



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE
PUEBLA**

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**DOCTORADO EN INGENIERÍA DEL LENGUAJE Y EL
CONOCIMIENTO**

Tesis Doctoral:

Dinámica y control de un robot bípedo de 18 grados de libertad

Presenta:

Miguel Angel Ortega Palacios

Directores de tesis:

Dra. Amparo Dora Palomino Merino

Dr. José Fernando Reyes Cortés

Puebla, Pue. Marzo de 2026

Agradecimientos

Deseo expresar mi más sincero agradecimiento a la Benemérita Universidad Autónoma de Puebla, institución que me brindó la oportunidad de realizar mis estudios de doctorado y que, a través de su compromiso con la investigación y la formación académica, ha contribuido de manera fundamental a mi desarrollo profesional.

De manera especial, agradezco al programa de posgrado Ingeniería del Lenguaje y del Conocimiento (Language and Knowledge Engineering, LKE) de la Facultad de Ciencias de la Computación, por proporcionar el entorno académico, los recursos y la formación necesarios para llevar a cabo esta investigación.

Expreso también mi profundo agradecimiento a mis directores de tesis, la Dra. Amparo Palomino Merino y el Dr. Fernando Reyes Cortés, por su orientación, paciencia, conocimiento y apoyo constante durante el desarrollo de este trabajo. Sus valiosas observaciones, su guía académica y su confianza fueron fundamentales para la culminación de este trabajo.

A mis padres, por su apoyo incondicional, sus palabras de aliento y por haber sido siempre una fuente constante de motivación a lo largo de mi formación académica y personal.

Finalmente, agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por la beca otorgada (CVU: 633719) para la realización de los estudios de Doctorado en Ingeniería del Lenguaje y del Conocimiento (Language and Knowledge Engineering, LKE) en la Benemérita Universidad Autónoma de Puebla, la cual fue fundamental para el desarrollo y conclusión de la tesis doctoral.

Resumen

La locomoción bípeda humana es el resultado de un proceso evolutivo altamente eficiente que permite el desplazamiento estable sobre dos extremidades, liberando simultáneamente las extremidades superiores para la manipulación e interacción con el entorno. La pérdida parcial o total de esta capacidad motriz tiene un impacto significativo en la movilidad del ser humano, lo que ha motivado el desarrollo de robots humanoides capaces de emular la marcha bípeda en entornos diseñados para la locomoción humana. No obstante, reproducir de manera fiel y estable la marcha humana en un robot bípedo continúa siendo un problema abierto, debido principalmente a la complejidad inherente de su modelado cinemático y dinámico, así como a los desafíos asociados con la estabilidad dinámica durante la locomoción.

En este contexto, la presente investigación se centra en el desarrollo del modelo dinámico completo de un robot humanoide de 18 grados de libertad (GDL), utilizando como plataforma física de referencia el robot Bioloid Premium. Se propone una metodología recursiva basada en la mecánica analítica de Euler, que permite obtener de manera sistemática y computacionalmente eficiente las ecuaciones dinámicas del sistema a partir de las ecuaciones de movimiento de Euler–Lagrange. El enfoque recursivo facilita el cálculo de la cinemática directa, la cinemática diferencial, las velocidades lineales y angulares, así como las energías cinética y potencial del robot.

El modelo dinámico obtenido se implementa en MATLAB mediante programación simbólica, lo que permite comparar el desempeño computacional del método recursivo propuesto frente al enfoque convencional de Euler–Lagrange. Asimismo, los resultados del modelado se validan mediante simulación en Simscape Multibody, donde se desarrolla un modelo virtual del robot humanoide. Sobre este modelo se implementan controladores de posición y se genera una trayectoria dinámica de marcha con el objetivo de evaluar el desempeño de tres esquemas de control: control Proporcional–Derivativo (PD), control con saturación y control basado en la función tangente hiperbólica. Los resultados de simulación permiten validar la consistencia del modelo dinámico propuesto y demostrar las ventajas computacionales y prácticas del enfoque recursivo.

Introducción

La incorporación progresiva de robots humanoides en distintos sectores de la sociedad como la industria, la medicina, la exploración y la asistencia a personas ha incrementado el interés por el desarrollo de sistemas capaces de interactuar de manera eficiente y segura en entornos diseñados originalmente para la locomoción humana. A diferencia de los robots industriales tradicionales, los robots humanoides deben ejecutar tareas complejas que implican locomoción bípeda, coordinación de múltiples articulaciones y operación bajo condiciones dinámicas, lo que representa un reto significativo desde el punto de vista del modelado y el control.

La locomoción bípeda humanoide constituye uno de los problemas más complejos en robótica móvil, debido a la necesidad de mantener el equilibrio dinámico durante el movimiento. El desempeño de los algoritmos de control depende en gran medida de la precisión de los modelos cinemáticos y dinámicos que describen el comportamiento del robot. Sin embargo, a medida que el número de grados de libertad aumenta, la obtención explícita de dichos modelos se vuelve algebraicamente compleja y computacionalmente costosa.

En la literatura, el modelado dinámico de robots bípedos se aborda comúnmente mediante formulaciones basadas en Euler–Lagrange o Newton–Euler. No obstante, en muchos casos estos enfoques se aplican a modelos simplificados o a subconjuntos del robot, y son escasos los trabajos que presentan el desarrollo completo del modelo dinámico de un robot humanoide de cuerpo completo con 18 grados de libertad. Esta limitación motiva la búsqueda de metodologías alternativas que permitan estructurar el proceso de modelado de forma sistemática y eficiente.

En esta tesis doctoral se propone una metodología recursiva basada en la mecánica analítica de Euler para la obtención del modelo dinámico completo de un robot humanoide de 18 grados de libertad, tomando como caso de estudio el robot Bioloid Premium. Mediante este enfoque se desarrollan de forma recursiva la cinemática directa, la cinemática diferencial, el cálculo de velocidades lineales y angulares, así como las energías cinética y potencial, culminando en la obtención de las ecuaciones de movimiento a través del formalismo de Euler–Lagrange.

El modelo dinámico resultante se implementa mediante programación simbólica en MATLAB, lo que permite evaluar su eficiencia computacional y compararla con el método convencional. Asimismo, el modelo se valida mediante simulaciones dinámicas en Simscape Multibody, donde se implementan controladores de posición y se generan trayectorias dinámicas de marcha para analizar el comportamiento del robot bajo condiciones realistas. De esta manera, la investigación aporta un modelo dinámico completo y validado, así como una metodología reproducible para el estudio y control de robots humanoides de múltiples grados de libertad. El Capítulo 1 presenta el estudio de la literatura relacionada con el presente trabajo, el capítulo 2 los fundamentos teóricos necesarios, revisando las formulaciones matemáticas, el Capítulo 3 detalla la metodología propuesta y el desarrollo del algoritmo recursivo, el Capítulo 4 describe la validación de la metodología propuesta, en el Capítulo 5 se explica la construcción del modelo virtual del robot Bioloid Premium en Simscape Multibody, así como el diseño de las trayectorias de marcha y la implementación del control de posición, y finalmente el capítulo 6 ofrece las conclusiones generales de la investigación y trabajo futuro de este estudio.

Contenido

1. Estado del arte	4
2. Marco teórico	9
2.1 Matrices de rotación	9
2.2 Matriz de rotación alrededor del eje z_0	11
2.3 Matriz de rotación alrededor del eje x_0	12
2.4 Matriz de rotación alrededor del eje y_0	12
2.5 Matriz antisimétrica	14
2.6 Propiedades de las matrices antisimétrica	14
2.7 Transformaciones de traslación y rotación	15
3. Modelo dinámico del robot Bioloid Premium	18
3.1 Cinemática directa recursiva	20
3.2 Componente lineal de la cinemática diferencial	22
3.3 Rapidez lineal recursiva	23
3.4 Matriz de inercia recursiva	23
3.5 Energía cinética	25
3.6 Energía potencial	25
3.7 Lagrangiano del robot	26
3.8 Ecuaciones de Euler-Lagrange	27
3.9 Propiedades del modelo dinámico	40
4. Análisis cuantitativo del método recursivo	46
5. Implementación del control de posición	48
6. Conclusiones y trabajo a futuro	67
7. Referencias	69
8. Publicaciones	71

Justificación

El estudio de la locomoción bípeda en robots humanoides continúa siendo uno de los desafíos más relevantes en la robótica moderna, debido a la complejidad inherente del movimiento humano y a la necesidad de garantizar estabilidad dinámica durante la marcha. La reproducción de este tipo de locomoción requiere modelos cinemáticos y dinámicos precisos que describan adecuadamente el comportamiento del robot, ya que cualquier imprecisión en estos modelos se refleja directamente en el desempeño de los algoritmos de control.

Esta problemática se intensifica conforme aumenta el número de grados de libertad del sistema, como ocurre en el caso de los robots humanoides de cuerpo completo. En particular, el robot Bioloid Premium cuenta con 18 grados de libertad, lo que implica un alto nivel de acoplamiento dinámico y una considerable complejidad computacional en la obtención de su modelo dinámico. Los enfoques convencionales basados en las ecuaciones de Euler–Lagrange suelen conducir a expresiones algebraicas extensas y difíciles de manejar, limitando su implementación práctica y su uso en simulaciones dinámicas y esquemas de control.

Ante este panorama, resulta necesario desarrollar metodologías de modelado que permitan obtener el modelo dinámico completo del robot de manera sistemática, eficiente y escalable. La metodología recursiva basada en la mecánica analítica de Euler propuesta en esta tesis ofrece una alternativa sólida para abordar este problema, ya que permite estructurar el cálculo de las variables cinemáticas y dinámicas de forma ordenada, reduciendo el costo computacional asociado al uso de métodos convencionales sin sacrificar precisión en el modelo resultante.

La presente investigación se justifica, además, por la implementación computacional del modelo dinámico mediante programación simbólica en MATLAB, lo que posibilita una comparación objetiva del tiempo de ejecución entre el método recursivo propuesto y el enfoque tradicional de Euler–Lagrange. Esta comparación proporciona evidencia cuantitativa de las ventajas computacionales del enfoque recursivo, particularmente relevante para robots con un elevado número de grados de libertad.

Finalmente, la validación del modelo dinámico a través de simulaciones en Simscape Multibody, junto con la implementación de controladores de posición y la generación de trayectorias dinámicas de marcha, permite evaluar de manera integral el comportamiento del robot bajo condiciones dinámicas realistas. Los resultados obtenidos no solo confirman la consistencia del modelo propuesto, sino que también aportan una guía metodológica reproducible que puede ser extendida a otros robots humanoides con características similares, fortaleciendo así la contribución científica y práctica de esta tesis doctoral.

Problema de investigación

El desarrollo del modelo dinámico de robots bípedos ha sido ampliamente abordado por la comunidad científica; sin embargo, gran parte de los trabajos existentes se apoyan en metodologías tradicionales, como la convención de Denavit–Hartenberg y las formulaciones clásicas de Euler–Lagrange o Newton–Euler, las cuales suelen conducir a modelos complejos y poco eficientes desde el punto de vista computacional cuando se aplican a robots

humanoides con un elevado número de grados de libertad. Esta situación limita la obtención explícita de modelos dinámicos completos y dificulta su implementación en algoritmos de control y simulación avanzada.

En particular, para robots humanoides de cuerpo completo, la derivación del modelo dinámico mediante enfoques convencionales implica un alto costo computacional y una complejidad algebraica significativa. Además, muchos de los estudios reportados en la literatura se centran en modelos simplificados o en subsistemas del robot, lo que restringe el análisis integral del comportamiento dinámico del sistema y su aplicación en esquemas de control de movimiento más generales.

Ante esta problemática, surge la necesidad de desarrollar una metodología alternativa que permita obtener el modelo dinámico completo del robot humanoide de 18 grados de libertad de manera sistemática, eficiente y escalable. En este contexto, el uso de la mecánica analítica de Euler, combinada con algoritmos recursivos, se plantea como una solución viable para expresar las ecuaciones del modelo dinámico de forma más compacta y estructurada, facilitando su implementación computacional y reduciendo el tiempo de ejecución respecto a los métodos convencionales.

Asimismo, resulta necesario contar con un modelo cinemático y dinámico preciso que permita el diseño e implementación de leyes de control de posición capaces de generar movimiento autónomo del robot en entornos controlados, sin depender de rutinas de movimiento preprogramadas. La ausencia de un modelo dinámico completo y validado limita la comprensión de las capacidades, restricciones y desempeño del sistema, lo que dificulta el desarrollo de estrategias de control más robustas y generalizables.

Por lo tanto, el problema de investigación de esta tesis se centra en cómo obtener y validar un modelo dinámico completo y computacionalmente eficiente de un robot humanoide de 18 grados de libertad, basado en el robot Bioloid Premium, mediante una metodología recursiva fundamentada en la mecánica analítica de Euler, y cómo utilizar dicho modelo para el diseño y evaluación de estrategias de control de posición que permitan generar movimiento dinámico en el robot.

Objetivo general

Desarrollar el modelo dinámico de un robot bípedo de 18 grados de libertad mediante algoritmos recursivos en un entorno simulado.

Objetivos específicos

1. Desarrollar y validar el modelo cinemático y dinámico del robot mediante técnicas recursivas basadas en la formulación de Euler–Lagrange y la mecánica analítica de Euler.
2. Implementar el modelo dinámico recursivo en MATLAB mediante programación simbólica.
3. Comparar los tiempos de ejecución del algoritmo recursivo frente al método convencional de Euler–Lagrange.
4. Proponer e implementar leyes de control de posición para las articulaciones del robot.
5. Evaluar y comparar la eficiencia de los controladores de posición propuestos mediante la generación y seguimiento de trayectorias dinámicas en un entorno de simulación.

Pregunta de investigación

¿En qué medida una metodología recursiva basada en la formulación de Euler–Lagrange y la mecánica analítica de Euler mejora la eficiencia computacional en el desarrollo del modelo dinámico de un robot humanoide de 18 grados de libertad (GDL), y cómo influye dicho modelo en el desempeño de controladores de posición durante la ejecución de trayectorias dinámicas?

Hipótesis

La implementación de una metodología recursiva basada en la formulación de Euler–Lagrange y la mecánica analítica de Euler permite desarrollar de manera computacionalmente más eficiente el modelo dinámico del robot humanoide Bioloid Premium de 18 GDL, en comparación con el método convencional, y proporciona una base adecuada para mejorar el desempeño de controladores de posición durante la ejecución de trayectorias dinámicas en simulación.

1. Estado del arte

El modelo cinemático y la dinámica inversa se usa ampliamente para el modelado de robots reales [1]. El método puede proporcionar una trayectoria de movimiento compatible y es robusto a perturbaciones externas, pero su rendimiento depende en gran medida de la calidad del modelo dinámico, lo cual es una tarea bastante compleja de calcular.

Existen tres metodologías que permiten desarrollar el modelo matemático de la dinámica de un robot industrial: La metodología de Newton-Euler, la metodología de Lagrange – Euler y el modelo matemático de Gibb [2]. En el presente trabajo se usa la metodología de Lagrange-Euler porque las expresiones resultantes son más compactas con respecto a las obtenidas con la metodología Newton-Euler; en la formulación Lagrange-Euler no se consideran todas las acciones que no generen trabajo, además con esta metodología se obtiene el modelo dinámico en términos de trabajo y de energías. La metodología de Lagrange-Euler analiza el sistema mecánico como un todo desde el punto de vista energético.

La mayor parte de los trabajos revisados en la literatura emplean la metodología de Euler-Lagrange [1] - [5], [7], [8], [10], [12] para obtener el modelo dinámico de los robots bípedos estudiados , y algunos otros emplean la metodología de Newton-Euler [6], [9]. Todos ellos se enfocan a la implementación de un modelo de marcha estable, empleando diversos métodos de control para lograr mantener la estabilidad del robot durante la marcha, ya sea mediante el moldeo de energía [4], control robusto de trayectoria [5], Cart-Table [6], [15], control PD [10] - [12]. Es importante señalar que todos los autores tienen el objetivo de modelar solamente la dinámica de las piernas del robot, y algunas veces tomando en cuenta el torso.

En los trabajos de Kajita se ha estudiado la serie de robots humanoides HRP, desde el año 2000 hasta actualidad, con el objetivo de emplear robots humanoides prácticos en lugar de humanos dentro de industrias de ensamblaje a gran escala, como sitios de construcción, instalaciones de aeronaves y astilleros. Sin embargo, todavía no se ha hecho un uso práctico de los robots humanoides que trabajan en sitios reales. En [24] el autor menciona que no se ha hecho un uso práctico de los robots humanoides, debido a que la mayoría de los robots de la serie HRP aún carecen de capacidad física para trabajar en sitios reales.

El desarrollo de modelos de marcha bípeda aún sigue siendo un tema de estudio. En [16] se propone emplear el modelo cart-table propuesto en [17]. La idea principal es aproximar al robot humanoide por una masa ubicada en su Centro de masas (CoM) y es igual a la masa total del robot humanoide. Por tanto, el complejo problema de controlar al robot humanoide se transforma en el control de un péndulo invertido. La principal ventaja de este modelo es no solo predecir la trayectoria *Zero Moment Point* (ZMP) del robot humanoide sin el cálculo de la dinámica multicuerpo, sino también proporcionar la trayectoria del CoM del robot humanoide para rastrear la trayectoria ZMP planificada. Este modelo es implementado en el robot humanoide HRP4-C [18].

En [14] plantean un modelo dinámico espacialmente cuantizado basado en la posición del ZMP (Punto de momento cero), la posición horizontal de la cadera y de su centro de masa. Pero mencionan que no está claro cuáles son los beneficios de su modelo propuesto con respecto a los métodos convencionales.

También hay múltiples trabajos revisados donde los autores no emplean robots comerciales, sino robots diseñados por ellos mismos, ya sea para implementarlos en

simulación o de forma experimental [4], [5], [7], [8], [10], [12].

Con respecto al robot Bioloid, ha sido empleado por la comunidad científica para realizar diversos estudios relativos a humanoides; se ha utilizado, por ejemplo, en investigaciones sobre estrategias de control de marcha bípeda basadas en redes neuronales [19], y en técnicas evolutivas [20]. Así mismo el Bioloid se ha aplicado en el análisis de la locomoción y auto-localización de agentes de equipos de futbol soccer [21]. En otros trabajos emplean el modelo cinemático inverso y la dinámica del robot Bioloid para generar la marcha. En [22] se calcula el valor de las variables que corresponden a la masa, centro de masa y tensor de inercia de cada eslabón mediante SolidWorks. Pero la marcha no se implementa en el robot, solo se limita a la simulación de la dinámica y del robot humanoide utilizando SimMechanics y Virtual Reality Toolbox. Además, se considera la relación que existe entre la velocidad angular y el torque de salida de los servomotores, por lo que se propone una estimación de la velocidad de los servomotores, a partir de la posición de estos, para estimar el torque de salida. Mientras que en [23] se obtienen las ecuaciones del modelo dinámico, pero no se indica como es que se implementó en el robot, puesto que el autor mismo menciona que los servomotores AX-12A tienen un control de posición PD en lazo cerrado de forma implícita, y no es posible implementar un control de torque.

2. Marco teórico

2.1 Matrices de rotación

En la figura 1 muestra dos sistemas de referencia cartesianos, asociados a un cuerpo rígido, representados por $\Sigma_0(x_0, y_0, z_0)$ y $\Sigma_1(x_1, y_1, z_1)$, ambos sistemas comparten el mismo origen. El sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ mantiene una orientación relativa al sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$ [25].

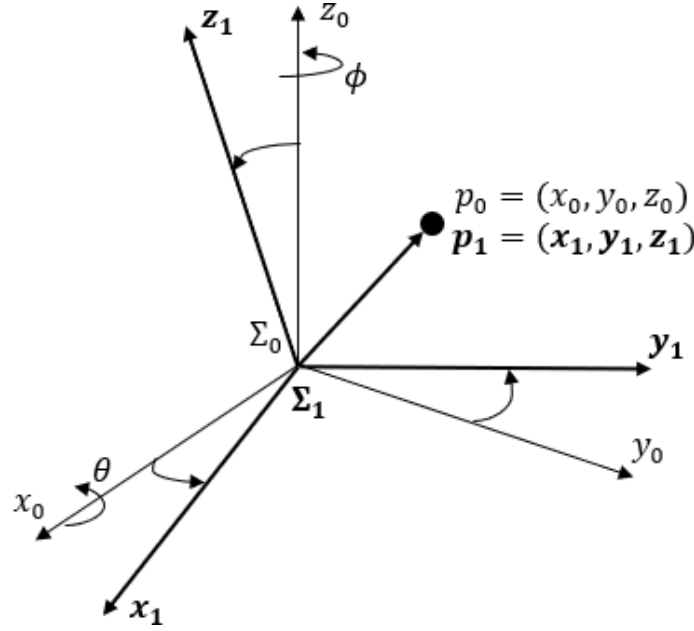


Figura 1. Sistemas de referencia fijo Σ_0 y rotado Σ_1 .

El vector $\mathbf{p} \in \mathbb{R}^{3 \times 1}$ contiene las coordenadas de un punto específico del cuerpo rígido que se muestra en la figura 1, con respecto al sistema $\Sigma_0(x_0, y_0, z_0)$ tiene coordenadas $\mathbf{p}_0 = [x_0, y_0, z_0]^T \in \mathbb{R}^{3 \times 1}$. El mismo vector $\mathbf{p} \in \mathbb{R}^{3 \times 1}$ con respecto al sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ adquiere una representación de $\mathbf{p}_1 = [x_1, y_1, z_1]^T \in \mathbb{R}^{3 \times 1}$. Para encontrar la relación que hay entre las coordenadas de $\mathbf{p}_1$ del sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ con el vector $\mathbf{p}_0$ definido en el sistema de referencia $\Sigma_0(x_0, y_0, z_0)$, se considera los vectores bases para cada sistema de referencia. Estos son $\{\mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0\}$ vectores unitarios a lo largo de los ejes x_0, y_0, z_0 , respectivamente. Es decir, $\mathbf{i}_0 = [1, 0, 0]^T$, $\mathbf{j}_0 = [0, 1, 0]^T$, $\mathbf{k}_0 = [0, 0, 1]^T$. Similarmente para el sistema de coordenadas $\Sigma_1(x_1, y_1, z_1)$ se definen sus vectores unitarios $\{\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1\}$.

Un vector que va desde el origen común para ambos sistemas hasta $\mathbf{p}$ puede expresarse en función de una de las dos bases de vectores unitarios de la siguiente forma [25],[26]:

$$\mathbf{p}_0 = p_{0x}\mathbf{i}_0 + p_{0y}\mathbf{j}_0 + p_{0z}\mathbf{k}_0 \text{ con respecto al sistema } \Sigma_0 \quad (1)$$

$$\mathbf{p}_1 = p_{1x}\mathbf{i}_1 + p_{1y}\mathbf{j}_1 + p_{1z}\mathbf{k}_1 \text{ con respecto al sistema } \Sigma_1 \quad (2)$$

Los vectores $\mathbf{p}_0, \mathbf{p}_1 \in \mathbb{R}^{3 \times 1}$ representan al mismo vector $\mathbf{p} \in \mathbb{R}^{3 \times 1}$. Tomando en cuenta las ecuaciones (1) y (2) la relación que hay entre sus componentes adquiere la siguiente forma (proyección del vector $\mathbf{p}_1$ sobre cada uno de los ejes del sistema $\Sigma_0(x_0, y_0, z_0)$ [25]:

$$\mathbf{p}_{0x} = p_0 \cdot \mathbf{i}_0 = p_1 \cdot \mathbf{i}_0 = p_{1x}\mathbf{i}_1 \cdot \mathbf{i}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{i}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{i}_0 \quad (3)$$

$$\mathbf{p}_{0y} = p_0 \cdot \mathbf{j}_0 = p_1 \cdot \mathbf{j}_0 = p_{1x}\mathbf{i}_1 \cdot \mathbf{j}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{j}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{j}_0 \quad (4)$$

$$\mathbf{p}_{0z} = p_0 \cdot \mathbf{k}_0 = p_1 \cdot \mathbf{k}_0 = p_{1x}\mathbf{i}_1 \cdot \mathbf{k}_0 + p_{1y}\mathbf{j}_1 \cdot \mathbf{k}_0 + p_{1z}\mathbf{k}_1 \cdot \mathbf{k}_0 \quad (5)$$

Estas ecuaciones pueden ser escritas de manera compacta como:

$$\mathbf{p}_0 = R_0^1 \mathbf{p}_1 \quad (6)$$

donde $R_0^1 \in \mathbb{R}^{3 \times 3}$ representa la siguiente matriz:

$$R_0^1 = \begin{bmatrix} i_1 \cdot i_0 & j_1 \cdot i_0 & k_1 \cdot i_0 \\ i_1 \cdot j_0 & j_1 \cdot j_0 & k_1 \cdot j_0 \\ i_1 \cdot k_0 & j_1 \cdot k_0 & k_1 \cdot k_0 \end{bmatrix} \quad (7)$$

La matriz R_0^1 es la matriz de transformación de las coordenadas del punto $\mathbf{p}$ del sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ hacia las coordenadas del sistema $\Sigma_0(x_0, y_0, z_0)$. Esto significa que dado un punto $\mathbf{p}_1$ en el sistema $\Sigma_1(x_1, y_1, z_1)$ entonces $R_0^1 \mathbf{p}_1$ representa el mismo vector expresado con respecto al sistema de referencia $\Sigma_0(x_0, y_0, z_0)$ [25].

Similarmente se puede obtener la proyección de $\mathbf{p}_1$ sobre el sistema $\Sigma_0(x_0, y_0, z_0)$ (es decir, el proceso inverso, $\mathbf{p}_1$ en función de $\mathbf{p}_0$). Entonces $\mathbf{p}_0 = R_0^1 \mathbf{p}_1$, donde R_0^1 es:

$$R_0^1 = \begin{bmatrix} i_0 \cdot i_1 & j_0 \cdot i_1 & k_0 \cdot i_1 \\ i_0 \cdot j_1 & j_0 \cdot j_1 & k_0 \cdot j_1 \\ i_0 \cdot k_1 & j_0 \cdot k_1 & k_0 \cdot k_1 \end{bmatrix} \quad (8)$$

La ecuación (8), satisface la siguiente propiedad:

Propiedad 1. [25] La matriz $R_0^1 \in \mathbb{R}^{3 \times 3}$ representa la matriz inversa de la transformación $R_0^1 \in \mathbb{R}^{3 \times 3}$. Debido a que el producto interno de vectores unitarios cumple la propiedad conmutativa, entonces $\mathbf{i}_1 \cdot \mathbf{j}_1 = \mathbf{j}_1 \cdot \mathbf{i}_1$, $\mathbf{i}_0 \cdot \mathbf{j}_0 = \mathbf{j}_0 \cdot \mathbf{i}_0$, etc., por lo que resulta:

$$R_0^1 = (R_0^1)^{-1} = (R_0^1)^T \quad (9)$$

2.2 Matriz de rotación alrededor del eje z_0

En la figura 2 se muestra al sistema de referencia $\Sigma_1 = (x_1, y_1, z_1)$ el cual esta rotado un ángulo θ alrededor del eje z_0 del sistema $\Sigma_0 = (x_0, y_0, z_0)$. El signo del ángulo θ está dado por la regla de la mano derecha. Por convención, un ángulo positivo es aquel cuyo sentido de rotación es contrario al movimiento de las manecillas del reloj. También se observa que los ejes z_0 y z_1 son paralelos [25], [26].

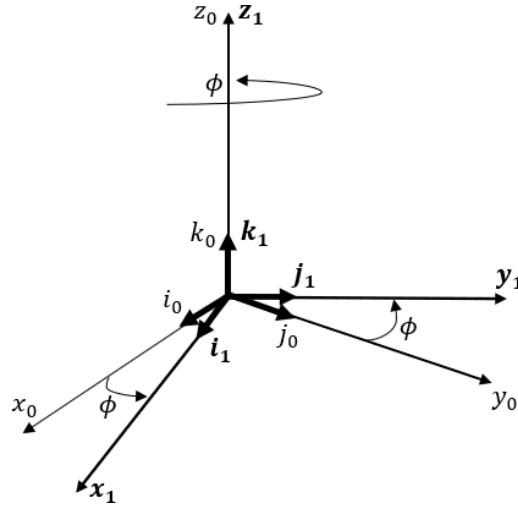


Figura 2. Rotación de un ángulo ϕ alrededor del eje z_0 .

La matriz $R_0^1 \in \mathbb{R}^{3 \times 3}$ correspondiente de rotación ϕ alrededor del eje z_0 se denota por $R_z(\phi) \in \mathbb{R}^{3 \times 3}$ con la siguiente estructura matemática [25],[26]:

$$R_z = \begin{bmatrix} \cos(\phi) & -\text{sen}(\phi) & 0 \\ \text{sen}(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Los ejes z_0 y z_1 coinciden entre sí, y el plano $x_1 - y_1$ se desplaza un ángulo ϕ de derecha a izquierda con respecto al plano $x_0 - y_0$.

2.3 Matriz de rotación alrededor del eje x_0

Si en la figura 3 en lugar de tener una rotación de un ángulo θ alrededor del eje z_0 , se tiene en el eje x_0 del sistema $\Sigma_0 = (x_0, y_0, z_0)$, entonces la matriz R_0^1 correspondiente de rotación θ alrededor del eje x_0 se denota por $R_x(\theta) \in \mathbb{R}^{3 \times 3}$ con la siguiente estructura matemática [25], [26]:

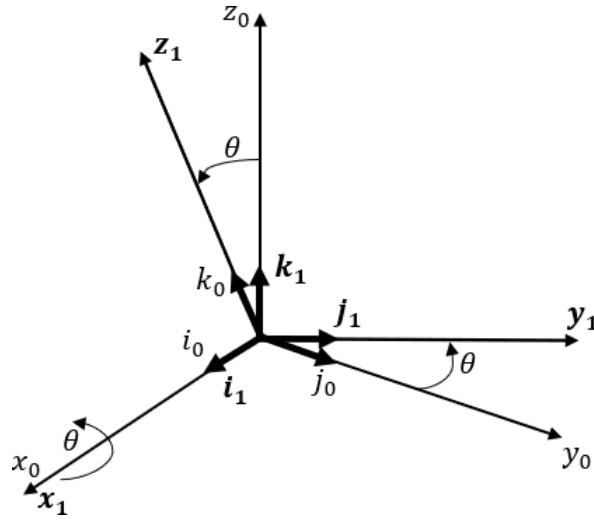


Figura 3. Rotación de un ángulo θ alrededor del eje x_0 .

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\text{sen}(\theta) \\ 0 & \text{sen}(\theta) & \cos(\theta) \end{bmatrix} \quad (11)$$

El ángulo de rotación θ gira alrededor del eje x_0 en sentido positivo (contrario a las manecillas del reloj). Los ejes x_0 y x_1 coinciden entre sí, y el plano $z_1 - y_1$ se desplaza un ángulo θ de derecha a izquierda con respecto al plano $z_0 - y_0$.

2.4 Matriz de rotación alrededor del eje y_0

Si en la figura 4 en lugar de tener una rotación de un ángulo ψ alrededor del eje z_0 ,

se tiene en el eje y_0 del sistema $\Sigma_0 = (x_0, y_0, z_0)$, entonces la matriz R_0^1 correspondiente de rotación ψ alrededor del eje y_0 se denota por $R_y(\psi) \in \mathbb{R}^{3 \times 3}$ con la siguiente estructura matemática [25], [26]:

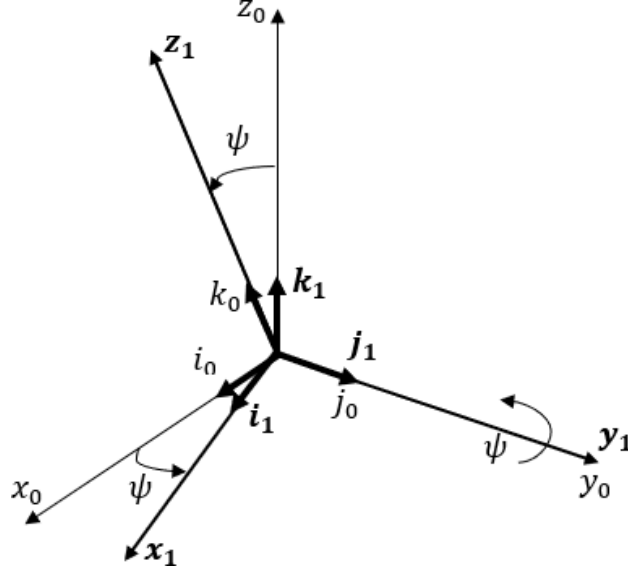


Figura 4. Rotación de un ángulo ψ alrededor del eje y_0 .

$$R_y = \begin{bmatrix} \cos(\psi) & 0 & -\text{sen}(\psi) \\ 0 & 1 & 0 \\ -\text{sen}(\psi) & 0 & \cos(\psi) \end{bmatrix} \quad (12)$$

Los ejes y_0 y y_1 coinciden entre sí, y el plano $z_1 - x_1$ se desplaza un ángulo ψ de derecha a izquierda con respecto al plano $z_0 - x_0$. El ángulo de rotación ψ gira alrededor del eje y_0 en sentido positivo (contrario a las manecillas del reloj).

Para una matriz de rotación $R_u(\alpha)$, donde $u = x, y, z$; α y β son los ángulos de rotación alrededor de u , se tienen varias propiedades importantes que a continuación se presentan:

Propiedad 2. [25] $R_u(0) = I$, $I \in \mathbb{R}^{3 \times 3}$ es la matriz identidad.

Propiedad 3. [25] $R_u(\alpha)R_u(\beta) = R_u(\beta)R_u(\alpha) = R_u(\alpha + \beta) = R_u(\alpha + \theta) \in \mathbb{R}^{3 \times 3}$.

Propiedad 4. [25] $R_u(\alpha)^{-1} = R_u(-\alpha) \in \mathbb{R}^{3 \times 3}$.

Propiedad 5. [25] $R_u(\alpha)^T = R_u(-\alpha) \in \mathbb{R}^{3 \times 3}$.

Propiedad 6. [25] $R_u(\alpha)R_u(\alpha)^T = R_u(\alpha)^TR_u(\alpha) = I \in \mathbb{R}^{3 \times 3}$.

Propiedad 7. [25] $\det[R_u(\theta)] = 1$, si el sistema de referencia cartesiano $\Sigma = (x, y, z)$ es seleccionado por la regla de la mano derecha, en otro caso $\det[R_u(\theta)] = -1$.

2.5 Matriz antisimétrica

Una matriz antisimétrica, es una matriz cuadrada, que es representada por $S \in \mathbb{R}^{3 \times 3}$; se define como [26], [28]:

$$S = \begin{bmatrix} 0 & -s_{1,2} & s_{1,3} \\ s_{2,1} & 0 & -s_{2,3} \\ -s_{3,1} & s_{3,2} & 0 \end{bmatrix} \quad (13)$$

donde

$s_{i,j} \in \mathbb{R}$ $i, j = 1, 2, 3$ son los elementos de la matriz S . Todos los elementos de la diagonal son $s_{i,i} = 0$, y fuera de la diagonal $s_{i,j}$ para $i \neq j$.

Los nueve elementos de la matriz S , se reducen a 3 elementos independientes, que denotamos por $s_{1,2}, s_{1,3}, s_{2,3} \in \mathbb{R}$, los cuales son los elementos de un vector.

La matriz antisimétrica S puede aceptar como argumento de entrada a vectores $\mathbf{p} \in \mathbb{R}^{3 \times 1}$ con tres coordenadas de los sistemas de referencia cartesianos tridimensionales: función matriz antisimétrica $S = S(\mathbf{p})$.

2.6 Propiedades de las matrices antisimétrica

Propiedad 8. [26], [28] Una matriz antisimétrica $S(\mathbf{p})$ satisface lo siguiente $\forall \mathbf{p} \in \mathbb{R}^{3 \times 1}$:

$$S(\mathbf{p}) + S(\mathbf{p})^T = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} + \begin{bmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

Propiedad 9. [26], [28] El producto de $S(\mathbf{p})S(\mathbf{p})^T$ es conmutativo, $S(\mathbf{p})^T S(\mathbf{p}) = -S(\mathbf{p})^2$; y el resultado es una matriz simétrica.

Propiedad 10. [26], [28] El determinante del producto de matrices antisimétricas $S(\mathbf{p})^T S(\mathbf{p}) = S(\mathbf{p})S(\mathbf{p})^T$ es cero. También se cumple que $\det[S(\mathbf{p})] = 0$ y $\det[S(\mathbf{p})] = 0$. Es decir, dada una matriz antisimétrica $S(\mathbf{p})$, no existen las siguientes matrices inversas, $S(\mathbf{p})^{-1}$, $[S(\mathbf{p})S(\mathbf{p})^T]^{-1}$ y $[S(\mathbf{p})^T S(\mathbf{p})]^{-1}$.

Propiedad 11. [26], [28] *Propiedad de escalamiento de $S(\mathbf{p})$: dada una matriz antisimétrica $S \in \mathbb{R}^{3 \times 3}$, el vector $\mathbf{p} \in \mathbb{R}^{3 \times 1}$, y el escalar $\alpha \in \mathbb{R}$, se cumple:*

$$S(\alpha \mathbf{p}) = \alpha S(\mathbf{p}) \quad (15)$$

Propiedad 12. [26], [28] *La derivada parcial de una matriz de rotación cumple con: $\frac{\partial}{\partial \theta} R(\theta) = S(\mathbf{n})R(\theta)$, donde $\mathbf{n} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario; cada matriz de rotación queda expresada de la siguiente forma:*

- $\frac{\partial}{\partial \theta} R_x(\theta) = S(\theta \mathbf{i})R(\theta)$, donde $\mathbf{i} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{i} = [1, 0, 0]^T$.
- $\frac{\partial}{\partial \theta} R_y(\theta) = S(\theta \mathbf{j})R(\theta)$, donde $\mathbf{j} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{j} = [0, 1, 0]^T$.
- $\frac{\partial}{\partial \theta} R_z(\theta) = S(\theta \mathbf{k})R(\theta)$, donde $\mathbf{k} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{k} = [0, 0, 1]^T$.

Propiedad 13. [26], [28] *La derivada con respecto al tiempo de una matriz de rotación cumple con: $\dot{R}(\theta) = \frac{d}{dt} R(\theta) = S(\boldsymbol{\omega}_0)R(\theta) = S(\dot{\theta} \mathbf{n})R(\theta)$, donde $\mathbf{n} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario, $\boldsymbol{\omega}_0 = \dot{\theta} \mathbf{n}$, $\theta = \theta(t)$; cada matriz de rotación queda expresada de la siguiente forma:*

- $\dot{R}_x(\theta) = \frac{d}{dt} R_x(\theta) = S(\boldsymbol{\omega}_{0x})R_x(\theta) = S(\dot{\theta} \mathbf{i})R_x(\theta)$, donde $\mathbf{i} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{i} = [1, 0, 0]^T$, $\boldsymbol{\omega}_{0x} = \dot{\theta} \mathbf{i}$, $\theta = \theta(t)$.
- $\dot{R}_y(\theta) = \frac{d}{dt} R_y(\theta) = S(\boldsymbol{\omega}_{0y})R_y(\theta) = S(\dot{\theta} \mathbf{j})R_y(\theta)$, donde $\mathbf{j} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{j} = [0, 1, 0]^T$, $\boldsymbol{\omega}_{0y} = \dot{\theta} \mathbf{j}$, $\theta = \theta(t)$.
- $\dot{R}_z(\theta) = \frac{d}{dt} R_z(\theta) = S(\boldsymbol{\omega}_{0z})R_z(\theta) = S(\dot{\theta} \mathbf{k})R_z(\theta)$, donde $\mathbf{k} \in \mathbb{R}^{3 \times 1}$ es un vector base unitario $\mathbf{k} = [0, 0, 1]^T$, $\boldsymbol{\omega}_{0z} = \dot{\theta} \mathbf{k}$, $\theta = \theta(t)$.

2.7 Transformaciones de traslación y rotación

Se tiene un sistema de referencia cartesiano fijo $\Sigma_0 = (x_0, y_0, z_0)$ y el sistema de referencia $\Sigma_1 = (x_1, y_1, z_1)$, donde sus respectivos orígenes son no coincidentes. El origen del sistema de referencia $\Sigma_1 = (x_1, y_1, z_1)$ se encuentra desplazado una distancia $d_0^1 \in \mathbb{R}^{n \times 1}$ con respecto al origen del sistema $\Sigma_0 = (x_0, y_0, z_0)$, como se observa en la figura 5:

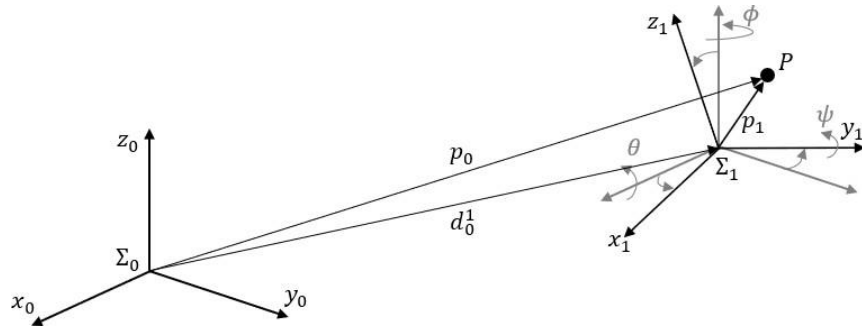


Figura 5. Transformaciones de traslación y rotación del sistema Σ_1 con respecto al sistema Σ_0 .

Observando la figura 5, se considera un punto arbitrario en el espacio P , el cual tiene representación $\mathbf{p}_0$ y $\mathbf{p}_1$. Donde $\mathbf{p}_0$ es el vector de coordenadas de P con respecto al sistema de referencia $\Sigma_0(x_0, y_0, z_0)$. Además, se considera otro sistema de referencia $\Sigma_1(x_1, y_1, z_1)$, donde $\mathbf{p}_1$ es el vector de coordenadas de P con respecto a Σ_1 . El vector $\mathbf{d}_0^1$ representa el desplazamiento del sistema Σ_1 con respecto al sistema Σ_0 , es decir: $\mathbf{d}_0^1 = [d_{0x}^1, d_{0y}^1, d_{0z}^1]^T$. Por otro lado, se considera R_0^1 como la matriz de rotación del sistema Σ_1 con respecto al sistema Σ_0 .

Por lo tanto, la posición del punto P con respecto al sistema de referencia Σ_0 puede ser expresado como en (16).

$$\mathbf{p}_0 = \mathbf{d}_0^1 + R_0^1 \mathbf{p}_1 = \begin{bmatrix} d_{0x}^1 \\ d_{0y}^1 \\ d_{0z}^1 \end{bmatrix} + R_0^1 \begin{bmatrix} p_{0x}^1 \\ p_{0y}^1 \\ p_{0z}^1 \end{bmatrix} \quad (16)$$

La expresión (16) representa las coordenadas de transformación que integran la traslación y rotación de un vector entre dos sistemas de referencia. Estas transformaciones de traslación y rotación se pueden llevar para un caso general, por ejemplo, si a la figura se le agrega un tercer sistema de referencia cartesiano, teniendo $\Sigma_0(x_0, y_0, z_0)$, $\Sigma_1(x_1, y_1, z_1)$ y $\Sigma_2(x_2, y_2, z_2)$, se obtienen las siguientes expresiones:

$$\mathbf{p}_0 = \mathbf{d}_0^1 + R_0^1 \mathbf{p}_1 \quad (17)$$

$$\mathbf{p}_1 = \mathbf{d}_1^2 + R_1^2 \mathbf{p}_2 \quad (18)$$

Reemplazando (18) en (17), se tiene:

$$\mathbf{p}_0 = \mathbf{d}_0^1 + R_0^1 (\mathbf{d}_1^2 + R_1^2 \mathbf{p}_2) \quad (19)$$

Por lo tanto, la regla de transformación de traslación y orientación de un punto $\mathbf{p}_2$ en el sistema $\Sigma_2(x_2, y_2, z_2)$ hacia un punto $\mathbf{p}_0$ en el sistema $\Sigma_0(x_0, y_0, z_0)$ adquiere la siguiente estructura:

$$\mathbf{p}_0 = \mathbf{d}_0^2 + R_0^2 \mathbf{p}_2 \quad (20)$$

donde:

$$R_0^2 = R_0^1 R_1^2 \quad (21)$$

$$\mathbf{d}_0^2 = \mathbf{d}_0^1 + R_0^1 \mathbf{d}_1^2 \quad (22)$$

La ecuación (16), puede ser también expresada con los ángulos de Euler $\mathbf{p}_0 = \mathbf{d}_0^1 + R_0^1(\phi, \theta, \psi) \mathbf{p}_1$, la cual es conocida como ecuación fundamental de la cinemática [25].

3. Modelo dinámico del robot Bioloid Premium

Se propone generar cuatro cadenas cinemáticas abiertas para describir la posición y orientación de cada uno de los eslabones del robot Bioloid Premium, considerando el torso del robot como sistema de referencia inicial $\Sigma_0(x_0, y_0, z_0)$. A partir del sistema inicial, la primera cadena cinemática considera como sistema de referencia final el pie derecho $\Sigma_{d7}(x_{d7}, y_{d7}, z_{d7})$, la segunda cadena cinemática toma como sistema de referencia final la pierna izquierda $\Sigma_{i7}(x_{i7}, y_{i7}, z_{i7})$, la tercera cadena tiene a $\Sigma_{bd4}(x_{bd4}, y_{bd4}, z_{bd4})$ como sistema de referencia final, el cual incluye el brazo derecho, y la cuarta cadena tiene como sistema de referencia final el brazo izquierdo $\Sigma_{bi4}(x_{bi4}, y_{bi4}, z_{bi4})$. En la figura 6 se muestra el modelo virtual del robot Bioloid Premium, y en la figura 7 se representan los sistemas de referencia asignados a cada una de las articulaciones del robot.

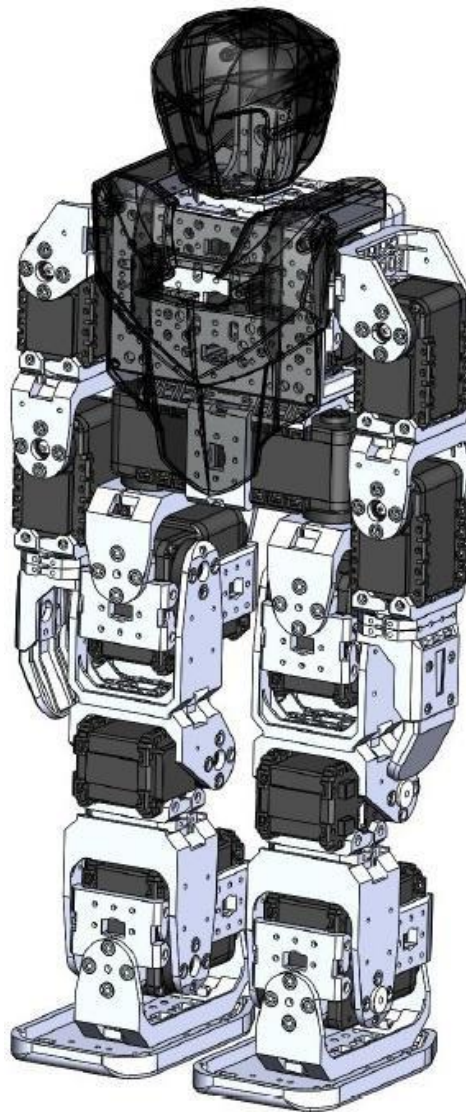


Figura 6. Modelo virtual del robot Bioloid Premium.

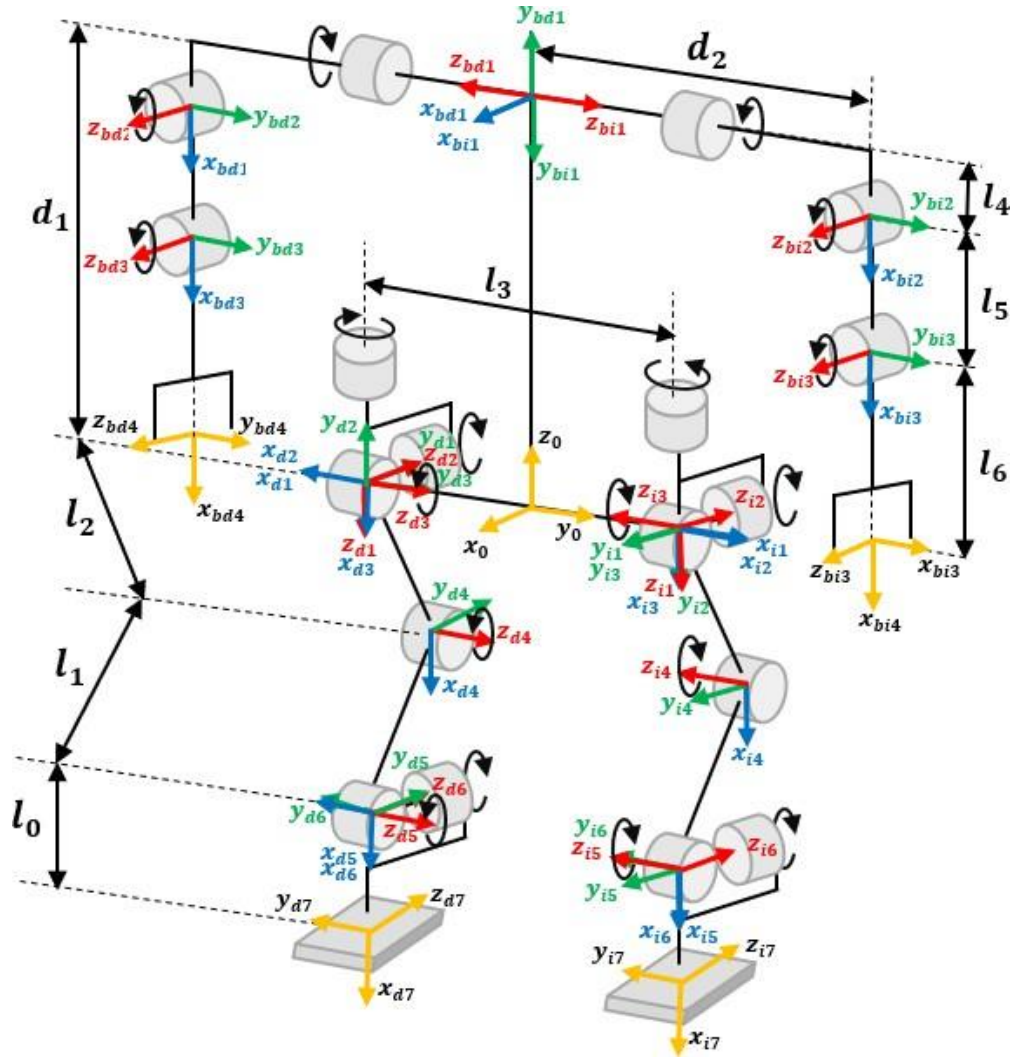


Figura 7. Sistemas de referencia asignados a las articulaciones del robot Bioloid.

Las dimensiones del robot son las siguientes: $l_0 = 33 \text{ mm}$, $d_1 = 118 \text{ mm}$, $d_2 = 73 \text{ mm}$, $l_1 = l_2 = 76 \text{ mm}$, $l_3 = 77 \text{ mm}$, $l_4 = l_6 = 108 \text{ mm}$, $l_5 = 66 \text{ mm}$.

3.1 Cinemática directa recursiva

Utilizando la convención de Denavit–Hartenberg y la correspondiente transformación de coordenadas [31]. Con el fin de calcular la posición cartesiana de cada eslabón del robot humanoide, se adopta una formulación recursiva de la cinemática directa. Sea $\mathbf{p}_i \in \mathbb{R}^3$ el vector de posición del eslabón i con respecto al sistema de referencia base. Cada eslabón está conectado mediante una rotación alrededor del eje z , seguida de una rotación alrededor del eje x , y un vector de desplazamiento que representa los parámetros geométricos del robot.

La orientación del eslabón i con respecto al sistema de referencia base se define como el producto acumulado de las rotaciones elementales de las articulaciones precedentes, como se muestra en (23):

$$R_0^i = \prod_{k=1}^i [R_{z_k} R_{x_k}] \quad (23)$$

donde R_{z_k} es la matriz de rotación alrededor del eje z del eslabón k , R_{x_k} es la matriz de rotación alrededor del eje x del eslabón k .

Para mayor claridad computacional, la rotación acumulada puede expresarse de forma recursiva, como se indica en (24):

$$R_0^i = R_0^{i-1} R_{z_i} R_{x_i} \quad (24)$$

donde $R_0^0 = I$, e I denota la matriz identidad. Esta recurrencia propaga la orientación desde la base hasta el efector final mediante multiplicaciones sucesivas de matrices.

Sea $\mathbf{d}_i \in \mathbb{R}^3$ el vector de desplazamiento desde el sistema de referencia $i - 1$ hasta el sistema i , expresado en el sistema de coordenadas local del eslabón i . La posición del eslabón i en el marco global se calcula de forma recursiva como en (25):

$$\mathbf{p}_i = \mathbf{p}_{i-1} + R_0^i R_{z_i} \mathbf{d}_i \quad (25)$$

con la condición inicial definida en (26):

$$\mathbf{p}_1 = R_{z_1} \mathbf{d}_1 \quad (26)$$

donde $\mathbf{p}_i$ es la posición cartesiana del eslabón i en el sistema de referencia base, $\mathbf{p}_{i-1}$ es la posición del eslabón anterior desde la cual se propaga la posición actual, R_0^i es la rotación total del eslabón i con respecto al sistema de referencia base, y $\mathbf{d}_i$ es el desplazamiento local desde la articulación $i - 1$ hasta la articulación i , que contiene la geometría del eslabón (longitudes y desplazamientos).

Esta formulación construye la posición cartesiana de cada eslabón de manera incremental, avanzando a lo largo de la cadena cinemática.

Utilizando como ejemplo los parámetros mostrados en las Tablas 1 y 2 puede obtenerse la cinemática directa de cada eslabón del robot. Dependiendo de la configuración

del robot, es importante considerar el centro de masa en el parámetro $\mathbf{d}_i$, cuando sea necesario.

También es importante mencionar que $\mathbf{p}_0$ corresponde a las coordenadas de transformación desde el sistema de referencia base (pelvis) hasta el primer sistema de referencia de cada extremidad del robot.

Tabla 1. Parámetros para calcular la cinemática de cada eslabón de la pierna derecha.

Posición p_{i-1}	Matriz $R_{xi-1}(\phi)$	Matriz $R_{zi-1}(\theta)$	Desplazamiento d_{i-1}
p_0	$R_{x0} = R_{x0}(\pi)$	$R_{z0} = R_{z0}(-\pi/2)$	$d_0 = [l_3/2 \quad 0 \quad 0]^T$
p_1	$R_{x1} = R_{x1}(-\pi/2)$	$R_{z1} = R_{z1}(q_{d1})$	$d_1 = [0 \quad 0 \quad 0]^T$
p_2	$R_{x2} = R_{x2}(\pi/2)$	$R_{z2} = R_{z2}(q_{d2})$	$d_2 = [0 \quad 0 \quad 0]^T$
p_3	$R_{x3} = R_{x3}(0)$	$R_{z3} = R_{z3}(q_{d3})$	$d_3 = [l_2 \quad 0 \quad 0]^T$
p_4	$R_{x4} = R_{x4}(0)$	$R_{z4} = R_{z4}(q_{d4})$	$d_4 = [l_1 \quad 0 \quad 0]^T$
p_5	$R_{x5} = R_{x5}(-\pi/2)$	$R_{z5} = R_{z5}(q_{d5})$	$d_5 = [0 \quad 0 \quad 0]^T$
p_6	$R_{x6} = R_{x6}(0)$	$R_{z6} = R_{z6}(q_{d6})$	$d_6 = [l_0 \quad 0 \quad 0]^T$

Tabla 2. Parámetros para calcular la cinemática de cada eslabón de la pierna izquierda.

Posición p_{i-1}	Matriz $R_{xi}(\phi)$	Matriz $R_{zi}(\theta)$	Desplazamiento d_i
p_0	$R_{x0} = R_{x0}(\pi)$	$R_{z0} = R_{z0}(\pi/2)$	$d_0 = [l_3/2 \quad 0 \quad 0]^T$
p_1	$R_{x1} = R_{x1}(\pi/2)$	$R_{z1} = R_{z1}(q_{i1})$	$d_1 = [0 \quad 0 \quad 0]^T$
p_2	$R_{x2} = R_{x2}(-\pi/2)$	$R_{z2} = R_{z2}(q_{i2})$	$d_2 = [0 \quad 0 \quad 0]^T$
p_3	$R_{x3} = R_{x3}(0)$	$R_{z3} = R_{z3}(q_{i3})$	$d_3 = [l_2 \quad 0 \quad 0]^T$
p_4	$R_{x4} = R_{x4}(0)$	$R_{z4} = R_{z4}(q_{i4})$	$d_4 = [l_1 \quad 0 \quad 0]^T$
p_5	$R_{x5} = R_{x5}(\pi/2)$	$R_{z5} = R_{z5}(q_{i5})$	$d_5 = [0 \quad 0 \quad 0]^T$
p_6	$R_{x6} = R_{x6}(0)$	$R_{z6} = R_{z6}(q_{i6})$	$d_6 = [l_0 \quad 0 \quad 0]^T$

Tabla 3. Parámetros para calcular la cinemática de cada eslabón del brazo derecho.

Posición p_{i-1}	Matriz $R_{xi-1}(\phi)$	Matriz $R_{zi-1}(\theta)$	Desplazamiento d_{i-1}
p_0	$R_{x0} = R_{x0}(\pi/2)$	$R_{z0} = R_{z0}(0)$	$d_0 = [0 \ 0 \ d_1]^T$
p_1	$R_{x1} = R_{x1}(-\pi/2)$	$R_{z1} = R_{z1}(q_{d7})$	$d_1 = [l_4 \ 0 \ d_2]^T$
p_2	$R_{x2} = R_{x2}(0)$	$R_{z2} = R_{z2}(q_{d8})$	$d_2 = [l_5 \ 0 \ 0]^T$
p_3	$R_{x3} = R_{x3}(0)$	$R_{z3} = R_{z3}(q_{d9})$	$d_3 = [l_6 \ 0 \ 0]^T$

Tabla 4. Parámetros para calcular la cinemática de cada eslabón del brazo izquierdo.

Posición p_{i-1}	Matriz $R_{xi-1}(\phi)$	Matriz $R_{zi-1}(\theta)$	Desplazamiento d_{i-1}
p_0	$R_{x0} = R_{x0}(-\pi/2)$	$R_{z0} = R_{z0}(0)$	$d_0 = [0 \ 0 \ d_1]^T$
p_1	$R_{x1} = R_{x1}(\pi/2)$	$R_{z1} = R_{z1}(q_{i7})$	$d_1 = [l_4 \ 0 \ d_2]^T$
p_2	$R_{x2} = R_{x2}(0)$	$R_{z2} = R_{z2}(q_{i8})$	$d_2 = [l_5 \ 0 \ 0]^T$
p_3	$R_{x3} = R_{x3}(0)$	$R_{z3} = R_{z3}(q_{i9})$	$d_3 = [l_6 \ 0 \ 0]^T$

3.2 Componente lineal de la cinemática diferencial

La cinemática diferencial lineal del robot se obtiene derivando el vector de posición $\mathbf{p}_i$. Para ello, se utiliza la matriz antisimétrica $S \in \mathbb{R}^{3 \times 3}$ mostrada en la expresión (27) [31, 32].

$$S = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (27)$$

Utilizando las propiedades de la matriz S , la velocidad lineal de cada eslabón puede calcularse mediante la ecuación (28):

$$\mathbf{v}_i = \mathbf{v}_{i-1} + R_0^{i-1} S \mathbf{p}_i \dot{q}_i \quad (28)$$

donde: $\mathbf{v}_i = \dot{\mathbf{p}}_i \in \mathbb{R}^{3 \times 1}$ es la velocidad lineal del eslabón i , $\mathbf{v}_{i-1}$ es la velocidad lineal del eslabón $i - 1$, S es la matriz antisimétrica, $\dot{q}_i$ es la derivada temporal de q_i , que corresponde al ángulo θ de la articulación i , con la condición inicial: $\mathbf{v}_0 = [0 \ 0 \ 0]^T$. El componente angular de la cinemática diferencial se desarrolla en las expresiones subsecuentes y, junto con los términos lineales y angulares, se completa el modelo cinemático diferencial del robot.

3.3 Rapidez lineal recursiva

Para determinar la rapidez lineal de cada eslabón, se considera el vector transpuesto de la ecuación (28). La velocidad lineal se calcula elevando al cuadrado cada componente de la velocidad $\|v_i\|^2 = (v_i)^T(v_i)$, dando como resultado la expresión (29):

$$v_i^2 = (v_{i-1})^2 + p_i^T S^T S p_i \dot{q}_i^2 + 2 \sum_{j=1}^{i-1} \left[p_j^T S^T \left(\prod_{m=j}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \dot{q}_i \dot{q}_j \right] \quad (29)$$

3.4 Matriz de inercia recursiva

Para calcular la matriz de inercia $M(q)$ del robot, primero deben determinarse los Jacobianos $J_{v_0}^i \in \mathbb{R}^{3 \times n}$, donde q denota la posición articular de cada ángulo θ , n es el número de eslabones del robot e i va desde 1 hasta n . Se propone una expresión general utilizando las matrices de rotación $R_{z_i}(\theta)$ and $R_{x_i}(\phi)$, el vector de desplazamiento d_i , y la matriz antisimétrica S . Cada $J_{v_0}^i$ puede expresarse como en la ecuación (30).

$$J_{v_0}^i = J_{v_0}^{i-1} + T_i \quad (30)$$

Considerando $J_{v_0}^1 = T_1$, donde $T_i \in \mathbb{R}^{3 \times n}$ es una matriz que añade términos a cada columna de $J_{v_0}^{i-1}$, con la columna j de T_i denotada por (31):

$$T_{i(1:3,j)} = \prod_{k=0}^{j-1} (R_{z_k} R_{x_k}) S \prod_{k=j}^{i-1} (R_{z_k} R_{x_k}) p_i \quad (31)$$

Adicionalmente, si $k \geq i - 1$, entonces el término $\prod_{k=j}^{i-1} (R_{z_k} R_{x_k})$ se convierte en la matriz identidad.

Al multiplicar la transpuesta del jacobiano de velocidad lineal por el mismo jacobiano $(J_{v_0}^i)^T (J_{v_0}^i) \in \mathbb{R}^{n \times n}$, tiene la ecuación (32):

$$(J_{v_0}^i)^T (J_{v_0}^i) = (J_{v_0}^{i-1})^T J_{v_0}^{i-1} + (J_{v_0}^{i-1})^T T_i + T_i^T J_{v_0}^{i-1} + T_i^T T_i \quad (32)$$

Posteriormente, se obtiene el jacobiano de velocidad angular de cada eslabón del robot. Primero, se determinan las matrices de rotación R_0^i que contienen las componentes de velocidad angular utilizando la ecuación (33):

$$R_0^i = \prod_{j=0}^i [R_{z_j} R_{x_j}] \text{ para } i = 0, 1, \dots, n \quad (33)$$

donde $R_0^1 \in \mathbb{R}^{3 \times 3}$ y n denota el número de eslabones. Las velocidades angulares $\omega_{z_0}^1$ respecto al eje z se extraen multiplicando la matriz resultante R_0^1 por el vector $z(k) = [0 \ 0 \ 1]^T$. Esto genera los vectores de velocidad angular de cada eslabón respecto al eje z , como se muestra en (34):

$$\omega_{z_0}^1 = R_0^1 z(k) \quad (34)$$

donde $\omega_{z_0}^1 \in \mathbb{R}^{3 \times 1}$. Los vectores $\omega_{z_0}^1$ se utilizan para construir los jacobianos de velocidad angular $J_{\omega_0}^i$ para cada eslabón. La ecuación (35) define la matriz Jacobiana antes mencionada:

$$J_{\omega_0}^i = [\omega_{z_0}^1 \ \omega_{z_0}^2 \ \dots \ \omega_{z_0}^i] \quad (35)$$

donde $J_{\omega_0}^i \in \mathbb{R}^{3 \times n}$. Utilizando las ecuaciones (30) y (32), la expresión (35) define la matriz de inercia del robot:

$$M(q) = \sum_{i=1}^n [m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i)] \quad (36)$$

donde m_i es la masa de inercia del i -ésimo eslabón; $J_{v_0}^i \in \mathbb{R}^{3 \times n}$ es el Jacobiano de velocidad lineal; $J_{\omega_0}^i \in \mathbb{R}^{3 \times n}$ es el Jacobiano de velocidad angular; e I_i es el tensor de inercia. La matriz $M(q) \in \mathbb{R}^{n \times n}$ es simétrica y definida positiva. Recursivamente se tiene la ecuación (37):

$$M_i(q) = M_{i-1}(q) + m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \quad (37)$$

para $i = 2$ hasta n ; $M(q) \in \mathbb{R}^{n \times n}$ representa la matriz de inercia, la cual es simétrica y definida positiva; m_i es la masa del i -ésimo eslabón; $J_{v_0}^i \in \mathbb{R}^{3 \times n}$ es la matriz que incluye los parámetros de velocidad lineal; $J_{\omega_0}^i \in \mathbb{R}^{3 \times n}$ es la matriz que tiene los parámetros de velocidad angular; I_0 es el tensor de inercia. Además, se considera la condición inicial (38):

$$M_1(q) = m_1 (J_{v_0}^1)^T (J_{v_0}^1) + (J_{\omega_0}^1)^T I_1 (J_{\omega_0}^1) \quad (38)$$

3.5 Energía cinética

La energía cinética se determina empleando la ecuación (39):

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^n \left(\frac{1}{2} m_i \mathbf{v}_i^2 + \frac{1}{2} I_i \boldsymbol{\omega}_i^2 \right) = \frac{1}{2} \sum_{i=1}^n (m_i \mathbf{v}_i^T \mathbf{v}_i + \boldsymbol{\omega}_i^T I_i \boldsymbol{\omega}_i) \quad (39)$$

donde $\mathbf{v}_i$ es la velocidad lineal del i -ésimo eslabón; $\boldsymbol{\omega}_i$ es la velocidad angular; n es el número de articulaciones del robot; $\dot{\mathbf{q}} \in \mathbb{R}^n$ es el vector velocidades articulares; $\mathbf{q} \in \mathbb{R}^n$ es el vector de posiciones articulares; $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ es el vector de energía cinética. La cual puede ser expresada con una estructura matemática de forma cuadrática, tal como en (40).

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \quad (40)$$

3.6 Energía potencial

Para obtener la energía potencial del robot, primero se determinan las alturas del centro de masa de cada eslabón con respecto a los pies del robot, para eso, es necesario emplear los parámetros de cinemática directa de la Tabla 1 - Tabla 4, posteriormente estos vectores se multiplican por el vector $\mathbf{z}(\mathbf{k})^T = [0 \ 0 \ 1]$. Por lo tanto, al emplear los vectores $\mathbf{p}_i$ de la Tabla 1 se definen las ecuaciones (41) - (46) para calcular las alturas de los centros de masa de cada eslabón de las piernas del robot, es importante considerar los centros de masa en cada vector de desplazamiento $\mathbf{d}_i$.

$$h_1(q) = -\mathbf{z}(\mathbf{k})^T \mathbf{p}_6 \quad (41)$$

$$h_2(q) = -\mathbf{z}(\mathbf{k})^T \mathbf{p}_6 \quad (42)$$

$$h_3(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_3 \quad (43)$$

$$h_4(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_4 \quad (44)$$

$$h_5(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_5 \quad (45)$$

$$h_6(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_6 \quad (46)$$

Empleando los vectores $\mathbf{p}_i$ de la Tabla 3, se definen las expresiones (47), (48) y (49) para determinar las alturas de los centros de masa de los eslabones de cada brazo del robot.

$$h_7(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_1 \quad (47)$$

$$h_8(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_2 \quad (48)$$

$$h_9(q) = h_1(q) + \mathbf{z}(\mathbf{k})^T \mathbf{p}_3 \quad (49)$$

Por lo tanto, la energía potencial del robot se determina mediante la ecuación (50), considerando $i = 1, 2, 3, \dots, n$.

$$\mathbf{u}(q) = \sum_{i=1}^n [m_i g h_i(q)] \quad (50)$$

donde $\mathbf{u}(q) \in \mathbb{R}^n$, la constante g representa la aceleración de la gravedad; $h_i(q)$ es la altura del centro de masa del i -ésimo eslabón del robot.

3.7 Lagrangiano del robot

El Lagrangiano del robot se obtiene como la diferencia entre la energía cinética y la energía potencial, tal como se presenta en la ecuación (51):

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{u}(\mathbf{q}) \quad (51)$$

donde $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$. Al expandir la expresión anterior, la formulación resultante viene dada por la ecuación (52):

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^n \left(\frac{1}{2} [m_i (\mathbf{v}_i^T \mathbf{v}_i) + \boldsymbol{\omega}_i^T I_i \boldsymbol{\omega}_i] - m_i g h_i(q) \right) \quad (52)$$

donde $\boldsymbol{\omega}_i \in \mathbb{R}^{3 \times 1}$ representa la velocidad angular del centro de masa del i -ésimo eslabón; que se calcula utilizando cada una de las matrices $J_{\omega_0}^1$, como se definió en la ecuación (35). Así, la velocidad angular $\boldsymbol{\omega}_i$ se obtiene mediante la expresión (53):

$$\boldsymbol{\omega}_i = (J_{\omega_0}^1) \dot{\mathbf{q}} \quad (53)$$

3.8 Ecuaciones de Euler-Lagrange

Las ecuaciones de movimiento del robot, basadas en la formulación de Euler-Lagrange, están dadas por la expresión (54):

$$\tau_i = \frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q_i} \quad (54)$$

para $i = 1, 2, \dots, n$. Donde τ_i denota el par motor (torque) de entrada aplicado a la i -ésima articulación. Adicionalmente, en la ecuación (54) se debe agregar el término $f_f(\dot{q}_i) \in \mathbb{R}^n$ que denota el vector de fricciones articulares del robot.

La fricción presente en el robot se modela como una combinación lineal, como se muestra en la ecuación (55).

$$f_f(\dot{q}_i) = b_i \dot{q}_i + f_{ci} \text{signo}(\dot{q}_i) \quad (55)$$

donde los términos b y f_c representan la fricción viscosa y de Coulomb, respectivamente.

Para determinar desarrollar la ecuación (54) es conveniente utilizar los componentes de la ecuación (51) por separado, e implementar el desarrollo analítico de Euler. El cual se explica a continuación.

Primero se toma la derivada parcial del lagrangiano con respecto a la i -ésima velocidad articular, la cual se expresa en (56):

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} = \sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{q}_i} - \sum_{j=1}^n \frac{\partial \mathcal{U}_j(q)}{\partial \dot{q}_i} \quad (56)$$

para $i = 1, 2, 3, \dots, n$. Donde n es el número de articulaciones del robot.

Empleando las ecuaciones (39) y (50), se prosigue a calcular las derivadas parciales con respecto a la i -ésima velocidad articular de la energía cinética y potencial de del robot. La derivada parcial de la energía cinética se obtiene a partir de (57):

$$\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{q}_i} = \frac{1}{2} \left(\sum_{j=1}^n m_j \frac{\partial (\mathbf{v}_j^T \mathbf{v}_j)}{\partial \dot{q}_i} + \sum_{j=1}^n \frac{\partial (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j)}{\partial \dot{q}_i} \right) \quad (57)$$

Puesto que la energía potencial $h_j(q)$ solo depende de las posiciones articulares y no de las velocidades, entonces la derivada parcial con respecto a la i -ésima velocidad articular de la energía potencial es igual a 0:

$$\sum_{j=1}^n \frac{\partial \mathbf{u}_j(q)}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n m_j g \frac{\partial (h_j(q))}{\partial \dot{\mathbf{q}}_i} = \mathbf{0} \quad (58)$$

Retomando la ecuación (57), se determinan las derivadas parciales de la rapidez lineal $\mathbf{v}_j^T \mathbf{v}_j$ y de las velocidades angulares $\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j$; como m_j e I_j son constantes, se consideran las propiedades de la derivada para determinar las expresiones (59) y (60):

$$\sum_{j=1}^n m_j \frac{\partial (\mathbf{v}_j^T \mathbf{v}_j)}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n m_j \left[\left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \mathbf{v}_j + \mathbf{v}_j^T \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \quad (59)$$

Además,

$$\sum_{j=1}^n \frac{\partial (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j)}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n \left[\left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \quad (60)$$

De la expresión (59), se tiene:

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} = \left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \quad (61)$$

para $i = 1, 2, 3, \dots, j$ y $p_{j+1} = [0 \ 0 \ 0]^T$. Esto permite calcular todas las derivadas parciales de cada velocidad lineal para cada eslabón del robot.

Si $i > j$, entonces:

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} = [0 \ 0 \ 0]^T$$

Para determinar la derivada parcial de cada velocidad angular se consideran las

ecuaciones (33) y (34) para desarrollar la expresión (62):

$$\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} = R_0^i \mathbf{z}(\mathbf{k}) = \left(\prod_{m=0}^i [R_{z_m} R_{x_m}] \right) \mathbf{z}(\mathbf{k}) \quad (62)$$

Donde se considera $i = 1, 2, \dots, j$, para obtener las derivadas parciales de las velocidades angulares de cada articulación del robot. Es importante señalar que si $i > j$, entonces:

$$\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} = [0 \quad 0 \quad 0]^T$$

Entonces las ecuaciones (59) y (60) se pueden reescribir como se observa en las expresiones (63) y (64), respectivamente:

$$\sum_{j=1}^n m_j \frac{\partial (\mathbf{v}_j^T \mathbf{v}_j)}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n m_j \left[\left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right)^T \mathbf{v}_j + \mathbf{v}_j^T \left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right) \right] \quad (63)$$

$$\sum_{j=1}^n \frac{\partial}{\partial \dot{\mathbf{q}}_i} (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j) = \sum_{j=1}^n \left[(R_0^i \mathbf{z}(\mathbf{k}))^T I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j (R_0^i \mathbf{z}(\mathbf{k})) \right] \quad (64)$$

El siguiente paso es determinar la derivada temporal de las ecuaciones (63) y (64), las cuales se representan en las ecuaciones (65) y (66), respectivamente:

$$\sum_{j=1}^n m_j \frac{d}{dt} \left[\frac{\partial (\mathbf{v}_j^T \mathbf{v}_j)}{\partial \dot{\mathbf{q}}_i} \right] = \sum_{j=1}^n m_j \frac{d}{dt} \left[\left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right)^T \mathbf{v}_j + \mathbf{v}_j^T \left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right) \right] \quad (65)$$

$$\sum_{j=1}^n \frac{d}{dt} \left[\frac{\partial (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j)}{\partial \dot{\mathbf{q}}_i} \right] = \sum_{j=1}^n \frac{d}{dt} \left[(R_0^i \mathbf{z}(\mathbf{k}))^T I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j (R_0^i \mathbf{z}(\mathbf{k})) \right] \quad (66)$$

Entonces la derivada temporal de la ecuación (65) puede reescribirse en (67):

$$\sum_{j=1}^n m_j \frac{d}{dt} \left[\left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right)^T v_j \right] + \sum_{j=1}^n m_j \frac{d}{dt} \left[v_j^T \left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right) \right] \quad (67)$$

Empleando las propiedades de la derivada, el primer término de la ecuación (67) se puede representar con la expresión (68):

$$\sum_{j=1}^n m_j \frac{d}{dt} \left(\frac{\partial v_j^T}{\partial \dot{q}_i} \right) v_j + \sum_{j=1}^n m_j \left(\frac{\partial v_j^T}{\partial \dot{q}_i} \right) \frac{dv_j}{dt} \quad (68)$$

Desarrollando el segundo término de la ecuación (67), se tiene la expresión (69):

$$\sum_{j=1}^n m_j \frac{dv_j^T}{dt} \left(\frac{\partial v_j}{\partial \dot{q}_i} \right) + \sum_{j=1}^n m_j (v_j^T) \frac{d}{dt} \left(\frac{\partial v_j}{\partial \dot{q}_i} \right) \quad (69)$$

El término $\frac{d}{dt} \left[\frac{\partial v_j}{\partial \dot{q}_i} \right]$ de la ecuación la expresión (69) se puede expresar como en (70):

$$\frac{d}{dt} \left[\frac{\partial v_j}{\partial \dot{q}_i} \right] = \sum_{r=1}^n \left[\frac{\partial}{\partial \dot{q}_r} \left(\frac{\partial v_j}{\partial \dot{q}_i} \right) \ddot{q}_r + \frac{\partial}{\partial q_r} \left(\frac{\partial v_j}{\partial \dot{q}_i} \right) \dot{q}_r \right] \quad (70)$$

donde:

$$\frac{\partial}{\partial \dot{q}_r} \left(\frac{\partial v_j}{\partial \dot{q}_i} \right) \ddot{q}_r = [0 \quad 0 \quad 0]^T \quad (71)$$

Por lo tanto, el término $\frac{d}{dt} \left[\frac{\partial v_j}{\partial \dot{q}_i} \right]$ de la ecuación la expresión (69) se puede desarrollar mediante la ecuación (72):

$$\frac{d}{dt} \left[\frac{\partial v_j}{\partial \dot{q}_i} \right] = \sum_{r=1}^n \left[\frac{\partial}{\partial q_r} \left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S p_i \right) \dot{q}_r \right] \quad (72)$$

donde $i = 1, 2, \dots, j$

De la expresión (72), la operación $\frac{\partial}{\partial \mathbf{q}_r}$ se puede expresar multiplicando la matriz S por $R_{z_r}(\phi)$ en cada uno de los términos de la sumatoria.

Por otra parte, tomando el término $\frac{dv_j}{dt}$ de la expresión (68), se puede representar con la ecuación (73):

$$\frac{d\mathbf{v}_j}{dt} = \sum_{r=1}^j \left[\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r + \frac{\partial(\mathbf{v}_j)}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right] \quad (73)$$

donde:

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r = \left(\prod_{m=0}^{r-1} [R_{z_m} R_{x_m}] \right) S \mathbf{p}_r \ddot{\mathbf{q}}_r \quad (74)$$

Para $r = 1, 2, \dots, j$. Si $r > j$, entonces:

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_r} = [0 \quad 0 \quad 0]^T$$

Además,

$$\frac{\partial(\mathbf{v}_j)}{\partial \mathbf{q}_r} = R_{z_0} R_{x_0} \sum_{u=1}^j \left(\prod_{m=1}^{u-1} [R_{z_m} R_{x_m}] S \prod_{n=u}^{r-1} [R_{z_n} R_{x_n}] \mathbf{p}_{\max(u,r)} \dot{\mathbf{q}}_u \right) \quad (75)$$

donde

- $\prod_{m=1}^0$ y $\prod_{n=u}^0$ representan la matriz identidad I .
- Además, si $u > r - 1$ entonces $\prod_{n=u}^{r-1}$ es la matriz identidad I .
- $\mathbf{p}_i$ es el vector de cinemática directa.

Es importante mencionar que es necesario multiplicar la matriz S por R_{z_r} en cada uno de los términos de la sumatoria de la ecuación (75) para obtener $\frac{\partial(\mathbf{v}_j)}{\partial \mathbf{q}_r}$.

Ahora se retoma la ecuación (66) y se aplican las propiedades de la derivada para obtener la expresión (76):

$$\sum_{j=1}^n \left[\left(\frac{\partial \omega_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \frac{d\omega_j}{dt} + \frac{d}{dt} \left(\frac{\partial \omega_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \omega_j \right] + \sum_{j=1}^n \left[\omega_j^T I_j \frac{d}{dt} \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) + \frac{d\omega_j^T}{dt} I_j \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \quad (76)$$

La derivada temporal de ω_j se puede determinar en (77):

$$\frac{d\omega_j}{dt} = \sum_{r=1}^j \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r \right) + \sum_{r=1}^j \left(\frac{\partial \omega_j}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right) \quad (77)$$

Si $r > j$, entonces:

$$\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_r} = \frac{\partial \omega_j}{\partial \mathbf{q}_r} = [0 \quad 0 \quad 0]^T$$

Cada uno de los elementos de la ecuación (77) se pueden reescribir en las ecuaciones (78) y (79):

$$\sum_{r=1}^j \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r \right) = \sum_{r=1}^j (R_0^r \mathbf{z}(\mathbf{k}) \ddot{\mathbf{q}}_r) \quad (78)$$

y

$$\sum_{r=1}^j \left(\frac{\partial \omega_j}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right) = \sum_{r=1}^j (D_r \dot{\mathbf{q}}_r) \quad (79)$$

donde:

$$\mathbf{D}_r = \sum_{m=r}^j \left(\prod_{n=0}^{r-1} [R_{z_n} R_{x_n}] S \left(\prod_{n=r}^j [R_{z_n} R_{x_n}] \right) \mathbf{z}(\mathbf{k}) \dot{\mathbf{q}}_m \right) \quad (80)$$

Por otra parte:

$$\frac{d}{dt} \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) = \frac{d}{dt} (R_0^i \mathbf{z}(\mathbf{k})) = \sum_{r=1}^j \left[\frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r + \frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right] \quad (81)$$

donde:

$$\frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \dot{\mathbf{q}}_r} \dot{\mathbf{q}}_r = [0 \quad 0 \quad 0]^T \quad (82)$$

Por lo tanto:

$$\frac{d}{dt} \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) = \sum_{r=1}^j \left[\frac{\partial}{\partial \mathbf{q}_r} \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) \dot{\mathbf{q}}_r \right] = \sum_{r=1}^j \left[\frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right] \quad (83)$$

$$= \sum_{r=1}^j \left[\prod_{m=0}^{r-1} [R_{z_m} R_{x_m}] S \left(\prod_{u=r}^i [R_{z_u} R_{x_u}] \right) \mathbf{z}(\mathbf{k}) \dot{\mathbf{q}}_r \right] \quad (84)$$

Si $j > i$ ó $r > i$ entonces:

$$\frac{d}{dt} \left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}_i} \right) = [0 \quad 0 \quad 0]^T \quad (85)$$

Como se mencionó anteriormente, la función $h_j(q)$ solo depende de las posiciones articulares, por lo que la expresión (86) es igual a cero, por lo tanto:

$$\sum_{j=1}^n m_j g \left[\frac{d}{dt} \left(\frac{\partial (h_j(q))}{\partial \dot{\mathbf{q}}_i} \right) \right] = 0 \quad (86)$$

Posteriormente, para determinar el término $\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \mathbf{q}_i}$ de la ecuación (54) es necesario obtener las derivadas parciales del lagrangiano, tal como se muestra en (87):

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \mathbf{q}_i} = \sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \mathbf{q}_i} - \sum_{j=1}^n \frac{\partial \mathcal{U}_j(q)}{\partial \mathbf{q}_i}; \text{ para } i = 1, 2, \dots, n \quad (87)$$

donde:

$$\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \mathbf{q}_i} = \frac{1}{2} \left(\sum_{j=1}^n m_j \frac{\partial}{\partial \mathbf{q}_i} (\mathbf{v}_j^T \mathbf{v}_j) + \sum_{j=1}^n \frac{\partial}{\partial \mathbf{q}_i} (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j) \right) \quad (88)$$

Para determinar el término $\frac{\partial}{\partial \mathbf{q}_i} (\mathbf{v}_j^T \mathbf{v}_j)$ se emplean nuevamente las propiedades de la derivada:

$$\frac{\partial}{\partial \mathbf{q}_i} (\mathbf{v}_j^T \mathbf{v}_j) = \frac{\partial (\mathbf{v}_j^T)}{\partial \mathbf{q}_i} \mathbf{v}_j + \mathbf{v}_j^T \frac{\partial (\mathbf{v}_j)}{\partial \mathbf{q}_i} \quad (89)$$

donde la derivada parcial $\frac{\partial (\mathbf{v}_j)}{\partial \mathbf{q}_i}$ se resuelve con la ecuación (75).

Para expresar el término $\frac{\partial}{\partial \mathbf{q}_i} (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j)$ en las propiedades de la derivada, tal como en (90):

$$\frac{\partial}{\partial \mathbf{q}_i} (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j) = \frac{\partial (\boldsymbol{\omega}_j^T)}{\partial \mathbf{q}_i} I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \frac{\partial (\boldsymbol{\omega}_j)}{\partial \mathbf{q}_i} \quad (90)$$

donde la operación $\frac{\partial (\boldsymbol{\omega}_j)}{\partial \mathbf{q}_i}$ se puede desarrollar mediante la ecuación (79).

Finalmente, la derivada parcial de la energía potencial con respecto a cada i-ésima posición articular, se expresa en (91):

$$\sum_{j=1}^n \frac{\partial \mathcal{U}_j(q)}{\partial \mathbf{q}_i} = \sum_{j=1}^n m_j g \frac{\partial (h_j(q))}{\partial \mathbf{q}_i} \quad (91)$$

La derivada parcial de $h_j(q)$ con respecto a $\mathbf{q}_i$ se determina con la expresión (92):

$$\frac{\partial (h_j(q))}{\partial \mathbf{q}_i} = \mathbf{z}(\mathbf{k})^T \prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] S p_i \quad (92)$$

Específicamente, para calcular las derivadas parciales de las alturas de los centros de masa de cada eslabón de las piernas, se propone la ecuación (93), considerando $j = 1, 2$:

$$\frac{\partial(h_j(q))}{\partial \mathbf{q}_i} = -\mathbf{z}(\mathbf{k})^T \prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] S p_i \quad (93)$$

Para determinar las derivadas parciales de las alturas restantes, tanto de las piernas como los brazos, se considera $j = 3, 4, 5, \dots, n$. en la ecuación (94):

$$\frac{\partial(h_j(q))}{\partial \mathbf{q}_i} = \frac{\partial(h_1(q))}{\partial \mathbf{q}_i} + \mathbf{z}(\mathbf{k})^T \prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] S p_i \quad (94)$$

Hay casos especiales, que se muestran en (95), (96) y (97):

$$\frac{\partial(h_3(q))}{\partial \mathbf{q}_5} = \frac{\partial(h_1(q))}{\partial \mathbf{q}_5} \quad (95)$$

$$\frac{\partial(h_3(q))}{\partial \mathbf{q}_6} = \frac{\partial(h_1(q))}{\partial \mathbf{q}_6} \quad (96)$$

$$\frac{\partial(h_4(q))}{\partial \mathbf{q}_6} = \frac{\partial(h_1(q))}{\partial \mathbf{q}_6} \quad (97)$$

Por lo tanto, las ecuaciones de movimiento de Euler-Lagrange del robot en (54) adquieren la forma de la ecuación (98).

$$\boldsymbol{\tau}_i = \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{\mathbf{q}}_i} - \sum_{j=1}^n \frac{\partial \mathcal{U}_j(q)}{\partial \dot{\mathbf{q}}_i} \right] - \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \mathbf{q}_i} - \sum_{j=1}^n \frac{\partial \mathcal{U}_j(q)}{\partial \mathbf{q}_i} \right] \quad (98)$$

donde:

$$\begin{aligned} \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{\mathbf{q}}_i} \right] &= \frac{1}{2} \sum_{j=1}^n \left[m_j \frac{d}{dt} \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \mathbf{v}_j + m_j \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \frac{d\mathbf{v}_j}{dt} + m_j \frac{d\mathbf{v}_j^T}{dt} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) + m_j (\mathbf{v}_j^T) \frac{d}{dt} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \\ &+ \frac{1}{2} \sum_{j=1}^n \left[\left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \frac{d\boldsymbol{\omega}_j}{dt} + \frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) + \frac{d\boldsymbol{\omega}_j^T}{dt} I_j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \end{aligned} \quad (99)$$

$$\frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathbf{u}_j(q)}{\partial \dot{\mathbf{q}}_i} \right] = 0 \quad (100)$$

Las derivadas parciales de la energía cinética y potencial se representan en (101) y (102), respectivamente:

$$\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \mathbf{q}_i} = \frac{1}{2} \left(\sum_{j=1}^n m_j \left[\left(\frac{\partial \mathbf{v}_j^T}{\partial \mathbf{q}_i} \right) \mathbf{v}_j + \mathbf{v}_j^T \left(\frac{\partial \mathbf{v}_j}{\partial \mathbf{q}_i} \right) \right] + \sum_{j=1}^n \left[\left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \mathbf{q}_i} \right) I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \mathbf{q}_i} \right) \right] \right) \quad (101)$$

$$\sum_{j=1}^n \frac{\partial \mathbf{u}_j(q)}{\partial \mathbf{q}_i} = \sum_{j=1}^n m_j g \mathbf{z}(\mathbf{k})^T \prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] S \mathbf{p}_i \quad (102)$$

Las derivadas parciales de la ecuación (101) se expresan en (103) y (104):

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} = \left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S \mathbf{p}_i \quad (103)$$

$$\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} = \left(\prod_{m=0}^i [R_{z_m} R_{x_m}] \right) \mathbf{z}(\mathbf{k}) \quad (104)$$

Por otro lado, las derivadas que involucran el vector de velocidades lineales $\mathbf{v}_j$ en la ecuación (99), se presentan en (105) - (108).

$$\frac{d}{dt} \left[\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right] = \sum_{r=1}^n \left[\frac{\partial}{\partial \mathbf{q}_r} \left(\left(\prod_{m=0}^{i-1} [R_{z_m} R_{x_m}] \right) S \mathbf{p}_i \right) \dot{\mathbf{q}}_r \right] \quad (105)$$

$$\frac{d\mathbf{v}_j}{dt} = \sum_{r=1}^j \left[\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r + \frac{\partial (\mathbf{v}_j)}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right] \quad (106)$$

$$\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_r} = \left(\prod_{m=0}^{r-1} [R_{z_m} R_{x_m}] \right) S \mathbf{p}_r \quad (107)$$

$$\frac{\partial(\mathbf{v}_j)}{\partial \mathbf{q}_r} = R_{z_0} R_{x_0} \sum_{u=1}^j \left(\prod_{m=1}^{u-1} [R_{z_m} R_{x_m}] S \prod_{n=u}^{r-1} [R_{z_n} R_{x_n}] \mathbf{p}_{\max(u,r)} \dot{q}_u \right) \quad (108)$$

Las derivadas que involucran el vector de velocidades angulares $\boldsymbol{\omega}_j$ en la ecuación (99), se presentan en (109) - (113).

$$\frac{d\boldsymbol{\omega}_j}{dt} = \sum_{r=1}^j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r \right) + \sum_{r=1}^j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right) \quad (109)$$

$$\sum_{r=1}^j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r \right) = \sum_{r=1}^j (R_0^r \mathbf{z}(\mathbf{k}) \ddot{\mathbf{q}}_r) \quad (110)$$

$$\sum_{r=1}^j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right) = \sum_{r=1}^j \left(\sum_{m=r}^j \left(\prod_{n=0}^{r-1} [R_{z_n} R_{x_n}] S \left(\prod_{n=r}^j [R_{z_n} R_{x_n}] \right) \mathbf{z}(\mathbf{k}) \dot{q}_m \right) \dot{\mathbf{q}}_r \right) \quad (111)$$

$$\frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) = \sum_{r=1}^j \left[\frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \dot{\mathbf{q}}_r} \ddot{\mathbf{q}}_r + \frac{\partial (R_0^i \mathbf{z}(\mathbf{k}))}{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r \right] \quad (112)$$

$$\frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) = \sum_{r=1}^j \left[\prod_{m=0}^{r-1} [R_{z_m} R_{x_m}] S \left(\prod_{u=r}^i [R_{z_u} R_{x_u}] \right) \mathbf{z}(\mathbf{k}) \dot{\mathbf{q}}_r \right] \quad (113)$$

Los parámetros empleados en el modelo dinámico del robot están enlistados en la Tabla 5.

Tabla 5. Parámetros de los eslabones del robot.

Eslabón	Significado	Notación
1 (Cadera derecha eje transversal)	Masa del eslabón 1	m_{d1}
	Inercia del eslabón 1	I_{d1}
	Posición articular del eslabón 1	q_{d1}
2 (Cadera derecha plano frontal)	Masa del eslabón 2	m_{d2}
	Inercia del eslabón 2	I_{d2}
	Posición articular del eslabón 2	q_{d2}
3 (Cadera derecha plano sagital)	Masa del eslabón 3	m_{d3}
	Inercia del eslabón 3	I_{d3}
	Posición articular del eslabón 3	q_{d3}
	Longitud del eslabón 3	l_2
	Centro de masa del eslabón 3	l_{c2}
4 (Rodilla derecha)	Masa del eslabón 4	m_{d4}
	Inercia del eslabón 4	I_{d4}
	Longitud del eslabón 4	l_1
	Centro de masa del eslabón 4	l_{c1}
	Posición articular del eslabón 4	q_{d4}
5 (Tobillo derecho plano sagital)	Masa del eslabón 5	m_{d5}
	Inercia del eslabón 5	I_{d5}
	Posición articular del eslabón 5	q_{d5}
6 (Tobillo derecho plano frontal)	Masa del eslabón 6	m_{d6}
	Inercia del eslabón 6	I_{d6}
	Posición articular del eslabón 6	q_{d6}
	Longitud del eslabón 6	l_0
	Centro de masa del eslabón 6	l_{c0}
7 (Cadera izquierda eje transversal)	Masa del eslabón 7	m_{i1}
	Inercia del eslabón 7	I_{i1}
	Posición articular del eslabón 7	q_{i1}
8 (Cadera izquierda plano frontal)	Masa del eslabón 8	m_{i2}
	Inercia del eslabón 8	I_{i2}
	Posición articular del eslabón 8	q_{i2}
9 (Cadera izquierda plano sagital)	Masa del eslabón 9	m_{i3}
	Inercia del eslabón 9	I_{i3}
	Posición articular del eslabón 9	q_{i3}
	Longitud del eslabón 9	l_2
	Centro de masa del eslabón 9	l_{c2}
10 (Rodilla izquierda)	Masa del eslabón 10	m_{i4}
	Inercia del eslabón 10	I_{i4}
	Longitud del eslabón 10	l_1
	Centro de masa del eslabón 10	l_{c1}

	Posición articular del eslabón 10	q_{i4}
11 (Tobillo izquierdo plano sagital)	Masa del eslabón 11	m_{i5}
	Inercia del eslabón 11	I_{i5}
	Posición articular del eslabón 11	q_{i5}
12 (Tobillo izquierdo plano frontal)	Masa del eslabón 12	m_{i6}
	Inercia del eslabón 12	I_{i6}
	Posición articular del eslabón 12	q_{i6}
	Longitud del eslabón 12	l_0
	Centro de masa del eslabón 12	l_{c0}
13 (Hombro derecho plano sagital)	Masa del eslabón 13	m_{d7}
	Inercia del eslabón 13	I_{d7}
	Longitud del eslabón 13	l_4
	Centro de masa del eslabón 13	l_{c4}
	Posición articular del eslabón 13	q_{d7}
14 (Hombro derecho plano frontal)	Masa del eslabón 14	m_{d8}
	Inercia del eslabón 14	I_{d8}
	Longitud del eslabón 14	l_5
	Centro de masa del eslabón 14	l_{c5}
	Posición articular del eslabón 14	q_{d8}
15 (Codo derecho)	Masa del eslabón 15	m_{d9}
	Inercia del eslabón 15	I_{d9}
	Longitud del eslabón 15	l_6
	Centro de masa del eslabón 15	l_{c6}
	Posición articular del eslabón 15	q_{d9}
16 (Hombro izquierdo plano sagital)	Masa del eslabón 16	m_{i7}
	Inercia del eslabón 16	I_{i7}
	Longitud del eslabón 16	l_4
	Centro de masa del eslabón 16	l_{c4}
	Posición articular del eslabón 16	q_{i7}
17 (Hombro izquierdo plano frontal)	Masa del eslabón 17	m_{i8}
	Inercia del eslabón 17	I_{i8}
	Longitud del eslabón 17	l_5
	Centro de masa del eslabón 17	l_{c5}
	Posición articular del eslabón 17	q_{i8}
18 (Codo izquierdo)	Masa del eslabón 18	m_{i9}
	Inercia del eslabón 18	I_{i9}
	Longitud del eslabón 18	l_6
	Centro de masa del eslabón 18	l_{c6}
	Posición articular del eslabón 18	q_{i9}

3.9 Propiedades del modelo dinámico

El modelo dinámico de un robot, particularmente un robot manipulador, se basa en las derivadas que involucran el lagrangiano del robot y la matriz de inercia. Estas ecuaciones describen cómo las fuerzas y los momentos afectan el movimiento del robot. Aquí están algunas de las propiedades fundamentales del modelo dinámico del robot:

Primero se demuestra la propiedad de la derivada de la matriz de inercia, esta expresión se puede representar en (114):

$$\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q})\ddot{\mathbf{q}} \quad (114)$$

Puesto que

$$\sum_{j=1}^n \frac{\partial \mathbf{u}_j(\mathbf{q})}{\partial \dot{\mathbf{q}}_i} = 0 \quad (115)$$

Entonces la parcial del lagrangiano se puede reescribir como (116):

$$\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q})\ddot{\mathbf{q}} \quad (116)$$

Reemplazando la energía cinética en su forma compacta se puede reducir a (118)

$$\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \frac{d}{dt} \left[\frac{1}{2} \frac{\partial (\dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q})\ddot{\mathbf{q}} \quad (117)$$

$$\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \frac{d}{dt} \left[\frac{1}{2} M(\mathbf{q}) \dot{\mathbf{q}} \right] - M(\mathbf{q})\ddot{\mathbf{q}} \quad (118)$$

La derivada de la matriz de inercia y del vector de velocidades, se obtiene con (119)

$$\frac{d}{dt} (M(\mathbf{q}) \dot{\mathbf{q}}) = M(\mathbf{q})\ddot{\mathbf{q}} + \dot{M}(\mathbf{q})\dot{\mathbf{q}} \quad (119)$$

Sustituyendo (119) en (118), se tiene:

$$M(\mathbf{q})\dot{\mathbf{q}} = [M(\mathbf{q})\ddot{\mathbf{q}} + M(\mathbf{q})\dot{\mathbf{q}}] - M(\mathbf{q})\ddot{\mathbf{q}} \quad (120)$$

Por lo tanto:

$$M(\mathbf{q})\dot{\mathbf{q}} = M(\mathbf{q})\dot{\mathbf{q}} \quad (121)$$

Posteriormente se demuestra la expresión (122), la cual toma en cuenta la parcial del lagrangiano con respecto a la velocidad.

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} = \frac{\partial}{\partial \dot{q}_i} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] = M(\mathbf{q})\dot{\mathbf{q}} \quad (122)$$

Ahora se reemplaza la forma compacta de la energía cinética, tal como se muestra en (123):

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} = \sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{q}_i} = \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \quad (123)$$

Simplificando ambos lados de la igualdad se obtiene la expresión (124).

$$\begin{aligned} \sum_{j=1}^n \left[m_j \frac{\partial (\mathbf{v}_j^T \mathbf{v}_j)}{\partial \dot{q}_i} + \frac{\partial (\boldsymbol{\omega}_j^T I_j \boldsymbol{\omega}_j)}{\partial \dot{q}_i} \right] &= \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \\ \frac{\partial}{\partial \dot{q}_i} \left(\sum_{j=1}^n \left[m_j (\dot{\mathbf{q}}^T (J_{v_0}^i)^T (J_{v_0}^i) \dot{\mathbf{q}}) + (\dot{\mathbf{q}}^T (J_{\omega_0}^i)^T I_j (J_{\omega_0}^i) \dot{\mathbf{q}}) \right] \right) &= \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \\ \frac{\partial}{\partial \dot{q}_i} \left(\sum_{j=1}^n \left[m_j \dot{\mathbf{q}}^T ((J_{v_0}^i)^T (J_{v_0}^i)) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T ((J_{\omega_0}^i)^T I_j (J_{\omega_0}^i)) \dot{\mathbf{q}} \right] \right) &= \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \\ \frac{\partial}{\partial \dot{q}_i} \left(\sum_{j=1}^n \dot{\mathbf{q}}^T \left[m_j (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_j (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \right) &= \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right] \dot{\mathbf{q}} \end{aligned}$$

$$\frac{\partial}{\partial \dot{q}_i} \left(\sum_{j=1}^n [m_j (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_j (J_{\omega_0}^i)] \dot{q} \dot{q} \right) = \sum_{i=1}^n [m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i)] \dot{q}$$

$$\sum_{j=1}^n [m_j (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_j (J_{\omega_0}^i)] \dot{q} = \sum_{i=1}^n [m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i)] \dot{q} \quad (124)$$

Por lo tanto:

$$M(q)\dot{q} = M(q)\dot{q} \quad (125)$$

Después se procede a demostrar la propiedad de conservación de la energía cinética y la correcta modelación de la matriz de Coriolis y centrífugas en el modelo dinámico del robot. Por lo cual, primero se toma la expresión (126)..

$$C(q, \dot{q})\dot{q} = \dot{M}(q)\dot{q} - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \quad (126)$$

Reemplazando la ecuación (117) en (126), se tiene:

$$C(q, \dot{q})\dot{q} = \frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} \right] - M(q)\ddot{q} - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \quad (127)$$

$$C(q, \dot{q})\dot{q} = \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(q, \dot{q})}{\partial \dot{q}_i} \right] - M(q)\ddot{q} - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \quad (128)$$

Por otro lado, se tiene la ecuación (129) que representa la propiedad de conservación de la energía cinética, que asegura que las fuerzas centrífugas y de Coriolis no afectan la energía cinética neta del sistema, garantizando así que el sistema no genera ni disipa energía interna debido a estas fuerzas.

$$N(q, \dot{q}) = M(q) - 2C(q, \dot{q}) \quad (129)$$

Para demostrar que se cumple la igualdad (130), se hace el desarrollo correspondiente:

$$\dot{\mathbf{q}}^T [N(\mathbf{q}, \dot{\mathbf{q}})] \dot{\mathbf{q}} \equiv 0 \quad (130)$$

Sustituyendo (129) en (130), se tiene lo siguiente:

$$\dot{\mathbf{q}}^T [M(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}})] \dot{\mathbf{q}} \equiv 0 \quad (131)$$

$$\dot{\mathbf{q}}^T [M(\mathbf{q}) \dot{\mathbf{q}} - 2C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}] \equiv 0 \quad (132)$$

Sustituyendo la (116) y (128) en (130) se tiene

$$\begin{aligned} \dot{\mathbf{q}}^T \left[\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q}) \ddot{\mathbf{q}} - 2 \left(\frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q}) \ddot{\mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] \right) \right] &\equiv 0 \\ \dot{\mathbf{q}}^T \left[\frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q}) \ddot{\mathbf{q}} - 2 \left(\frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - M(\mathbf{q}) \ddot{\mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \left[\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right] \right) \right] &\equiv 0 \end{aligned}$$

Eliminando términos, se tiene la ecuación (133):

$$\dot{\mathbf{q}}^T \left[\frac{\partial}{\partial \mathbf{q}} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] \right] \equiv 0 \quad (133)$$

La expresión (133) se puede desarrollar mediante las ecuaciones (134) y (135).

$$\begin{aligned} \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] &= \sum_{j=1}^n \left[m_j \frac{d}{dt} \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \mathbf{v}_j + m_j \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \frac{d\mathbf{v}_j}{dt} + m_j \frac{d\mathbf{v}_j^T}{dt} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) + m_j (\mathbf{v}_j^T) \frac{d}{dt} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \\ &+ \sum_{j=1}^n \left[\left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \frac{d\boldsymbol{\omega}_j}{dt} + \frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \frac{d}{dt} \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) + \frac{d\boldsymbol{\omega}_j^T}{dt} I_j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \end{aligned} \quad (134)$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{q}_i} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] &= \sum_{j=1}^n \left[m_j \frac{\partial}{\partial \mathbf{q}_i} \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \mathbf{v}_j + m_j \left(\frac{\partial \mathbf{v}_j^T}{\partial \dot{\mathbf{q}}_i} \right) \frac{\partial \mathbf{v}_j}{\partial \mathbf{q}_i} + m_j \frac{\partial \mathbf{v}_j^T}{\partial \mathbf{q}_i} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) + m_j (\mathbf{v}_j^T) \frac{\partial}{\partial \mathbf{q}_i} \left(\frac{\partial \mathbf{v}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \\ &+ \sum_{j=1}^n \left[\left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \frac{\partial \boldsymbol{\omega}_j}{\partial \mathbf{q}_i} + \frac{\partial}{\partial \mathbf{q}_i} \left(\frac{\partial \boldsymbol{\omega}_j^T}{\partial \dot{\mathbf{q}}_i} \right) I_j \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T I_j \frac{\partial}{\partial \mathbf{q}_i} \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) + \frac{\partial \boldsymbol{\omega}_j^T}{\partial \mathbf{q}_i} I_j \left(\frac{\partial \boldsymbol{\omega}_j}{\partial \dot{\mathbf{q}}_i} \right) \right] \end{aligned} \quad (135)$$

Observando cada uno de los términos, son muy parecidos. Los cuales se pueden eliminar siempre y cuando se cumpla que las derivadas parciales se anulen, esto implica que $r > j$ en las ecuaciones (70) y (77). De esta forma, se cumple la expresión (136).

$$\frac{\partial}{\partial \mathbf{q}_i} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] - \frac{d}{dt} \left[\sum_{j=1}^n \frac{\partial \mathcal{K}_j(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_i} \right] = 0 \quad (136)$$

Dada la matriz de inercia

$$M(\mathbf{q}) = \sum_{i=1}^n \left[m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i) \right]$$

Se va a demostrar que $M(\mathbf{q})$ es una matriz definida positiva.

Primero se verifica que $(J_{v_0}^i)^T (J_{v_0}^i)$ es una matriz definida positiva, entonces se considera la expresión:

$$\begin{aligned} \dot{\mathbf{q}}^T (J_{v_0}^i)^T (J_{v_0}^i) \dot{\mathbf{q}} &= \dot{\mathbf{q}}^T (p_i^T R_{x_{i-1}}(\theta)^T + d_{i-1}^T) I_d (d_{i-1} + R_{x_{i-1}}(\theta) p_i) \dot{\mathbf{q}} \\ \dot{\mathbf{q}}^T (J_{v_0}^i)^T (J_{v_0}^i) \dot{\mathbf{q}} &= (p_i^T R_{x_{i-1}}(\theta)^T + d_{i-1}^T) I_d (d_{i-1} + R_{x_{i-1}}(\theta) p_i) \dot{\mathbf{q}}^2 \end{aligned}$$

Esta expresión es un producto cuadrático que incluye las velocidades articulares $\dot{\mathbf{q}}$ y está definida por operaciones lineales y propiedades de rotación ortogonales, por lo que sus autovalores son positivos, ya que la matriz resultante preserva la definición positiva durante las operaciones matriciales. Además:

$$I_d = R_{z_i}^T S^T S R_{z_i} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \quad (137)$$

donde I es la matriz identidad y es definida positiva.

Puesto que R_{z_i} y R_{x_i} son matrices ortogonales, tienen la propiedad de que $R_{z_i}^T R_{z_i} = I$ y $(R_{x_{i-1}})^T R_{x_{i-1}} = I$. Esto implica que el producto de estas matrices también sea definido positivo.

Por lo tanto, podemos concluir que $(J_{v_0}^i)^T (J_{v_0}^i)$ es definida positiva, ya que para todo

$\dot{\mathbf{q}} \neq 0$, se cumple que $\dot{\mathbf{q}}^T (J_{v_0}^i)^T (J_{v_0}^i) \dot{\mathbf{q}} > 0$.

También se considera que $(J_{v_0}^i)^T (J_{v_0}^i)$ es invertible, puesto que $S(k_i)^T S(k_i)$ es invertible, $R_{z_i}(\phi)$ y $R_{x_i}(\theta)$ son matrices ortogonales, sus transpuestas y productos son invertibles. Dado que cada componente de $(J_{v_0}^i)^T (J_{v_0}^i)$ es invertible, se concluye que la matriz resultante es invertible.

Para demostrar que $(J_{\omega_0}^i)^T I_i (J_{\omega_0}^i)$ es una matriz definida positiva, es importante mencionar que el tensor de inercia I_i es una matriz definida positiva. Esto se debe a que representa la distribución de masa y forma de un cuerpo rígido, cuyos autovalores son positivos. El jacobiano de velocidad angular $J_{\omega_0}^i$ es una matriz de transformación lineal, conformada por las matrices de rotación y el vector $\mathbf{z}(\mathbf{k})$, los cuales son constantes. Dado que la estructura y propiedades de $J_{\omega_0}^i$ no introduce valores negativos en la multiplicación de la matriz resultante, ya que la multiplicación de una matriz definida positiva por otra matriz no puede introducir autovalores negativos.

Puesto que los componentes de $M(q)$ son matrices definidas positivas, se concluye que la matriz $\sum_{i=1}^n [m_i (J_{v_0}^i)^T (J_{v_0}^i) + (J_{\omega_0}^i)^T I_i (J_{\omega_0}^i)]$ es definida positiva.

4. Análisis cuantitativo del método recursivo

Se propone una comparación cuantitativa entre el método recursivo propuesto y la formulación convencional de Euler-Lagrange, ambos implementados mediante programación simbólica en MATLAB. Las pruebas se realizaron en una computadora portátil estándar, ejecutando los códigos que calculan la cinemática directa, las matrices Jacobianas y la matriz de inercia del robot humanoide. Cada programa se ejecutó 20 veces para obtener el tiempo promedio de ejecución y su correspondiente desviación estándar. El análisis considera únicamente la computación simbólica hasta la generación de la matriz de inercia, excluyendo las etapas de evaluación numérica o simulación dinámica.

El rendimiento se evaluó para configuraciones de robot con 3 GDL (grados de libertad), 6 GDL, 9 GDL y 18 GDL, correspondientes a diferentes subconjuntos del modelo del robot humanoide. Para cada configuración, se registró el tiempo de ejecución de ambos métodos y se calculó el factor de aceleración (speed-up) del método recursivo con respecto al enfoque convencional.

La Tabla 6 indica que el método recursivo logra consistentemente una aceleración computacional de aproximadamente $2\times$ en comparación con la derivación simbólica convencional. Además, el algoritmo recursivo exhibe un crecimiento casi lineal en el tiempo de computación a medida que aumenta el número de grados de libertad, mientras que el enfoque convencional muestra una tendencia cuadrática, lo que confirma la mejor escalabilidad y eficiencia computacional de la formulación recursiva.

Tabla 6. Comparación cuantitativa entre los métodos recursivo y convencional

GDL	Método Recursivo (Tiempo $\pm$ Desv. Est.) (s)	Conventional method Time $\pm$ Std. Deviation (s)	Aceleración (Speed-up)
3	0.0320 $\pm$ 0.1430	0.0663 $\pm$ 0.2965	2.07
6	0.0781 $\pm$ 0.3492	0.1560 $\pm$ 0.6976	2.00
9	0.0920 $\pm$ 0.4117	0.1943 $\pm$ 0.8692	2.11
18	0.1714 $\pm$ 0.7665	0.3662 $\pm$ 1.6377	2.14

Las Figuras 2 y 3 comparan el rendimiento computacional de la formulación recursiva propuesta y el enfoque convencional de Euler-Lagrange. El método recursivo logra aproximadamente el doble de eficiencia computacional, con un tiempo de ejecución que aumenta linealmente $O(n)$ a medida que se incrementa el número de GDL (grados de libertad), mientras que la derivación simbólica convencional muestra una tendencia casi cuadrática $O(n^2)$.

Estos resultados demuestran la superior escalabilidad y eficiencia del algoritmo recursivo para el modelado dinámico simbólico de robots humanoides.

La Figura. 2 ilustra el tiempo promedio de computación simbólica para ambos métodos, incluyendo barras de error que representan la desviación estándar tras 20 ejecuciones. El método recursivo exhibe consistentemente tiempos de ejecución menores que la formulación convencional para todas las configuraciones de robot (3, 6, 9 y 18 GDL).

La Fig. 3 muestra la relación de aceleración (factor de aceleración) del método recursivo en comparación con el convencional. El algoritmo recursivo mantiene una aceleración aproximadamente constante de unas 2 veces más rápido en todas las

configuraciones, lo que confirma su crecimiento computacional lineal respecto al número de grados de libertad.

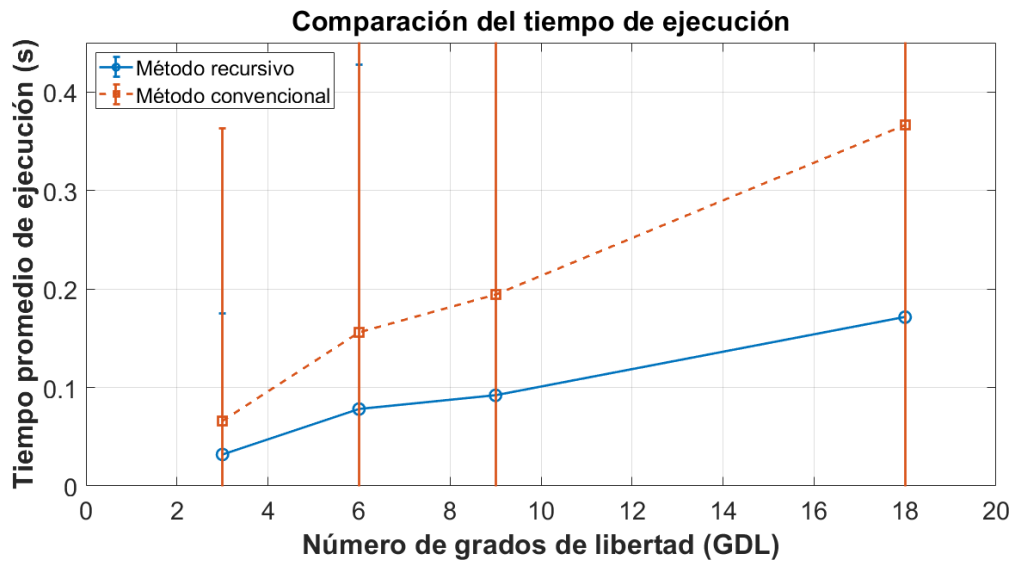


Figura 8. Tiempo promedio de ejecución de los métodos recursivo y convencional para diferentes números de grados de libertad (GDL).

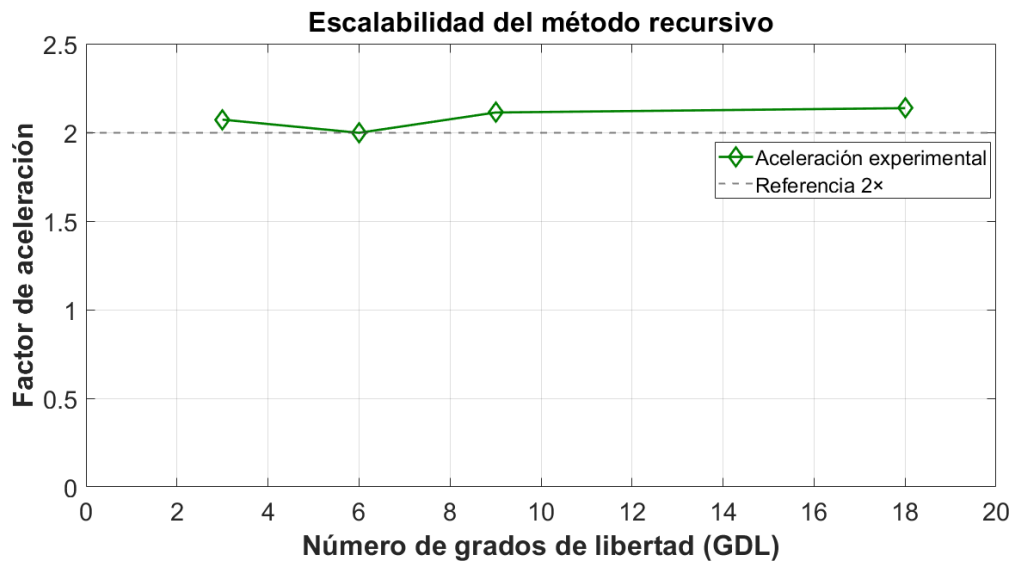


Figura 9. Aceleración computacional (speed-up) del método recursivo en relación con el enfoque convencional.

5. Implementación del control de posición

La siguiente sección presenta los resultados de simulación obtenidos en Simulink para el control de posición del robot bípedo de 18 grados de libertad (DoF) utilizando tres controladores diferentes.

Los algoritmos de control no pudieron implementarse en el robot físico Bioloid Premium debido a que sus articulaciones están equipadas con servomotores estándar en lugar de motores de CC (corriente continua) con codificadores (encoders). Estos servomotores no cuentan con una arquitectura abierta que permita la implementación directa de algoritmos de control personalizados para la salida de par motor (torque).

Por lo tanto, se desarrolló un modelo virtual del robot Bioloid Premium en Simscape Multibody, lo que permitió la prueba y validación de las estrategias de control propuestas en un entorno simulado antes de cualquier aplicación potencial en el mundo real.

Se presentan los resultados del control de posición implementado dentro del método recursivo para obtener el modelo dinámico del robot. Para resumir los hallazgos, se muestra la respuesta de las articulaciones correspondientes a la pierna derecha y el brazo derecho del robot Bioloid Premium. Esta selección se justifica porque estas articulaciones representativas permiten ilustrar el desempeño de las estrategias de control sin redundancia, proporcionando al mismo tiempo una visión clara del comportamiento dinámico del sistema.

La saturación ocurre cuando la ley de control requiere un par motor (torque) que supera los límites del servoamplificador electrónico. En otras palabras, dado que el amplificador no puede exceder sus límites físicos, simplemente proporciona el par máximo τ_{max} .

Para abordar esto, se propone una estrategia de control acotada de tal manera que el error de posición $\tilde{q}$ y la velocidad $\dot{q}$ converjan asintóticamente a cero, independientemente de las condiciones iniciales del robot. Se implementaron tres tipos de controladores para su comparación: un controlador PD estándar, un control saturado y un controlador de tangente hiperbólica. El controlador PD se describe mediante la ecuación (138):

$$\tau = K_p \tilde{q} - K_v \dot{q} + g(q) \quad (138)$$

donde τ denota la entrada de control en el modelo de la ecuación (138). Adicionalmente, K_p es la ganancia proporcional y K_v es la ganancia derivativa. Las ganancias se seleccionaron de la siguiente manera: $K_p = 15$, $K_v = 1.5$.

El segundo controlador propuesto utiliza un controlador saturado, descrito por la ecuación (139):

$$\tau = K_p \left[\frac{\arctan(\alpha \tilde{q})}{\sqrt{1 + \tanh^2(\alpha \tilde{q})}} \right] - K_v \left[\frac{\arctan(\alpha \dot{q})}{\sqrt{1 + \tanh^2(\alpha \dot{q})}} \right] + g(q) \quad (139)$$

donde α es una constante positiva. Los valores se seleccionaron de la siguiente manera: $K_p = K_v = 1.5$, $\alpha = 50$ para las piernas, y $\alpha = 4$ para los brazos.

El tercer controlador propuesto utiliza una función de tangente hiperbólica, descrita por la ecuación (140):

$$\tau = K_p \tanh(\Lambda \tilde{q}) - K_v \tanh(\Gamma \dot{q}) + g(q) \quad (140)$$

donde $\Lambda, \Gamma \in \mathbb{R}^{n \times n}$ son matrices de ganancia diagonales definidas positivas. Las ganancias se seleccionaron de la siguiente manera: $K_p = 1.6$, $K_v = 1$, $\Lambda = 10$, $\Gamma = 1$.

La sintonización de las ganancias depende del par motor (torque) máximo que los motores pueden entregar. Para verificar el desempeño de la estrategia de control de posición propuesta, se realizó una simulación en MATLAB.

La figura 4 ilustra el modelo virtual del robot Bioloid Premium en su posición inicial (home), el cual fue ensamblado primero en SolidWorks y luego importado a Simscape Multibody dentro de Simulink para crear una representación virtual del robot. Este modelo virtual permite realizar pruebas de los comportamientos cinemáticos y dinámicos en un entorno controlado, garantizando la precisión y facilitando el desarrollo de algoritmos de control.

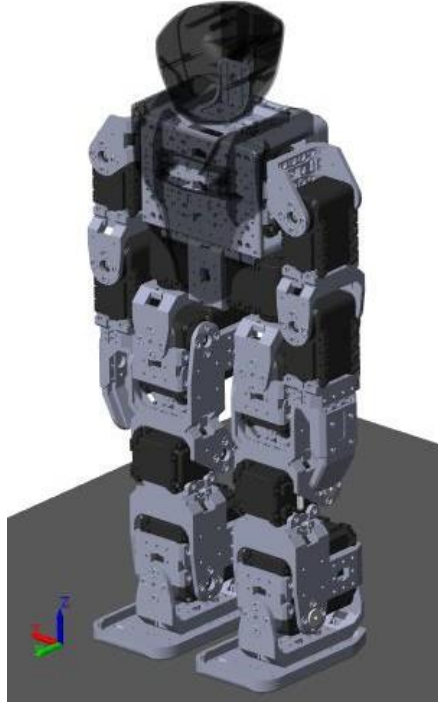


Figura 10. Modelo virtual del robot Bioloid Premium importado en Simscape Multibody.

Para validar el modelo dinámico propuesto del robot, se implementaron y evaluaron los tres controladores de posición articular descritos anteriormente. Se generó un movimiento de caminata dinámica definiendo una trayectoria sinusoidal paramétrica para cada articulación, con el objetivo de producir locomoción hacia adelante. No se incorporaron estrategias de equilibrio o de estabilización de la marcha de alto nivel (como el control basado en ZMP, modelos de péndulo invertido o MPC). Por lo tanto, el movimiento del robot se rige únicamente por las trayectorias comandadas en el espacio articular, y el desempeño de los controladores de posición permite que la validación se centre estrictamente en la precisión del modelo dinámico y el comportamiento del controlador, sin la influencia de algoritmos de estabilización adicionales. La trayectoria de referencia para cada articulación $q_i(t)$ se define

en la ecuación (141):

$$q_i(t) = A_i \sin(\omega_i t + \phi_i) + q_{0,i} \quad (141)$$

donde A_i es la amplitud de oscilación, ω_i es la frecuencia angular, ϕ_i es el desfase de fase utilizado para coordinar las articulaciones simétricas (por ejemplo, las piernas izquierda y derecha), y $q_{0,i}$ es la posición central nominal de la articulación.

Esta función fue seleccionada por tres razones principales: Proporciona una referencia de articulación suave, continua y diferenciable, lo cual es esencial para evaluar el desempeño del seguimiento bajo un movimiento dinámico bien definido. La parametrización sinusoidal permite generar movimientos periódicos similares a la marcha sin la necesidad de un controlador de equilibrio explícito, reduciendo la complejidad del modelado y la implementación. Este enfoque es ampliamente utilizado en las etapas iniciales del desarrollo de robots humanoides cuando la intención es analizar el desempeño del controlador en lugar de la estabilidad dinámica de todo el cuerpo.

Utilizando estas trayectorias, el robot realiza un movimiento de caminata continua hacia adelante durante seis segundos. Durante este intervalo, los 18 controladores articulares operan en modo de posición de lazo cerrado. Las tres estrategias de control: PD convencional, control saturado y control de tangente hiperbólica, se ejecutan bajo condiciones de movimiento idénticas. Su desempeño se compara analizando los errores de seguimiento articular obtenidos de simulaciones del modelo completo del robot en Simscape Multibody. Esta configuración de validación permite una evaluación cuantitativa de la eficiencia del controlador bajo una tarea dinámica, manteniendo la estructura del sistema enfocada en el control a nivel articular en lugar del control de estabilidad de todo el cuerpo. La figura 5 ilustra la implementación de la trayectoria de caminata dinámica propuesta en el modelo humanoide virtual desarrollado en Simscape Multibody. Las referencias articulares generadas se aplicaron al robot simulado, y el movimiento resultante demuestra la ejecución completa del ciclo de marcha bajo las condiciones dinámicas definidas en el marco de control. Esta visualización confirma que las coordenadas articulares, velocidades y pares motor (torques) calculados a partir del modelo dinámico recursivo pueden impulsar con éxito la simulación multicuerpo, validando la viabilidad de la generación de trayectorias propuesta y el enfoque de control en un entorno realista basado en la física.

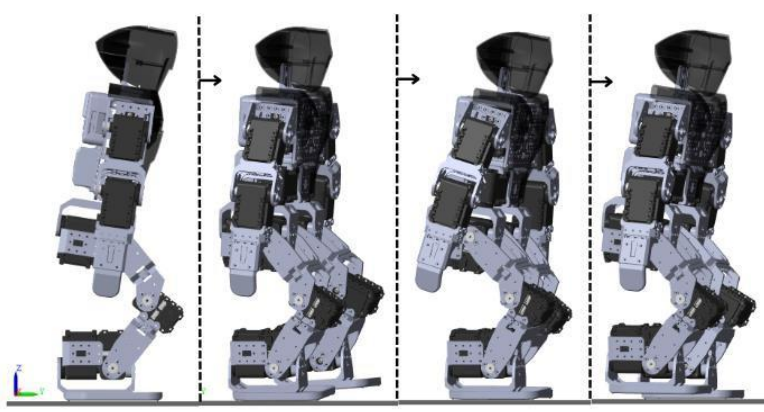


Figura 11. Marcha dinámica deseada propuesta para el robot

Para evaluar la precisión del modelo dinámico propuesto, se realizó un análisis comparativo entre los pares articulares (torques) generados por el modelo analítico y aquellos aplicados al robot virtual implementado en Simscape. Con este propósito, se simuló una secuencia de caminata de seis segundos, en la cual se aplicó la trayectoria de referencia propuesta a ambos sistemas. En ambos casos, se utilizó la misma ley de control saturado, definida en el manuscrito, para realizar el seguimiento del movimiento deseado. Los pares articulares se recolectaron tanto del modelo dinámico como del robot virtual, y la comparación se llevó a cabo únicamente durante el intervalo de caminata real, excluyendo intencionalmente las transiciones de postura iniciales y finales.

De la Fig. 6 a la Fig. 11 se muestran las curvas de error de torque obtenidas al comparar los pares articulares calculados a partir del modelo dinámico analítico con los torques generados por la simulación en Simscape Multibody. Solo se incluyen las articulaciones de las piernas, ya que desempeñan el papel más crítico en la generación de trayectorias y el control de seguimiento articular durante la caminata dinámica. La figura destaca la estrecha similitud con la que los torques aplicados siguen las predicciones basadas en el modelo a lo largo de todo el movimiento. Adicionalmente, la Tabla 7 reporta los valores de RMSE (Error Cuadrático Medio) para cada articulación, proporcionando una medida cuantitativa del desempeño del seguimiento de torque y confirmando la consistencia entre el modelo analítico y el robot simulado.

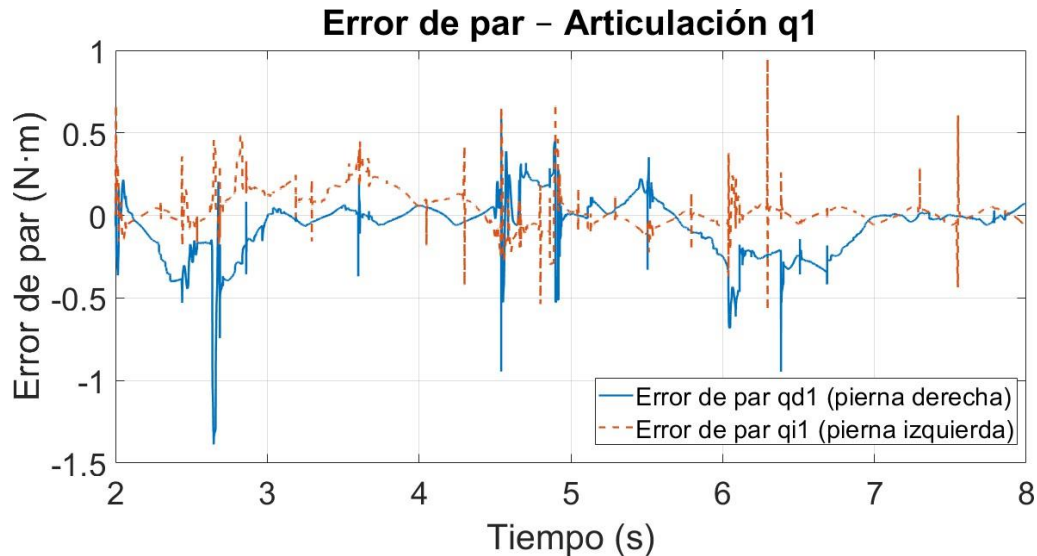


Figura 12. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d1} y q_{i1} .

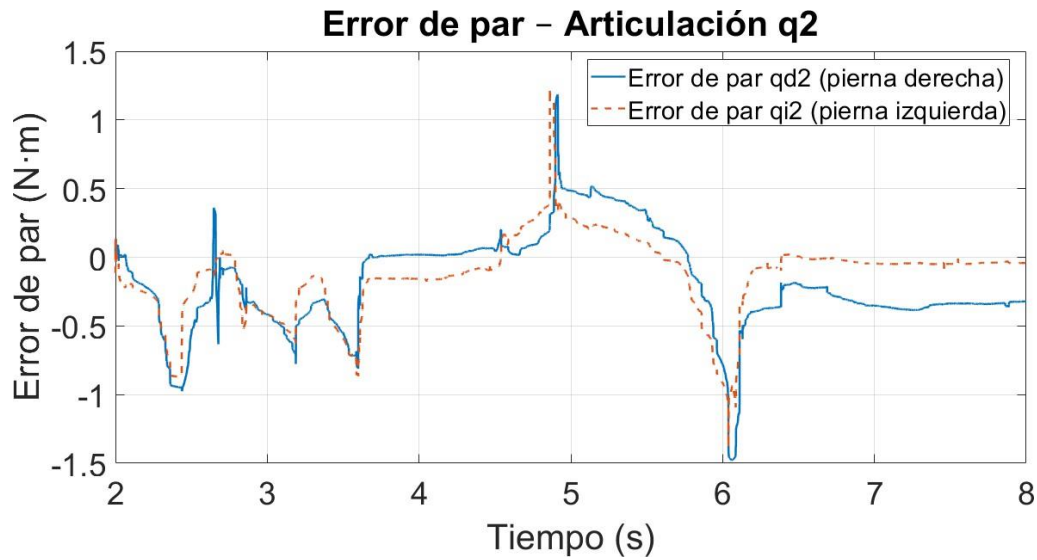


Figura 13. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d2} y q_{i2} .

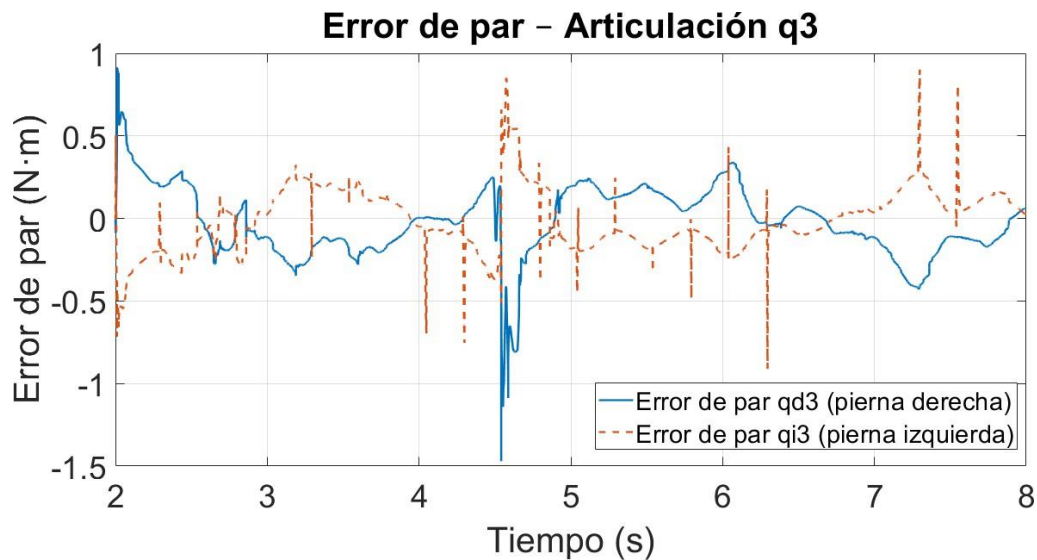


Figura 14. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d3} y q_{i3} .

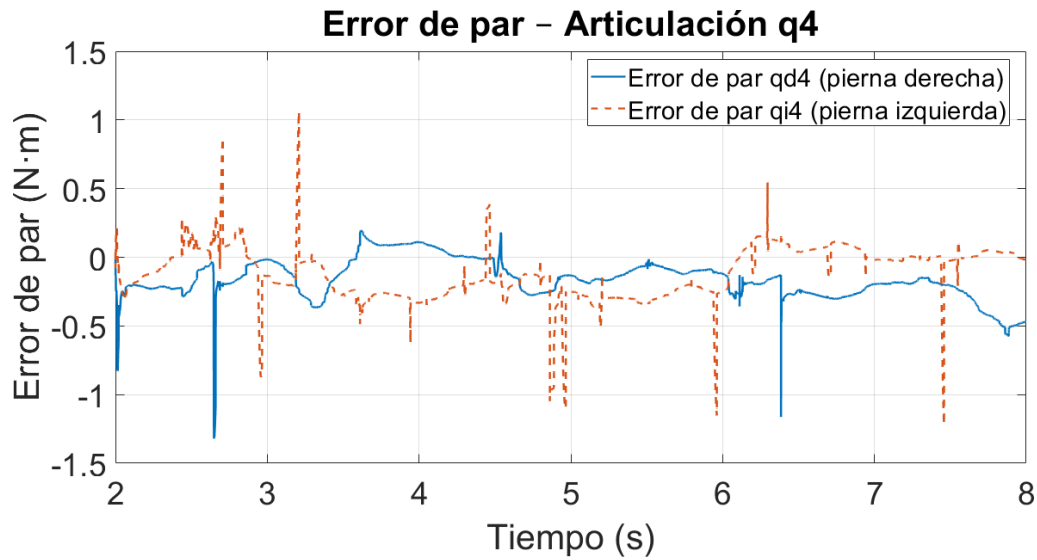


Figura 15. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d4} y q_{i4} .

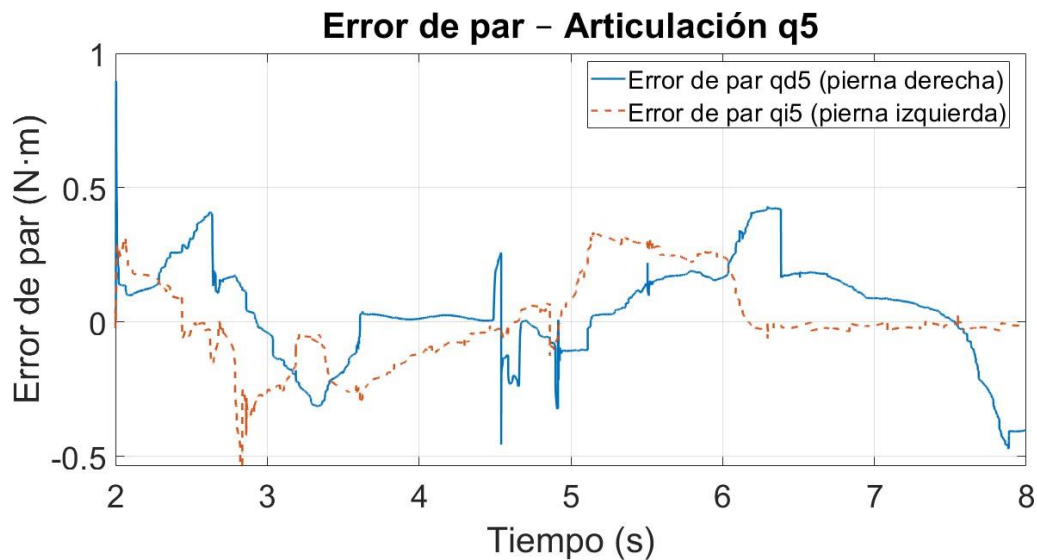


Figura 16. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d5} y q_{i5} .

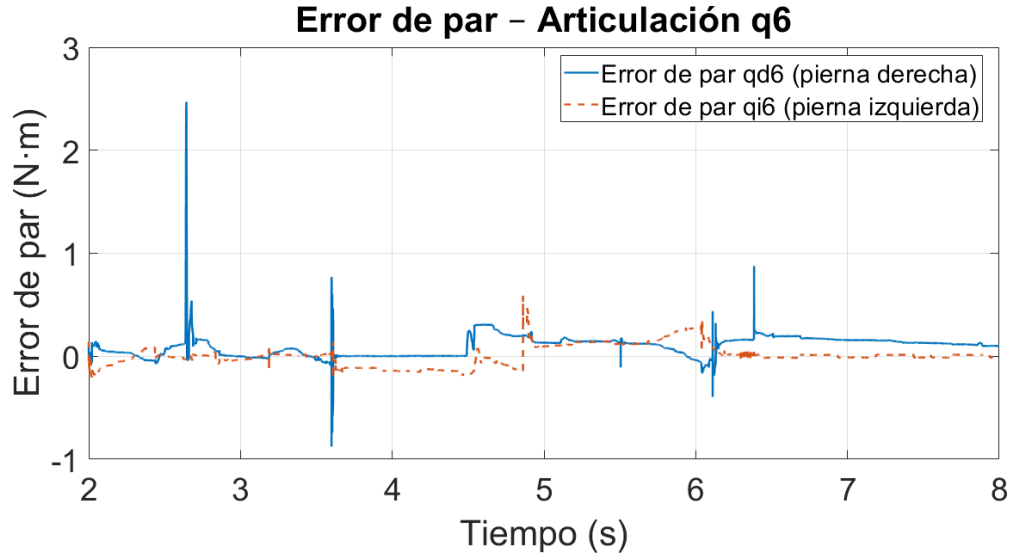


Figura 17. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d6} y q_{i6} .

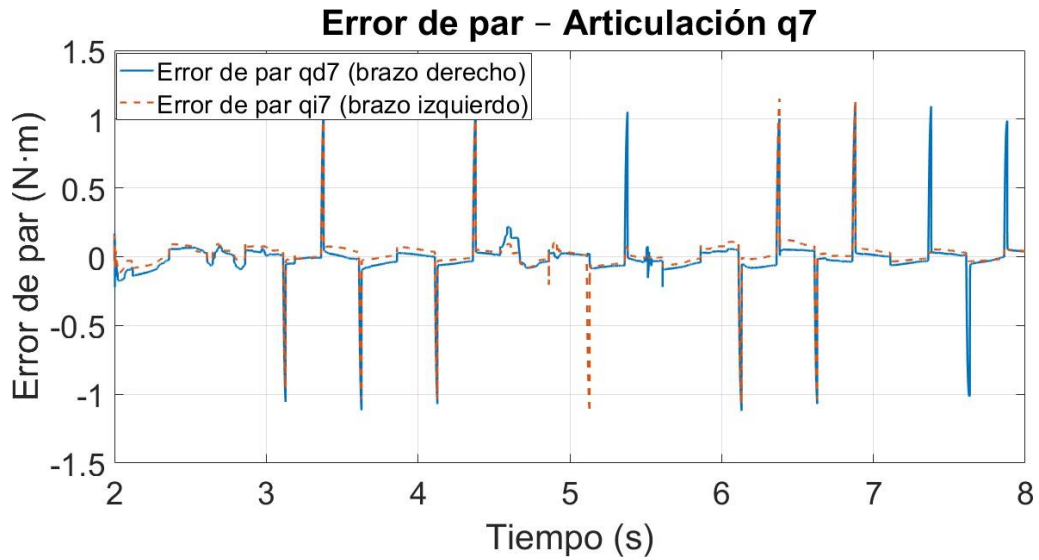


Figura 18. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d7} y q_{i7} .

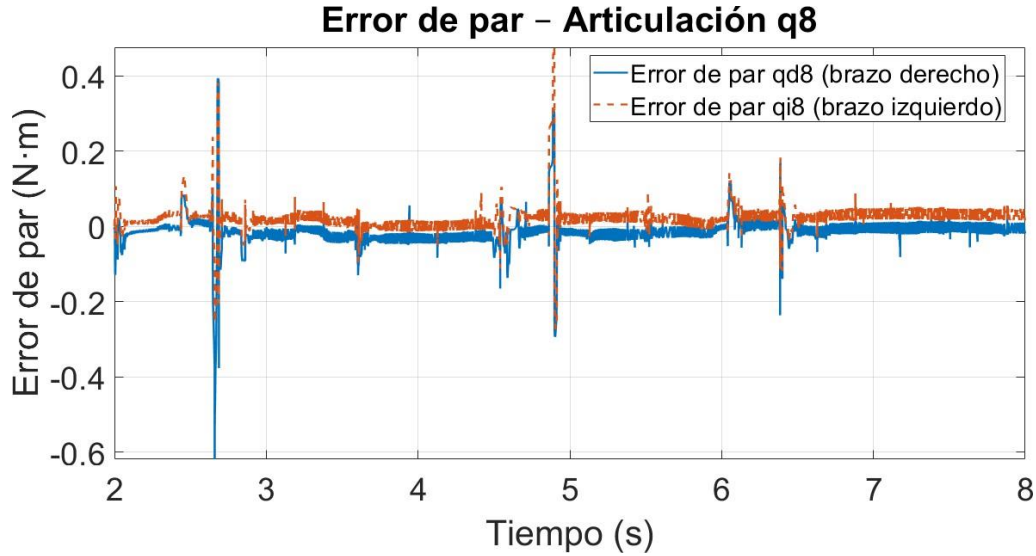


Figura 19. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d8} y q_{i8} .

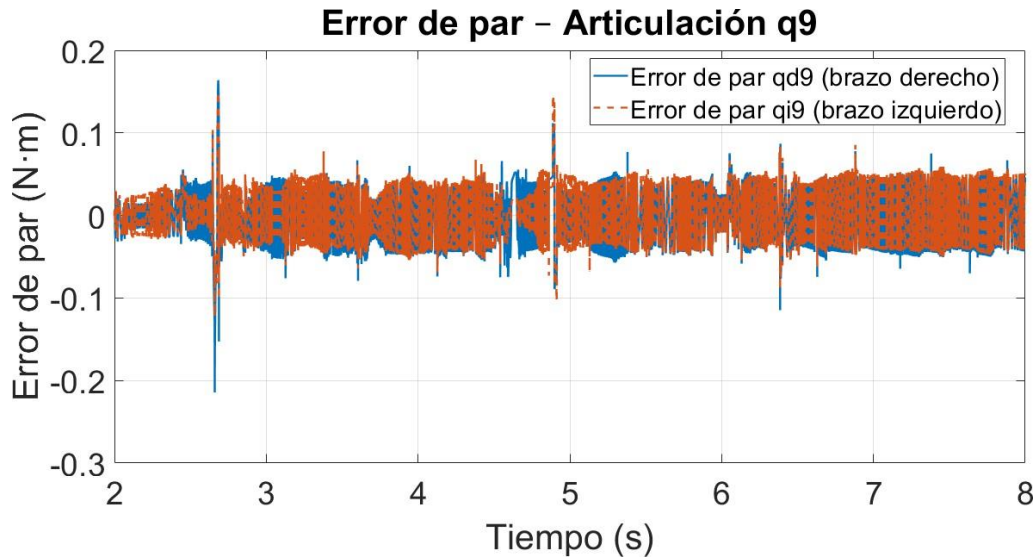


Figura 20. Comparación de los errores de torque entre el modelo dinámico analítico y la simulación de Simscape Multibody para las articulaciones q_{d9} y q_{i9} .

El desempeño se cuantificó utilizando el Error Cuadrático Medio (RMSE) para cada una de las 18 articulaciones (9 derechas y 9 izquierdas), obteniendo valores como 0.1906 para la rotación de la cadera derecha (hip roll, Articulación 1) y 0.1330 para su contraparte en el lado izquierdo, entre otros. Estos resultados se muestran en la Tabla 7 demuestran que los pares aplicados por el modelo dinámico coinciden estrechamente con los obtenidos del robot virtual, validando la precisión del marco de modelado y control utilizado en este estudio.

Tabla 7. RMSE medido en cada articulación del robot.

Articulación	RMSE – Lado derecho (Nm)	RMSE - Lado izquierdo (Nm)
1	0.19062	0.13304
2	0.38839	0.31689
3	0.195664	0.18962
4	0.22268	0.21168
5	0.19307	0.16367
6	0.15134	0.10375
7	0.14327	0.12724
8	0.03932	0.04034
9	0.03091	0.03082

La evaluación del desempeño de los tres controladores se realizó cuantitativamente utilizando el criterio de la norma L_2 (o norma euclidiana), la cual es una herramienta ampliamente utilizada por la comunidad científica en robótica para medir el desempeño del control. Está dada por la expresión (142):

$$L_2 = \sqrt{\frac{1}{T} \int_0^T \|\tilde{q}(\sigma)\| d\sigma} \quad (142)$$

donde $T \in \mathbb{R}_+$ representa el tiempo de simulación, en este caso 6 segundos. Una norma menor significa un menor error de posición y representa el mejor desempeño del esquema evaluado. Basado en este análisis, el controlador saturado exhibió el mejor desempeño general en comparación con los controladores de tangente hiperbólica y PD. A nivel global, el controlador saturado alcanza el mejor desempeño, con un valor normalizado de aproximadamente el 32%, lo que corresponde a una reducción del 68% en el error de seguimiento en relación con el controlador PD. Esto confirma que acotar el comando de control mediante una función de saturación mejora significativamente la precisión del seguimiento en lazo cerrado, particularmente ante la presencia de movimientos cíclicos rápidos y grandes demandas de par motor (torque). El controlador de tangente hiperbólica también mejora el seguimiento de trayectoria en comparación con el PD, con un desempeño normalizado de aproximadamente el 90%, lo que representa una reducción del 10% en el error de seguimiento global. Aunque la función $\text{Tanh}(\cdot)$ introduce un amortiguamiento no lineal suave que modera las acciones de control agresivas, la mejora es más modesta que la lograda por el controlador saturado. El controlador de tangente hiperbólica aún permite señales de control de gran magnitud cuando se requieren, lo que explica por qué no proporciona el mismo nivel de reducción de error. En contraste, el controlador PD exhibe el error de seguimiento más alto (establecido como el 100%), confirmando las limitaciones de una ley de control puramente lineal para movimientos articulares altamente dinámicos. Sin ninguna no linealidad o límite de amplitud, el controlador PD genera mayores sobreimpulsos (overshoots) y picos de esfuerzo de control durante el ciclo de marcha, resultando en una precisión de seguimiento inferior. La figura 21 presenta la comparación del desempeño

global de los tres controladores durante la prueba de caminata dinámica, evaluada mediante la métrica de la norma L_2 .

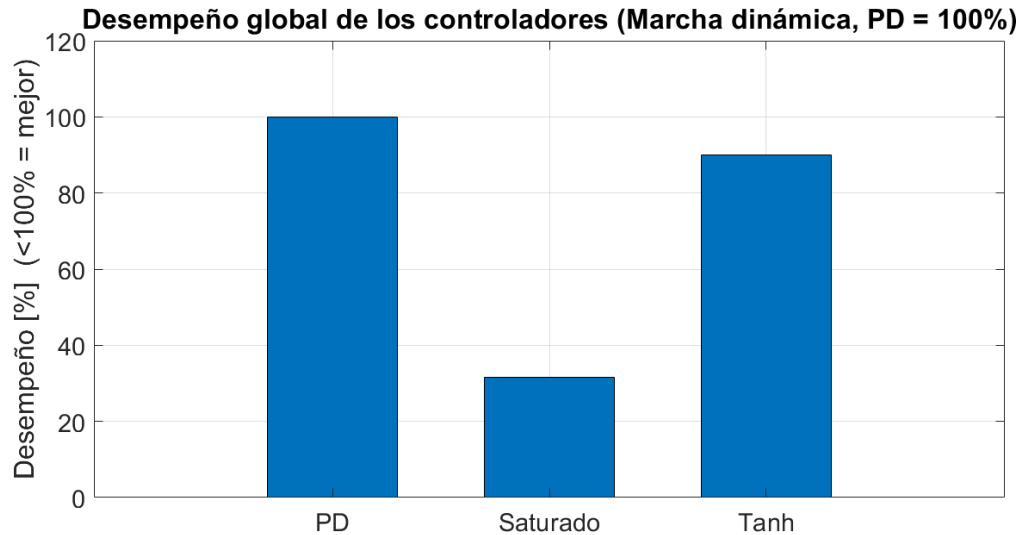


Figura 21. Índices de desempeño global de los controladores.

Por diseño, los valores por debajo del 100% indican un mejor seguimiento general que la referencia (línea base) del PD (es decir, un menor error acumulado), mientras que los valores por encima del 100% indican un desempeño inferior.

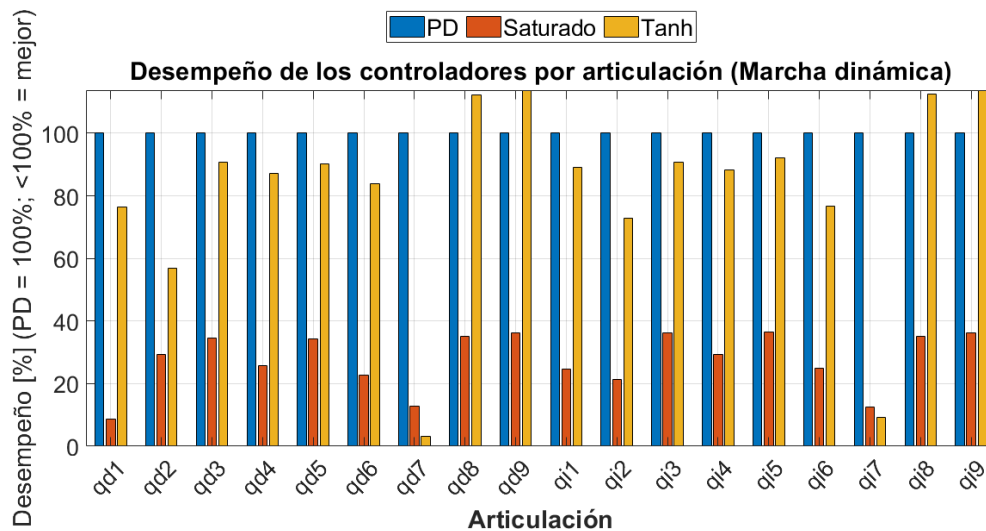


Figura 22. Índices de desempeño por cada articulación.

Los resultados de desempeño por articulación basados en la norma L_2 se muestran en la figura 22 donde cada articulación incluye tres barras que corresponden al controlador PD (referencia al 100%), el controlador saturado y el controlador de tangente hiperbólica. Dado que los valores inferiores al 100% indican un menor error de seguimiento acumulado, los

resultados demuestran que el controlador saturado logra el mejor desempeño en la mayoría de las articulaciones, particularmente en las piernas. Las reducciones consistentes en grados de libertad clave, como $q_{d1} - q_{d6}$ y $q_{i1} - q_{i6}$, confirman que el control saturado mejora la precisión del seguimiento y la estabilidad durante la trayectoria dinámica.

El controlador tangente hiperbólica muestra un desempeño mixto: supera al PD en varias articulaciones, pero excede la línea base del 100% en otras, lo que refleja su sensibilidad a la dinámica de los grados de libertad individuales. Por su parte, el controlador PD se mantiene en el nivel de referencia y compite de manera efectiva en articulaciones específicas; sin embargo, en términos generales, es superado por el controlador saturado en la mayor parte de la cadena articulada.

A continuación, se muestran los resultados obtenidos de la simulación en Simscape del control de posición del robot bípedo de 18 grados de libertad implementando la ley de control de la ecuación

Durante el intervalo total de 10 segundos, las Figuras 23 a 26 muestran de forma consistente la respuesta articular del robot bajo un control saturado durante la ejecución de la marcha dinámica. En una primera etapa, de 0 a 2 s, el robot parte desde la posición inicial y converge suavemente hacia una configuración estática de equilibrio, lo cual se observa en todas las articulaciones de las piernas (Figuras 23 y 24) y de los brazos (Figuras 25 y 26) mediante transiciones mayormente monótonas, sin oscilaciones significativas, indicando una correcta gestión de la saturación del actuador. Posteriormente, en el intervalo de 2 a 8 s, el robot entra en la fase de marcha dinámica, caracterizada por señales articulares periódicas y acotadas, especialmente visibles en las articulaciones de cadera, rodilla y tobillo de ambas piernas (Figuras 23 y 24), las cuales presentan oscilaciones cuasi-senoidales asociadas al ciclo de paso. Durante esta fase, los brazos (Figuras 25 y 26) exhiben movimientos coordinados y de menor amplitud, actuando como mecanismos de compensación dinámica para mejorar la estabilidad del sistema. Finalmente, de 8 a 10 segundos, el robot retorna de manera progresiva a la posición de equilibrio previa a la marcha, observándose una atenuación gradual de las oscilaciones y una convergencia estable de todas las articulaciones. En conjunto, los resultados evidencian que el control saturado permite una transición estable entre fases estáticas y dinámicas, mantiene las variables articulares dentro de límites físicos razonables y garantiza una marcha periódica y simétrica, sin pérdida de estabilidad ni comportamientos divergentes.

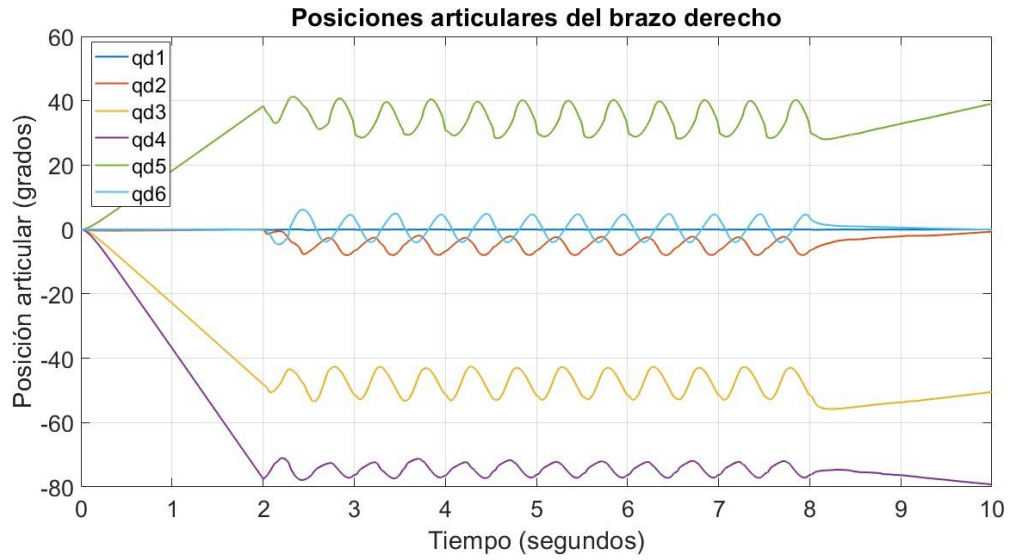


Figura 23. Posiciones articulares de la pierna derecha.

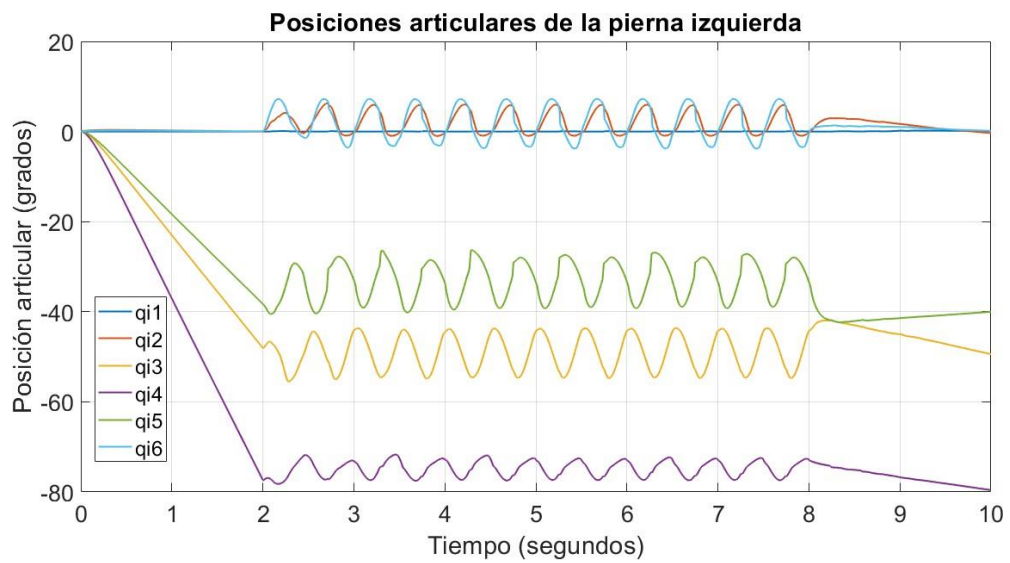


Figura 24. Posiciones articulares de la pierna izquierda.

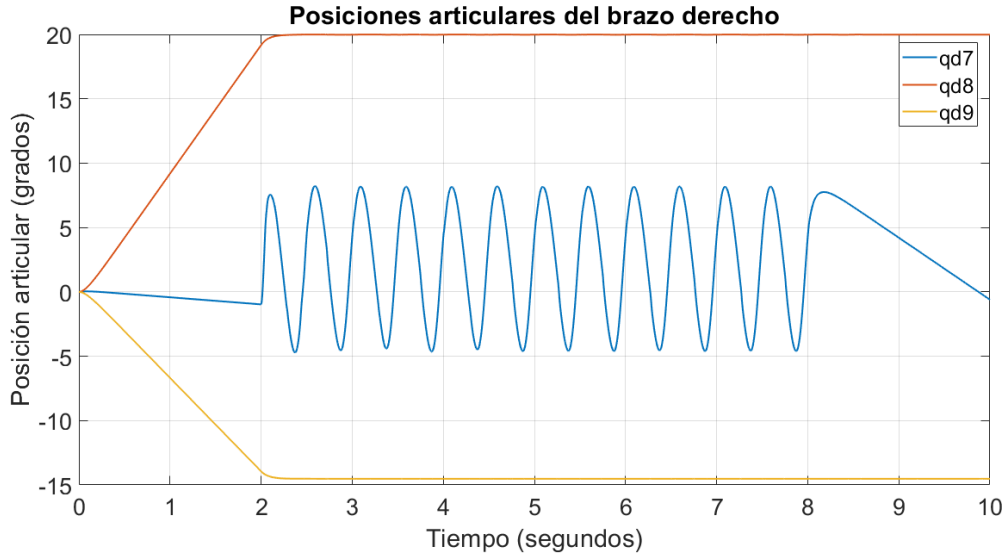


Figura 25. Posiciones articulares del brazo derecho.

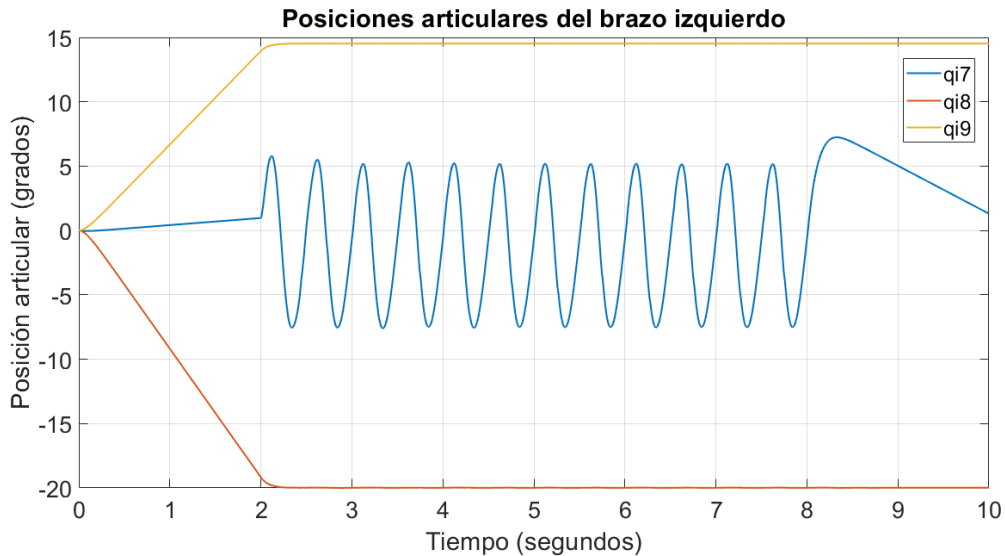


Figura 26. Posiciones articulares del brazo izquierdo.

En las Figuras 27 a 30 se observa el comportamiento del error de posición articular para las extremidades del robot, evidenciando una respuesta coherente y consistente del sistema de control en todos los grados de libertad. En particular, tanto en la pierna derecha (Figura 27) como en la pierna izquierda (Figura 28), los errores articulares presentan oscilaciones transitorias claramente acotadas, asociadas a la dinámica del movimiento coordinado de múltiples articulaciones, pero permanecen dentro de márgenes aceptables y tienden a converger hacia valores cercanos a cero una vez que el régimen transitorio finaliza. De forma análoga, en el brazo derecho (Figura 29) y el brazo izquierdo (Figura 30) se aprecia que, aunque algunas articulaciones exhiben mayores amplitudes de error debido a su

participación más activa o a efectos de acoplamiento dinámico, el sistema mantiene estabilidad global y no se presentan errores crecientes ni comportamientos divergentes. En conjunto, los resultados indican un adecuado seguimiento articular, buena sincronización entre extremidades y una correcta compensación de los efectos dinámicos, lo que valida el desempeño del esquema de control y del modelo cinemático-dinámico empleado.

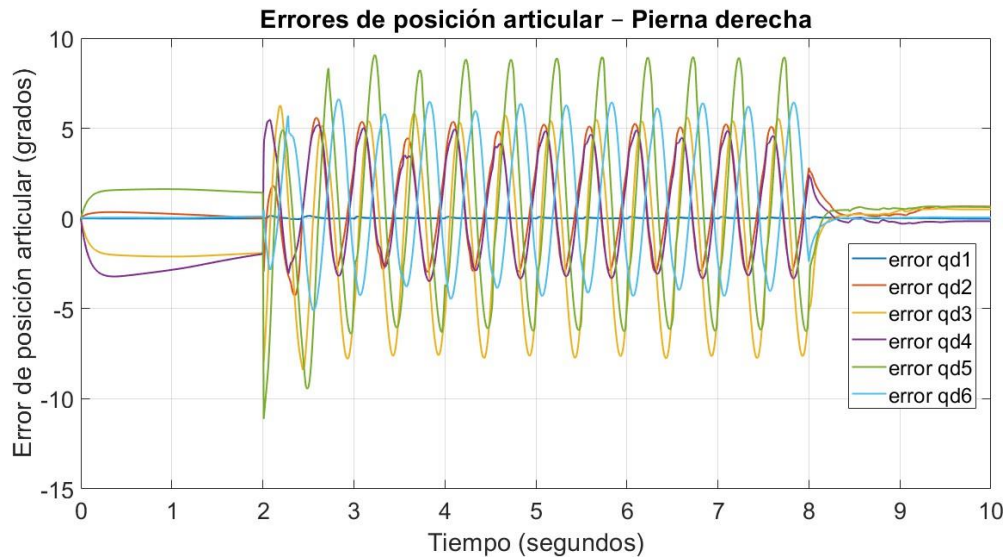


Figura 27. Errores de posición de las articulaciones de la pierna derecha.

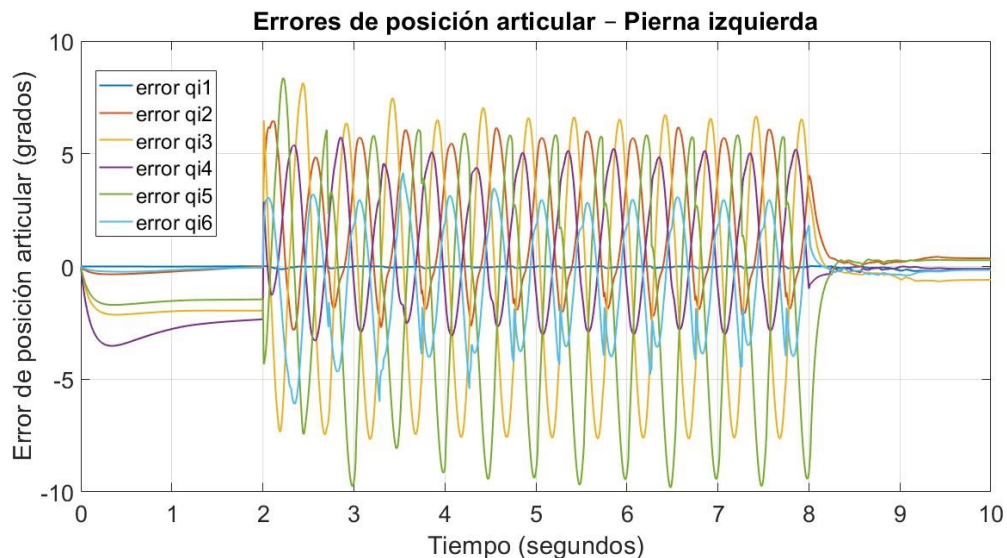


Figura 28. Errores de posición de las articulaciones de la pierna izquierda.

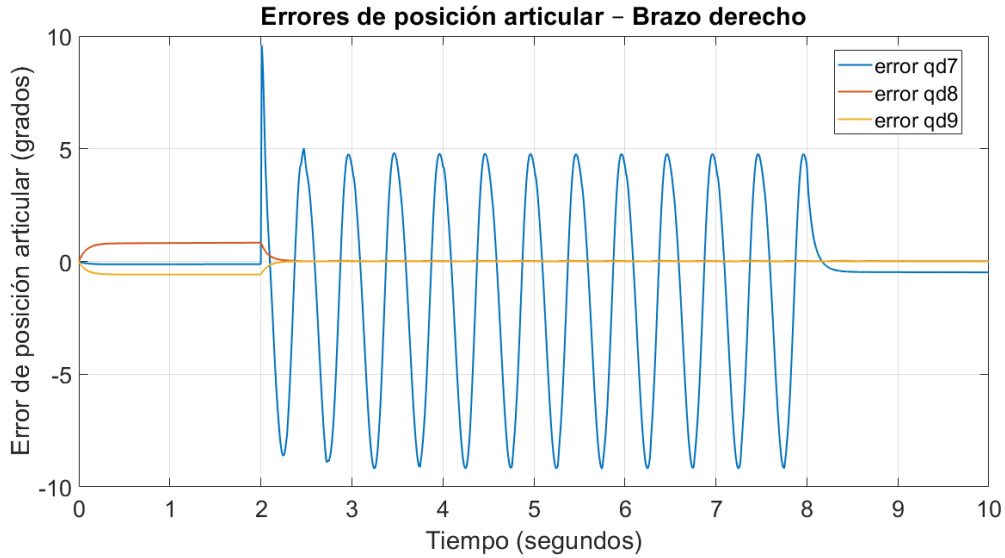


Figura 29. Errores de posición de las articulaciones del brazo derecho.

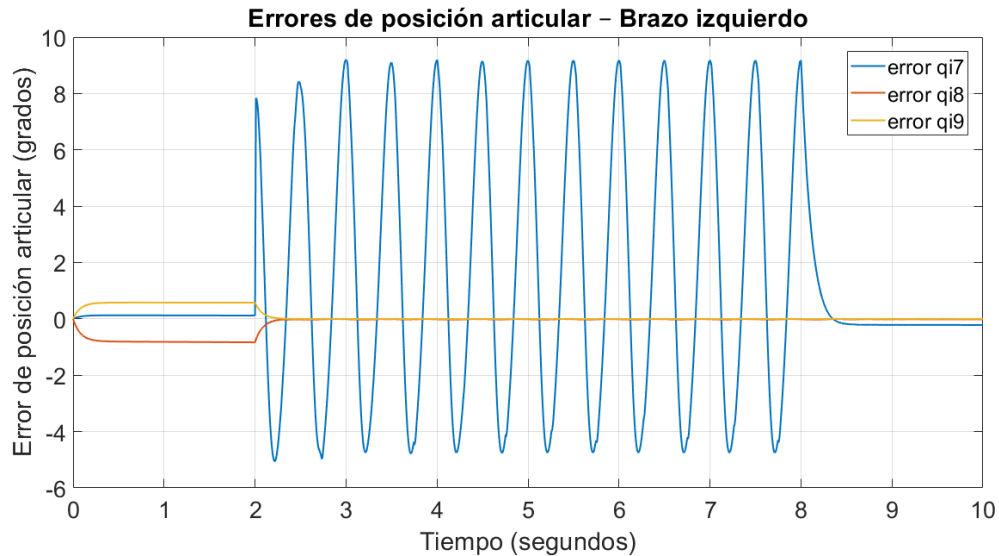


Figura 30. Errores de posición de las articulaciones del brazo izquierdo.

El comportamiento de las velocidades articulares, ilustrado en las Figuras 31 a 34, demuestra la efectividad del control saturado para generar perfiles de movimiento suaves y acotados a lo largo de todas las etapas de la simulación. Durante la fase de marcha dinámica (2 a 8s), se observa en las articulaciones de las piernas (Figuras 31 y 32) un régimen periódico y estable, donde las magnitudes de velocidad se mantienen dentro de límites físicos seguros, evitando picos abruptos o discontinuidades que podrían comprometer la estabilidad mecánica del robot. De manera complementaria, las articulaciones de los brazos (Figuras 33 y 34) exhiben una velocidad menor a la de las piernas, puesto que solamente las articulaciones q_{d7} y q_{i7} presentan movimiento, mientras que las demás permanecen estáticas. Es particularmente notable la respuesta en las etapas de transición: tanto en la fase inicial de

convergencia al equilibrio (0 a 2s) como en la etapa final de retorno a la postura estática (8 a 10s), las velocidades decaen asintóticamente hacia cero sin presentar vibraciones de alta frecuencia ni sobreimpulsos bruscos, lo que confirma que la ley de control regula eficientemente la energía cinética del sistema y garantiza transiciones fluidas entre el reposo y el movimiento dinámico.

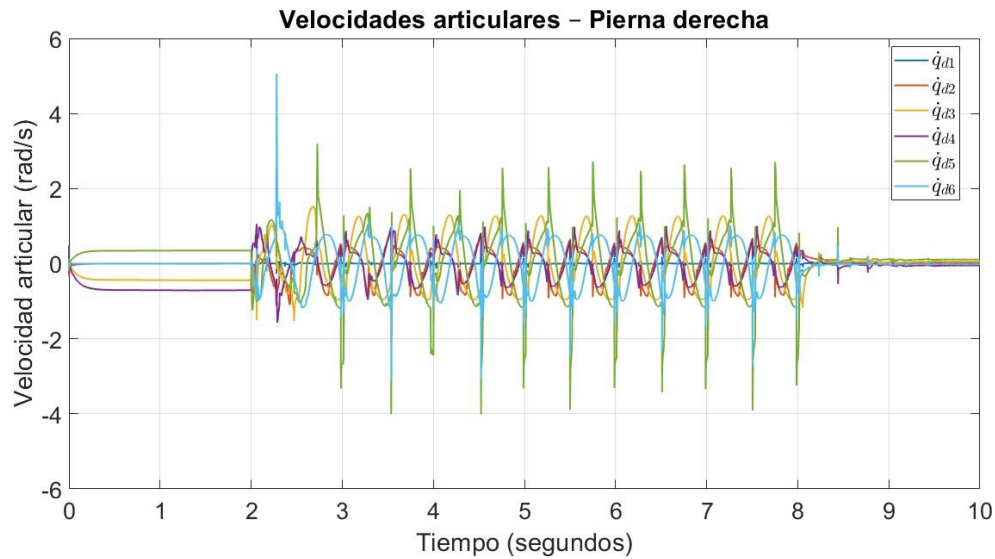


Figura 31. Velocidades articulares de la pierna derecha.

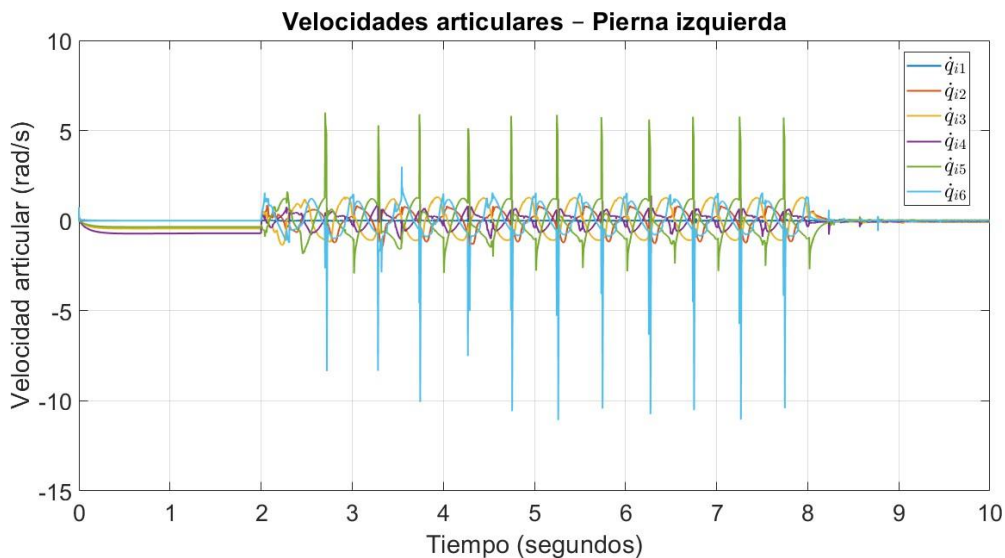


Figura 32. Velocidades articulares de la pierna izquierda.

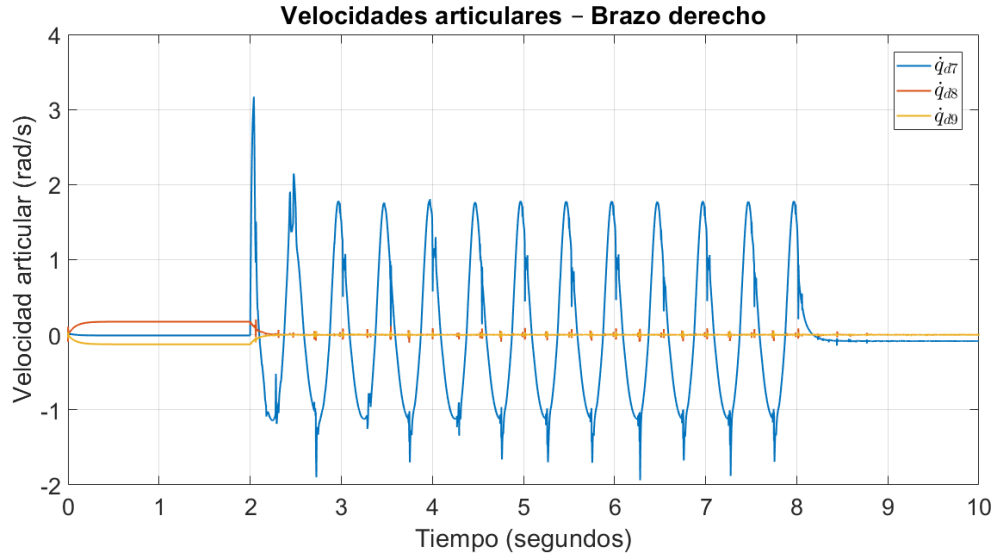


Figura 33. Velocidades articulares del brazo derecho.

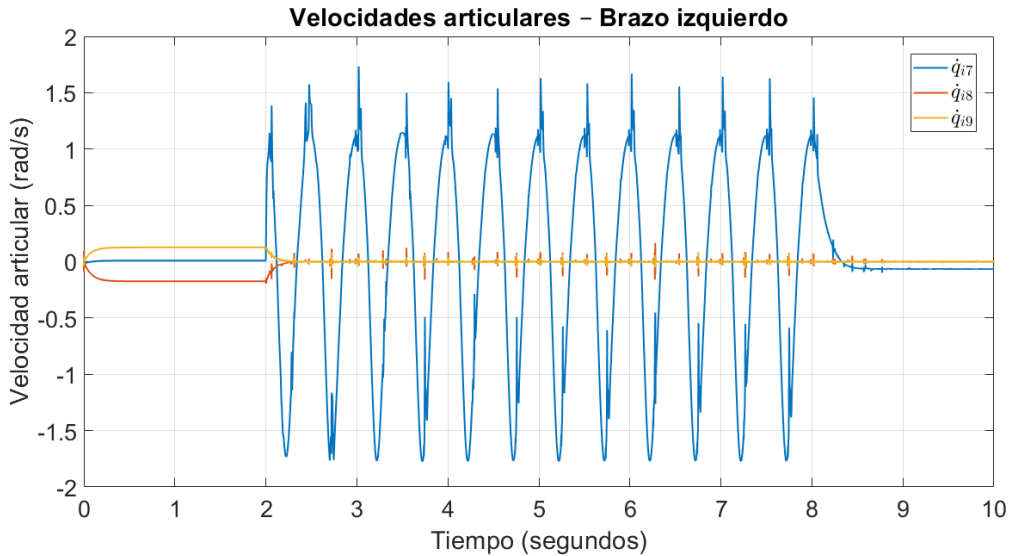


Figura 34. Velocidades articulares del brazo izquierdo.

El comportamiento de los pares articulares, presentado en las Figuras 35 a 38, refleja fielmente las demandas dinámicas variables a lo largo de las tres etapas del movimiento programado y demuestra la capacidad del esquema de control saturado para gestionar estas cargas sin exceder los límites operativos.

En las articulaciones de las piernas (Figura 35 para la derecha y Figura 36 para la izquierda), se observan las magnitudes de par más significativas, como es esperado debido a que estas articulaciones soportan el peso total del robot y las fuerzas de reacción del suelo. Durante las fases estáticas inicial (0 a 2s) y final (8 a 10s), los torques convergen a valores constantes no nulos, necesarios para compensar la gravedad y mantener la postura de

equilibrio. En contraste, la etapa de marcha dinámica (2 a 8s) presenta demandas de par altas y cíclicas, con picos pronunciados (notablemente en q_{d3} , q_{d4} y sus contrapartes izquierdas) que corresponden a las fases de soporte, impacto del talón y despegue del pie. Es importante notar que, a pesar de la alta demanda, las señales de control permanecen suaves y acotadas, sin mostrar saturación abrupta ni vibraciones de alta frecuencia, lo que valida la efectividad de la acción de control suave proporcionada por la función de saturación.

Por otro lado, los torques en los brazos (Figura 37 derecha y Figura 38 izquierda) son considerablemente menores en magnitud comparados con los de las piernas. Su comportamiento durante la marcha (2 a 8s) es oscilatorio y suave, cumpliendo la función de conservación del momento angular y estabilización del balanceo, en lugar de soporte de carga. La simetría desfasada observable entre las extremidades derechas e izquierdas confirma la correcta ejecución de un patrón de marcha bípeda coordinado. En conjunto, los resultados indican que los actuadores operan dentro de un rango seguro y que el controlador gestiona adecuadamente las transiciones entre el par estático requerido para el equilibrio y el par dinámico necesario para la locomoción.

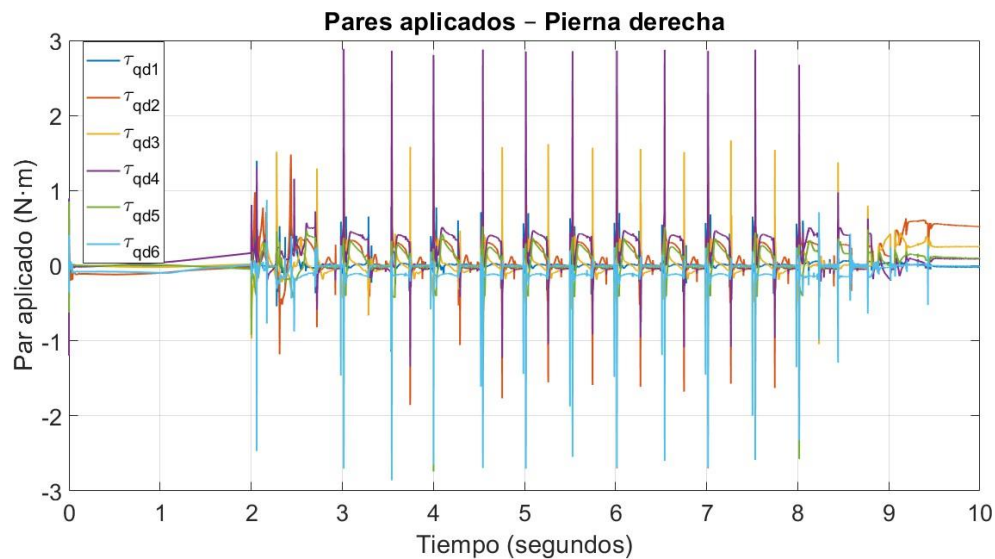


Figura 35. Pares aplicados a las articulaciones de la pierna derecha.

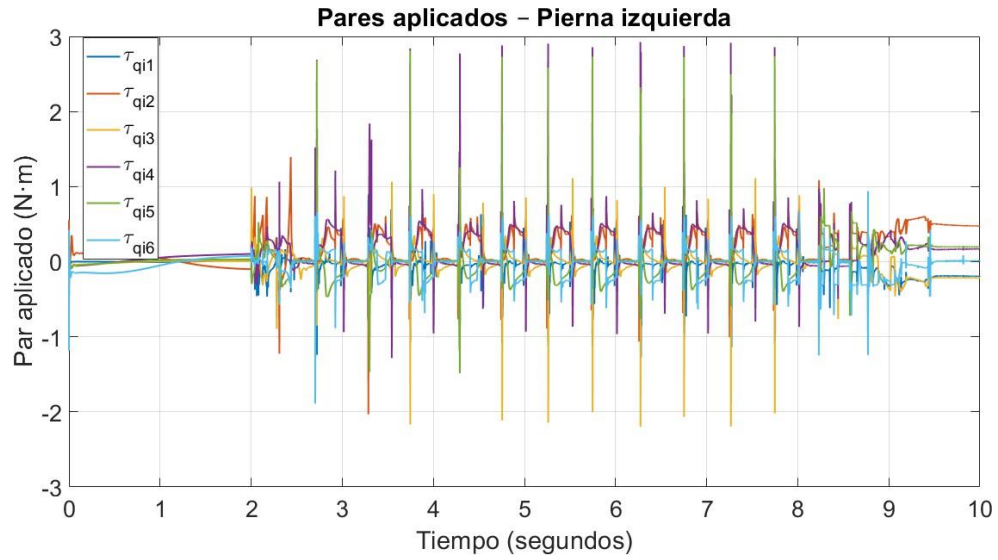


Figura 36. Pares aplicados a las articulaciones de la pierna izquierda.

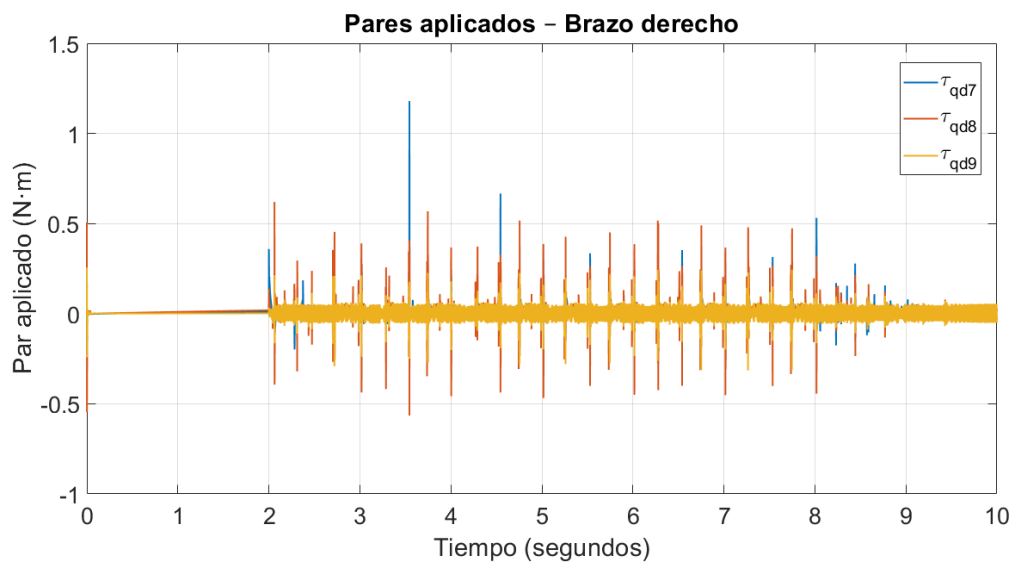


Figura 37. Pares aplicados a las articulaciones del brazo derecho.

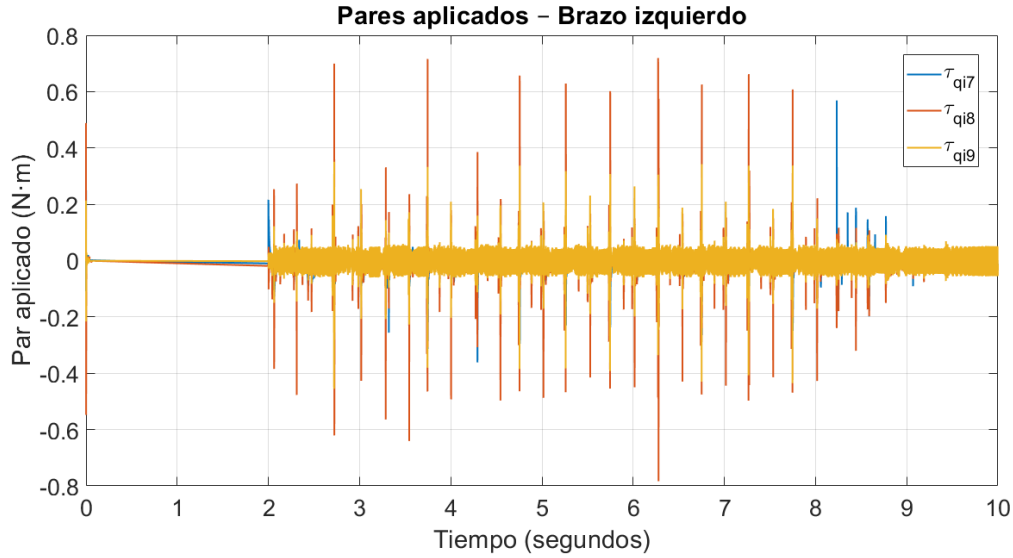


Figura 38. Pares aplicados a las articulaciones del brazo izquierdo.

6. Conclusiones y trabajo a futuro

El método analítico basado en la formulación de Euler demostró ser una solución efectiva y robusta para resolver el problema de modelado dinámico del robot Bioloid Premium de 18 grados de libertad (GDL). La implementación del enfoque recursivo permitió gestionar la complejidad computacional de manera significativamente más eficiente que los métodos convencionales.

Los resultados de las pruebas de rendimiento computacional validaron que, mientras la derivación simbólica clásica exhibe un crecimiento casi cuadrático, la metodología recursiva propuesta mantiene una complejidad lineal $O(n)$ respecto al número de grados de libertad. Esto se tradujo en tiempos de cómputo drásticamente reducidos para la obtención de la cinemática, jacobianos y matrices de inercia, confirmando la escalabilidad del algoritmo para sistemas de alta complejidad.

La novedad de este trabajo radica en la aplicación inédita de esta formulación recursiva para derivar, implementar y validar integralmente el modelo dinámico completo de la plataforma Bioloid, un aporte no documentado previamente en la literatura para este robot específico. Al explotar las propiedades de la matriz antisimétrica $\underline{S}$ y evitar el cálculo explícito de derivadas analíticas parciales, se logró un marco de modelado apto para aplicaciones en tiempo real.

La validez del modelo dinámico obtenido se corroboró mediante un entorno de física virtual en Simscape Multibody. El análisis de los pares articulares durante una secuencia de marcha dinámica arrojó valores de Error Cuadrático Medio (RMSE) consistentemente bajos en todas las articulaciones, demostrando una alta fidelidad entre las predicciones del modelo matemático y la respuesta física simulada.

Adicionalmente, el estudio evaluó la aplicabilidad del modelo en esquemas de control de movimiento. A través de la métrica de la norma $\mathcal{L}_2$, se determinó que la ley de control saturado superó tanto al controlador PD convencional como al basado en tangente hiperbólica. El control saturado demostró ser la estrategia más eficaz para gestionar las no-linealidades y las demandas de par durante las fases críticas de impacto y soporte en la locomoción bípeda, garantizando la estabilidad global y minimizando el error de seguimiento sin exceder los límites físicos de los actuadores.

Este trabajo sienta bases sólidas para el desarrollo de algoritmos eficientes en robótica humanoide. En cuanto al trabajo futuro, se proponen las siguientes líneas de investigación: Implementación física para implementar los algoritmos de control validados en simulación al hardware real de un robot que pueda ser diseñado con características similares a las del Bioloid Premium, abordando los retos de comunicación y procesamiento embebido. Estrategias de equilibrio dinámico para integrar el modelo dinámico recursivo con criterios de estabilidad como el Punto de Momento Cero (ZMP) o el Modelo de Péndulo Invertido Lineal (LIPM) para generar marchas robustas ante perturbaciones externas. Control de par computado con el objetivo de utilizar la eficiencia del cálculo de la dinámica inversa para implementar esquemas de control de par computado (Computed Torque Control) en tiempo real, mejorando la suavidad y precisión del movimiento.

7. Referencias

- [1] Leng, X., Piao, S., Chang, L., He, Z., & Zhu, Z. (2020). Universal Walking Control Framework of Biped Robot Based on Dynamic Model and Quadratic Programming Complexity.
- [2] Rojas, J. H. C., Hernández, R. D. G., & Quintero, W. R. (2009). Proyecto mecatrónico de brazo robot cartesiano integrado a una celda de almacenamiento y recuperación automatizada AS/RS de un Sistema Flexible de Manufactura FMS. *ITECKNE*, 6(1), 35-49.
- [3] Narváez-Aroche, O., Rocha-Cózatl, E., & Cuenca-Jiménez, F. (2011). Modelo dinámico de un robot bípedo de 12 GDL internos. In *Memorias del XVII Congreso Internacional Anual de la SOMIM* (pp. 992-1001).
- [4] Hiraes, D. A. (2017) Modelado y control de un robot bípedo basado en conceptos de energía.
- [5] Machado, J. E. (2014). Modelación y Control de un Robot Bípedo (Doctoral dissertation, Centro de Investigación en Matemáticas).
- [6] Bravo Montenegro, D. A. (2016). Generación de trayectorias para un robot bípedo basadas en captura de movimiento humano.
- [7] Velázquez, J. G. R., Díaz, E. O., & Díaz, A. M. (2009). Modelo dinámico de un robot bípedo planar de 9 grados de libertad en fase de vuelo y fases de contacto.
- [8] Aguilar, D., & Vivas, A. (2009). Modelado y control de un robot bípedo de nueve grados de libertad. In *VIII Congreso de la Asociación Colombiana de Automática*.
- [9] Domínguez, C. A. M., & Sánchez, E. M. (2017). Análisis estático y dinámico de un robot bípedo durante la fase de soporte simple de un ciclo de marcha. In *Memorias del XXIII Congreso Internacional Anual de la SOMIM*.
- [10] Martínez, I. C. (2008). Análisis cinemático y dinámico de un robot bípedo.
- [11] Rodríguez Rodríguez, J. S. (2017). Modelado matemático de un robot bípedo con equilibrio dinámico. Facultad de Ingeniería.
- [12] Shih, C. L., & Gruver, W. A. (1992). Control of a biped robot in the double-support phase. *IEEE transactions on systems, man, and cybernetics*, 22(4), 729-735.
- [13] Kun, A., & Miller, W. T. (1996). Adaptive dynamic balance of a biped robot using neural networks. In *Proceedings of IEEE international conference on robotics and automation* (Vol. 1, pp. 240-245). IEEE.
- [14] Kajita, S., Benallegue, M., Cisneros, R., Sakaguchi, T., Nakaoka, S. I., Morisawa, M., ... & Kanehiro, F. (2018). Biped gait control based on spatially quantized dynamics. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (pp. 75-81). IEEE.
- [15] Shigemi, S., Goswami, A., & Vadakkepat, P. (2018). ASIMO and humanoid robot research at Honda. *Humanoid robotics: A reference*, 55-90.
- [16] Suleiman, W., Kanehiro, F., Miura, K., & Yoshida, E. (2011). Enhancing zero moment point-based control model: system identification approach. *Advanced Robotics*, 25(3-4), 427-446.
- [17] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation* (Cat. No. 03CH37422) (Vol. 2, pp. 1620-1626). IEEE.

- [18] Kaneko, K., Kanehiro, F., Morisawa, M., Miura, K., Nakaoka, S. I., & Kajita, S. (2009). Cybernetic human HRP-4C. In 2009 9th IEEE-RAS International Conference on Humanoid Robots (pp. 7-14). IEEE.
- [19] Hein, D., Hild, M., Berger, R. (2008) "Evolution of biped walking using neural oscillators and physical simulation", RoboCup 2007: Robot Soccer World Cup XI, Springer Berlin, Heilderberg.
- [20] Eaton, M., Evolutionary humanoid Robotics: Past, Present and Future (2007). 50 Years of Artificial Intelligence, Springer Berlin, Heilderberg.
- [21] Wolf, J.C., Hall, P., Culverhouse, P., "Bioloid based Humanoid Soccer Robot Design", Proc. Of the 2nd IEEE/RAS International Workshop on humanoid Soccer Robots, U.S.A. (2007).
- [22] Teodoro, P. (2007). Humanoid robot: Development of a simulation environment of an entertainment humanoid robot. Instituto Superior Tecnico Universidad Tecnica de Lisboa, Lisboa, 1-15.
- [23] Bravo Montenegro, D. A. (2016). Generación de Trayectorias para un Robot Bípedo basadas en Captura de Movimiento Humano (Doctoral dissertation).
- [24] Kaneko, K., Kaminaga, H., Sakaguchi, T., Kajita, S., Morisawa, M., Kumagai, I., & Kanehiro, F. (2019). Humanoid robot HRP-5P: An electrically actuated humanoid robot with high-power and wide-range joints. IEEE Robotics and Automation Letters, 4(2), 1431-1438.
- [25] Cortés, F. R. (2012). Matlab aplicado Robótica y Mecatrónica. México DF, México: Grupo Editorial Alfaomega.
- [26] Jazar, R. N. (2010). Theory of applied robotics: Kinematics, dynamics, and control (2nd Edition). New York, US: Springer.
- [27] Kelly, R. and Santibáñez, V. (2003). Contol de Movimiento de Robots Manipuladores. Madrid, España: Pearson Prentice–Hall.
- [28] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). Robotics: Modelling, Planning and Control. Springer.

8. Publicaciones

INVERSE KINEMATICS MODEL FOR A 18 DEGREES OF FREEDOM ROBOT

Submitted: 1st September 2022; accepted: 17th July 2023

Miguel Angel Ortega-Palacios, Amparo Dora Palomino-Merino, Fernando Reyes-Cortes

DOI: 10.14313/IAMRIS/1-2023/3

Abstract:

The study of humanoid robots is still a challenge for the scientific community, although there are several related works in this area, several limitations have been found in the literature that drive the need to develop an inverse kinematic modeling of biped robots. This paper presents a research proposal for the Bioloid Premium robot. The objective is to propose a complete solution to the inverse kinematics model for a 18 DOF (Degrees Of Freedom) biped robot. This model will serve as a starting point to obtain the dynamic model of the robot in a subsequent work. The proposed methodology can be extended to other biped robots.

Keywords: *bioloid premium robot, forward kinematics, inverse kinematic, kinematic chain.*

1. Introduction

The problem of study related to the kinematics of biped robots has been widely studied in the scientific community; in the literature have been found several limitations in the models of biped robot kinematics. This drives the need to develop an inverse kinematics model for the Bioloid robot of 18 DOF. Specifically, the problem of the lack of study of the kinematics of the upper train in biped robots arises [1-13, 19-26]. Due to the high number of degrees of freedom and the complexity involved in the calculation of the inverse and forward kinematics equations, most authors have the objective of modeling only the lower train of the robots, either using commercial robots such as Nao with 12 DOF legs [1], HYDROID which has 8 active DOF per leg [2], Scout [3] and NWPUBR-1 [4] with 12 DOF legs, Ostrich Bionic with 13 DOF legs [5], Cassie with 20 DOF legs [6], or robots wich are author's design with 12 DOF [7-9] 10 DOF [10-12] and 9 DOF [13]. All of these research papers calculate the forward kinematics model by taking one of the robot's feet as supporting foot.

In other works, it is possible to obtain the forward and inverse kinematics solution for both legs and arms, using the HRP-2 robot with 12 DOF legs [14], DARwin-OP with 6 DOF per leg [15], AXIS with 12 DOF legs [16], NAO with 21 DOF [17], Digit robot with 20 DOF [18], but these models propose the torso or pelvis of the robot as the initial frame.

The Bioloid robot has been used by the scientific community to perform several studies related to kinematics, dynamics and control. Most of the works

obtain the kinematic model of the legs, taking into account only one foot as the initial frame [19-24]; proposing two different cases where the supporting foot is either the right or the left foot [25], in [26] the torso is taken as the initial frame. In [27] the kinematic model of the robot legs and arms is obtained but uses the torso and pelvis as initial frames.

All the works mentioned previously calculate the kinematic modeling considering the Denavit-Hartenberg method to represent the position and orientation of the end-effector.

On the other hand, the authors have not established a complete inverse kinematic model for a 18 DOF bipedal robot. Therefore, the Bioloid Premium robot with 18 DOF is proposed as a study target. The main motivation in this paper is to develop a methodology based on the Denavit-Hartenberg method to obtain the forward and inverse kinematic model for a 18 DOF Bioloid Premium robot.

In the present work we propose to obtain the complete kinematic model of the Bioloid robot, considering four open kinematic chains, where the initial frames are the support feet, and we have the left and right pelvis as end-effector frames; the pelvis is also proposed as another initial frame to have the left and right hand as the other end-effector frames.

The paper is organized as follows. In Section 2 the Denavit-Hartenberg method is applied to calculate the geometric parameters of the robot. In Section 3, forward kinematic model is obtained. The equations of inverse kinematics of the robot are computed in Section 4. Finally, the conclusions are given in Section 5.

2. Denavit-Hartenberg Parameters

The key idea is to generate four open kinematic chains to describe the position and orientation of each link of the Bioloid Premium robot. Using the Denavit-Hartenberg method, the frames and parameters of each link, as well as the position and orientation of each joint of the robot are presented in Figure 1.

We can observe that the supporting right and left feet are proposed as the initial frames $\Sigma_{d0}(x_{d0}, y_{d0}, z_{d0})$ and $\Sigma_{l0}(x_{l0}, y_{l0}, z_{l0})$, then the first two kinematic chains goes up to the pelvis frame, from this point three open kinematic chains can be considered, one of them has the left foot end-effector frame $\Sigma_{12}(x_{12}, y_{12}, z_{12})$, while the second chain takes into account the right hand end-effector frame $\Sigma_{d3}(x_{d3}, y_{d3}, z_{d3})$ and finally, the third chain considers the left hand end-effector frame $\Sigma_{l3}(x_{l3}, y_{l3}, z_{l3})$.

Modeling of an 18 DoF Humanoid Robot Using a Recursive Analytical Method

Miguel Angel Ortega-Palacios , Amparo Palomino-Merino , and Fernando Reyes-Cortés 

Abstract— Humanoid robots have been widely studied by the scientific community to understand and reproduce human motion in a variety of tasks. Over the past decades, numerous modeling strategies have been proposed for describing humanoid dynamics, including analytical formulations, geometric modeling, and the Denavit-Hartenberg convention. In this work, a recursive formulation based on Euler's analytical mechanics is presented, and leverages the properties of the skew-symmetric matrix (S) to address key modeling challenges, including forward kinematics, differential kinematics, linear and angular velocity computation, kinetic and potential energy calculation, and the Euler-Lagrange equations of an 18-degree-of-freedom (DoF) humanoid robot. Furthermore, three position controllers—Proportional-Derivative (PD), Hyperbolic Tangent, and Saturation-based control—are implemented and evaluated using the virtual model of robot. Their performance is validated in simulation using a virtual model of the humanoid robot developed in Simscape Multibody, where a dynamic walking gait is executed. A comparative analysis is presented to assess tracking performance for each controller. Finally, the simulation results in MATLAB and Simscape demonstrate the applicability and computational advantages of the proposed modeling and control framework.

Link to graphical and video abstracts, and to code: <https://latam.tiiceer9.org/index.php/transactions/article/view/10058>

Index Terms— Dynamics, Euler, humanoid robot, recursive method, skew-symmetric matrix.

I. INTRODUCCIÓN

THE study of the dynamics of biped robots has been extensively investigated in the scientific community. However, several limitations have been identified in the modeling approaches currently available in the literature, highlighting the need for alternative formulations. In this regard, we propose the use of Euler's analytical mechanics, which allows the kinematic and dynamic modeling to be addressed through a recursive approach. This is particularly relevant from a computational standpoint, as the development of a recursive algorithm capable of obtaining the dynamic model of any biped robot is envisioned for future work.

The modeling of humanoid robots has been approached

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Moreno-Valenzuela (Corresponding author: Miguel Angel Ortega Palacios).

Miguel Angel Ortega Palacios, A. Palomino-Merino, and F. Reyes-Cortés are with the Benemérita Universidad Autónoma de Puebla, Puebla, México (e-mails: miguel.ortega@alumno.buap.mx, amparo.palomino@correo.buap.mx, and fernando.reyes@correo.buap.mx).

through dynamic analysis [1–3]. Most of the reviewed works apply the Euler-Lagrange methodology [1, 4–9] to derive the dynamic model, while others rely on the Newton-Euler formulation [10–13]. All of them aim to implement stable gait patterns using control techniques that ensure robot balance during locomotion, such as PD control strategies [14–16].

It is important to note that most authors focus primarily on modeling the dynamics of the legs, occasionally including the torso, with the main goal of achieving stable walking trajectories.

Kajita's work [17] extensively studied the HRP humanoid robot series from 2000 to the present. The objective is to deploy practical humanoid robots in large-scale assembly industries such as construction, aerospace, and shipbuilding. Nevertheless, practical deployment in real environments remains unachieved, largely due to the insufficient physical capabilities of current HRP robots.

The development of biped gait models continues to be an active research area. The challenge of humanoid control is often simplified using an inverted pendulum approach. This model not only predicts the humanoid robot's ZMP trajectory without requiring full multibody dynamics but also generates the CoM trajectory for ZMP tracking. This approach has been implemented in the HRP-4C humanoid robot [18].

Despite the extensive research on humanoid robot dynamics, finding studies that explicitly develop the complete dynamic model for humanoid robots with 18 degrees of freedom (DoF) remains challenging. Most recent works focus on general whole-body dynamics or simplified models with fewer joints (e.g., 6-DoF or 12-DoF systems), while others address higher-DoF structures without publishing the full derivation of their dynamic equations. Recursive formulations, such as the Newton-Euler or Euler-Lagrange approaches, are often applied to subsystems or simplified kinematic chains, leaving few references that detail the full 18-DoF configuration. The limited availability of such studies is mainly due to the computational complexity and the numerical instability that arise when modeling and solving the coupled dynamic equations of multibody humanoid systems. One of the few known works addressing an 18-DoF humanoid robot [19], which focuses on gait generation rather than on the analytical derivation of the complete dynamic model. Therefore, the development of an explicit and computationally efficient dynamic model for a humanoid robot with 18 DoF—particularly through recursive and matrix-based methods such as the skew-symmetric formulation—represents a relevant and still underexplored