiten ajustar los controladores de manipuladores robóticos. Las simulaciones virtuales desarrolladas buscan superar las limitaciones de las tecnologías de simulación actuales. El enfoque se ha probado con éxito en la simulación virtual de un robot manipulador de dos grados de libertad.

Palabras Clave:

^a Tecnológico Nacional de México | IT de Tlalnepantla | Estado de México | México, <https://orcid.org/0009-0007-4924-2261>, email: michelalejandrocruzmartinez@gmail.com

^b Universidad Autónoma de la Ciudad de México | Colegio de Ciencia y Tecnología | Ciudad de México | México, <https://orcid.org/0000-0003-3310-6605>, Email: alejandro.rodriguez.molina@uacm.edu.mx

^c Autor de Correspondencia, Tecnológico Nacional de México | IT de Tlalnepantla | Estado de México | México, <https://orcid.org/0000-0003-3310-6605>, Email: oscar.gv@tlalnepantla.tecnm.mx

^d Tecnológico Nacional de México | IT de Tlalnepantla | Estado de México | México, <https://orcid.org/0009-0002-1887-8148>, Email: allan.rg@tlalnepantla.tecnm.mx

^e Tecnológico Nacional de México | IT de Tlalnepantla | Estado de México | México, <https://orcid.org/0000-0002-2483-083X>, Email: jose.sr@tlalnepantla.tecnm.mx

^f Tecnológico Nacional de México | IT de Tlalnepantla | Estado de México | México, Email: <https://orcid.org/0000-0002-5011-1417>, email: miguel.pr@tlalnepantla.tecnm.mx

Introducción

Los manipuladores robóticos son sistemas dinámicos que combinan elementos eléctricos y mecánicos para realizar movimientos similares a los de las extremidades humanas, en particular del brazo (Ceccarelli, 2022). Estos sistemas están formados por cadenas de eslabones, abiertas o cerradas, conectados mediante uniones que permiten diferentes movimientos, ya sea inducidos o actuados, y coordinados para ejecutar una tarea específica mediante un efector final (Spong et al., 2020). Gracias a su capacidad para llevar a cabo tareas que requieren movimientos complejos con mayor agilidad, fuerza y precisión que los humanos, los manipuladores robóticos son ampliamente utilizados en diversas áreas, desde la industria (Gaber et al., 2021) hasta la salud (Jayaswal et al., 2021).

Para que los manipuladores robóticos puedan realizar sus tareas con un rendimiento adecuado, deben ser controlados por un sistema de control. El controlador es el dispositivo encargado de gestionar y coordinar los movimientos de las uniones actuadas del manipulador para completar una tarea específica (Lewis et al., 2023). Actualmente, existen muchos tipos de controladores capaces de gestionar el comportamiento de manipuladores robóticos con distintos niveles de complejidad (Spong, 2022). Independientemente del tipo de controlador, todos ellos tienen parámetros ajustables que influyen en el rendimiento del robot durante la ejecución de una tarea. Si estos parámetros se ajustan correctamente, la tarea puede completarse de manera eficiente. De lo contrario, la tarea podría no completarse, lo que podría generar comportamientos no deseados en el manipulador, con riesgos y costos considerables.

El proceso de ajuste de controladores se conoce como sintonización (Rodríguez et al., 2020), y típicamente consiste en un proceso iterativo en el que se prueba el rendimiento del controlador de un sistema dinámico utilizando distintas combinaciones de parámetros, hasta encontrar una que cumpla con los criterios de rendimiento establecidos para una aplicación específica. Debido a los riesgos y costos asociados con la experimentación en el sistema dinámico real, es preferible realizar este proceso mediante simulaciones.

La simulación de manipuladores implica el uso de un modelo, ya sea computacional o matemático, que describe los aspectos más importantes del

comportamiento del robot real. Este modelo permite estudiar su funcionamiento y predecir su respuesta bajo diversas condiciones, como las distintas combinaciones de parámetros del controlador, sin la necesidad de experimentar directamente con el sistema físico. El objetivo es tomar decisiones que reduzcan costos y riesgos, y maximicen los beneficios en tareas como el diseño, optimización, planificación y control de estos sistemas (Choi et al., 2021).

Actualmente, las simulaciones de robots se llevan a cabo en plataformas especializadas que requieren una inversión considerable en software propietario y hardware específico. Plataformas como MATLAB/Simulink y ROS (Robot Operating System) han sido ampliamente utilizadas para estas tareas (Cai et al., 2021; Pratheep et al., 2021; Maldonado-Valencia et al., 2022; Marian et al., 2020), pero su alto costo o la necesidad de equipos con altas prestaciones limitan su accesibilidad, especialmente en entornos educativos y de investigación con recursos limitados. Además, estas plataformas generalmente carecen de características avanzadas de visualización e interactividad, lo que dificulta la creación de experiencias inmersivas que facilitarían el ajuste de controladores.

La llegada de tecnologías emergentes como la realidad virtual (VR), la realidad aumentada (AR) y la realidad extendida (XR) ha abierto nuevas posibilidades para mejorar la experiencia de simulación robótica (Makhataeva et al., 2020). Estas tecnologías permiten a los usuarios visualizar de manera inmersiva el comportamiento de un sistema robótico en un entorno 3D, brindando una percepción más realista y precisa de cómo se comportaría el robot en el mundo físico (Lin et al., 2020; Alonso et al., 2022; Ostanin et al., 2020). Sin embargo, el uso de VR, AR y XR tradicionalmente se ha limitado a sistemas propietarios que requieren hardware especializado, como costosos cascos de realidad virtual, y software dedicado, lo que dificulta su implementación masiva en proyectos con presupuestos ajustados.

En contraste con las tecnologías de simulación tradicionales, la realidad virtual en la web (WebVR) emerge como una alternativa prometedora, innovadora y accesible (Feng et al., 2022). Esta tecnología permite desarrollar entornos de realidad virtual directamente en navegadores web, eliminando la necesidad de software propietario y hardware costoso, lo que incrementa significativamente la accesibilidad y escalabilidad de las simulaciones (Neelakantam & Pant, 2017). Con WebVR,

cualquier dispositivo compatible con un navegador moderno puede convertirse en una herramienta de simulación inmersiva, facilitando el acceso a estas tecnologías tanto en entornos académicos como industriales.

Este trabajo presenta un enfoque para el desarrollo de una simulación de realidad virtual basada en la web, diseñada para permitir el ajuste de controladores de un manipulador robótico. La simulación utiliza tecnologías accesibles como WebVR, eliminando la dependencia de software y hardware costosos. Además, proporciona una experiencia inmersiva e interactiva que facilita el ajuste de controladores en tareas complejas con sistemas robóticos. Este enfoque tiene como objetivo superar las limitaciones de las plataformas de simulación tradicionales, ofreciendo una solución más flexible y accesible tanto para la educación como para la investigación. Como caso de estudio, se emplea la simulación y control de un manipulador robótico de dos grados de libertad para validar el enfoque propuesto.

El resto del trabajo se organiza de la siguiente manera: la Sección II describe el enfoque propuesto para la generación de simulaciones virtuales en la web. En la Sección III, se presenta el caso de estudio que involucra un manipulador robótico de dos grados de libertad. Finalmente, la Sección IV expone las conclusiones y el trabajo futuro derivado del enfoque presentado.

Metodología

A continuación, se describen las principales actividades necesarias para desarrollar una simulación de realidad virtual en la web destinada al control de un manipulador robótico.

A. Diseño de la plataforma

La plataforma hace referencia al sitio o página web que permitirá la visualización e interacción con la simulación virtual 3D del manipulador robótico. Esta plataforma se desarrolla utilizando las siguientes tecnologías web (Ranjan et al., 2020):

- HTML (HyperText Markup Language): Se utiliza para implementar la estructura y disposición de los componentes en la interfaz de usuario, como botones, campos de texto, menús y áreas de contenido, estableciendo la base de la página y organizando la información. Una de estas áreas se destina a la visualización de la simulación virtual del robot y su entorno. El resto de los elementos se utiliza para configurar la simulación (es decir, las condiciones de operación), ajustar los parámetros del controlador y gestionar

interacciones básicas, como el inicio y finalización de una tarea con el manipulador.

- CSS (Cascading Style Sheets): Define el diseño visual de los componentes de la interfaz de usuario implementados con HTML, incluyendo aspectos como colores, tipografías y la disposición de los elementos en distintos tamaños de pantalla, ofreciendo un control detallado sobre la presentación y el estilo de la página.
- JavaScript (JS): Permite la actualización dinámica y la interactividad de la página al gestionar en tiempo real las acciones del usuario, como clics, desplazamientos o entradas de datos. Además, permite la manipulación directa del contenido y facilita la ejecución de eventos, mejorando la experiencia del usuario con interacciones fluidas y dinámicas, sin necesidad de recargar la página. En este contexto, JS se utiliza para controlar la simulación del manipulador robótico en función de las entradas y acciones del usuario en tiempo real. También se encarga de generar y actualizar la visualización de la simulación en la página.

B. Modelado del manipulador robótico

Esta etapa consiste en obtener una representación de dos aspectos clave del manipulador robótico: su comportamiento y su apariencia. Para ello, se emplean los siguientes dos tipos de modelado:

1) *Modelado matemático del manipulador robótico*: En esta actividad se describe el comportamiento cinemático y dinámico del manipulador:

- Modelo cinemático: Describe la relación entre los movimientos de las articulaciones del manipulador y la posición y orientación del efector final en el espacio. Este modelo permite determinar cómo deben moverse las articulaciones del robot para alcanzar una posición deseada, sin tener en cuenta las fuerzas que generan el movimiento. Es esencial para tareas como la planificación de trayectorias, donde es necesario conocer cómo deben moverse las uniones para que el efector final siga una trayectoria específica en el espacio de trabajo. En el caso de los robots manipuladores, existen metodologías bien establecidas, como la de Denavit-Hartenberg (DH), para obtener su modelo cinemático (Craig, 2026).
- Modelo dinámico: Describe las fuerzas y torques necesarios para generar el movimiento en el manipulador. Estas fuerzas dependen de diversos aspectos físicos del robot y su entorno,

como la masa, inercia, fricción y gravedad. Este modelo es fundamental para diseñar y ajustar los controladores que gobiernan el comportamiento del robot, ya que estos calculan, en cada instante de tiempo, las entradas que deben aplicarse a los actuadores para mover el robot de manera correcta bajo diferentes condiciones de operación. Por esta razón, el modelo dinámico del robot se acopla con el del controlador utilizado. El método de Euler-Lagrange, una generalización de las leyes de Newton, permite obtener el modelo dinámico de sistemas mecánicos, incluidos los manipuladores, a partir de un análisis de sus energías (Ogata & Katsuhiko, 2024).

Los modelos matemáticos obtenidos deben ser codificados en JS para su uso en la página web descrita anteriormente.

2) *Modelado 3D del manipulador robótico*: Este tipo de modelado permite obtener una representación visual del sistema y su entorno en un espacio tridimensional. A través de esta actividad, se crean geometrías simples que representan las diferentes partes del robot, como eslabones, uniones y el efector final, así como los objetos estáticos del entorno, replicando su estructura física y proporcionando una base visual para simulaciones y análisis. El modelo 3D del sistema y su entorno se elabora utilizando herramientas de diseño asistido por computadora (CAD). Dicho modelo se exporta a un formato adecuado para su visualización en la página web mencionada anteriormente, como glTF (GL Transmission Format) o GLB (GL binario). Estos formatos permiten la manipulación de los modelos 3D mediante JS (Choi & Park, 2023).

C. Simulación y control

La simulación del sistema se refiere al uso de los modelos obtenidos en la etapa anterior para predecir, en tiempo real, el comportamiento del manipulador físico bajo condiciones de operación específicas y ajustes particulares de su controlador. La simulación del robot se inicia en la página web a petición del usuario y se lleva a cabo en dos fases, utilizando JS:

- 1) Los modelos matemáticos del manipulador, junto con la información sobre las condiciones de operación y la configuración del controlador, proporcionada a través de las entradas del usuario, se utilizan para predecir su comportamiento en un periodo corto de tiempo, que coincide con el tiempo real.
- 2) La información obtenida a partir de los modelos matemáticos se utiliza de inmediato para

manipular la visualización del sistema y su entorno, es decir, para ajustar las posiciones y orientaciones de los elementos 3D del manipulador, generando así la sensación de movimiento.

Las dos fases anteriores se repiten de manera continua cada vez que transcurre el mismo periodo corto en tiempo real, hasta que el usuario decide detener la simulación.

Al finalizar esta etapa, se dispone de una simulación virtual 3D del manipulador robótico y su entorno en la web, que permite al usuario probar diferentes ajustes del controlador y analizar su comportamiento bajo diversas condiciones de operación.

Simulación y control de un manipulador de dos grados de libertad

En esta sección se utiliza el enfoque previamente expuesto para desarrollar la simulación virtual en la web de un manipulador robótico de dos grados de libertad. Este caso de estudio considera un sistema como el que se presenta en la Figura 1. El manipulador tiene una cadena cinemática compuesta por dos eslabones que parten de una base fija. Las uniones entre el primer eslabón y la base, así como entre los dos eslabones, son de tipo rotacional y están actuadas por dos motores eléctricos, capaces de suministrar torques de hasta $10(Nm)$. El efector final se encuentra en el extremo más alejado de la cadena cinemática. El robot opera sobre el plano horizontal XY y no está afectado por la aceleración de la gravedad. Los parámetros físicos de este sistema son conocidos y se muestran en la Tabla I. En esta tabla, l_1 y l_2 representan las longitudes de los eslabones, m_1 y m_2 sus masas, l_{c1} y l_{c2} las distancias a sus centros de masa medidas desde sus uniones más cercanas a la base, I_1 e I_2 sus momentos de inercia, y b_1 y b_2 sus coeficientes de fricción viscosa.



Figura 1. Robot serial de dos grados de libertad. Fuente: Elaboración propia.

TABLA I. Parámetros Físicos del robot manipulador.
Fuente: Elaboración propia.

Parámetro	Valor nominal
l_1	0.4017(m)
l_2	0.2423(m)
m_1	0.8000(kg)
m_2	0.4000(kg)
l_{c_1}	0.2678(m)
l_{c_2}	0.1615(m)
I_1	0.0206(kg.m ²)
I_2	0.0009(kg.m ²)
b_1	0.0120(N.m.s/rad)
b_2	0.0230(N.m.s/rad)

Se asume que el manipulador parte desde una posición de reposo, con sus eslabones apuntando hacia la derecha. El efector final debe llevar a cabo las siguientes tareas independientes en el plano horizontal XY con el menor error posible:

- Seguimiento de trayectoria circular:

$$\begin{aligned} x_d &= r_1 \cos(\alpha) \\ y_d &= r_0 + r_1 \sin(\alpha) \end{aligned} \quad (1)$$

- Seguimiento de trayectoria hipocicloide:

$$\begin{aligned} x_d &= (r_1 - r_2) \cos(\alpha) + r_2 \cos\left(\left(1 - \frac{r_1}{r_2}\right)\alpha\right) \\ y_d &= r_0 + (r_1 - r_2) \sin(\alpha) + r_2 \sin\left(\left(1 - \frac{r_1}{r_2}\right)\alpha\right) \end{aligned} \quad (2)$$

- Seguimiento de trayectoria lemniscata:

$$\begin{aligned} x_d &= r_1 \frac{\cos(\alpha)}{\sin^2(\alpha) + 1} \\ y_d &= -\frac{4}{10} + r_1 \frac{\cos(\alpha) \sin(\alpha)}{\sin^2(\alpha) + 1} \end{aligned} \quad (3)$$

En las tareas anteriores, $\alpha = \frac{2t\pi}{10}$, $r_0 = -\frac{4}{10}$, $r_1 = \frac{15}{10}$, $r_2 = \frac{15}{100}$ y t representa el tiempo. Las componentes de la posición deseada del efector final, x_d y y_d , están expresadas en metros (m) y el tiempo t en segundos (s).

El controlador utilizado para gobernar el manipulador es de tipo Proporcional-Integral-Derivativo (PID). De manera general, el controlador PID calcula las acciones de control para el actuador j -ésimo como se muestra en (4), donde u_j es la entrada de control, e_j es el error en la realización de una tarea en el espacio articular, y $k_{p,j}$, $k_{i,j}$ y $k_{d,j}$ son las ganancias proporcional, integral y derivativa, respectivamente.

$$u_j = k_{p,j} e_j + k_{i,j} \int e_j dt + k_{d,j} \dot{e}_j \quad (4)$$

Se asume que las ganancias del controlador PID se encuentran en el rango de [0,100].

A. Diseño de la plataforma

Con base en la información del caso de estudio, se construye la página web que albergará la simulación virtual del manipulador robótico, junto con otros elementos interactivos. La página implementada se muestra en la Fig. 2. Esta fue desarrollada utilizando el framework Bootstrap, que ofrece componentes preconstruidos de interfaz de usuario en HTML, CSS y JS, lo que facilita y acelera la creación de páginas web responsivas con diseños atractivos (Spurlock, 2023).

La plataforma mostrada en la Figura 2, incluye un panel principal en color blanco para visualizar la simulación virtual del robot y su entorno. Además, incorpora un panel colapsable (es decir, que permite mostrar u ocultar contenido de forma dinámica a elección del usuario) con elementos interactivos. Estos elementos permiten configurar manualmente los dos controladores PID que utiliza el sistema (uno por cada actuador), seleccionar una de las tres tareas que debe realizar el robot (seguimiento de trayectorias circular, hipocicloide y lemniscata), así como iniciar o finalizar la simulación.



Figura 2. Plataforma web de la simulación virtual. Fuente: Elaboración propia.

B. Modelado del manipulador robótico

En esta etapa se obtiene tanto la descripción matemática del comportamiento del sistema como su representación visual.

1) *Modelado matemático del manipulador robótico.* Para facilitar el modelado matemático del robot, sus componentes más importantes se abstraen en el diagrama esquemático de la Figura 3. En dicha figura, $q_1(t)$ y $q_2(t)$ representan las posiciones angulares de las uniones actuadas del manipulador en cualquier instante de tiempo t . A continuación, se derivan los modelos matemáticos necesarios para simular numéricamente el comportamiento del robot:

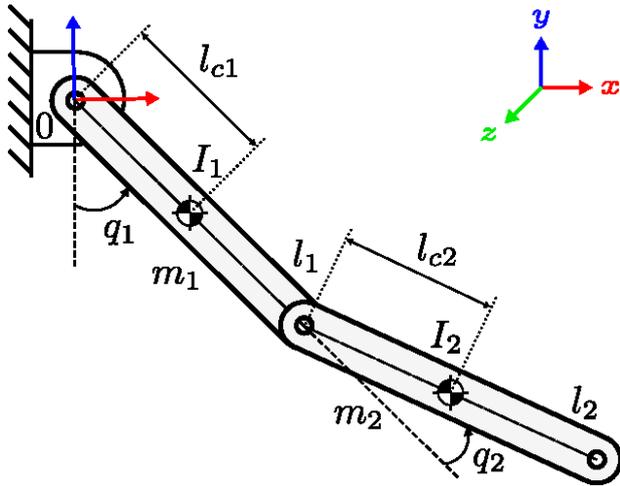


Figura 3. Diagrama esquemático del robot serial de dos grados de libertad. Fuente: Elaboración propia.

- Cinemática directa del robot: Determina la posición de cualquier punto de interés en el manipulador para cualquier configuración articular $q = [q_1, q_2]^T$. En particular, las posiciones de los centros de masa de cada eslabón, representadas por $\xi_1 = [x_1, y_1]^T$ y $\xi_2 = [x_2, y_2]^T$, pueden calcularse mediante un análisis geométrico simple, utilizando las ecuaciones (5) y (6).

$$\xi_1 = \begin{bmatrix} l_{c1} \sin(q_1) \\ -l_{c1} \cos(q_1) \end{bmatrix} \quad (5)$$

$$\xi_2 = \begin{bmatrix} l_1(q_1) + l_{c2} \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_{c2} \cos(q_1 + q_2) \end{bmatrix} \quad (6)$$

- Dinámica del robot: Describe el movimiento del robot en función de los torques que lo generan. El método de Euler-Lagrange se utiliza para obtener la ecuación de movimiento del manipulador. Usando la información del análisis anterior, es posible calcular las energías cinética y potencial totales del sistema, que son necesarias para aplicar dicho método. Estas energías se muestran en las ecuaciones (7) y (8). Cabe destacar que la energía potencial en este sistema es cero, ya que no se ve afectada por la aceleración de la gravedad.

$$\mathcal{K} = \frac{1}{2} m_1 \dot{\xi}_1^T \dot{\xi}_1 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} m_2 \dot{\xi}_2^T \dot{\xi}_2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \quad (7)$$

$$U = m_1 g y_1 + m_2 g y_2 = 0 \quad (8)$$

Utilizando ambas energías, se establece el Lagrangiano del sistema en la ecuación (9).

$$\mathcal{L} = \mathcal{K} - U = \mathcal{K} \quad (9)$$

Por otro lado, la función de disipación en la ecuación (10) considera los efectos de las fuerzas no conservativas en el sistema, que son producidas por la fricción viscosa en cada eslabón.

$$\mathcal{D} = \frac{1}{2} b_1 \dot{q}_1^2 + \frac{1}{2} b_2 \dot{q}_2^2 + \mu_1 |\dot{q}_1| + \mu_2 |\dot{q}_2| \quad (10)$$

La ecuación de movimiento no conservativa del robot manipulador está dada por (11), donde $\tau = [\tau_1, \tau_2]^T$ es el vector de torques generalizados (Spong et al., 2020).

$$\frac{\partial \mathcal{L}}{\partial q} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} + \frac{\partial \mathcal{D}}{\partial \dot{q}} = \tau \quad (11)$$

La forma matricial de la ecuación de movimiento se muestra en (12), donde $M(q)\ddot{q} \in R^{2 \times 2}$ es la matriz de inercia, $C(q, \dot{q}) \in R^{2 \times 2}$ es la matriz de fuerzas centrífugas y de Coriolis, $f_{nc}(\dot{q}) \in R^{2 \times 1}$ es el vector de fuerzas no conservativas, y $q, \dot{q}, \ddot{q} \in R^{2 \times 1}$ son los vectores de posiciones, velocidades y aceleraciones generalizadas, respectivamente. Finalmente, $\tau = u$, donde $u = [u_1, u_2]^T$ es la acción de control calculada por el controlador PID.

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + f_{nc}(\dot{q}) \quad (12)$$

- Cinemática directa del efector final: Además de la cinemática y dinámica del manipulador, es importante conocer el comportamiento del efector final para una configuración articular dada $q = [q_1, q_2]^T$, ya que las tareas requeridas dependen de este punto del robot. La relación entre la posición cartesiana del efector final, $\xi = [x, y]^T$, y las coordenadas articulares q se expresa mediante la ecuación (13).

$$\xi = \begin{bmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) \end{bmatrix} \quad (13)$$

Las velocidades cartesianas $\dot{\xi} = [\dot{x}, \dot{y}]^T$ del efector final se obtienen utilizando la expresión en (14), donde $J(q)$ es la matriz Jacobiana definida en (15).

$$\dot{\xi} = J(q)\dot{q} \quad (14)$$

$$J(q) = \frac{\partial \xi}{\partial q} \quad (15)$$

Finalmente, las aceleraciones cartesianas $\ddot{\xi} = [\ddot{x}, \ddot{y}]^T$ pueden calcularse derivando $\dot{\xi}$ con respecto al tiempo t , como se muestra en la ecuación (16).

$$\ddot{\xi} = \dot{J}(q)\dot{q} + J(q)\ddot{q} \quad (16)$$

- Cinemática inversa del efector final: A diferencia de la cinemática directa, este modelo determina las posiciones, velocidades y aceleraciones articulares necesarias para alcanzar una posición, velocidad y aceleración específicas en el efector final. Estos valores articulares establecen los objetivos de control del sistema para la realización de una tarea. Geométricamente, la posición del efector final ξ puede expresarse en el espacio de coordenadas articulares $q = [q_1, q_2]^T$ utilizando la ecuación (17).

$$q = \begin{bmatrix} \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \\ \frac{\pi}{2} + \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)} \right) \end{bmatrix} \quad (17)$$

Las velocidades articulares se obtienen en la ecuación (18) a partir de la ecuación (14).

$$\dot{q} = J^{-1}(q)\dot{\xi} \quad (18)$$

Por último, las aceleraciones articulares se calculan en la ecuación (19) derivando (18) con respecto al tiempo t .

$$\ddot{q} = J^{-1}(q)\ddot{\xi} - J^{-1}(q)\dot{J}(q)J^{-1}(q)\dot{\xi} \quad (19)$$

2) Modelado 3D del manipulador robótico: Para representar el aspecto físico del robot y los elementos presentes en su entorno de operación, se utilizó Blender, una herramienta gratuita y de código abierto para diseño asistido por computadora (CAD), especializada en modelado y animación 3D (Mullen, 2011).

Las geometrías principales del manipulador robótico fueron modeladas en Blender utilizando primitivas 3D simples, como prismas rectangulares y cilindros, aplicando materiales básicos de un solo color para mantener la simplicidad visual. Además, como parte del entorno, se crearon representaciones tridimensionales del suelo, una mesa y una silla, proporcionando un contexto de operación más realista. Es importante señalar que las unidades en el espacio virtual 3D se ajustaron para coincidir con las del mundo real, lo que facilita su manipulación en etapas posteriores del desarrollo. Asimismo, las relaciones espaciales entre el robot y los objetos de su entorno se mantuvieron coherentes con la realidad, con el objetivo de ofrecer a los usuarios una experiencia de simulación familiar y realista. Los modelos 3D se muestran en la Figura 4.

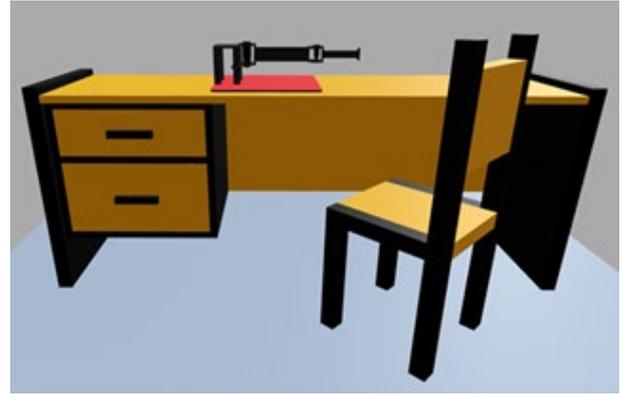


Figura 4. Simulación virtual del robot manipulador de dos grados de libertad en la web. Fuente: Elaboración propia.

Una vez generados los modelos en Blender, se exportan en formato GLB para su visualización en la página web. Antes de su implementación en la plataforma, se configuran otros aspectos del entorno, como la iluminación, utilizando la herramienta Three.js Editor (consultado el 20 de octubre de 2024). En la plataforma web, Three.js se utiliza para facilitar la manipulación y visualización de los modelos en el panel designado. Three.js es una biblioteca de JS que simplifica la creación y renderizado de gráficos 3D en navegadores mediante WebGL, permitiendo desarrollar visualizaciones tridimensionales interactivas de manera eficiente (Danchilla, 2012). La visualización de la simulación virtual en la web se muestra en la Figura 5.



Figura 5. Simulación virtual del robot manipulador de dos grados de libertad en la web. Fuente: Elaboración propia.

C. Simulación y control

Una vez que los elementos de la simulación están desplegados en la página web, como se muestra en la Figura 5, los modelos matemáticos derivados anteriormente se utilizan para predecir el comportamiento del manipulador bajo condiciones de operación específicas y ajustes particulares de su controlador. Las condiciones de operación, en este caso, están relacionadas con la tarea seleccionada por el usuario para el robot (es decir, el seguimiento de una trayectoria circular, hipocicloide o lemniscata) y sus condiciones

iniciales (el robot siempre comienza desde el reposo con una posición articular $q(0) = [q_1(0), q_2(0)]^T = [\pi/2, 0]^T$, es decir, con ambos eslabones apuntando hacia la derecha). Por otra parte, el ajuste particular del controlador está determinado por las ganancias de los controladores PID que el usuario introduce manualmente. Cuando el usuario inicia la simulación, se ejecutan las dos fases siguientes para simular numéricamente los modelos matemáticos del robot y actualizar la visualización del entorno virtual:

- 1) En esta fase, el modelo dinámico del manipulador, dado en (12), se establece en el espacio de estado como se muestra en (20). En esta ecuación, $z(t)$ representa el estado del sistema, $u(t)$ es la entrada de control, y las funciones $f(z(t), u(t))$ y $h(z(t), u(t))$ definen su dinámica y cómo los estados afectan su salida, respectivamente. El estado de un sistema es el conjunto mínimo de variables que describen su comportamiento en un instante dado (Ogata, 1967). Para sistemas mecánicos, como los manipuladores, el estado incluye las coordenadas generalizadas (configuraciones articulares) y sus primeras derivadas con respecto al tiempo (Ogata, 2004). Por lo tanto, el estado del sistema está representado por $z = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T$.

$$\begin{aligned} \dot{z}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \end{aligned} \quad (20)$$

Por otro lado, la ecuación de estado del sistema, dada por $\dot{z}(t)$ en (20), describe la evolución del estado $z(t)$ en el tiempo cuando se aplica la entrada de control $u(t)$. Dado que $z = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T$, la ecuación de estado puede derivarse de las ecuaciones de movimiento del mecanismo en (12). Teniendo en cuenta que la salida $y(t)$ del manipulador incluye las posiciones y velocidades de las articulaciones, el modelo dinámico en el espacio de estado es:

$$\begin{aligned} \dot{z}(t) &= \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ M(q)^{-1}(u - C(q, \dot{q})\dot{q} - f_{nc}(\dot{q})) \end{bmatrix} \\ y(t) &= x(t) \end{aligned} \quad (21)$$

En el modelo anterior, $u(t) = [u_1(t), u_2(t)]^T$ se calcula utilizando la ecuación del controlador PID en (4), donde $e_j(t) = \bar{q}_j(t) - q_j(t)$ es el error de posición en el espacio articular en el instante t . Aquí, $\bar{q}_j(t)$ es la posición articular deseada del j -ésimo eslabón, la cual se obtiene a partir de la trayectoria seleccionada por el usuario y la cinemática inversa del efector final en (17). Las ganancias $k_{p,j}$, $k_{i,j}$ y $k_{d,j}$ corresponden a las proporcionadas por el usuario desde la interfaz web.

El modelo en (21) es una ecuación diferencial ordinaria de primer orden que, sujeta a la condición inicial $q(0)$, define un problema de valor inicial. Este problema puede resolverse mediante un método de integración numérica para predecir la evolución del estado dentro de un breve intervalo de tiempo futuro dt , es decir, $z(t + dt)$. Debido a su simplicidad, en esta fase se utiliza el método *ode1* (Griffiths et al., 2010) junto con un $dt = 0.005$ (s).

- 2) La información del estado futuro obtenida en la fase anterior, es decir, $z(t + dt)$, se utiliza para actualizar la visualización del sistema y su entorno. En este contexto, los valores de las posiciones articulares del robot $q_1(t)$ y $q_2(t)$ se reemplazan por aquellos correspondientes $az(t + dt)$, es decir, $q_1(t + 1)$ y $q_2(t + 1)$, lo que produce cambios en las posiciones y orientaciones de los eslabones del robot, generando la sensación de movimiento. Three.js se emplea para facilitar la implementación de estos cambios.

Estas dos fases se repiten cada vez que transcurre un intervalo de $dt = 0.005$ (s) en tiempo real, y la simulación finaliza cuando el usuario lo solicita.

Finalmente, la Figura 6 muestra un ejemplo de la ejecución de la simulación virtual, utilizando los parámetros de control predefinidos en la página web y el seguimiento de la trayectoria circular por parte del manipulador.



Figura. 6. Simulación virtual del robot manipulador en el seguimiento de una trayectoria circular. Fuente: Elaboración propia.

Al concluir esta fase, se dispone de una simulación 3D en línea del manipulador robótico y su entorno, lo que permite al usuario experimentar con diferentes configuraciones del controlador y evaluar su desempeño en diversas condiciones de operación. La simulación completa está disponible en *Simulación* (consultado el 20 de octubre de 2024).

Conclusiones y trabajo a futuro

Este trabajo presenta una plataforma accesible y eficiente para la simulación y ajuste de controladores de manipuladores robóticos mediante tecnologías web. El enfoque desarrollado permite al usuario interactuar con una simulación inmersiva de un manipulador de dos grados de libertad, facilitando la experimentación con diferentes configuraciones de control en tiempo real. La implementación elimina la necesidad de software propietario y hardware costoso, abriendo el acceso a herramientas avanzadas de simulación a un público más amplio, incluidos estudiantes y profesionales en entornos de recursos limitados.

El caso de estudio demostró que la simulación es capaz de replicar con precisión el comportamiento dinámico del manipulador bajo distintas condiciones de operación. Los usuarios pueden observar cómo diferentes ajustes del controlador afectan el desempeño del sistema, lo que permite realizar ajustes más informados antes de implementar el controlador en un robot físico. Esto subraya la viabilidad y el potencial del uso de tecnologías web para tareas de simulación que tradicionalmente requerían plataformas especializadas y costosas.

El trabajo futuro se centrará en expandir la plataforma para manipuladores robóticos con más grados de libertad y en integrar la simulación con hardware físico, permitiendo transferir parámetros optimizados al robot real en tiempo real. Además, se explorará el uso de realidad aumentada para mejorar la experiencia inmersiva y se optimizará el desempeño de la simulación mediante algoritmos avanzados. Finalmente, se planea ampliar la interfaz de usuario con herramientas de análisis de datos y automatización del ajuste de controladores, mejorando la experiencia general y aumentando su aplicabilidad en diversos entornos.

Referencias

- Alonso, R., Bonini, A., Reforgiato Recupero, D., & Spano, L. D. (2022). Exploiting virtual reality and the robot operating system to remote-control a humanoid robot. *Multimedia Tools and Applications*, 81(11), 15565-15592.
- Ceccarelli, M. (2022). *Fundamentals of mechanics of robotic manipulation* (Vol. 112, pp. 1-381). Cham, Switzerland: Springer.
- Cai, J., Deng, J., Zhang, W., & Zhao, W. (2021). Modeling Method of Autonomous Robot Manipulator Based on D-H Algorithm. *Mobile Information Systems*, 2021(1), 4448648.
- Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., ... & Trinkle, J. (2021). On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118(1), e1907856118.
- Choi, J. M., & Park, K. H. (2023). Performance Analysis of GLTF/GLB to Improve 3D Content Rendering Performance. *Journal of Platform Technology*, 11(4), 13-18.
- Craig, J. (2004). *Introduction to Robotics: Mechanics and Control*, Textbook. International Edition, Prentice Hall.
- Danchilla, B., & Danchilla, B. (2012). Three.js framework. *Beginning WebGL for HTML5*, 173-203.
- Feng, Y., You, C., Li, Y., Zhang, Y., & Wang, Q. (2022). Integration of computer virtual reality technology to college physical education. *Journal of web engineering*, 21(7), 2049-2071.
- Gaber, A. N. A., Eldrainy, Y. A., & Awad, T. H. (2021). Uncertainty solution of robot parameters using fuzzy position control applied for an automotive cracked exhaust system inspection. *Alexandria Engineering Journal*, 60(2), 2355-2367.
- Griffiths, D. F., & Higham, D. J. (2010). *Numerical methods for ordinary differential equations: initial value problems* (Vol. 5). London: Springer.
- Jayaswal, K., Palwalia, D. K., & Kumar, S. (2021). Analysis of robust control method for the flexible manipulator in reliable operation of medical robots during COVID-19 pandemic. *Microsystem Technologies*, 27(5), 2103-2116.
- Lewis, F. L., Dawson, D. M., & Abdallah, C. T. (2003). *Robot manipulator control: theory and practice*. CRC Press.
- Lin, M., San, L., & Ding, Y. (2020, March). Construction of robotic virtual laboratory system based on Unity3D. In *IOP Conference Series: Materials Science and Engineering* (Vol. 768, No. 7, p. 072084). IOP Publishing.
- Luenberger, D. (1967). State space analysis of control systems. *IEEE Transactions on Automatic Control*, 12(6), 805-805.
- Makhataeva, Z., & Varol, H. A. (2020). Augmented reality for robotics: A review. *Robotics*, 9(2), 21.
- Maldonado-Valencia, R. I., Rodriguez-Garavito, C. H., Cruz-Perez, C. A., Hernandez-Navas, J. S., & Zabala-Benavides, D. I. (2022). Planning and visual-servoing for robotic manipulators in ROS. *International Journal of Intelligent Robotics and Applications*, 6(4), 602-614.
- Marian, M., Stîngă, F., Georgescu, M. T., Roibu, H., Popescu, D., & Manta, F. (2020, October). A ROS-based control application for a robotic platform using the gazebo 3D simulator. In *2020 21th International Carpathian Control Conference (ICCC)* (pp. 1-5). IEEE.
- Mullen, T. (2011). *Mastering blender*. John Wiley & Sons.
- Neelakantam, S., & Pant, T. (2017). Learning web-based virtual reality: build and deploy web-based virtual reality technology. Apress.
- Ogata, K. (2004). *System dynamics*.
- Ostanin, M., Mikhel, S., Evlampiev, A., Skvortsova, V., & Klimchik, A. (2020, May). Human-robot interaction for robotic manipulator programming in Mixed Reality. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2805-2811). IEEE.
- Pratheep, V. G., Chinnathambi, M., Priyanka, E. B., Ponnuragan, P., & Thiagarajan, P. (2021, February). Design and Analysis of six DOF robotic manipulator. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1055, No. 1, p. 012005). IOP Publishing.

- Ranjan, A., Sinha, A., & Battice, R. (2020). JavaScript for modern web development: building a web application using HTML, CSS, and JavaScript. BPB Publications.
- Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., & Aldape-Pérez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem. *Applied Soft Computing*, 93, 106342.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). Robot modeling and control. John Wiley & Sons.
- Spong, M. W. (2022). An historical perspective on the control of robotic manipulators. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1), 1-31.
- Spurlock, J. (2013). Bootstrap: responsive web development. " O'Reilly Media, Inc."