CENTRO DE INVESTIGACIÓN
Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE MECATRÓNICA

Control óptimo de sistemas mecatrónicos basado en datos.

Tesis que presenta:
Luis Eduardo Solís Herrera.

Para obtener el grado de:
Maestro en Ciencias.

En la especialidad de:
Ingeniería Eléctrica.

Directores de Tesis:
Dr. Carlos Alberto Cruz Villar
Dr. Héctor Cervantes Culebro.

Ciudad de México Febrero, 2023

Agradecimientos

A mis padres: porque gracias a su excelente guía, apoyo, cariño incondicional y enseñarme valores humanos contribuyendo al legado más grande que pudiera recibir y por el cual viviré eternamente agradecido.

A mis hermanos y mis sobrinas: por su confianza e interés, por su cálido cariño y curiosidad en mi trabajo cuando estábamos confinados por la pandemia.

Al Dr. Carlos Alberto Cruz Villar: por la excelente guía y libertad otorgada en el desarrollo de éste trabajo, sus comentarios críticos complementaron mis ideas y me invitaron a la reflexión, además, de todas las conversaciones de vida que sin duda sumaron a mi desarrollo integral.

Al Dr. Héctor Cervantes Culebro: por su excelente guía y libertad otorgada en el desarrollo de éste trabajo, por sus comentarios críticos de gran aportación y por recibirme en una estancia corta en el laboratorio del Instituto Tecnológico y de Estudios Superiores de Monterrey; sin dudarlo dicha estancia fue importante en el desarrollo de la plataforma experimental.

A los miembros del jurado: por el tiempo dedicado a la lectura del documento, sus valiosos comentarios, críticas constructivas y correcciones en este trabajo que sin duda fueron de gran valor.

A mis compañeros de generación: que colaboramos de forma parcial en clases virtuales y de manera presencial a la mitad de nuestro trabajo de tesis, con quienes desarrollé una excelente amistad y sé que perdurará durante mucho tiempo.

A mis compañeros de doctorado Arturo y Daniel: que gracias a sus consejos y puntos de vista, ayudaron en complementar mi formación, pero principalmente por siempre estar dispuestos en ayudar y compartir su experiencia y conocimiento.

A mis amigos de Preparatoria: por su amistad de tanto tiempo, por las excelentes personas que son y por su éxito profesional, mantener su hermandad en este proceso y poder compartirles de forma verbal mi trabajo fue muy importante en la búsqueda del equilibrio personal.

Al CONACYT: Por haberme otorgado una beca para realizar mis estudios de maestría en el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV).

Resumen

En este trabajo de tesis, se aborda el diseño e implementación de un controlador basado en aquellos datos que se obtienen con técnicas de procesamiento de señales, permitiendo a cualquier sistema mecatrónico, estimar su dinámica y lograr el comportamiento deseado. Dicho enfoque de control se conoce como Deep Reinforcement Learning, que es el uso del enfoque de aprendizaje por refuerzo (Reinforcement Learning), potencializado con la utilización de redes neuronales informáticas (Deep Learning).

En la revisión de la literatura, se identifica que la comunidad de investigadores, propusieron diferentes metodologías de diseño de controladores en sistemas mecatrónicos, según sea la tarea a realizar y su dinámica particular. Sin embargo, todas estas dependen de un modelado preciso, aumentando la complejidad del diseño e implementación de un controlador, que sea capaz de funcionar adecuadamente ante ambientes cambiantes con incertidumbre paramétrica. Por ello, en el trabajo que se desarrolló, se eligió introducir una metodología general del diseño del controlador basado en aquellos datos que puede obtenerse en un sistema instrumentado. Con el enfoque de Deep Reinforcement Learning, de forma iterativa, un agente computacional es capaz de estimar la dinámica del sistema explorando posibles acciones para un controlador y satisfacer la tarea a realizar.

En este trabajo, se propone el diseño, construcción y control mediante Reinforcement Learning de un robot bípedo, sin rodillas, con dos grados de libertad actuados, capaz de realizar un ciclo de caminado semi-pasivo en suelo horizontal. Dicho sistema se compone de la actuación de un péndulo en el plano transversal que permite desplazar el centro de masa y generar movimiento transitorio en la cadera; un segundo actuador, genera el movimiento articular en la pierna. El ciclo de marcha se logra si se puede sincronizar la conmutación de accionamiento entre actuadores y sus cambios de sentido de giro, esto dependerá de la fase instantánea en el caminado realizada por el robot bípedo.

Del robot bípedo, se obtuvo su modelo dinámico con el propósito de tener mayor comprensión de la complejidad que conllevaba diseñar un controlador basado en el modelo. Con el resultado anterior, se abordó el problema de locomoción con una estrategia de diseño de controlador donde nos permita prescindir del modelo. Para disminuir la complejidad del aprendizaje, el tiempo y número de iteraciones, se combina Deep Reinforcement Learning con la utilización de máquinas de estados para sincronizar varias subtareas y lograr el trabajo conmutado que nos permita completar el ciclo de caminado semi-pasivo.

Por último se presentan y se analizan los resultados experimentales del ciclo de caminado del robot bípedo logrado con el enfoque de Reinforcement Learning.

Abstract

This thesis work addresses the design and implementation of a data-driven controller obtained with signal processing techniques, allowing any mechatronic system to estimate its dynamics and achieve the desired behavior. This control approach is known as Deep Reinforcement Learning, which is the use of the Reinforcement Learning approach, enhanced with the use of computer neural networks (Deep Learning).

In the literature review, it is identified that the research community proposed different methodologies for the design of controllers in mechatronic systems, depending on the task to be performed and its particular dynamics. However, all these depend on an accurate modeling, increasing the complexity of the desing and implementation of a controller that is able to perform adequately in changing environments with parametric uncertainty. Therefore, in this work, it was chosen to introduce a general methodology of the controller desing based on those data that can be obtained in an instrumented system. With the Deep Reinforcement Learning approach, iteratively, a computational agent is able to estimate the dynamics of the system exploring possible actions for a controller and satisfy the task to be performed.

In this work, we propose the design, construction and control by Reinforcement Learning of a biped robot, without knees, with two actuated degrees of freedom, capable of performing a semi-passive walking cycle on horizontal ground. This system is composed of the actuation of a pendulum in the transverse plane that allows to move the center of mass and generate transient motion in the hip; a second actuator generates the joint motion in the leg. The walking cycle is achieved if the synchronize between actuators and their changes of direction of rotation, this will depend on the instantaneous walking phase performed by the biped robot.

From the bipedal robot, its dynamic model is obtained in order to have a better understanding of the complexity involved in designing a controller based on the model. With the previous result, the locomotion problem is addressed with a controller design strategy that allows us to do without the model. To reduce the complexity of learning, the time and number of iterations, Deep Reinforcement Learning is combined with the use of state machines to synchronize various subtasks and achieve the commuted task that allows us to complete the semi-passive walking cycle.

Finally, the experimental results of the walking cycle of the bipedal robot achieved with the Reinforcement Learning approach are shown and analyzed.

Contenido

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En el presente capítulo se describen las generalidades del trabajo de tesis. En la sección 1.1 se detalla todas aquellas aportaciones ya documentadas en sistemas mecatrónicos utilizando algún control basado en datos. Para la sección 1.2 se plantea la problemática que se busca resolver en el presente trabajo de tesis. En la sección 1.3 se retoma el problema planteado y se enuncia el objetivo general y los objetivos particulares. Para la sección 1.4 se describe las aportaciones que se obtuvieron en este trabajo de tesis y por último, para la sección 1.5 se da una descripción de la organización de todo el documento.

1.1. Estado del Arte

El diseño de estrategias de control basado en datos para sistemas mecatrónicos es un tema de reciente relevancia, comenzó a proponerse aproximadamente en el año 2003, pero desde el año 2016 con la influencia de la ciencia de datos, industria 4.0 y el aprendizaje automático ha impulsado este enfoque [4]. Su objetivo principal es la capacidad de estimar modelos dinámicos proporcionados por la manipulación de datos obtenidos en el sistema.

El control de sistemas dinámicos basados en datos podemos clasificarlos en dos principales vertientes [5]

1. Representación teórica de operadores infinitesimales de Koopman.
2. Aprendizaje automático

Cabe aclarar de forma explícita que este trabajo se ha desarrollado únicamente con el enfoque de control basado en datos a partir del algoritmo de aprendizaje automático conocido como Deep Reinforcement Learing (DRL). DRL consiste en la implementación de un agente computacional capaz de estimar dinámicas de un sistema sin requerir un modelo que lo describa. Su obtención es por medió de un proceso iterativo de prueba y error, donde se propone acciones al sistema y se evalúa su desempeño. A largo plazo, de cada intento que realicé, se estima que conjunto de acciones permiten lograr la tarea a realizar.

El enfoque de aprendizaje automático se extiende en aplicaciones de mecatrónica, control automático, telecomunicaciones, neurociencias, sistemas de transporte, aeroespacial, entre otros. La principal razón con la que se comenzó a optar por este enfoque, fue prescindir del modelo dinámico de forma total o parcial al momento de diseñar controladores, ya que en la gran mayoría de las ocasiones, la manipulación matemática suele requerir sistemas computacionales y de difícil implementación en un controlador. En [6] se propone la estimación de la dinámica para el acoplamiento entre un manipulador y la nave espacial, esté trabajo prescinde del modelo dinámico sustituyéndolo por un controlador basado en la telemetría de trayectorias efectuadas en

anteriores misiones espaciales. Además del acoplamiento, como resultado de implementación del enfoque, se pudo suprimir las vibraciones que se generaban entre estructuras en el momento de la maniobra. En poco tiempo, el enfoque de aprendizaje automático de Reinforcement Learning sin modelo fue potencializado por el uso de enfoque Deep Learning, en los trabajos de [7] y de [8] se ejemplifica el uso de Deep Reinforcement Learning libre de modelo aplicado a vehículos marinos con agentes computacionales tipo actor-crítico [1], estos primeros trabajos validaron sus objetivos de control en simuladores. Los simuladores son de gran utilidad para realizar aplicaciones de este enfoque, principalmente para investigaciones en aplicaciones de sistemas de transporte, en [9], [10] y [11] describen como el enfoque ayudó a que los vehículos sean capaces de conducir en forma autónoma en carreteras desconocidas, previniendo situaciones extraordinarias como choques, cambio de carril o presencia de tráfico, todo esto utilizando datos recopilados previamente por ejercicios de manejo hechos por humanos. Sin embargo, el uso de simuladores y su verdadera eficacia en sistemas reales dependen de la complejidad de la tarea a realizar y lo más cercano que este sea a un gemelo digital, donde contemple las dinámicas del entorno como gravedad, presión, velocidades de viento o fricción del suelo por mencionar algunos ejemplos; aún hay un campo de investigación enfocado a los vehículos autónomos y resolver la mayoría de las situaciones dinámicas que se presentan en la conducción.

Para la reducción de la brecha existente entre simuladores y los sistemas físicos, en [12] se utilizó lo que se conoce como *modelado diferencial*, dicho enfoque reduce la demanda de datos reales, aprendiendo únicamente aquellas dinámicas donde existan diferencias mayores a la tolerancia permitida de la comparación entre el simulador y el sistema físico. Al poder comparar resultados entre simuladores y sistemas físicos, en [13] realizaron un estudio que permitía la evaluación entre el enfoque de DRL con metodologías de control basadas por modelo como un controlador predictivo no lineal, se validaron las respuestas de cada enfoque en un sistema de carro con péndulo invertido e incertidumbres paramétricas; se concluyó que Deep Reinforcement Learning funciona de mejor manera que el controlador predictivo no lineal. También en [14], comparan el desempeño de un controlador robusto de histéresis contra un controlador por Reinforcement Learning para un sistema de calefacción de suelo por radiación, de igual forma se concluyó que el controlador por RL tiene mejor desempeño. Estos trabajos además de la comparación entre controladores, generaron motivación para estudiar el efecto combinado de Reinforcement Learning y controladores clásicos. Se han hecho esfuerzos de desarrollar controles híbridos que combinen el aprendizaje por refuerzo y alguna metodología de control clásico, en [15] se detalla la teoría y los teoremas necesarios para sinergia entre Reinforcement Learning y control por modos deslizantes, la complejidad matemática en las operaciones fue alta, por lo que no se logró realizar una validación experimental. En [16] se muestra la combinación del aprendizaje por refuerzo y control predictivo por modelo, en este trabajo enfatizaron que la ventaja de implementación fue la reducción de iteraciones durante el entrenamiento, es decir, se logra convergencia con mayor rapidez. Por último, en [17] utiliza la combinación de Deep Reinforcement Learning y control adaptativo para mejorar la precisión de seguimiento de trayectoria ante perturbaciones en un brazo robótico industrial.

En años recientes, a partir del 2021, además de buscar enfoques que combinen el aprendizaje por refuerzo y controladores provenientes de la teoría de control clásica, como solución al desempeño de los sistemas, se ha propuesto para tareas que requieran trabajo conmutado, un enfoque que combina Reinforcement Learning con la teoría de máquinas de estados. En [18] propone la estructura de la función de recompensa para máquinas de estados finitos, con agentes que aceptan redes neuronales y acciones de tiempo discreto. En [19] describe la forma de realizar estimaciones de acciones futuras, utilizando Reinforcement Learning y una máquina de estados probabilista, el enfoque ayudó a optimizar el número de estados necesarios, reduciendo el tiempo

[1] Agentes computacionales que acepta de redes neuronales, para más detalles ver Tabla 2.5

de ejecución computacional y satisfacer la tarea de control; dicho trabajo fue validado en simulación para robots tipo enjambre como drones. Para vehículos autónomos, en [20] el enfoque se utiliza como alternativa para el aprendizaje en la conducción, tratando de proveer acciones para todas las posibles situaciones que se presenten. En [21], con un esquema híbrido de máquinas de estados con DRL, se estima y sistematiza el proceso de cambio de carril mejorando la toma de decisiones de cuando se debe realizar dicha acción.

Poco a poco, el aprendizaje por refuerzo se ha implementado en sistemas de alta dimensionalidad, aumentando el número robots a controlar mientras trabajan de forma colaborativa. En [22] se evaluó el rendimiento del agente computacional PPO [2] verificando la cantidad de información que pueden compartirse si se varía el número de robots a utilizar; se logró compartir información de hasta 10 robots bípedos. Otra tarea de alta dimensionalidad y complejidad es la teleoperación, en [23] se describe el entrenamiento por Reinforcement Learning con teleoperación para un sistema maestro-esclavo, se comprobó que el retardo de tiempo existente afecta la respuesta de acciones del sistema esclavo. Además, se concluyé que aquellos entrenamientos donde se proporciona un modelo parcial del entorno, tendrán menores afectaciones que los entrenamientos sin modelo. Para resolver el problema de afectación en el sistema esclavo, en [24] se utiliza un compensador de identificación difusa que ayuda a retroalimentar la variable de fuerza al sistema maestro sin que este identifique los retrasos. El tema de teleoperación con aprendizaje automático sigue siendo un tema de investigación actual.

Una desventaja del enfoque de Reinforcement Learning es que dependiendo de la complejidad de la dinámica del sistema, aumenta el número de intentos que tendrá que realizar para que logre su estimación. En teleoperación, en los trabajos de [23] y [24], cada sistema requirió más de 100,000 intentos, esto es porque el algoritmo requiere de una etapa de exploración de acciones; si el sistema es complejo requerirá mayor número de intentos para lograr estimar la dinámica, por lo tanto, el entrenamiento puede ser un proceso largo. Muchas veces, durante la etapa de exploración del entrenamiento, el agente propone acciones de alto riesgo para el sistema, ocasionando daños físicos en ellos. En [25] se realizó un estudio completo de fatiga en la caja de engranes de robot bípedo y de las caídas en la etapa de exploración; se evaluó entre los agentes DDPG [3], TD3 [4] y PPO, cuál de estos ocasionaría menor daño a la caja de engrane, se concluyó que el agente TD3 ocasiona menor daño. Para contrarrestar el riesgo de daño físico de los sistemas a controlar, [26] propone un enfoque de seguridad llamado *Safe Reinforcement Learning* donde el agente genere datos con algoritmos de estimación, y de forma paralela, proponer acciones al sistema que sean seguras desde el comienzo del entrenamiento.

En los últimos dos años, se identifica que la línea de investigación para el control basado en datos por aprendizaje automático se enfoca en las siguientes vertientes:

1. mejorar el algoritmo de Reinforcement Learning a través de algoritmos evolutivos, mejorando la rapidez de convergencia de una solución.

2. mejorar la respuesta del sistema en el cumplimiento de tareas multiobjetivo.

Para el enfoque de mejora del algoritmo de Reinforcement Learning se ha optado por la combinación con algoritmos evolutivos. En [27] se propone el uso del algoritmo Surrogate con Deep Reinforcement Learning, este permitió disminuir el tiempo de entrenamiento principalmente para un sistema de alta dimensión. En [28] utiliza un algoritmo independiente para estimar los

[2]Proximal Policy Optimization (PPO).- agente computacional utilizado en Deep Reinforcement Learning. Para más detalles ver Tabla 2.7

[3]Deep Deterministic Policy Gradient (DDPG).- agente computacional utilizado en Deep Reinforcement Learning. Para más detalles ver Tabla 2.8

[4]Twind-Delayed Deep Deterministic Policy Gradient (TD3).- agente computacional utilizado en Deep Reinforcement Learning. Para más detalles ver Tabla 2.8

estados futuros para un control predictivo, por modelo que retroalimentan de forma paralela a Reinforcement Learning, con esto aumentan la rapidez de convergencia del entrenamiento. En [29], para no depender del diseño de una función de recompensa que evalúe el desempeño del sistema, se sustituyó por una función de recompensa que se modifica durante el entrenamiento, permitiendo que un bípedo pueda realizar actividades como saltar, correr, agacharse o girar, validando su propuesta únicamente en simulación, mencionando que la implementación en un sistema real es una tarea compleja y que requiere gran número de recursos informáticos.

Para mejorar la respuesta de los sistemas con tareas multiobjetivo, en [30] planteó un primer esquema multiobjetivo, sin modelo, que sea capaz de llevar al levitador a la posición deseada, efectuando su movimiento descrito por trayectoria suave y en el menor tiempo posible. En [31] detallan un esquema generalizado para la planeación de trayectorias de mínimo consumo energético y en [32] presenta un vehículo aéreo no tripulado(UAV) con envió de datos IoT con el mínimo consumo de energía y maximizando el envió de datos IoT; su solución fue proponer trayectorias donde le permitiera maximizar la conexión para un mayor envió de información cumpliendo con las restricciones del sistema.

En el presente trabajo de investigación de tesis, al obtener el modelo dinámico de un robot bípedo con dos grados de libertad actuados capaz de lograr un ciclo de caminado semi-pasivo, se optó por incursionar en la métodologia de Deep Reinforcement Learning. Se eligió utilizar varios agentes computacionales (DDPG y SAC [5]) que aceptan redes neuronales y permitan la estimación de acciones en los motores, que sincronizado con una máquina de estados, se logre la conmutación requerida para el ciclo de marcha del robot bípedo.

1.2. Planteamiento del problema

En la literatura, se presentan varias estrategias de control basados principalmente en el modelo dinámico de diferentes sistemas. Sin embargo, no es posible obtener el modelo dinámico exacto o de alta precisión, donde se contemple todas aquellas no linealidades que produzcan comportamientos especiales del sistema como holguras mecánicas y algunos parámetros que se modifiquen en el tiempo, como coeficiente de fricción, además, de tener sistemas de alta dimensionalidad [33]. Para resolver la siguiente problemática, se ha optado por un cambio de paradigma, con el objetivo principal de diseñar un controlador que no dependa del uso necesario del modelo dinámico, ya que este aumenta la complejidad de obtención y de implementación cuando más precisión se requiera. En su lugar, se deberá estimar el comportamiento deseado contemplando de forma implícita, aquellas dinámicas especiales que es difícil incluir en el modelo dinámico. Con Deep Reinforcement Learning, se estimaran aquellos comportamientos provenientes de la lectura de datos de sensores hasta obtener la ejecución de la tarea deseada. Hoy en día, el nuevo enfoque ha generado gran interés en investigación, impulsado por el aumento en la capacidad de almacenamiento y manejo de datos, técnicas que permiten manipular información con mayor rapidez, bajo costo de sensores que permiten maximizar el rendimiento del sistema y minimizar el tiempo de desarrollo.

En el presente trabajo de tesis se aborda el diseño de una metodología general de un control basado en datos para sistemas mecatrónicos, que se pueda implementar el objetivo de control de posición y velocidad en un robot con trabajo en conmutación. Para no contemplar el modelado dinámico, se utilizarán los datos obtenidos a través de sensores y su manipulación con la metodología de Deep Reinforcement Learning, este algoritmo iterativo permitirá encontrar las señales de control a los actuadores y cumplir el objetivo de control.

[5] Soft (SAC).- agente computacional DDPG retardado utilizado en Deep Reinforcement Learning. Para más detalles ver la Tabla 2.7

1.3. Objetivos

Es esta sección se describe el objetivo general y los objetivos particulares del presente trabajo de tesis.

1.3.1. Objetivo General

Introducir una metodología general para la solución de problemas de sistemas mecatrónicos y evaluar el desempeño de controladores en el contexto de Deep Reinforcement Learning.

1.3.2. Objetivos Particulares

1. Diseñar y construir un bípedo con ciclo de marcha semi-pasiva, que sea capaz de realizar su tarea en un plano horizontal.

2. Diseñar e implementar un controlador en el contexto de Deep Reinforcement Learning para el problema de control de regulación, que le permita al bípedo realizar un ciclo de marcha semi-pasiva.

3. Identificar del enfoque de aprendizaje automático, patrones que permitan extrapolar su estructura de forma parcial o total, para la solución de sistemas de trabajo conservativo [6] o trabajo colaborativo.

1.4. Contribución

Las principales contribuciones del trabajo de tesis se presentan a continuación:

1. Se describe una la guía general del diseño de controladores en el contexto de Reinforcement Learning y recomendaciones de configuración de acuerdo con el sistema mecatrónico que se desee utilizar.

2. Se implementa un nuevo diseño conceptual en la composición estructural de un bípedo que pueda realizar un ciclo de caminado semi-pasivo sin articulaciones de rodilla.

3. Diseño de controlador por Reinforcement Learning para robot bípedo, sin rodillas, con ciclo de caminado semi-pasivo.

4. Incorporación de máquinas de estados para el robot bípedo que realiza trabajo conmutado.

1.5. Organización de la tesis

La presente tesis está compuesta por seis capítulos organizados de la siguiente forma:

1. **Capítulo 2** Se expone la formalización del enfoque de Reinforcement Learning con sus características principales y parámetros necesarios para su funcionamiento, después se exponen las mejoras e integración de redes neuronales profundas en Deep Reinforcement Learning con sus parámetros de diseño.

2. **Capítulo 3** Se expone y se describe a detalle el diseño mecánico, electrónico y de software para el bípedo con caminado semi-pasivo.

[6] energía mecánica que prevalecerá durante la operación del sistema sin interrupciones.

3. **Capítulo 4** Se desarrolla y se detallan los modelos cinemáticos y dinámicos del robot bípedo construido. Se muestra la justificación de no considerar la obtención de un controlador basado en el modelado dinámico, sustituyéndolo por la utilización del diseño de un controlador basado en datos.

4. **Capítulo 5** Se desarrolla la estrategia de control para el robot bípedo con ciclo de marcha semipasivo, se describe la construcción de redes neuronales profundas, tipo de agentes utilizados, parámetros de entrenamiento para Deep Reinforcement Learning, por último se analizan los resultados experimentales.

5. **Capítulo 6** Se presentan conclusiones generales e implementación del controlador generado por Deep Reinforcement Learning y se proponen algunas tareas como trabajo futuro.

Capítulo 2

Marco teórico y antecedentes

El presente capítulo describe a detalle los conceptos principales y definiciones matemáticas que intervienen en el enfoque de Deep Reinforcement Learning y el método de optimización multiobjetivo utilizado. En la sección 2.1 se ha desarrollado utilizando como principales referencias [34], [5] y [35] describiendo de manera general el algoritmo de Reinforcement Learning y conceptos principales como: política, acción, observación, función de recompensa, función del valor de observación y agente. En la sección 2.2 se definen aquellos parámetros de redes neuronales como: función de activación, topología, capas y nodos, además, de la combinación con Reinforcement Learning con los conceptos de actor y crítico.

2.1. Reinforcement Learning

Machine Learning es un enfoque que busca de forma específica crear algoritmos computacionales, que sean capaces de resolver problemas donde se requiere que exista un control, a partir de los datos que se puedan obtener del sistema. Machine Learning se divide en tres enfoques principales:

1. *Aprendizaje supervisado.* El diseñador tiene una influencia directa sobre la decisión de qué datos recolectar y como etiquetarlos. Se utilizan técnicas clásicas como árboles de decisión, máquinas de soporte vectorial, bosques aleatorios o técnicas más actualizadas como redes neuronales, redes convolucionales y redes recurrentes.

2. *Aprendizaje no supervisado.* El diseñador tiene una influencia directa ya que decide qué datos recolectar, pero no es necesario etiquetarlos. Se utilizan técnicas como algoritmos de clustering o de agrupación, algoritmos de reducción de dimensionalidad.

3. *Aprendizaje por refuerzo.* No tiene ninguna influencia el diseñador, un agente computacional determina de forma autónoma qué datos recolectar y cómo usarlos para su aprendizaje.

El enfoque de interés en este trabajo es el aprendizaje por refuerzo o reforzado (Reinforcement Learning), este consiste en implementar un algoritmo computacional para un sistema mecatrónico con múltiples situaciones a resolver, el sistema de forma autónoma, deberá encontrar en un proceso iterativo de prueba y error, aquellas acciones que maximizarán una función de recompensa, dicha función evalúa si se han logrado los objetivos de la interacción del sistema con el entorno. El agente computacional preferirá a largo plazo el conjunto de acciones que se hayan intentado en el pasado en el proceso de entrenamiento, y resultaron ser eficaz para maximizar la función de recompensa.

Una característica relevante de Reinforcement Learning es que considera la totalidad del problema, no se tiene permitido subdividir el problema. Los agentes computacionales del algoritmo

de Reinforcement Learning operarán a pesar de la incertidumbre que existe en el problema, ya que dichos agentes tendrán objetivos explícitos.

2.1.1. Elementos de Reinforcement Learning

Para el funcionamiento del algoritmo de Reinforcement Learning se requieren los siguientes elementos:

Entorno

El entorno será el mundo físico y sus perturbaciones en las que interactúa el sistema mecatrónico. El entorno define si el objetivo de control es posible físicamente, las reglas existentes y las acciones permitidas.

Se le conoce como *modelo del entorno* a aquella predicción futura de una acción a tomar por parte del sistema mecatrónico, y su repercusión con la interacción con el entorno.

Si conocemos completamente el modelo del entorno, conocemos completamente las reglas físicas que rigen al entorno, y se realizará Reinforcement Learning basado en modelo del entorno. Se considera como ventaja realizar Reinforcement Learning basado en modelo del entorno ya que se requerirá menor entrenamiento (intentos a prueba y error), para lograr el objetivo, sin embargo, pocas veces se conoce el entorno y sus reglas completas, en la mayoría de los casos solo se conoce de manera parcial. Cuando conocemos de manera parcial los detalles del entorno, se realizará Reinforcement Learning libre de modelo, para este caso el algoritmo realizará acciones que le permitan encontrar aquel conjunto solución que produzca maximizar la función de recompensa, para este caso requiere mayor entrenamiento.

Observaciones (S)

Son aquellas variables que permitirán al sistema mecatrónico comprender las repercusiones de las acciones en el entorno. La decisión del número y cuáles serán variables observables dependerán del diseñador, este parámetro debe contener todas aquellas variables que nos permitirán lograr el objetivo de control. En el caso particular de los sistemas mecatrónicos, los estados provendrán de aquellas variables observables y las lecturas de los diferentes sensores que podamos colocar en el sistema, y nos permitan entender de mejor forma el entorno.

Política (π)

La política π es un aproximador de funciones con parámetros ajustables que podría ser una función simple, tabla de búsqueda o un cálculo extenso que definirá la forma de comportarse del agente computacional, en un tiempo de muestreo dado. Si buscamos un elemento análogo de la política en la teoría clásica de control, y que realice función similar, sería el controlador.

La política π dada a una observación que proviene de las lecturas de los sensores del entorno y una acción, será la probabilidad de tomar acción, dado que actualmente se encuentre en la observación s perteneciente al conjunto de observaciones S, y se define con la siguiente ecuación:

$$\pi\left(s,a\right)=\Pr\left(a\in A\mid s\in S\right) \tag{2.1}$$

donde

$$\begin{aligned} a &= \text{acción a tomar} \\ A &= \text{conjunto de acciones} \\ s &= \text{observación actual} \\ S &= \text{conjunto de observaciones} \end{aligned}$$

Se conocen las siguientes técnicas de programación para políticas: programación diferencial (teoría de juegos), método de Monte Carlo, diferencia temporal, exploración vs explotación, gradiente descendiente y política iterativa.

Función de recompensa (γ)

La función de recompensa es un indicador numérico que evalúa las acciones tomadas de forma inmediata en un tiempo de muestreo del entrenamiento, y define el objetivo del problema a resolver. Si la señal de recompensa es negativa indicará que los estados no son los deseados, ya que existirán desviaciones mayores a las permitidas, si es positiva, el entrenamiento adquiere velocidad positiva y requerirá menores episodios. El diseño de esta función es de forma heurística, en la literatura, muy pocos detallan la obtención de la función de recompensa.

La función de recompensa en conjunto con el vector de observación, son los parámetros principales para modificar la política. Durante el entrenamiento, por medio de la evaluación de los parámetros instantáneos de la función de recompensa, el algoritmo verificará si la acción estimada por el agente ayudó a cumplir la tarea de control a través del valor numérico [36]. Si la representación del valor numérico no cumple con la tarea objetivo, el agente modificará la política y estimará una nueva acción para el sistema. El objetivo principal del algoritmo de Reinforcement Learning es optimizar la política (optimizar el controlador) maximizando la función de recompensa a largo plazo.

Función del valor (V_π)

La función de valor es un indicador de evaluación de las acciones realizadas en el entrenamiento a largo plazo. El valor numérico de la función de valor de una política π, será la expectativa de la recompensa que se obtendrá en el futuro. Se comienza en una observación fija para cada episodio y se realizan las políticas que se obtienen, descontando ligeramente una tasa de valor numérico de las recompensas futuras en comparación con el valor de recompensas inmediatas, esto se expresa con la ecuación 2.2

$$V_\pi(s) = \sum \gamma^t r_t \mid s_0 \in s \tag{2.2}$$

donde:

$$\begin{aligned} r_t &= \text{valor numérico de la recompensa en el tiempo de muestreo inmediato} \\ \gamma^t &= \text{tasa de descuento en el tiempo de muestreo inmediato} \end{aligned}$$

Agentes

Los agentes son programas informáticos que recibirán un entrenamiento para completar una tarea en un entorno. Los agentes contienen dos elementos: la política y al algoritmo de Reinforcement Learning.

Los agentes pueden basarse en:

1. *función de valor*.- Estos agentes aprenden funciones de calidad Q, como consecuencia de ello, no requieren un modelo del entorno que indique cuál será la próxima observación, este agente define de forma explícita el valor de lo que se sabe en función de adónde irá en el futuro.

2. *gradiente de la política*.- Estos agentes actualizan los parámetros de su política y su función de valor con un algoritmo de optimización del gradiente, todas las funciones de recompensas futuras estarán en una de las funciones de los parámetros actuales que parametrizará a la política.

Se tienen los siguientes aproximadores de funciones parametrizados:

1. *actor*.- Se utilizan para una observación particular, devolverá como resultado la acción que maximiza la señal de recompensa numérica acumulada esperada a largo plazo.

2. *críticos*. Se utilizan para una observación y acción particular, devolverá como resultado el valor esperado de la función de recompensa numérica acumulada a largo plazo.

De manera general, se muestra el diagrama del funcionamiento del aprendizaje reforzado que se aplica a los sistemas mecatrónicos. En la imagen 2.1 se encuentra al agente computacional encerrada en un recuadro color verde. Los elementos del agente computacional son: la política y el algoritmo de aprendizaje Reinforcement Learning, dicho agente, de forma probabilística enviará una acción a realizar al sistema. Éste interactuará con el entorno y enviará a través de las lecturas de los sensores aquellas mediciones que permitan entender lo sucedido en el entorno, dichos elementos son las observaciones. Para evaluar si la acción enviada por el agente cumple el objetivo de control, se tiene una función de recompensa, está en conjunto con el vector de observaciones ingresarán al agente en cada tiempo de muestreo modificando la acción a tomar por la política. Al finalizar cada episodio, se tendrá una función de valor que a largo plazo será nuestro indicador del aprendizaje del sistema. Por lo tanto, el entrenamiento se juzga promediando un número de varios episodios durante el entrenamiento.

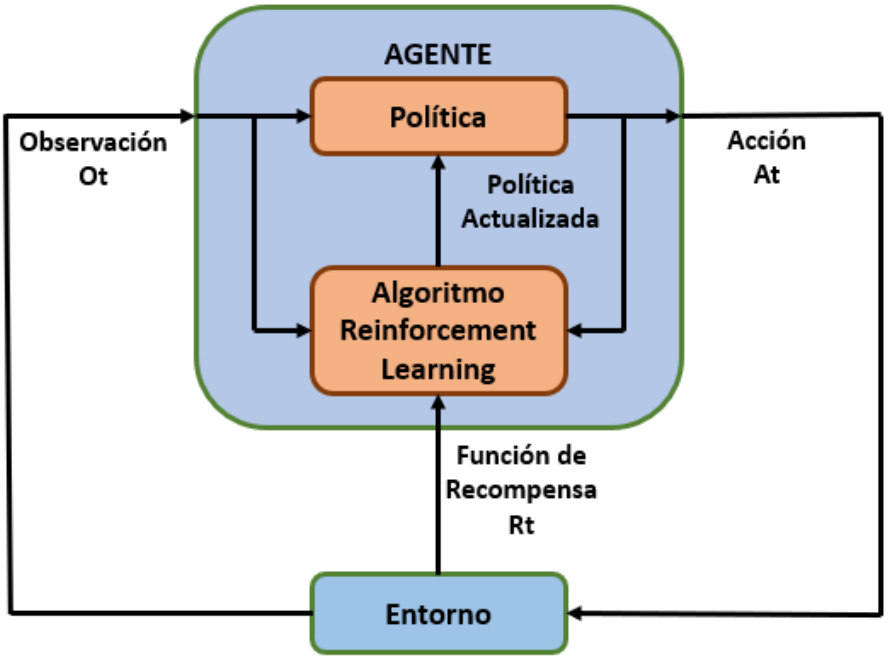


Figura 2.1: Diagrama de funcionamiento general de Reinforcement Learning

Tipos de agentes

Los agentes se subdividen en *agentes discretos*, *agentes continuos* y *agentes híbridos*. Sus nombres están dados respecto al conjunto del espacio de acciones que analizar y enviar al sistema físico.

La tabla 2.1 muestra los diferentes agentes existentes libres de modelo de entorno, si se basan en la función de valor o en el gradiente de la política y su espacio de acción.

Agente	**Basándose en**	**Espacio de Acción**	**Tipo de Agente**
Q-Learning (Q)	Función de valor	Discreto	crítico
SARSA	Función de valor	Discreto	crítico
Policy Gradient (PG)	Gradiente de la política	Discreto o Continuo	actor - crítico

Tabla 2.1: Agentes libres de modelo de entorno y espacio de acción

Las principales características de los agentes libres de modelo de entorno respecto al entrenamiento y acciones a tomar se muestra en la siguiente tabla 2.2.

Agente	**Características principales**
Q-Learning (Q)	-Durante el entrenamiento aprenden la política y las funciones de valor de forma simultánea. -La función observación-acción $Q(s, a)$ describe la cualidad de estar en una observación y tomar una acción determinada combinando la función de valor y la política. -Cuando se completa el entrenamiento, el aproximador de la función de valor entrenado se almacena en el elemento crítico.
Estado de acción Recompensa - estado acción (SARSA)	-Durante el entrenamiento aprenden la política de forma simultánea entre la función de recompensa y algoritmos de programación específicas como diferencia temporal o método de Monte Carlo. -La función observación-acción $Q(s, a)$ describe la cualidad de estar en una observación y tomar una acción determinada combinando la función de recompensa y la política. -Cuando se completa el entrenamiento, el aproximador de la función de valor entrenado se almacena en la sección crítica del agente.
Policy Gradient (PG)	-Durante el entrenamiento estima las probabilidades de tomar una acción que pertenecen al conjunto de acciones A y selecciona aleatoriamente una de ellas basándose en la distribución de probabilidades. -La sección del actor reducirá la varianza durante la estimación del gradiente, mientras que la sección del crítico calculará la función de valor para un estado u observación determinado.

Tabla 2.2: Características principales para los agentes libres de modelo de entorno

2.1.2. Entrenamiento de agentes

Se le conoce como entrenamiento de agentes al conjunto de intentos a prueba y error para que un sistema mecatrónico logré uno o múltiples objetivos de funcionamiento. Dicho entrenamiento

especifica como se modifican los parámetros en respuesta a la función de recompensa otorgadas en cada intervalo de tiempo y de esa forma actualizar la política del agente.

Por lo general, se desea maximizar la función de recompensa a medida que avanza el entrenamiento del agente. Sin embargo, la aleatoriedad de la elección de la acción del agente y su interacción con el entorno darán como resultado una variación de la función de recompensa. Durante el entrenamiento, como parámetros para todos los agentes se puede establecer el número de episodios (intentos prueba y error), el tiempo de muestreo o frecuencia con la que el agente interactúa con el entorno y el factor de descuento que es el que determina cómo se pondera la función de recompensa en cada tiempo de muestreo.

Algunas características específicas durante el entrenamiento son las siguientes:

1. El agente computacional almacena y utiliza experiencias previas.
2. El agente actualiza la política en cada tiempo de muestreo de cada episodio. En agentes del tipo actor-crítico, se actualiza la frecuencia de la política que realiza, únicamente para la sección del crítico.
3. El agente explora el espacio de posibles acciones dadas a observaciones particulares, eligiendo aquellas de acuerdo con la política o al azar.

Para entrenamientos donde los agentes se basan en el gradiente de la política se utiliza el algoritmo de gradiente descendiente para modificar parámetros del entrenamiento. Por lo tanto, cuando se modifican dichos parámetros se modifica la tasa de aprendizaje y el tamaño del gradiente. Por lo general en Reinforcement Learning, durante el entrenamiento se desea explorar más al comienzo y menos cuando el agente ha comenzado en generar una estimación de la función de calidad Q.

Los criterios de finalización del entrenamiento del agente computacional, son las siguientes condiciones:

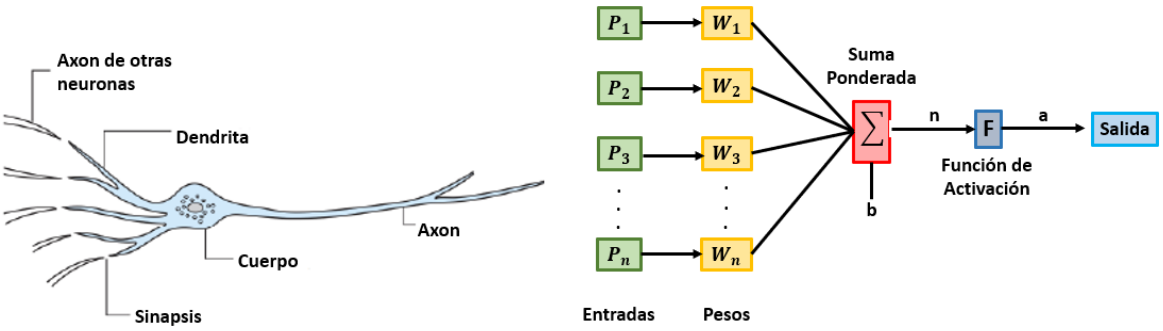
1. que el entrenamiento alcance el número máximo de episodios establecidos como parámetro.
2. que en el entrenamiento logre alcanzar un umbral determinado por el promedio de un número determinado de episodios de la función de recompensa, es decir, el valor numérico promedio de la evaluación del desempeño del sistema respecto al objetivo de control.

2.2. Deep Reinforcement Learning

El aprendizaje de refuerzo profundo o Deep Reinforcement Learning es la sinergia de enfoques entre Deep Learning y Reinforcement Learning. Este enfoque ayudó a que el aprendizaje por refuerzo necesitará un menor número de episodios en el entrenamiento y resolver problemas de sistemas más complejos que solo utilizar Reinforcement Learning. Deep Learning es un área del aprendizaje automático que, haciendo una extensión del concepto de red neuronal biológica, interconecta nodos y neuronas artificiales para el procesamiento de información de forma paralela, que les permite producir una salida muy similar al cerebro humano o al sistema nervioso.

En la figura 2.2 (a) se muestra que la composición de una red neuronal biológica es por medio de millones de neuronas, que pueden estar físicamente conectadas entre ellas a través de los axones hacia las dendritas y, de esta forma, compartir información por medio del envío y recepción de señales eléctricas de diferente amplitud, permitiendo entender la influencia que tiene dicha neurona hacia sus vecinos conectados. Realizando la analogía a una red neuronal artificial, en la figura 2.2 (b) se observa que cada neurona artificial se le asocia un peso numérico indicando su

influencia cuando envía información hacia las neuronas artificiales con las que se tenga conexión; la neurona recibe información a través de valores donde calcula su activación, tomando la suma ponderada de sus señales de entrada determinando su salida [1].



(a) red neuronal biológica (b) red neuronal artificial

Figura 2.2: Elementos de una red neuronal biológica y una red neuronal artificial [1]

2.2.1. Elementos de Deep Learning

Los principales elementos de una red neuronal artificial simple son las entradas, los pesos relativos, la suma ponderada, la función de activación, la función de escalado o de acotación y la salida. A continuación se describen de forma general [37]:

1. *pesos relativos*.- Son coeficientes que indican el impacto que tiene cada entrada que requiere la función de suma ponderada, que se adaptan y se modifican de acuerdo a la respuesta del entrenamiento y la topología especifica de la red, o por medio de sus reglas de aprendizaje.

2. *suma ponderada*.-De forma matemática, la suma ponderada es el producto interno entre el vector de entradas y el vector de pesos relativos, con la finalidad de tener una métrica de similitud desde el punto de vista geométrico entre vectores y así determinar el mínimo, el máximo, el producto, el mayor o el menor de acuerdo a un algoritmo de normalización y así tener la capacidad de determinar la arquitectura de la red y sus reglas de aprendizaje.

3. *bias o sesgo*.- Señal del tipo ON/OFF que permite controlar la activación de una neurona independientemente de los pesos relativos asociados a los datos de entrada. El método más utilizado para realizar la inicialización de este elemento es el llamado *normal y uniforme Glorot/Xavier*, y para redes neuronales profundas el más utilizado es el llamado *normal y uniforme de Kaiming He* [38].

4. *función de activación*.-Permite transformar la suma ponderada de la combinación lineal de pesos y entradas en una función de trabajo acotada, determinando la salida de la neurona artificial. Las más utilizadas son Sigmoidal y tanh, para redes neuronales profundas es más común utilizar la función de activación ReLU [1].

5. *escalado*.- Después de la función de activación, el resultado puede pasar por un proceso adicional de escalamiento, que permita que la salida se encuentre dentro de un umbral designado por el diseñador [1].

Clasificación de redes neuronales artificiales según su topología

La tabla 2.3 muestra la clasificación de las principales redes neuronales de acuerdo a su topología y sus característica operativas principal.

Topología de red	Características de la topología
Perceptrón simple	-Red neuronal más simple compuesta por una capa de neuronas con función de activación sigmoidea, ayudando a la convergencia de la función de perdida hacia los mínimos globales. [39].
Perceptrón multicapa	-Son redes neuronales de tipo perceptrón anidados con múltiples capas ocultas entre la capa de entrada y la capa de salida aumentando, las dificultades del entrenamiento y requiriendo mayor número de recursos informáticos, además, acepta más funciones de activación como Relu, Hard Limit, Lineal y Saturación Lineal. [39].
Red neuronal convolucional	-Son redes neuronales profundas donde cada neurona unicamente se agrupa con un subgrupo de neuronas de ellas, reduciendo el número de neuronas requeridas, la complejidad computacional en su ejecución produce resultados independientes. [39].
Red neuronal recurrente	Son redes que no tienen una estructura de capas, sino que comparte información desde la neurona inicial a conexiones de forma arbitraria o creando ciclos, consiguiendo tener temporalidad y memoria. [39].
Red de base radial	-Son redes que calculan la salida de la neurona en función de la distancia al centro. Tiene como ventaja de no converger a mínimos locales y la retropropagación quede bloqueada. [39].

Tabla 2.3: Clasificación y descripción de redes neuronales artificiales de acuerdo a su topología

Clasificación de redes neuronales artificiales según su tipo de aprendizaje

Tipo de aprendizaje	Formas de implementación	Algoritmos
Supervisado [37]	-Corrección del error	-Regla del perceptrón -Regla del mínimo error cuadrado -Backpropagation ó regla delta generalizada
	-Aprendizaje por refuerzo	-Linear Reward-Penalty -Critica heurística adaptativa
	-Aprendizaje estocástico	-Máquina de Boltzman -Máquina Cauchy
No Supervisado [37]	-Aprendizaje Hebbiano	-Red Hopfield -Matriz de aprendizaje -Adición Grossberg -Memoria asociativa bidireccional -Memoria asociativa temporal
	-Aprendizaje competitivo y cooperativo	-Red de resonancia dirigida -Learning Vector Quantization (LVQ)

Tabla 2.4: Clasificación y algoritmos de mejor funcionamiento de acuerdo a su tipo de aprendizaje

Se sabe que en general las redes neuronales tienen los mismos elementos sin importar que estas tengas diferentes estructuras, sin embargo, las variaciones principales que permite identificarlas

entre unas y otras son por sus reglas de aprendizaje, que rigen su funcionamiento y modifican la topología de la red. En la tabla 2.4 se muestra una clasificación de los algoritmos más utilizados con redes neuronales que se ha verificado con las que se cumple el objetivo y tiene mayor eficacia para aplicaciones de clasificación, predicción y filtrado de datos.

2.2.2. Funcionamiento de Deep Reinforcement Learning

La sinergia de Reinforcement Learning con Deep Learning produjo algunos cambios significativos en funcionamiento para la solución de problemas, a diferencia de únicamente utilizar el enfoque de Reinforcement Learning. Dichos cambios se describen de forma general a continuación:

1. De Reinforcement Learning se conoce que se tiene una política probabilística, retomando la arquitectura de Reinforcement Learning presentado en la figura 2.1, la variación principal en Deep Reinforcement Learning, es que dicha política probabilística se remplaza con una red neuronal profunda $\pi_\theta(s, a)$, esta red no tiene limitaciones de acuerdo a los valores de sus elementos, su arquitectura y de las reglas de aprendizaje.

2. En Reinforcement Learning se deseaba explorar más al comienzo del entrenamiento tratando de estimar una función de calidad Q, sin embargo, dicha función solía ser muy complejas. En Deep Reinforcement Learning se puede remplazar con redes neuronales profundas la función de calidad Q. Una cualidad de las redes neuronales profundas es que permiten aproximar funciones muy complejas y proponer acciones que no se habían probado únicamente con aprendizaje por refuerzo, además de que no se tenía la capacidad de explorar dicho conjunto solución. Si se desea modificar el resultado de algún entrenamiento, es suficiente el variar cualquier parámetro de la estructura de la red, como la topología y/o regla de aprendizaje.

3. El algoritmo de Reinforcement Learning, una vez que recibe como entradas la observación del sistema y la función de recompensa se actualizará a la política, esta ingresará a las redes neuronales con el fin de evaluar la efectividad propuesta y estimarla como una acción anterior, la salida de la red será una nueva propuesta de acción y dirigiéndola para su ejecución en el entorno.

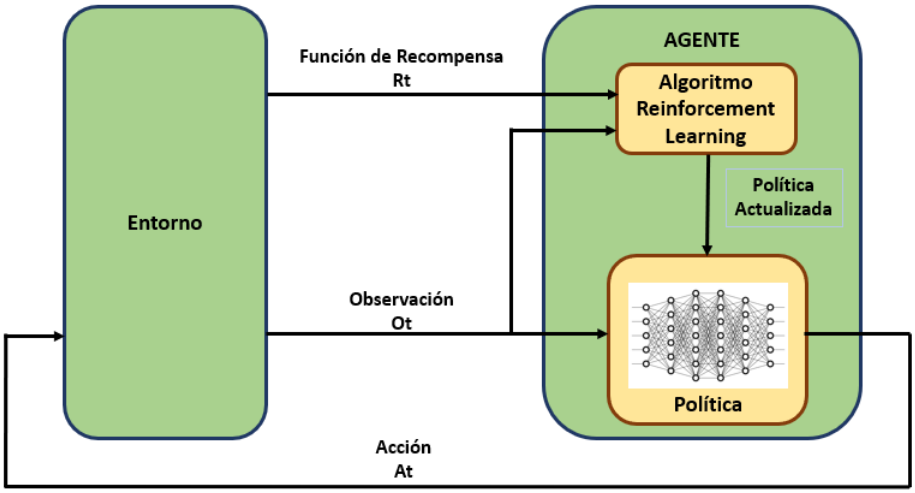


Figura 2.3: Diagrama de funcionamiento general para Deep Reinforcement Learning

En Deep Reinforcement Learning los tipos de agentes actor, críticos y actores-críticos modificarán de forma parcial sus características, en la tabla 2.5 se describen las de mayor prioridad y el espacio de acción donde mejor se desempeñan.

Tipo de Agente	Características en Deep Reinforcement Learning
Agentes críticos	-Pueden utilizar una política indirecta o utilizar una red neuronal que aproxime la representación una función de valor llamada función de calidad Q. -Estos agentes funcionan mejor con espacio de acción discreta.
Agentes actores	-Son agentes simples que utilizan una política directa y funcionan mejor con espacios de acción continua. -Tienen como desventaja que la sección en el algoritmo de Reinforcement Learning es sensible a observaciones proveniente de los sensores con mediciones ruidosas, a consecuencia de esto, el resultado no deseable es que el resultado converge a mínimos locales.
Agentes críticos-actores o híbridos	-Son agentes simples que utilizan una política directa que aprovecha las mejores características de los agentes anteriores. -Durante el entrenamiento la sección del actor aprende cuál es la mejor acción utilizando retroalimentación de la sección del crítico (en lugar de la función de recompensa directamente), a su vez, la sección del crítico aprende la función de valor y retroalimenta a la sección de actor.

Tabla 2.5: Características principales para los agentes libres de modelo de entorno

2.2.3. Agentes de Deep Reinforcement Learning

De los agentes mostrados en la tabla 2.1 de Reinforcement Learning, el único que admite redes neuronales profundas para la sección del actor y del crítico son los que se basan en el gradiente de la política (*Policy Gradient (PG)*).

En la tabla 2.6, en la primera columna se muestran los agentes computacionales basados en el gradiente de su política que admiten redes neuronales profundas en su estructura, en la segunda columna se muestra el tipo de agente conforme a la tabla 2.5, en la tercera columna y última columna, se menciona su espacio de acción y espacio de observación de cada agente.

Agente	Tipo	Espacio de Acción	Espacio de Observación
Deep Q-Network (DQN)	Crítico	Discreto	Continuo o Discreto
Proximal Policy Optimization (PPO)	Híbrido	Discreto o Continuo	Continuo o Discreto
Deep Deterministic Policy Gradient(DDPG)	Híbrido	Continuo	Continuo o Discreto
Twin-Delayed Deep Deterministic Policy Gradient (TD3)	Híbrido	Continuo	Continuo o Discreto
Soft Actor-Critic (SAC)	Híbrido	Continuo	Continuo o Discreto

Tabla 2.6: Agentes que admiten redes neuronales profundas, tipo de agentes y espacio de acción

Las tablas 2.7 y 2.8 muestran los agentes utilizados en capítulos futuros y las características principales durante el entrenamiento. Para la plataforma experimental que se describirá en el siguiente capítulo, se utilizarón los agentes DDPG y SAC, la elección y utilización de dichos agentes se describen en el Capítulo 5.

Agente	Descripción del entrenamiento	Funciones de redes neuronales
Deep Q-Network (DQN) *Estiman la función de recompensas futuras y la función de valor*	-En cada tiempo de muestreo se actualizan las propiedades del crítico basándose en las experiencias creadas. -El agente seleccionará una acción entre la probabilidad ϵ y la función de valor con probabilidad $1-\epsilon$, este último con la tarea de hacer la elección localmente óptima. -Al finalizar el entrenamiento, la función de valor que se aproximó se almacenará en la sección del crítico $Q(S,A)$.	-La función de valor se realiza con las redes neuronales siguientes: *Crítico* $Q(S,A)$. Producirá la expectativa de la función de recompensa a largo plazo tomando como entradas la observación S y la acción A. *Crítico objetivo* $Q^{`}(S,A)$. Actualiza periódicamente el objetivo del crítico basándose en los últimos valores de sus parámetros con la tarea de mejorar la estabilidad del algoritmo de optimización. -$Q(S,A)$ y $Q^{`}(S,A)$ pueden ser de estructura semejante.
Proximal Policy Optimization (PPO) *Estiman la política y la función de valor.*	-El cambio de política se alterna entre la función objetivo acotada utilizando el algoritmo de gradiente descendiente y el muestreo de las lecturas de los sensores de la interacción del sistema con el entorno. La finalidad de que la función objetivo sea acotada es mejorar la estabilidad del entrenamiento al limitar el cambio del tamaño en la política en cada tiempo de muestreo. -Al finalizar el entrenamiento, la función de valor que se aproximó se almacenará en $\mu(S)$.	La función de valor se realiza con las redes neuronales siguientes: *Actor* $\mu(S)$. Emitirá las probabilidades de acción cuando se encuentra en la observación S en cada tiempo de muestreo. *Crítico* $V(S)$. Emitirá la expectativa de la función de recompensa a largo plazo. -$\mu(S)$ y $V(S)$ no será necesario que sean de estructura semejante.
Soft Actor-Critic (SAC) *Estiman la política y la función de valor al seleccionar acciones de forma aleatoria en función de la distribución de probabilidades.*	-Su objetivo es optimizar la política directamente y entrenar un crítico para estimar las recompensas futuras. -El agente interactúa con el entorno durante varios pasos utilizando la política actual, antes de actualizar las propiedades del actor. -Al finalizar el entrenamiento, la función de valor que se aproximó se almacenará en $\mu(S)$.	La función de valor se realiza con las redes neuronales siguientes: *Actor* $\mu(S)$ Toma la observación S y emite las probabilidades de elegir una acción permitida cuando está en la observación S. *Crítico* $V(S)$: el crítico toma la observación S y emite la expectativa correspondiente de la recompensa descontada a largo plazo.

Tabla 2.7: Características de los agentes durante su entrenamiento

Agente	Descripción del entrenamiento	Funciones de redes neuronales
Deep Deterministic Policy Gradient (DDPG) *Sobreestima la función de valor que maximizará a largo plazo la función de recompensa.*	-Durante el entrenamiento, perturba la acción elegida por la política utilizando un modelo de ruido estocástico y modifica las propiedades del actor-crítico en cada tiempo de muestreo. - Al finalizar el entrenamiento, la función de valor que se aproximó se almacenará en $\mu(S)$.	La función de valor se realiza con las redes neuronales siguientes: *Actores* $\mu(S)$ *y* $\mu'(S)$. Se elegirá la acción que maximice la función de recompensa a largo plazo tomando la observación S, después actualizará el objetivo del actor y así mejorar la estabilidad del algoritmo de optimización. *Críticos* $Q(S,A)$ *y* $Q'(S,A)$. Se produce la expectativa de la función de recompensa a largo plazo tomando la observación S y la acción A, después se actualizará el objetivo del crítico y así mejorar la estabilidad del algoritmo de optimización -$Q(S,A)$, $Q'(S,A)$, $\mu(S)$ y $\mu'(S)$ deben ser de estructura semejante.
Twin-Delayed Deep Deterministic Policy Gradient (TD3) *Este agente es una extensión del algoritmo DDPG enfocado en reducir la sobrestimación de la función de valor que maximizará a largo plazo la función de recompensa*	Para reducir la sobrestimación el agente durante el entrenamiento realizará las siguientes acciones: -El agente deberá aprender dos funciones de valor y las utilizará durante la actualización de la política y de la estimación de la función de valor. -Se actualiza la política y los objetivos de optimización con menor frecuencia que las funciones de valor. -Durante la actualización de la política se añade ruido a la acción, ocasionando que la política tenga menores probabilidades de explolación. Si solo se utiliza una función de valor en lugar de dos, se obtiene un agente DDPG con una política con actualizaciones retrasadas y acciones suavizadas. -Al finalizar el entrenamiento, la función de valor que se aproximó se almacenará en $\mu(S)$	La función de valor se realiza con las redes neuronales siguientes: *agente determinista* $\mu(S)$. El agente realiza una acción que maximizará la función de recompensa a largo plazo. *Actor* $\mu(S)$. Actualizará el objetivo del actor y así mejorar la estabilidad del algoritmo de optimización. -*Uno o dos valores críticos* $Q_i(S,A)$ y $Q_i'(S,A)$. Producirá la expectativa de la función de recompensa a largo plazo tomando la observación S y la acción A, después se actualizará el objetivo del crítico y así mejorar la estabilidad del algoritmo de optimización -$Q_k(S,A)$, $Q_k'(S,A)$, $\mu(S)$ y $\mu(S)$ deben ser de estructura semejante. Si se utilizan dos críticos $Q_1(S,A)$ y $Q_2(S,A)$,estos pueden tener una estructura idéntica o diferente. Si tienen la misma estructura, deberán tener diferentes valores de parámetros iniciales.

Tabla 2.8: Características de los agentes durante su entrenamiento (2ª parte)

Capítulo 3

Diseño paramétrico de la plataforma experimental

En el presente capítulo se describe a detalle el diseño paramétrico e implementación de un robot bípedo semipasivo como plataforma experimental para este trabajo de tesis, se mencionan características esenciales de la estructura mecánica, circuito de potencia e instrumentación requerida para la recepción y envió de datos, y finalmente se describe la programación del software implementado en el sistema embebido.

3.1. Estado del arte

A comienzos de la década de los años 80's, con la intención de que los robots creados para asistencia del humano realizarán las mismas tareas que un humano, se comenzó con la investigación de los robots bípedos por la misma línea de cada vez tener mayores similitudes. Una de las principales líneas de investigación en los robots bípedos se extiende a diseñar controladores para emular el caminado de un humano de forma más natural posible, por ello, la gran mayoría de los bípedos, consta de un alto número de grados de libertad siendo análogo a las articulaciones posibles que intervienen en el ciclo de marcha. Las principales articulaciones, se ubican en las piernas y en el torso, actuando adecuadamente de forma independiente y sincronizando sus movimientos se logra el objetivo de caminado, sin embargo, esto no es una tarea sencilla, por ello, en la actualidad sigue siendo un tema de investigación.

Con la finalidad de estudiar bípedos de menor número de grados de libertad, a principio del año 2000, se propuso estudiar la dinámica de bípedos con ciclo de caminado aprovechando la dinámica natural del mecanismo del bípedo a favor de su movimiento, a esto se le conoce como *caminado semi-pasivo.*

De los primeros artículos relacionados de ciclo de marcha semi-pasiva y quienes son pioneros en su estudio se tiene el trabajo de [2] se propone un bípedo con la reducción de grados de libertad, pero capaz de realizar el ciclo de caminado, para ello, integro una base de pie curva que le permite aprovechar de mejor forma la dinámica natural del mecanismo, la planta de pie curvo le ayuda al robot bípedo a generar un movimiento transitorio en el plano frontal, permitiendole realizar el ciclo de caminado en un suelo inclinado y proponiendo algunas estrategias de control no lineal. [40] es otro trabajo que aprovecha la dinámica natural gracias a la gravedad en un plano inclinado como se muestra en la figura 3.1, éste trabajo propone que su controlador converge al ciclo límite del caminado, además, implementan en simulación el algoritmo *'Ensemble Kalman Filter'* que permite eliminar en todo momento las divergencias de la respuesta del controlador ayudando que la velocidad de convergencia en el ciclo limite se logre en menor tiempo.

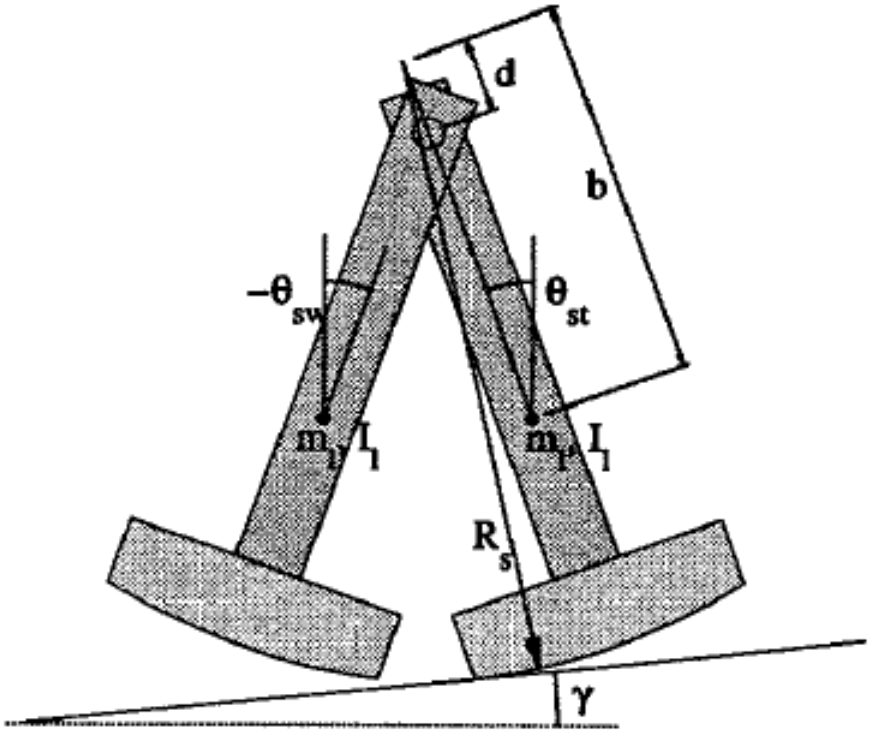


Figura 3.1: Vista de plano Sagital de bípedo con ciclo de marcha en plano inclinado [2]

Con el trabajo de [41] se ofreció un panorama diferente en el estudio del ciclo de marcha como fenómeno caótico con entornos de mayor complejidad, uno de los que se proponen es realizar el ciclo de marcha sin la pendiente. Una de las maneras de realizar una simplificación en el diseño del controlador como en la obtención de su modelo dinámico fue realizar la analogía en comportamiento y principio del ciclo de caminado como un péndulo invertido con acción de un carro. Con este enfoque existe un número importante de artículos científicos referente al caminado semi-pasivo. En [33] se propone una estrategia de ciclo de marcha enfocado en el conocimiento del centro de masa analizando la fase activa y pasiva del bípedo. En [42] se agrega el estudio del caminado en las rodillas, determinando el comportamiento deseado dependiendo de la fase del ciclo de marcha en la que se encuentre el bípedo. En [43] se describe la forma de calcular ZMP (Zero Moment Point por sus siglas en inglés) a partir del estado del centro de masa sin que se tome como referencia alguna trayectoria deseada. Por último, en [44] se diseña como un mecanismo de eslabones la pierna de un robot bípedo, se subdivide el mecanismo en en tres: dos son mecanismos de cuatro barras y el otro es un mecanismo de cinco barras; utilizando el algoritmo de evolución diferencial se realiza la síntesis del mecanismo que permite estudiar el ciclo de marcha de un bípedo donde la planta del pie sigue una trayectoria elíptica.

Además del estudio de locomoción del ciclo de marcha en los bípedos, en los últimos años comenzó el interés en el estudio de impacto cuando la planta del pie del bípedo tiene contacto con diferentes coeficientes de fricción en el suelo, y su repercusión de las vibraciones formadas en la estructura de la pierna, encontrando instantes en el que el movimiento experimenta bloqueos en los que podría convertir un ciclo de marcha estable en inestable. El estudio de la fuerza de contacto entre el bípedo y el suelo ha ayudado en realizar estudios de locomoción en terrenos difíciles, en [45] se desarrolló un controlador en el torso generando energía activa en sustitución de la gravedad y el diseño de un controlador en la pierna basculante.

Como trabajo más reciente que ayuda en el estudio del ciclo de caminado en bípedos con caminado semi-pasivo, en [46] desarrollaron un gemelo digital con validación de modelos dinámicos

independientes y de forma experimental, destacando la alta coincidencia entre la comparación de respuesta del gemelo digital y el bípedo físico, exhortando al uso de gemelos digitales para futuros trabajos e investigaciones.

3.2. Diseño paramétrico

El diseño se subdivide en tres principales sistemas que son:

1. Estructura mecánica
2. Diseño electrónico
3. Diseño de software

Los principales requerimientos de diseño para el robot bípedo semi-pasivo son:

1. robot bípedo capaz de realizar su ciclo de caminado en un suelo plano (que no tenga necesidad de operar en un plano inclinado).
2. la plataforma debe contar con la posibilidad de envío y recepción de datos.
3. el sistema debe contar con un sistema de paro de emergencia.
4. la plataforma experimental debe contar con la posibilidad de guardar datos de desempeño en tiempo real.

3.2.1. Estructura mecánica

Para la estructura mecánica se utilizó el software de diseño asistido por computadora Solidworks; el sistema se conforma de los siguientes componentes mecánicos principales que se pueden apreciar en la figura 3.2: piernas (color azul), housing motor pierna (color verde), housing motor péndulo transversal (color rojo), péndulo transversal (color amarillo), base de pie curvo (color naranja), y por último, los componentes electrónicos (color negro).

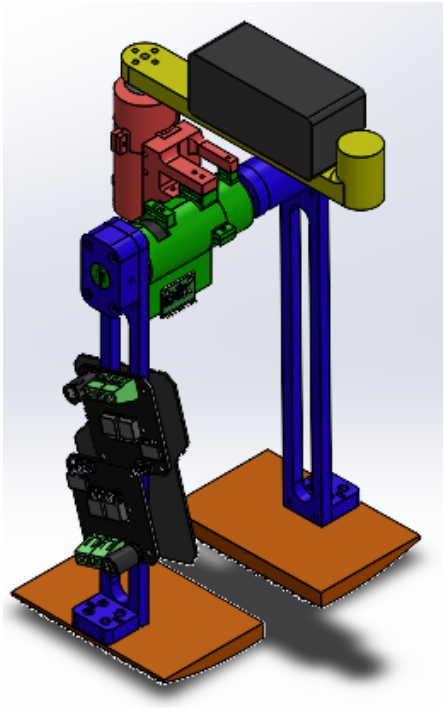


Figura 3.2: Componentes mecánicos principales del bípedo semipasivo en vista isométrica

Los diferentes componentes del ensamble que conforman al sistema fueron manufacturados en impresoras 3d de material PLA. Las dimensiones generales del robot se muestran en el apéndice A.

En los modelos estudiados en la sección 3.1, se mencionó que para el robot bípedo, se requiere generar movimiento oscilatorio en el plano frontal, de esta forma, el robot se beneficia de la dinámica del movimiento permitiéndole realizar el ciclo de caminado en un plano inclinado sin obstrucción física en los pies. Sin embargo, en este trabajo de tesis se propone inducir el movimiento necesario del plano frontal adicionando un actuador con movimiento angular en el plano transversal, provocando desplazamientos del centro de masa en cada fase del ciclo de marcha del robot. En la imagen 3.3 se muestra en vista transversal, la región del polígono de estabilidad del centro de masa del robot bípedo que permite realizar el ciclo de marcha, si el centro de masa supera los límites, el robot bípedo caerá.

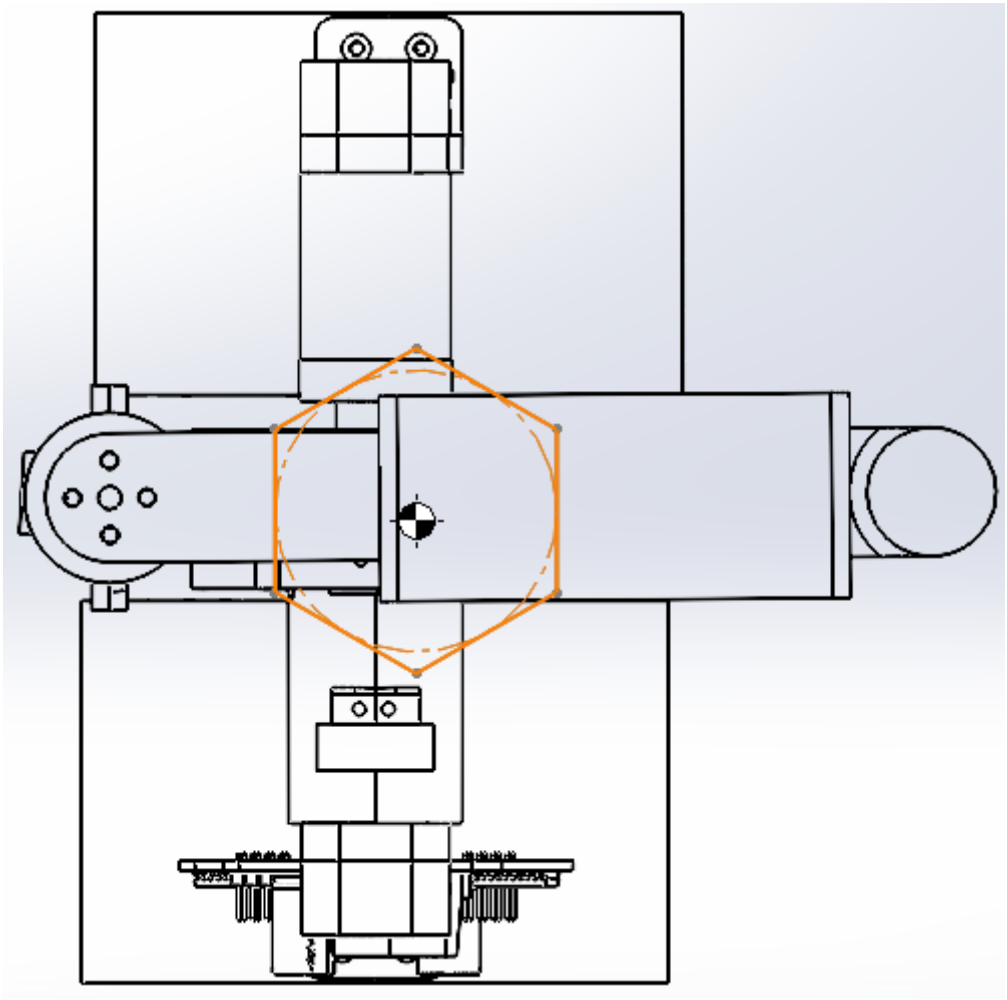


Figura 3.3: Región de estabilidad en el plano transversal

Para garantizar el movimiento transitorio del plano frontal requerido por el bípedo y este pueda ejecutar su ciclo de caminado sin ninguna obstrucción física, se retoma el diseño de las plantas del pie curvas de [3], este trabajo también describe a detalle la obtención del radio de curvatura reescrito en el Apéndice A. Nuestra aportación al diseño, en la figura 3.2.1 de la planta de pie curva recae en la adición de un estriado en forma de zig-zag de material de foami que permitirá al bípedo tener la fricción necesaria cuando la planta del pie curva este en contacto con el suelo permitiendo el caminado, sin ella, el robot se deslizaría en su lugar sin lograr avanzar.

(a) base de pie curvo en CAD (b) base de pie curvo con foami

Figura 3.4: Base de pie curvo con foami para tener la fricción necesaria en el ciclo de caminado

En los modelos existentes de robots bípedos, normalmente se desea que la cadera, sea articulada y actuada por dos motores (uno para cada pierna) o un solo actuador haciendo una elección de una pierna móvil. Retomando el trabajo [3], se tiene únicamente una pierna móvil. Como parte del rediseño, se eligió un motor de corriente directa con escobillas que tuviera la menor relación de engranaje posible y contará con encoder para la medición de posición angular θ_c y velocidad angular $\dot{\theta}_c$. El actuador deberá entregar el mayor par posible dentro de una gama de motores comerciales con voltaje de 12V y 2[A] de corriente continua. Para el accionamiento del péndulo transversal, se eligió un motor de corriente directa con escobillas con mayor relación de engranaje, con la capacidad de entregar el par suficiente para mover el péndulo transversal con masa, generando el momento suficiente que permita obtener el movimiento oscilatorio del plano frontal. Por medio del encoder, se pueden obtener los datos de posición angular θ_p y velocidad angular $\dot{\theta}_p$. Las especificaciones generales de los motores se muestran en la tabla 3.1.

	Motor péndulo	**Motor cadera**
Voltaje [V]	12	12
Relación engranaje	1:9.7	1:45
Velocidad [rpms]	1030	330
Torque [kg-cm]	3.2	4.3
Corriente pico [A]	5.6	2.5
Corriente sin carga [mA]	300	150
Resolución encoder	465.6	493.9
Peso [g]	180	128

Tabla 3.1: Características de los motores seleccionados

3.2.2. Diseño electrónico

Para el subsistema electrónico del bípedo semi-pasivo se diseñaron dos placas PCBs que se interconectan entre ellas compartiendo voltajes y señales. Los componentes utilizados se mues-

tran en la tabla 3.2

Componente	**Cantidad**	**Descripción General**
Microcontrolador ESP32	1	Sistema embebido programable de la lógica completa del robot
Batería LIPO	1	Batería LIPO de 4S, 1550 mAh y 100C
Modulo LM2596	2	Fuente de alimentación step down DC-DC, ajusta voltajes de 16.1V-12V y 12V a 5V
Vnh2sp30 Puente H	1	Controlador de 2 motores programables y sensor de corriente
Schmitt Tigger Sn74hc14n	1	filtro de histéresis de entrada para señales cuadradas deformadas provenientes del encoder
Modulo MPU6050	1	A través del giroscopio y acelerómetro se obtienen los ángulos de giro en los ejes x,y y z.
Modulo INA219	2	Sensor de consumo de voltaje en los actuadores.
Receptor Rf 433MHz	1	Recepción de paros de emergencia
Medidor voltaje LIPO	1	Medidor de voltaje en cada celda de batería LIPO.

Tabla 3.2: Componentes electrónicos utilizados en el diseño electrónico y características generales

Las placas electrónicas se diseñaron en el software EAGLE con las siguientes características.

1. Distribución de voltajes para todos componentes que provee la batería a 12V y 5V.
2. Adquisición de las señales de control que provienen de los encoders de los motores convertidos en posición y velocidad angular.
3. Adquisición de datos del acelerómetro y giroscopio del módulo MPU6050, y datos de voltaje del módulo INA219
4. Adquisición de datos de corriente en los motores desde el controlador puente H Vnh2sp30.
5. Distribución de las señales de comunicación I2C y UART.
6. Distribución de las señales de control entregadas por el microcontrolador ESP32 Wifi + Bluetooth Dual Core
7. Acondicionamiento de las señales que provienen del encoder con amplificadores operacionales Schmitt Tigger 74ls14.
8. Facilidad de montaje del circuito electrónico con la estructura mecánica.
9. Soportes en conexiones resistente a movimientos no deseados que provoquen desconexión.
10. Posibilidad de conexión alámbrica por comunicación UART e inalámbrica por diferentes protocolos de comunicación (WIFI y Bluetooth).

Para la elección del microcontrolador ESP32 se priorizó en elegir un sistema embebido dual-core, que tuviera los pines suficientes GPIO, comunicación I2C y protocolo de comunicación alámbrica UART y protocolos de conexión inalámbrica por WIFI y Bluetooth. La figura 3.5 muestra la

distribución general de las señales de control y de potencia de las pcbs diseñadas, mientras que los diagramas eléctricos de las PCBs diseñadas se muestran en el apéndice A.

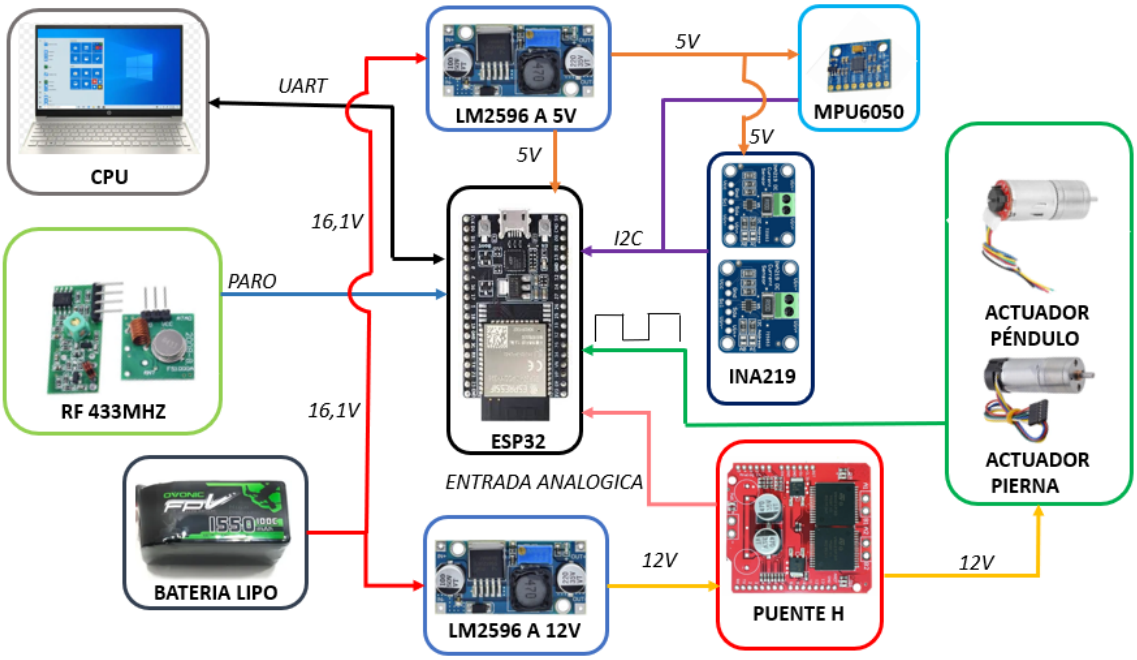
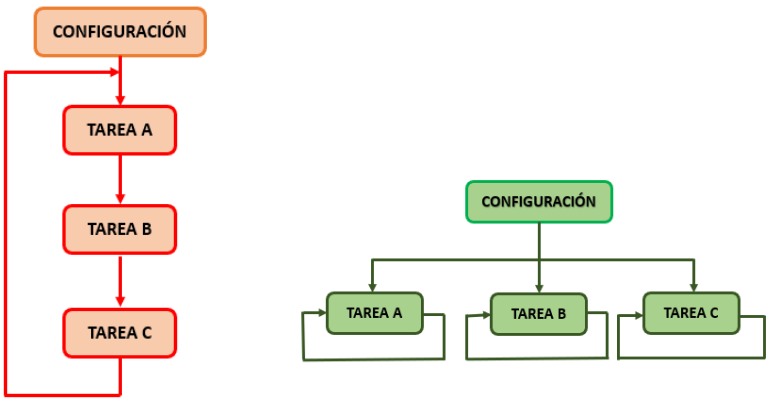


Figura 3.5: Señales de potencia, comunicación y control de circuitos electrónicos

3.2.3. Diseño de software

El microcontrolador ESP32 tiene la posibilidad de programarse con varios lenguajes de programación: C, C++ y Python. Se eligió el lenguaje C porque se utiliza el paradigma de programación RTOS (Real Time Operating System por sus siglas en inglés); dicho paradigma propone la distribución de tareas independientes en subprogramas que se ejecutan de forma paralela, que pueden compartir mensajes o información de forma síncrona o asíncrona teniendo la posibilidad de manejar y reducir latentes existentes (retardos de ejecución de programa) en comparación con la programación habitual (ejecución de forma serial). La figura 3.6 muestra de forma general la diferencia de paradigmas entre la programación serial y programación RTOS.



(a) Programación paradigma serial

(b) Programación paradigma RTOS

Figura 3.6: Comparación entre paradigmas de programación

Para acelerar las operaciones que requiere el ciclo de marcha en el microcontrolador, se utilizan y se configuran tareas en la memoria RAM General y se hacen uso de ellas con llamadas desde Dual-Core del microcontrolador obteniendo la ejecución *"Multitask"*. Las tareas una vez que son creadas se ejecutaran de forma infinita, para evitar que dos o más tareas se traslapen y creen una competición entre ellas, cada tarea está clasificada en función de prioridad de ejecución donde 0 es la más baja prioridad y 5 es la más alta prioridad, de esta forma se evita comportamientos no deseados.

El diagrama de la figura 3.7 muestra la planificación de las tareas y su predicción de ejecución en el microcontrolador, dicho diagrama se complementa con la tabla 3.3 que describe a detalle sus objetivos al ejecutarse. La predicción de ejecución que realiza como ejemplo el paradigma de programación RTOS considera que se ha activado alguna interrupción de seguridad programa, sin embargo, esto no siempre será así, si es que no se cumplen las condiciones de paro de emergencia.

Item	**Tarea**	**Prioridad**	**Descripción General**
A	*Recepción y envió de datos*	4	Recepción y envió de trama de datos del CPU al microcontrolador.
B	*Accionamiento de motor de pierna*	3	Recepción de datos de voltaje mapeados a señales de PWM para el actuador del bípedo.
C	*Adquisición de datos de encoder del actuador de la pierna*	2	Obtención de pulsos cuadráticos mapeados a posición angular, velocidad angular y dirección del actuador.
D	*Adquisición de datos de voltaje en los actuadores.*	0	Obtención de datos de voltaje a través de una resistencia shunt conectado directamente en los. actuadores.
E	*Adquisición de datos de corriente de los actuadores*	0	Obtención de datos de corriente utilizado por los actuadores a través del controlador puente H.
F	*Obtención de datos de módulo MPU6050*	1	Recepción de trama de datos del CPU al microcontrolador.
G	*Accionamiento de motor de péndulo*	3	Recepción de datos de voltaje mapeados a señales de PWM para el actuador del péndulo.
H	*Adquisición de datos de encoder del actuador del péndulo*	4	Obtención de pulsos cuadráticos mapeados a posición angular, velocidad angular y dirección del actuador del péndulo.
I	*Interrupciones por* software	5	Paros de emergencia accionados cuando se presentan anomalías como caídas, paro por sobre paso de límites del encoder, paro de cambio de orientación de centro de masa ó paro de emergencia accionada por el usuario.

Tabla 3.3: Descripción de tareas a realizar y sus prioridad de ejecución

De la figura 3.7 se puede observar aquellas tareas que requiere el microcontrolador para su correcto funcionamiento indicadas de color azul, además en esta sección se consideran los retardos programados para sincronizar el funcionamiento del sistema embebido.

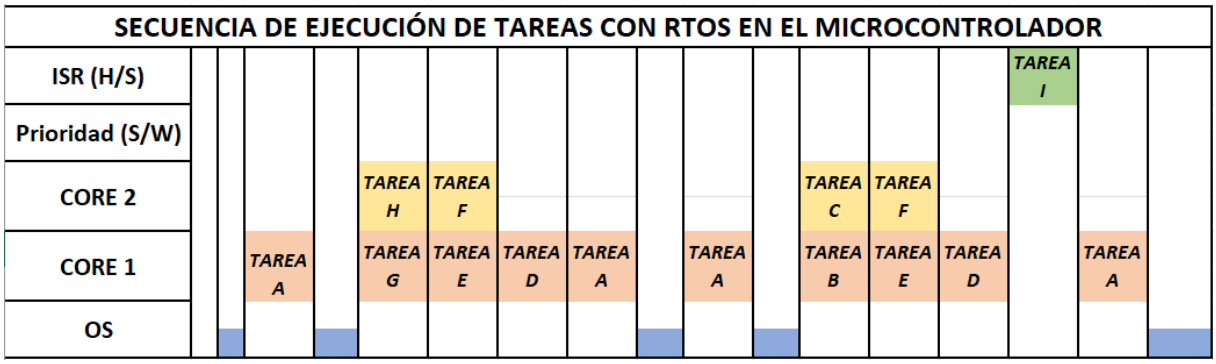


Figura 3.7: Ejecución de tareas con paradigma de programación RTOS

Para el programa principal de ejecución del sistema embebido, se diseñó una máquina de estados de tipo Mealy [47], dicha máquina es encargada de sincronizar la ejecución en cada episodio durante el entrenamiento. La figura 3.8 muestra el diagrama de funcionamiento de la máquina de estados y la tabla 3.4 indica la descripción de los estados y sus transiciones.

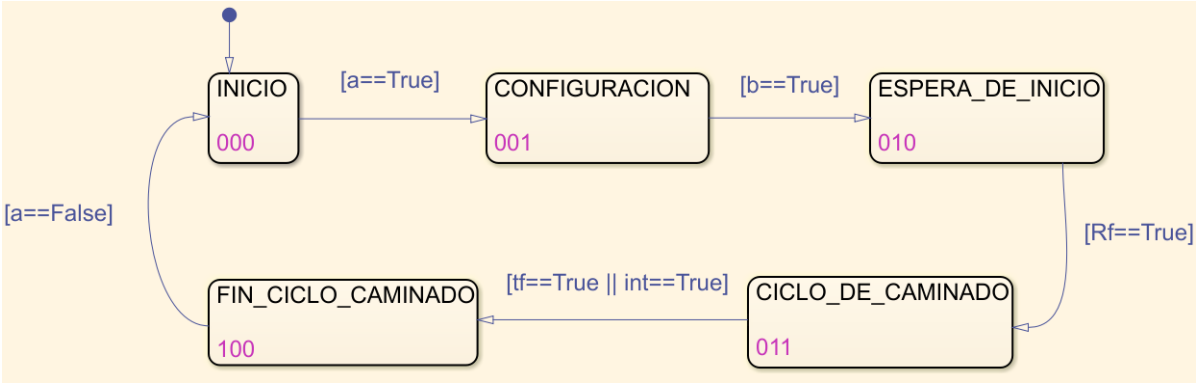


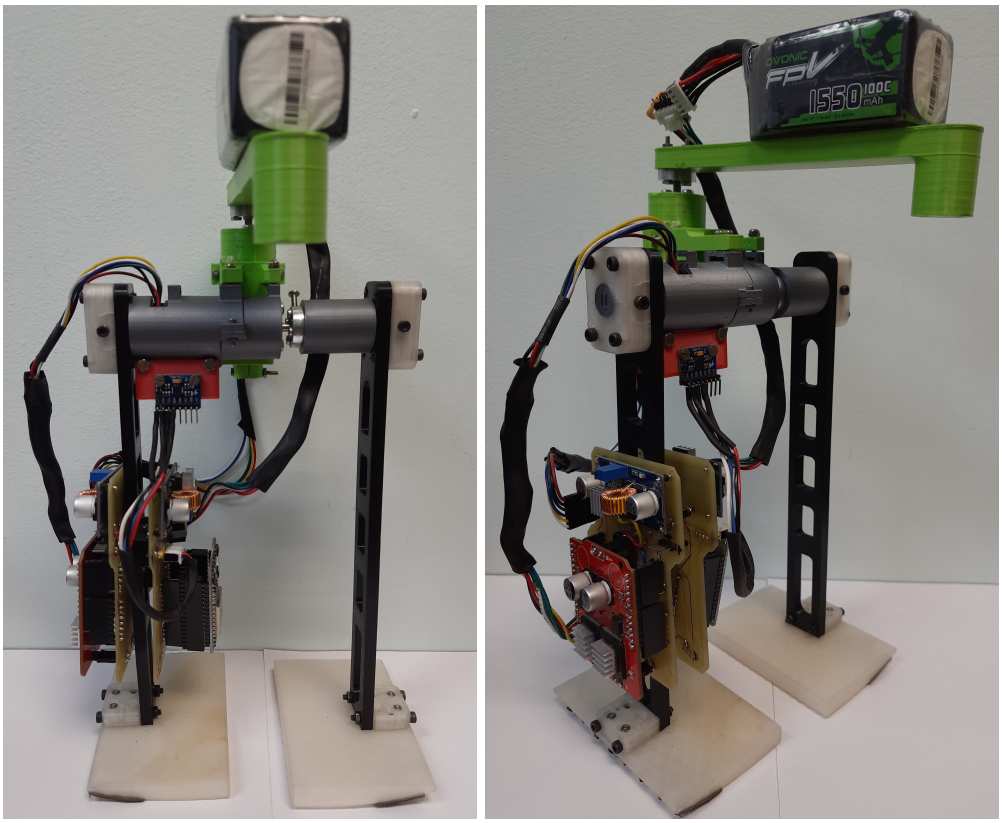
Figura 3.8: Máquina de estados de funcionamiento principal del sistema embebido

Estado	Transición	Descripción General
000	a = True	Arranque del microcontrolador, una vez que finalice dicho proceso la bandera *a* será verdadera.
001	b = True	Configuración del Filtro de Kalman y módulo MPU6050, módulos INA219 y conexión serial, una vez que finalice dicho proceso la bandera *b* será verdadera
010	RF = True	Estado de espera hasta accionar algún botón del control RF modificando la bandera *RF* a verdadera.
011	tf = True ó int = True	Estado de ciclo de caminado, el proceso finalizará si se activa el paro de emergencia ó se activan las interrupciones de seguridad del bípedo, ó termina el tiempo otorgado de 15[s].
100	a = False	Estado fin de caminado a espera de colocar al robot bípedo de forma manual en la condición de inicio modificando la bandera **a** a Falso.

Tabla 3.4: Descripción de la máquina de estados y sus transiciones

3.3. Ensamble de plataforma experimental

En la figura 3.3 se muestran una fotografías con vista en el plano sagital e isométrico del prototipo del robot bípedo semipasivo analizado en el presente trabajo de tesis. Se observa la estructura mecánica y las tarjetas PCBs ya ensambladas e interactuando.



(a) Vista plano sagital (b) Vista isométrica

Figura 3.9: Prototipo de bípedo

El robot bípedo construido, se utilizará para validar el enfoque de control basado en datos por Reinforcement Learning. En el siguiente capítulo se muestra el modelo dinámico con la morfología propuesta.

Capítulo 4

Modelado cinemático y dinámico

En el presente capitulo se presentan los conceptos necesarios para la obtención del modelo matemático cinemático y dinámico mostrando que no es trivial su obtención, aumentando la complejidad del diseño y funcionamiento correcto de un controlador basado en el modelo. Para la subsección 4.1 se muestra el modelo cinemático de la plataforma experimental obtenido por la multiplicación de transformaciones homogéneas provenientes del método de parámetros de Denavit Hartenberg. Para la subsección 4.2 se obtiene el modelo dinámico en el plano frontal del movimiento oscilatorio requerido para la marcha del bípedo sin obstrucción. Por último, en la subsección 4.3 se muestra el modelo dinámico de la plataforma experimental obtenido por el método Euler-Lagrange y trabajo virtual durante el ciclo de marcha. En el apéndice B, se muestran los cálculos y los conceptos que intervienen en la obtención de los modelos de forma detallada.

4.1. Modelo cinemático

La cinemática directa es el problema de encontrar la posición y orientación del marco final F con respecto al marco base B. En los robots bípedos, la representación cinemática puede realizarse en dos principales formas:

1. Robot serial donde uno de los pies se asigna como eslabón base y el otro pie será el efector final. En este enfoque se considera que el ciclo de marcha del bípedo es respecto a un pie. [48]
2. Árbol cinemático, donde las piernas se consideran como cadenas cinemáticas abiertas conectadas en la cadera del bípedo asignada como eslabón base. Con este enfoque se considera que durante el ciclo de caminado es respecto a la cadera.[49]

Las matrices de transformaciones homogéneas que describen la posición y orientación general de un sistema tienen la forma

$$^{0}A_{f} = \begin{bmatrix} ^{0}R_{f} & ^{0}O_{f} \\ 0 & 1 \end{bmatrix} =^{0} A_{1}^{0}A_{2}...^{i-1}A_{i}, j = i$$

donde $^{0}R_{f}$ es la matriz de rotación que relaciona al marco final $\{f\}$ respecto al marco base $\{0\}$, $^{0}O_{f}$ el vector de posición del origen del marco final $\{f\}$ respecto al origen del marco base $\{0\}$

Las matrices de rotación que generan giros en los ángulo θ respecto al sistema de coordenadas cartesianas x, y, z se representan de la siguiente forma:

$$R_{x\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}; R_{y\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}; R_{z\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La convención de Denavit-Hartenberg tiene como objetivo obtener una matriz A como la composición de dos transformaciones de traslación y dos de rotación obteniendo:

$$ {}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

donde:

1. α_{i-1} es ángulo formado entre Z_{i-1} y Z_i medido alrededor de X_{i-1}
2. θ_i es el ángulo de articulación formado entre X_{i-1} y X_i medido alrededor de Z_i
3. a_{i-1} es la longitud del eslabón entre Z_{i-1} y Z_i medida a lo largo de X_{i-1}.
4. d_i es la distancia entre X_{i-1} y X_i medida a lo largo de Z_i

Considerando la figura 4.1 se asignan los sistemas de coordenadas a los eslabones del robot. Para la obtención del modelo cinemático se consideró seguir el enfoque de árbol cinemático, con ello, se realizará el análisis en 3 cadenas cinemáticas, una cadena cinemática para cada pie y otra cadena cinemática adicional para el sistema del péndulo transversal.

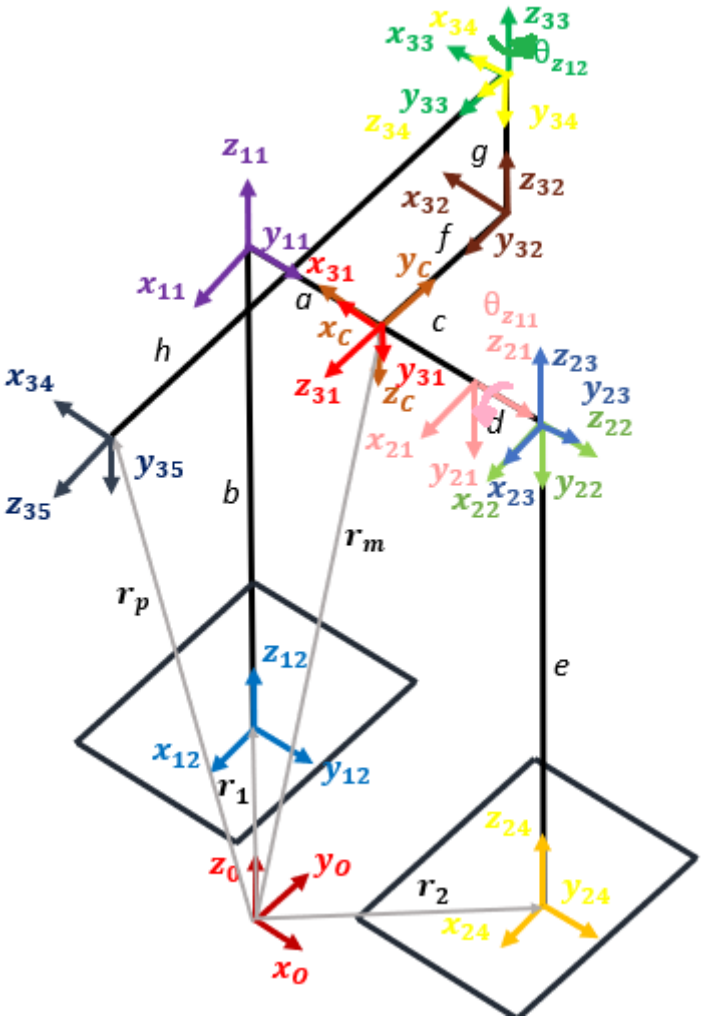


Figura 4.1: Sistemas de coordenadas con la postura de referencia

Los parámetros D-H para la cadena cinemática de la pierna izquierda y de la pierna derecha son respectivamente:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$\frac{\pi}{2}$	$-c$	0	$-\frac{\pi}{2}$
2	0	0	d	$\theta_{z_{21}}$
3	$\frac{\pi}{2}$	0	0	0
4	0	0	$-e$	0

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	π	a	0	$-\frac{\pi}{2}$
2	0	0	$-b$	0

Los parámetros D-H para la cadena cinemática del péndulo transversal son:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$\frac{\pi}{2}$	0	$-f$	0
2	$\frac{\pi}{2}$	0	0	0
3	0	0	g	$\theta_{z_{12}}$
4	$-\frac{\pi}{2}$	0	0	0
5	0	0	h	0

Las transformaciones homogéneas de las cadenas cinemáticas se realizan de acuerdo con las tablas de parámetros de D-H, teniendo la siguiente forma general:

$$\begin{aligned} {}^{12}T_C &= T_C^{11}T_{11}^{12} \\ {}^{24}T_C &= T_C^{21}T_{21}^{22}T_{22}^{23}T_{23}^{24} \\ {}^{35}T_C &= T_C^{31}T_{31}^{32}T_{32}^{33}T_{33}^{34}T_{34}^{35} \end{aligned}$$

donde

${}^{12}T_C$ transformación homogénea de posición en la pierna derecha
${}^{24}T_C$ transformación homogénea de posición en la pierna izquierda
${}^{35}T_C$ transformación homogénea de posición del péndulo

Aplicando el método de propagación de velocidades, se determinan las relaciones de la forma matricial entre las velocidades articulares y las velocidades traslacionales. Para ello se utilizaron las matrices homogéneas obtenidas por el método de D-H anteriormente.

Para la cadena cinemática de la pierna derecha se tiene que la propagación de velocidades es de la siguiente forma:

Para $i = 0$:

$$\begin{aligned} {}^{11}\omega_{11} &= {}^{11}_{10}R\,{}^{10}\omega_{10} \\ {}^{11}\nu_{11} &= {}^{11}_{10}R\left({}^{10}\nu_{10} + {}^{10}\,\omega_{10} \times {}^{10}\,P_{11}\right) \end{aligned}$$

Para $i = 1$

$$\begin{aligned} {}^{12}\omega_{12} &= {}^{12}_{11}R\,{}^{11}\omega_{11} \\ {}^{12}\nu_{12} &= {}^{12}_{11}R\left({}^{11}\nu_{11} + {}^{11}\,\omega_{11} \times {}^{11}\,P_{12}\right) \end{aligned}$$

Para la cadena cinemática de la pierna izquierda se tiene que la propagación de velocidades es de la siguiente forma:

Para $i = 0$:

$$\begin{aligned} {}^{21}\omega_{21} &= {}^{21}_{20}R\,{}^{20}\omega_{20} \\ {}^{21}\nu_{21} &= {}^{21}_{20}R\left({}^{20}\nu_{20} + {}^{20}\,\omega_{20} \times {}^{20}\,P_{21}\right) \end{aligned}$$

Para $i = 1$:

$$\begin{aligned} {}^{22}\omega_{22} &= {}^{22}_{21}R\,{}^{21}\omega_{21} + \dot{\theta}_{z_{21}} z_1 \\ {}^{22}\nu_{22} &= {}^{22}_{21}R\left({}^{21}\nu_{21} + {}^{21}\,\omega_{21} \times {}^{21}\,P_{22}\right) \end{aligned}$$

Para $i = 2$:

$$\begin{aligned} {}^{23}\omega_{23} &= {}^{23}_{22}R\,{}^{22}\omega_{22} \\ {}^{23}\nu_{23} &= {}^{23}_{22}R\left({}^{22}\nu_{22} + {}^{22}\,\omega_{22} \times {}^{22}\,P_{23}\right) \end{aligned}$$

Para $i = 3$:

$$\begin{aligned} {}^{24}\omega_{24} &= {}^{24}_{23}R\,{}^{23}\omega_{23} \\ {}^{24}\nu_{24} &= {}^{24}_{23}R\left({}^{23}\nu_{23} + {}^{23}\,\omega_{23} \times {}^{23}\,P_{24}\right) \end{aligned}$$

Para la cadena cinemática del péndulo transversal se tiene que la propagación de velocidades es de la siguiente forma:

Para $i = 0$:

$$\begin{aligned} {}^{31}\omega_{31} &= {}^{31}_{30}R\,{}^{30}\omega_{30} \\ {}^{31}\nu_{31} &= {}^{31}_{30}R\left({}^{30}\nu_{30} + {}^{30}\,\omega_{30} \times {}^{30}\,P_{31}\right) \end{aligned}$$

Para $i = 1$:

$$\begin{aligned} {}^{32}\omega_{32} &= {}^{32}_{31}R\,{}^{31}\omega_{31} \\ {}^{32}\nu_{32} &= {}^{32}_{31}R\left({}^{31}\nu_{31} + {}^{31}\,\omega_{31} \times {}^{31}\,P_{32}\right) \end{aligned}$$

Para $i = 2$:

$$\begin{aligned} {}^{33}\omega_{33} &= {}^{33}_{32}R\,{}^{32}\omega_{32} + \dot{\theta}_{z_{32}} z_3 \\ {}^{33}\nu_{33} &= {}^{33}_{32}R\left({}^{32}\nu_{32} + {}^{32}\,\omega_{32} \times {}^{32}\,P_{33}\right) \end{aligned}$$

Para $i = 3$:

$$\begin{aligned} {}^{34}\omega_{34} &= {}^{34}_{33}R\,{}^{33}\omega_{33} \\ {}^{34}\nu_{34} &= {}^{34}_{33}R\left({}^{33}\nu_{33} + {}^{33}\,\omega_{33} \times {}^{33}\,P_{34}\right) \end{aligned}$$

Para $i = 4$:

$$\begin{aligned} {}^{35}\omega_{35} &= {}^{35}_{34}R\,{}^{34}\omega_{34} \\ {}^{35}\nu_{35} &= {}^{35}_{34}R\left({}^{34}\nu_{34} + {}^{34}\,\omega_{34} \times {}^{34}\,P_{35}\right) \end{aligned}$$

transformando el resultado de la propagación de velocidades en forma matricial, se tiene que la estructura general es:

$$\dot{\chi} = J_p V \begin{pmatrix} \dot{\theta}_{z_{12}} \dot{\theta}_{z_{12}} \end{pmatrix}$$

donde:

J_p es el Jacobiano geométrico

V vector de velocidades articulares

4.2. Modelo dinámico

La obtención del modelo dinámico se realizó de forma independiente en los dos principales movimientos requeridos para asegurar el caminado dinámico del robot bípedo semi-pasivo.

1. Modelo dinámico del movimiento transitorio en el plano frontal
2. Modelo dinámico del ciclo de caminado

4.2.1. Modelo dinámico del plano frontal

El modelo dinámico que representa el movimiento transitorio del bípedo se basó del trabajo de [2].

De la figura 4.2 se consideran dos posibles casos en el movimiento transitorio que requiere el bípedo por sus plantas del pie curvas. Los detalles del cálculo basado en el trabajo de [2], se encuentra en el apéndice B.

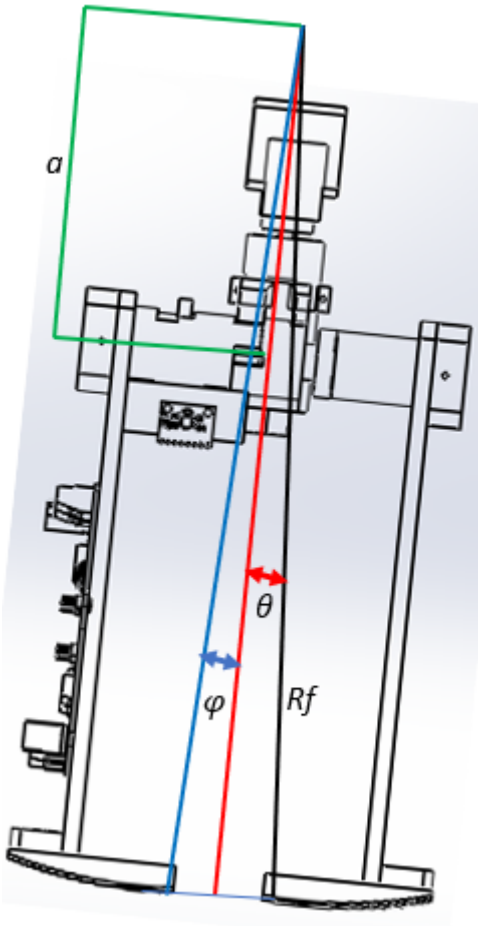


Figura 4.2: Plano frontal del robot bípedo pasivo

donde en [3]:

a es la distancia del centro del radio al centro de masa
R_f Radio del pie en el plano frontal
θ Ángulo de giro
φ Ángulo que se forma de la línea media del robot y R_f

La ecuación de movimiento de Lagrange general dada por:

$$\tau = M(q)\,\ddot{q} + C\left(q, \dot{q}\right)\dot{q} + G(q)$$

donde:

M es la matriz de inercia
C es la matriz de Coriolis
G es el vector de términos gravitacionales
q coordenadas generalizadas de posición
$\dot{q}$ coordenadas generalizadas de velocidad

1. **CASO 1** $|\theta| > \varphi$
   Se tiene la ecuación de movimiento general de Lagrange

$$\begin{aligned} M(\theta) &= m\left(R_f^2 + a^2 - 2R_f a\cos(\theta)\right) + I \\ C\left(\theta, \dot{\theta}\right) &= mR_f a\sin(\theta)\,\dot{\theta} \\ G(\theta) &= mg\left(a\sin(\theta)\right) \end{aligned}$$

   donde

   m masa total del bípedo
   I Tensor de inercia total del robot

2. **CASO 2** $|\theta| \leq \varphi$
   Se tiene la ecuación de movimiento general de Lagrange

$$\begin{aligned} M(\theta) &= m\left(R_f^2 + a^2 - 2R_f a\cos(\varphi)\right) + I \\ C\left(\theta, \dot{\theta}\right) &= 0 \\ G(\theta) &= mg\left(R_f\sin(\theta - \varphi) - a\sin(\theta)\right) \end{aligned}$$

   Se considera que el signo φ cambia $\theta < 0$, al tener dicho comportamiento se considera que los elementos de la ecuación de movimiento general de Lagrange son:

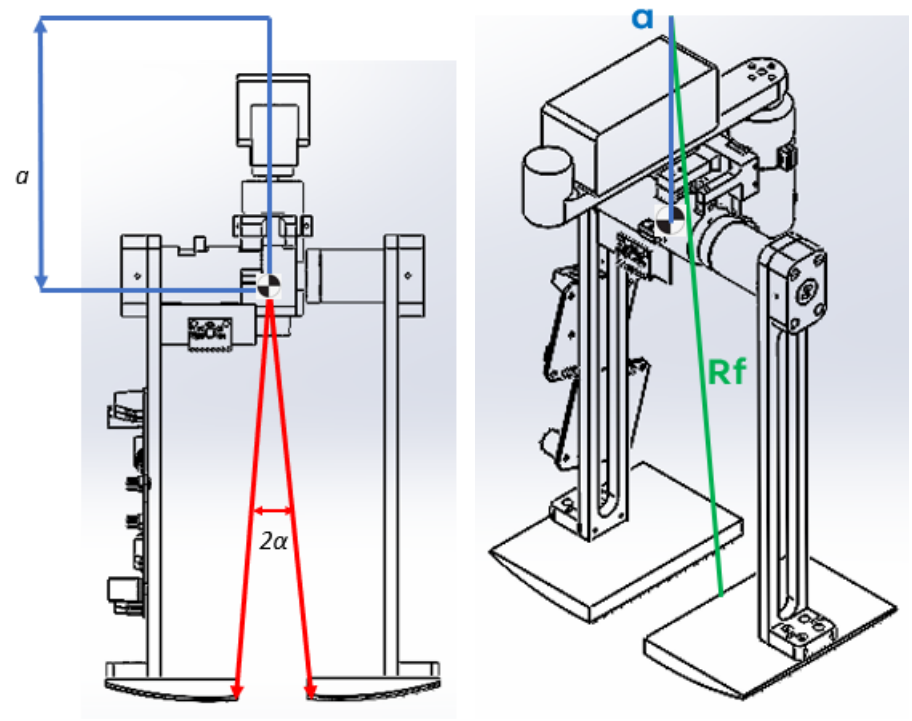
$$\begin{aligned} M(\theta) &= m\left(R_f^2 + a^2 - 2R_f a\cos(\varphi)\right) + I \\ C\left(\theta, \dot{\theta}\right) &= 0 \\ G(\theta) &= mg\left(R_f\sin(\theta - sign(\theta)\,\varphi) - a\sin(\theta)\right) \end{aligned}$$

   La transferencia de velocidades en el momento de impacto al dar un paso se rige por la ecuación

$$\dot{\theta}^{+} = \dot{\theta}^{-}\cos(2\alpha)$$

   donde 2α se muestra en la figura 4.3

$$2\alpha = 2\tan^{-1}\left(\frac{R_f\sin(\varphi)}{Rf\cos(\varphi) - a}\right)$$



(a) Parámetro de transferencia de velocidad vista plano frontal (b) Parámetro de transferencia de velocidad

Figura 4.3: Transferencia de velocidad [1]

4.2.2. Modelo dinámico del ciclo de marcha

Para deducir un modelo dinámico, se requiere encontrar las relaciones matemáticas entre las coordenadas generalizadas q y las fuerzas generalizadas Q. En este trabajo de tesis se muestra la deducción por el método basado en energías de Euler-Lagrange.

De tal forma que se definen las coordenadas generalizadas $q_j = 1 \leq i \leq 9$ del robot bípedo y sus respectivas derivadas como:

$$\begin{aligned}
q_1 &= x_B \quad \dot{q_1} = \dot{x_B} \\
q_2 &= y_B \quad \dot{q_2} = \dot{y_B} \\
q_3 &= z_B \quad \dot{q_3} = \dot{z_B} \\
q_4 &= \theta_B \quad \dot{q_4} = \dot{\theta_B} \\
q_5 &= \phi_B \quad \dot{q_5} = \dot{\phi_B} \\
q_6 &= \psi_B \quad \dot{q_6} = \dot{\psi_B} \\
q_7 &= \theta_P \quad \dot{q_7} = \dot{\theta_P} \\
q_8 &= \theta_i \quad \dot{q_8} = \dot{\theta_i} \\
q_9 &= \theta_d \quad \dot{q_9} = \dot{\theta_d}
\end{aligned}$$

Para la función de Lagrange, se suman las energías relacionadas al centro de masa L_B, energía relacionada al péndulo L_P y la correspondiente energía de cada pie (L_i para el pie izquierdo y L_d para el pie derecho). La función de Lagrange L del sistema está dada por:

$$L = L_B + L_P + L_i + L_d \tag{4.1}$$

El lagrangiano de cada energía de la ecuación (4.1) constituye a la diferencia entre la energía cinética y la energía potencial. Las energías asociadas a la cadera y a cada pie están dadas por las expresiones (4.2), (4.3), (4.4) y (4.5).

$$L_B\left(q_B,\dot{q}_B\right) = K\left(q_B,\dot{q}_B\right) - V\left(q_B\right) \tag{4.2}$$
$$L_P\left(q_P,\dot{q}_P\right) = K\left(q_P,\dot{q}_P\right) - V\left(q_P\right) \tag{4.3}$$
$$L_i\left(q_i,\dot{q}_i\right) = K\left(q_i,\dot{q}_i\right) - V\left(q_i\right) \tag{4.4}$$
$$L_d\left(q_d,\dot{q}_d\right) = K\left(q_d,\dot{q}_d\right) - V\left(q_d\right) \tag{4.5}$$

Para la obtención de la energía cinética, de acuerdo con [50] y [51], tiene la siguiente forma:

$$K_B = \frac{1}{2}\left(m_B\left(\dot{b}_B^0\right)^T b_B^0 + \left(\omega_B^0\right)^T\left(I_B^0\omega_B^0\right)\right) \tag{4.6}$$
$$K_P = \frac{1}{2}\left(m_P\left(\dot{b}_P^0\right)^T b_P^0 + \left(\omega_P^0\right)^T\left(I_P^0\omega_P^0\right)\right) \tag{4.7}$$
$$K_i = \frac{1}{2}\left(m_i\left(\dot{b}_i^0\right)^T b_i^0 + \left(\omega_i^0\right)^T\left(I_i^0\omega_i^0\right)\right) \tag{4.8}$$
$$K_d = \frac{1}{2}\left(m_d\left(\dot{b}_d^0\right)^T b_d^0 + \left(\omega_d^0\right)^T\left(I_d^0\omega_d^0\right)\right) \tag{4.9}$$

donde

K_B es la energía cinética del centro de masa
K_P es la energía cinética del péndulo transversal
K_i es la energía cinética de la pierna izquierda
K_d es la energía cinética de la pierna derecha
m_b masa del eslabón de la cadera
m_p masa del eslabón del péndulo transversal
m_i masa del eslabon de la pierna izquierda
m_d masa del eslabon de la pierna derecha
$\dot{b}_B^0$velocidad lineal del centro de gravedad del centro de masa en el marco inercial
$\dot{b}_P^0$velocidad lineal del centro de gravedad del péndulo transversal en el marco inercial
$\dot{b}_i^0$ velocidad lineal del centro de gravedad de la pierna izquierda en el marco inercial
$\dot{b}_d^0$velocidad lineal del centro de gravedad de la pierna derecha en el marco inercial
b_C^0 vector de posición del centro de gravedad de la cadera en el marco inercial
b_i^0 vector de posición del centro de gravedad de la pierna izquierda en el marco inercial
b_d^0 vector de posición del centro de gravedad de la pierna derecha en el marco inercial
ω_B^0 velocidad angular del centro de masa referido al marco inercial
ω_P^0 velocidad angular del péndulo transversal referido al marco inercial
ω_i^0 velocidad angular de la pierna izquierda referido al marco inercial
ω_d^0 velocidad angular de la pierna derecha referido al marco inercial

I_B^0 momento de inercia del centro de masa
I_P^0 momento de inercia del péndulo transversal
I_i^0 momento de inercia de la pierna izquierda
I_i^0 momento de inercia de la pierna derecha

siendo matrices simétricas I_B^B, I_P^P, I_i^i y I_d^d de la siguiente forma:

$$I_B^B = \begin{bmatrix} I_{xx_B} & I_{xy_B} & I_{xz_B} \\ I_{yx_B} & I_{yy_B} & I_{yz_B} \\ I_{zx_B} & I_{zy_B} & I_{zz_B} \end{bmatrix}; I_P^P = \begin{bmatrix} I_{xx_P} & I_{xy_P} & I_{xz_P} \\ I_{yx_P} & I_{yy_P} & I_{yz_P} \\ I_{zx_P} & I_{zy_P} & I_{zz_P} \end{bmatrix};$$

$$I_i^i = \begin{bmatrix} I_{xx_i} & I_{xy_i} & I_{xz_i} \\ I_{yx_i} & I_{yy_i} & I_{yz_i} \\ I_{zx_i} & I_{zy_i} & I_{zz_i} \end{bmatrix}; I_d^d = \begin{bmatrix} I_{xx_d} & I_{xy_d} & I_{xz_d} \\ I_{yx_d} & I_{yy_d} & I_{yz_d} \\ I_{zx_d} & I_{zy_d} & I_{zz_d} \end{bmatrix}$$

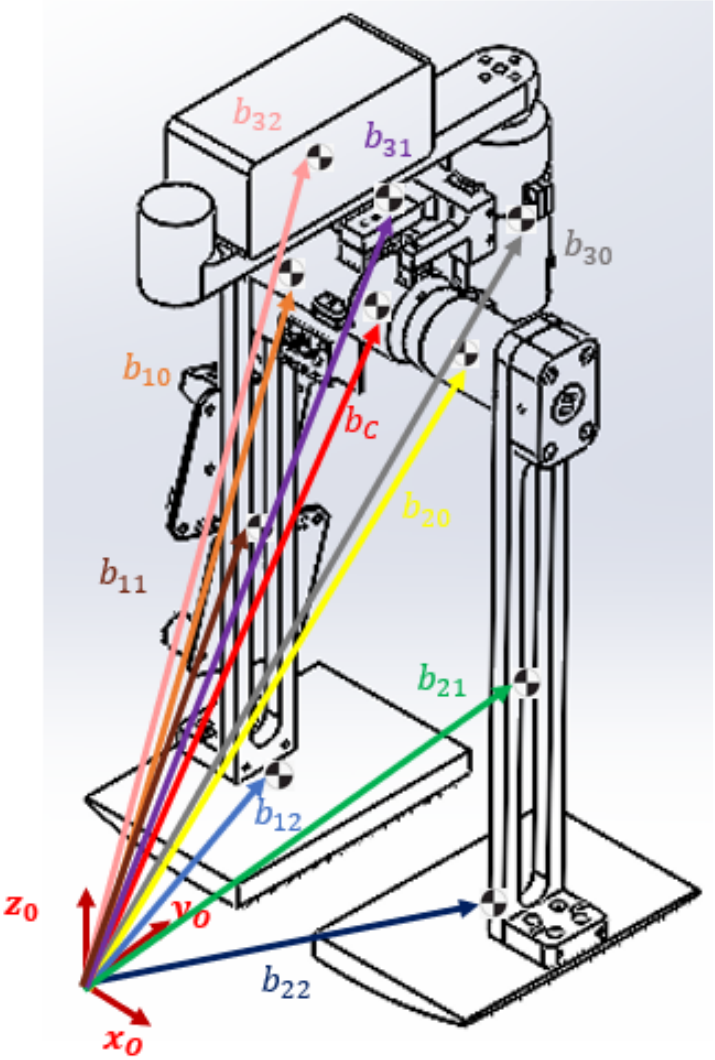


Figura 4.4: Vectores de posición del marco inercial a los centros de masa de cada componente

Por otro lado, en la figura 4.4 los vectores de posición en el marco inercial de los centros de gravedad de la cadera y de cada pierna está dada por:

$$\begin{aligned} b_B^0 &= r_B^0 + r_{GB}^0 \\ b_P^0 &= r_B^0 + r_P^0 + r_{GP}^0 \\ b_i^0 &= r_B^0 + r_i^0 + r_{Gi}^0 \\ b_d^0 &= r_B^0 + r_d^0 + r_{Gd}^0 \end{aligned}$$

donde $r_{GB} = R_B^0 r_{GB}^B$, $r_{GP} = R_P^0 r_{GP}^P$, $r_{Gi} = R_i^0 r_{Gi}^i$ y $r_{Gd} = R_d^0 r_{Gd}^d$ son las transformaciones de los vectores r_{GB}^B, r_{GP}^P, r_{Gi}^i y r_{Gd}^d en sus marcos locales.

Derivando los vectores b_B^0, b_P^0, b_i^0 y b_d^0 con respecto al tiempo se tiene:

$$\begin{aligned}
\dot{b}_B^0 &= v_B^0 + \omega_B^0 \times r_{GB}^0 \\
\dot{b}_P^0 &= v_B^0 + \omega_P^0 \times r_{GP}^0 \\
\dot{b}_i^0 &= v_B^0 + v_i^0 + \omega_i^0 \times r_{Gi}^0 \\
\dot{b}_d^0 &= v_B^0 + v_d^0 + \omega_d^0 \times r_{Gd}^0
\end{aligned}$$

La energía potencial se expresa de la siguiente forma:

$$\begin{aligned}
V_B &= -m_B g^T b_B^0 \\
V_P &= -m_P g^T b_P^0 \\
V_i &= -m_i g^T b_i^0 \\
V_d &= -m_d g^T b_d^0
\end{aligned}$$

donde

V_B energía potencial de la cadera
V_P energía potencial del péndulo transversal
V_i energia potencial de la pierna izquierda
V_d energia potencial de la pierna derecha
g^T vector de aceleración de gravedad en el marco inercial

sustituyendo las ecuaciones de la energía cinética de (4.6)-(4.9), y la ecuación de la energía potencial en el Lagrangiano de la ecuación 4.1 se obtiene la función Lagrangiana de la ecuación con la forma siguiente:

$$\begin{aligned}
L(q,\dot{q}) = &\left(\frac{1}{2}\left(m_B\left(\dot{b}_B^0\right)^T b_B^0 + \left(\omega_B^0\right)^T\left(I_B^0\omega_B^0\right)\right) + m_B g^T b_B^0\right) + \\
&\left(\frac{1}{2}\left(m_P\left(\dot{b}_P^0\right)^T b_P^0 + \left(\omega_P^0\right)^T\left(I_P^0\omega_P^0\right)\right) + m_P g^T b_P^0\right) \\
&\left(\frac{1}{2}\left(m_i\left(\dot{b}_i^0\right)^T b_i^0 + \left(\omega_i^0\right)^T\left(I_i^0\omega_i^0\right)\right) + m_i g^T b_i^0\right) + \\
&\left(\frac{1}{2}\left(m_d\left(\dot{b}_d^0\right)^T b_d^0 + \left(\omega_d^0\right)^T\left(I_d^0\omega_d^0\right)\right) + \ m_d g^T b_d^0\right)
\end{aligned}$$

Si se desarrolla y se simplifica las velocidades de los centros de gravedad b_B^0, b_P^0, b_i^0 y b_d^0 con respecto al tiempo en forma matricial, se tiene:

$$\begin{aligned}
\dot{b}_B^0 &= \left[\begin{array}{cccccccccc} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & 0 & 0 \end{array}\right]\dot{q} = J_B\dot{q} \\
\dot{b}_P^0 &= \left[\begin{array}{cccccccccc} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & e_{\theta P}^0 & 0 & 0 \end{array}\right]\dot{q} = J_P\dot{q} \\
\dot{b}_i^0 &= \left[\begin{array}{cccccccccc} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & e_{\theta i}^0 & 0 \end{array}\right]\dot{q} = J_i\dot{q} \\
\dot{b}_d^0 &= \left[\begin{array}{cccccccccc} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & 0 & e_{\theta d}^0 \end{array}\right]\dot{q} = J_d\dot{q}
\end{aligned}$$

donde

$$\begin{array}{ll} e^B_{\theta B} = i^0_{\theta B} \times r^0_{GB} & e^B_{\theta i} = i^0_{\theta B} \times \left(r^0_i + r^0_{Gi}\right) \\ e^B_{\phi B} = j^0_{\phi B} \times r^0_{GB} & e^B_{\phi i} = j^0_{\phi B} \times \left(r^0_i + r^0_{Gi}\right) \\ e^B_{\psi B} = k^0_{\psi B} \times r^0_{GB} & e^B_{\psi i} = k^0_{\psi B} \times \left(r^0_i + r^0_{Gi}\right) \\ e^B_{\theta P} = i^0_{\theta P} \times \left(r^0_P + r^0_{GP}\right) & e^B_{\theta d} = i^0_{\theta B} \times \left(r^0_d + r^0_{Gd}\right) \\ e^B_{\phi P} = j^0_{\phi P} \times \left(r^0_P + r^0_{GP}\right) & e^B_{\phi d} = j^0_{\phi B} \times \left(r^0_d + r^0_{Gd}\right) \\ e^B_{\psi P} = k^0_{\psi P} \times \left(r^0_P + r^0_{GP}\right) & e^B_{\psi d} = k^0_{\psi B} \times \left(r^0_d + r^0_{Gd}\right) \\ e^0_{\theta P} = \left(k^0_{\theta P} \times r^0_{GP}\right) & \\ e^0_{\theta d} = \left(k^0_{\theta d} \times r^0_{Gd}\right) & e^0_{\theta i} = \left(k^0_{\theta i} \times r^0_{Gd}\right) \end{array}$$

Las velocidades angulares de forma matricial se expresan como

$$\begin{array}{rcl} \omega_B & = & \dot{\theta_B} i^0_{0B} + \dot{\phi_B} j^0_{0B} + \dot{\psi_B} k^0_{0B} \\ \omega_B & = & \left[\begin{array}{ccccccccc} 0 & 0 & 0 & i^0_{0B} & j^0_{0B} & k^0_{0B} & 0 & 0 & 0 \end{array}\right] \dot{q} = W_B \dot{q} \\ \omega_P & = & \dot{\theta_B} i^0_{0B} + \dot{\phi_B} j^0_{0B} + \dot{\psi_B} k^0_{0B} + \dot{\theta_P} k^0_{0P} \\ \omega_P & = & \left[\begin{array}{ccccccccc} 0 & 0 & 0 & i^0_{0B} & j^0_{0B} & k^0_{0B} & k^0_{0P} & 0 & 0 \end{array}\right] \dot{q} = W_P \dot{q} \\ \omega_i & = & \dot{\theta_B} i^0_{0B} + \dot{\phi_B} j^0_{0B} + \dot{\psi_B} k^0_{0B} + \dot{\theta_i} k^0_{0i} \\ \omega_i & = & \left[\begin{array}{ccccccccc} 0 & 0 & 0 & i^0_{0B} & j^0_{0B} & k^0_{0B} & 0 & k^0_{0i} & 0 \end{array}\right] \dot{q} = W_i \dot{q} \\ \omega_d & = & \dot{\theta_B} i^0_{0B} + \dot{\phi_B} j^0_{0B} + \dot{\psi_B} k^0_{0B} + \dot{\theta_d} k^0_{0d} \\ \omega_d & = & \left[\begin{array}{ccccccccc} 0 & 0 & 0 & i^0_{0B} & j^0_{0B} & k^0_{0B} & 0 & k^0_{0d} & 0 \end{array}\right] \dot{q} = W_d \dot{q} \end{array}$$

Sustituyendo los términos del lagrangiano y aprovechando las notaciones en la ecuación de la función Lagrangiana se reescribe de la siguiente forma:

$$\begin{array}{rcl} L\left(q, \dot{q}\right) & = & \left(\frac{1}{2}\dot{q}^T N_B \dot{q} + m_B g^T b^0_B\right) + \left(\frac{1}{2}\dot{q}^T N_P \dot{q} + m_P g^T b^0_P\right) \\ & & + \left(\frac{1}{2}\dot{q}^T N_i \dot{q} + m_i g^T b^0_i\right) + \left(\frac{1}{2}\dot{q}^T N_d \dot{q} + m_B g^T b^0_d\right) \end{array}$$

donde

$$\begin{array}{rcl} N_B & = & m_B \left(J_B\right)^T J_B + \left(W_B\right)^T \left(I^0_B W_B\right) \\ N_P & = & m_P \left(J_P\right)^T J_P + \left(W_P\right)^T \left(I^0_P W_P\right) \\ N_i & = & m_i \left(J_i\right)^T J_i + \left(W_i\right)^T \left(I^0_i W_i\right) \\ N_d & = & m_d \left(J_d\right)^T J_d + \left(W_d\right)^T \left(I^0_d W_d\right) \end{array}$$

Las ecuaciones de movimiento de Euler-Lagrange son:

$$\frac{d}{dt}\left(\frac{\partial L\left(q, \dot{q}\right)}{\partial \dot{q}}\right) - \frac{\partial L\left(q, \dot{q}\right)}{\partial q} = Q - \Gamma^k_j \tag{4.10}$$

Donde Q es el vector restringido a los efectos provocados por los actuadores τ_f y τ_p, Γ^k_j son elementos que constituyen al vector de fuerzas de restricción relacionados con las fuerzas y momentos derivados del contacto del bípedo y el suelo durante su ciclo de marcha.

La ecuación de Euler-Lagrange es

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = D_j \ddot{q} + V_j \dot{q} + C_j$$

donde

$$\begin{aligned} D_j &= D_B + D_P + D_i + D_d \\ V_j &= V_B + V_P + V_i + V_d - \left(\acute{V}_B + \acute{V}_P + \acute{V}_i + \acute{V}_d\right) \\ C_j &= -\left(\acute{C}_B + \acute{C}_P + \acute{C}_i + \acute{C}_d\right) \end{aligned}$$

que a su vez son:

$$\begin{array}{cccc} D_B = \frac{\partial \dot{q}^T}{\partial \dot{q}} N_B & V_B = \frac{\partial \dot{q}^T}{\partial \dot{q}} \dot{N}_B & \acute{V}_B = \frac{1}{2}\dot{q}^T \frac{\partial N_B}{\partial q} & \acute{C}_B = m_B g^T \frac{\partial b_B^0}{\partial q} \\ D_P = \frac{\partial \dot{q}^T}{\partial \dot{q}} N_P & V_P = \frac{\partial \dot{q}^T}{\partial \dot{q}} \dot{N}_P & \acute{V}_P = \frac{1}{2}\dot{q}^T \frac{\partial N_P}{\partial q} & \acute{C}_P = m_P g^T \frac{\partial b_P^0}{\partial q} \\ D_i = \frac{\partial \dot{q}^T}{\partial \dot{q}} N_i & V_i = \frac{\partial \dot{q}^T}{\partial \dot{q}} \dot{N}_i & \acute{V}_i = \frac{1}{2}\dot{q}^T \frac{\partial N_i}{\partial q} & \acute{C}_i = m_i g^T \frac{\partial b_i^0}{\partial q} \\ D_d = \frac{\partial \dot{q}^T}{\partial \dot{q}} N_d & V_d = \frac{\partial \dot{q}^T}{\partial \dot{q}} \dot{N}_d & \acute{V}_d = \frac{1}{2}\dot{q}^T \frac{\partial N_d}{\partial q} & \acute{C}_d = m_d g^T \frac{\partial b_d^0}{\partial q} \end{array}$$

Fuerzas generalizadas por trabajo virtual

En las fuerzas generalizadas, en [52], el trabajo virtual δW que se encuentra restringido al par ejercido por los motores se expresa con

$$\delta W = F^T \delta r + M^T \delta Q = Q \delta q \tag{4.11}$$

donde F es el vector asociado a la fuerza que se aplica en el punto del cuerpo con el vector de posición r y M es el momento de la coordenada generalizada relacionada al cambio de orientación en el cuerpo que son contempladas en el vector de cambios virtuales en las coordenadas generalizadas δQ. Las fuerzas de reacción del piso se consideran en términos de los multiplicadores de Lagrange [51], el cálculo del trabajo virtual se simplifica a la siguiente expresión.

$$\delta W = \tau_f \delta Q_f^0 + \tau_p \delta Q_p^0 = Q^T \delta q \tag{4.12}$$

De la figura 4.5 las direcciones de los torques de los actuadores en las juntas rotacionales del bípedo respecto al marco inercial son:

$$\begin{aligned} \tau_f &= \tau_f k_3^0 \\ \tau_p &= \tau_p k_7^0 \end{aligned}$$

Donde el vector de cambios virtuales en las coordenadas generalizadas δq es:

$$\delta q = \begin{bmatrix} \delta x_B & \delta y_B & \delta z_B & \delta\theta_B & \delta\beta_B & \delta\gamma_B & \delta\theta_p & \delta\theta_i & \delta\theta_d \end{bmatrix}^T$$

Los vectores de velocidad angular se relacionan con los cambios virtuales de coordenadas generalizadas referenciadas a la orientación de cada eslabón del bípedo con las expresiones:

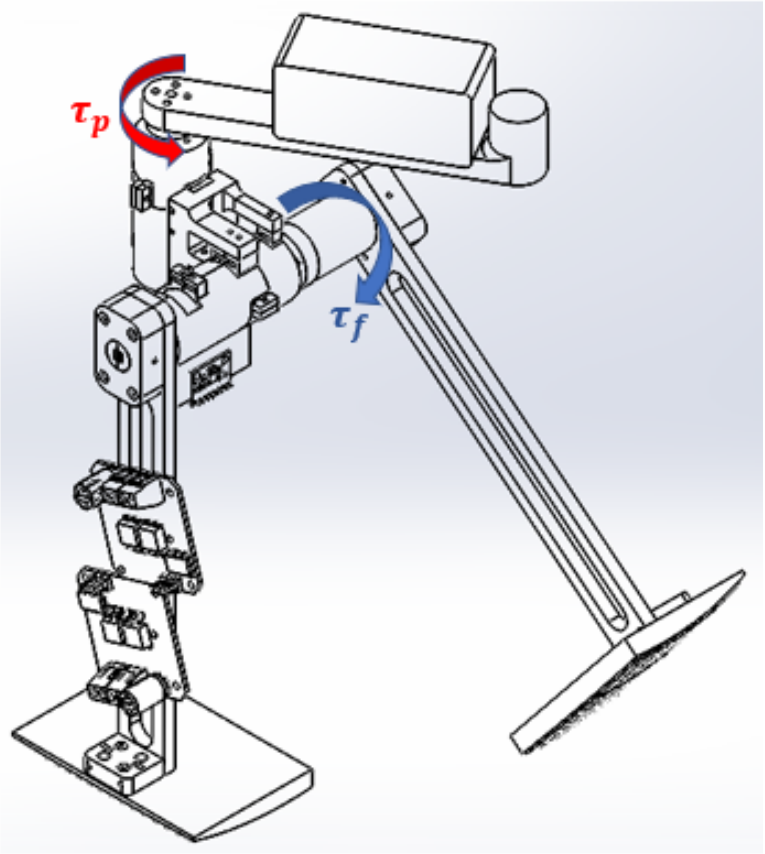


Figura 4.5: Torques ejercidos por los actuadores en las juntas rotacionales

$$
\begin{aligned}
\delta Q_p^0 &= \sum_{j=1}^{9} \frac{\partial \omega_p^0}{\partial \dot{q}_j}\delta q_j = \sum_{j=1}^{9} \frac{\partial\left(\dot{\theta}_B i_{\theta P}^B + \dot{\phi}_B j_{\phi P}^B + \dot{\psi}_B k_{\psi P}^B + \dot{\theta}_P k_{\theta P}^0\right)}{\partial \dot{q}}\delta q_j \\
&= i_{\theta P}^B \delta\theta_B + j_{\theta P}^B \delta\phi_B + k_{\theta P}^B \delta\psi_B + k_{\theta P}^0 \delta\theta_P \\
\delta Q_i^0 &= \sum_{j=1}^{9} \frac{\partial \omega_i^0}{\partial \dot{q}_j}\delta q_j = \sum_{j=1}^{9} \frac{\partial\left(\dot{\theta}_B i_{\theta i}^B + \dot{\phi}_B j_{\phi i}^B + \dot{\psi}_B k_{\psi i}^B + \dot{\theta}_i k_{\theta i}^0\right)}{\partial \dot{q}}\delta q_j \\
&= i_{\theta i}^B \delta\theta_B + j_{\theta i}^B \delta\phi_B + k_{\theta i}^B \delta\psi_B + k_{\theta i}^0 \delta\theta_i \\
\delta Q_d^0 &= \sum_{j=1}^{9} \frac{\partial \omega_d^0}{\partial \dot{q}_j}\delta q_j = \sum_{j=1}^{9} \frac{\partial\left(\dot{\theta}_B i_{\theta d}^B + \dot{\phi}_B j_{\phi d}^B + \dot{\psi}_B k_{\psi d}^B + \dot{\theta}_d k_{\theta d}^0\right)}{\partial \dot{q}}\delta q_j \\
&= i_{\theta d}^B \delta\theta_B + j_{\theta d}^B \delta\phi_B + k_{\theta d}^B \delta\psi_B + k_{\theta d}^0 \delta\theta_d
\end{aligned}
$$

Una vez que se han desarrollado los terminos δQ_p^0, δQ_i^0, δQ_d^0 y los vectores par τ_f y τ_p, se determina el trabajo virtual

$$
\begin{aligned}
SW &= \left(\tau_p k_7^0\right)^T \delta Q_p^0 + \left(\tau_f k_8^0\right)^T \delta Q_i^0 + \delta Q_d^0 \\
SW &= \left(\tau_p k_7^0\right)\left(i_{\theta B}^B \delta\theta_B + j_{\theta B}^B \delta\phi_B + k_{\theta B}^B \delta\psi_B + k_{\theta P}^0 \delta\theta_P\right) + \\
&\quad \left(\tau_f k_8^0\right)\left(i_{\theta B}^B \delta\theta_B + j_{\theta B}^B \delta\phi_B + k_{\theta B}^B \delta\psi_B + k_{\theta i}^0 \delta\theta_i\right) + i_{\theta B}^B \delta\theta_B + j_{\theta B}^B \delta\phi_B + k_{\theta B}^B \delta\psi_B + k_{\theta d}^0 \delta\theta_d \\
SW &= \left(\tau_p k_7^0 i_{\theta B}^B + \tau_f k_8^0 i_{\theta B}^B + i_{\theta B}^B\right)\delta\theta_B + \left(\tau_p k_7^0 j_{\theta B}^B + \tau_f k_8^0 j_{\theta B}^B + j_{\theta B}^B\right) j_{\theta B}^B \delta\phi_B + \\
&\quad \left(\tau_p k_7^0 k_{\theta B}^B + \tau_f k_8^0 k_{\theta B}^B + k_{\theta B}^B\right)\delta\psi_B + \tau_p k_7^0 k_{\theta P}^0 \delta\theta_P + \tau_f k_8^0 k_{\theta i}^0 \delta\theta_i + k_{\theta d}^0 \delta\theta_d
\end{aligned}
$$

Definiendo el vector τ

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \tau_p & \tau_f & 0 \end{bmatrix}$$

y el vector A_τ es

$$A_\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{71} & A_{72} & A_{73} & 0 & 0 & 0 & A_{77} & 0 & 0 \\ A_{81} & A_{82} & A_{83} & 0 & 0 & 0 & 0 & A_{88} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

donde

$$\begin{array}{llll} A_{71} = k_7^0 i_{\theta B}^B & A_{72} = k_7^0 j_{\theta B}^B & A_{73} = k_7^0 k_{\theta B}^B & A_{77} = k_7^0 k_{\theta P}^0 \\ A_{81} = k_8^0 i_{\theta B}^B & A_{82} = k_8^0 j_{\theta B}^B & A_{83} = k_8^0 k_{\theta B}^B & A_{88} = k_8^0 k_{\theta i}^0 \end{array}$$

$$B = \begin{bmatrix} i_{\theta B}^B & j_{\theta B}^B & k_{\theta B}^B & 0 & 0 & 0 & 0 & 0 & k_{\theta d}^0 \end{bmatrix}$$

resuelve δW de forma matricial con la forma

$$\delta W = \tau^T \left(A_\tau + B\right)^T \ \delta q = Q^T \delta q$$

El vector de fuerzas generalizadas Q^T será

$$Q^T = \left(\tau^T \left(A_\tau + B\right)^T\right)^T = \left(A_\tau + B\right)\tau$$

por lo tanto, el vector de fuerzas generalizadas se denota como:

$$Q = \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 & Q_5 & Q_6 & Q_7 & Q_8 & Q_9 \end{bmatrix}^T$$

Fuerzas de restricción

El ciclo de caminata del bípedo semi-pasivo se puede describir como las siguientes fases de contacto entre el piso y los pies [53]:

1. Soporte sencillo (SS).- El bípedo cumple con alguna de las dos condiciones descrito en el modelo dinámico del plano frontal, donde tiene un solo pie haciendo contacto con el piso, el bípedo tiene un comportamiento de una cadena cinemática abierta.

2. -Soporte doble (SD).- Al dar el paso, el bípedo mantiene los dos pies en contacto con el suelo, el bípedo evoluciona a un comportamiento de cadena cinemática cerrada.

Las condiciones cinemáticas del ciclo de caminado corresponden al vector de fuerzas de restricción Γ_k impuesta por el patrón de contacto de la base curva del pie con el suelo. La figura 4.6 se muestra el patrón de caminado para la deducción de fuerzas de restricción propuesto por [53], donde entre los instantes t_0 y t_3 se tiene el caso de soporte simple (SS), en los demás instantes de tiempo se efectuará el caso de soporte doble (SD).

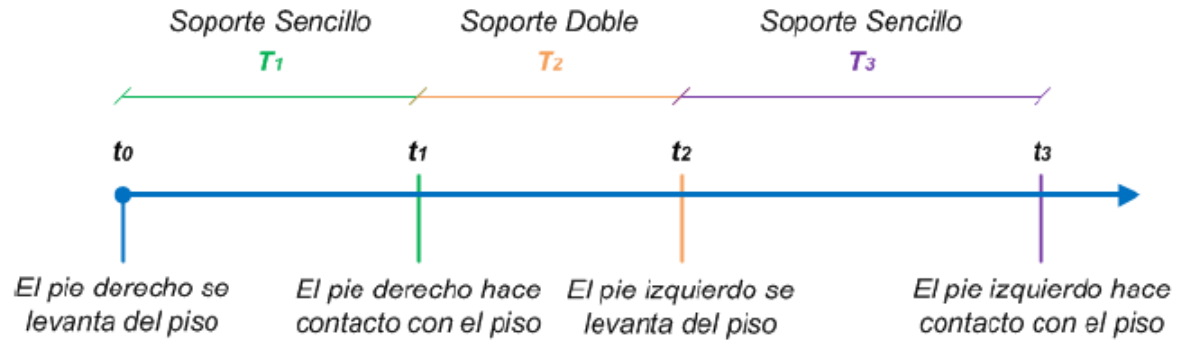


Figura 4.6: Secuencia de ciclo de caminado

A pesar de que el diseño de la base del pie es curvo, cuando se encuentra en SD durante su ciclo de caminado ambos pies permanecen paralelos al plano sagital.

Para cualquier tiempo del ciclo de caminado, se define el vector de fuerzas de restricción como:

$$\Gamma_k = J^T \lambda_k^0 = \left(\frac{\partial \Phi_k^0}{\partial q} \right)^T \lambda_k^0 \tag{4.13}$$

donde

J^T es la matriz Jacobiana transpuesta
Φ_k^0 vector de ecuaciones de restricción holónomicas
λ_k^0 vector de multiplicadores de Lagrange

El vector de multiplicadores de Lagrange reúne los valores relativos al marco inercial de las fuerzas y momentos de reacción por el contacto entra el pie y el suelo.

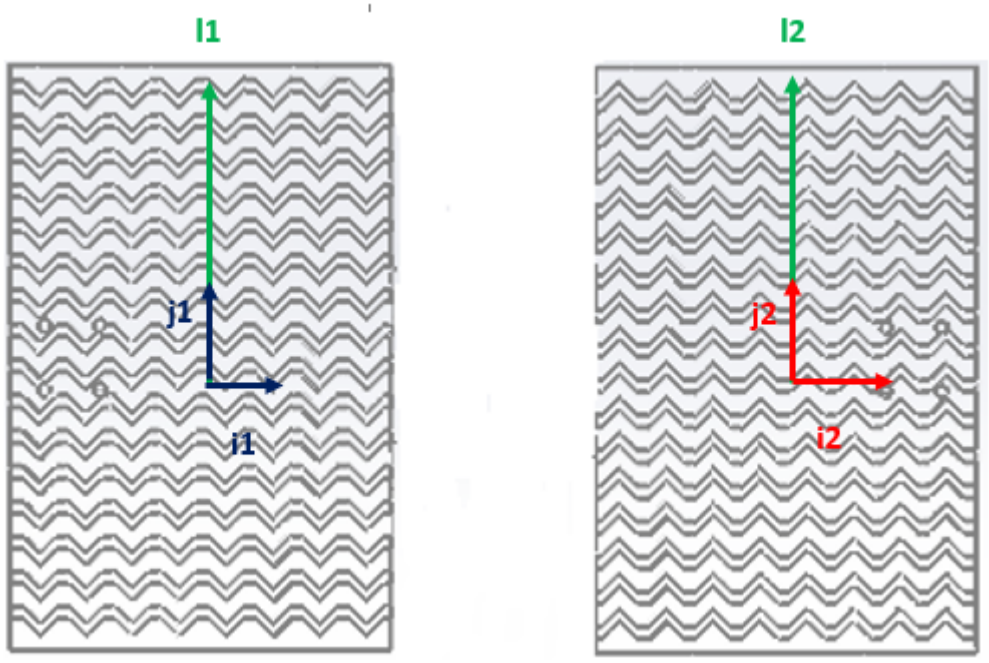


Figura 4.7: Planteamiento de las ecuaciones de restricción en la planta del pie

se define el vector

$$l_k^0 = b_{k_y} j_k^0, \text{ para } k = 1, 2$$

La representación gráfica de los vectores l_k^0 se muestran en la figura (TAL FIGURA 4.7), los valores de b_{k_y}es la definición de cada vector de posición por la descomposición de puntos para la planta del pie.

Las restricciones en el intervalo SS

$$\begin{aligned}\varphi_1^1(q_j(t)) &= \left(l_k^0(q_j(t))\right)^T k_1 = 0\\ \varphi_2^1(q_j(t)) &= \left(l_k^0(q_j(t))\right)^T j_1 = 0\\ \varphi_3^1(q_j(t)) &= \left(l_k^0(q_j(t))\right)^T i_1 = 0\end{aligned}$$

donde $q_j(t)$ es la orientación de los vectores l_k^0 respecto a la base inercial y su dependencia con los valores asumidos por las coordenadas generalizadas en función del tiempo t. El vector de ecuaciones de restricciones holonómicas se escribe de la siguiente forma:

$$\Phi_1^0 = \begin{bmatrix}\varphi_1^1(q_j(t))\\ \varphi_2^1(q_j(t))\\ \varphi_3^1(q_j(t))\end{bmatrix} = 0$$

La matriz jacobiana del vector de restricciones holonómicas se define como:

$$\frac{\partial \Phi_1^0}{\partial q_i} = \begin{bmatrix}\frac{\partial \varphi_1^1}{\partial q_1} & \cdots & \frac{\partial \varphi_1^1}{\partial q_9}\\ \frac{\partial \varphi_2^1}{\partial q_1} & \cdots & \frac{\partial \varphi_2^1}{\partial q_9}\\ \frac{\partial \varphi_3^1}{\partial q_1} & \cdots & \frac{\partial \varphi_3^1}{\partial q_9}\end{bmatrix} \text{ para } i = 1,2,3,4,5,6,7,8,9$$

con el vector de multiplicadores de Lagrange de la siguiente forma:

$$\lambda_1^0 = \begin{bmatrix}\lambda_1^1 & \lambda_2^1 & \lambda_3^1\end{bmatrix}$$

Para el intervalo T2 del ciclo de caminado, al tener doble soporte (los dos pies en contacto con el suelo), la ecuación que describe las fuerzas de restricción se escribe tiene

$$\Gamma_2 = \left(\frac{\partial \Phi_2^0}{\partial q}\right)^T \lambda_2^0$$

La matriz jacobiana del vector de restricciones holonómicas se define como:

$$\frac{\partial \Phi_2^0}{\partial q_i} = \begin{bmatrix}\frac{\partial \varphi_1^2}{\partial q_1} & \cdot & \cdot & \cdot & \frac{\partial \varphi_1^2}{\partial q_9}\\ \cdot & \cdot & & & \cdot\\ \cdot & & \cdot & & \cdot\\ \cdot & & & \cdot & \cdot\\ \frac{\partial \varphi_6^2}{\partial q_1} & \cdot & \cdot & \cdot & \frac{\partial \varphi_6^2}{\partial q_9}\end{bmatrix} \text{ para } i = 1,2,...,9$$

donde el vector de multiplicadores de Lagrange es

$$\lambda_2^0 = \begin{bmatrix}\lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \lambda_4^2 & \lambda_5^2 & \lambda_6^2\end{bmatrix}$$

Para el intervalo T3 del ciclo de caminado, vuelve a tener soporte sencillo. La ecuación que describe las fuerzas de restricción son:

$$\Gamma_3 = \left(\frac{\partial \Phi_3^0}{\partial q}\right)^T \lambda_3^0$$

con el vector de multiplicadores de Lagrange de la siguiente forma:

$$\lambda_3^0 = \begin{bmatrix}\lambda_1^3 & \lambda_2^3 & \lambda_3^3\end{bmatrix}$$

El vector de fuerzas de restricción se denota como

$$\Gamma_k = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix}$$

Las restricciones holonómicas durante el ciclo de caminado en cualquier instante de tiempo definen el cambio de las coordenadas generalizadas respecto al tiempo y la orientación del bípedo respecto al marco inercial.

De la ecuacion de movimiento de Euler - Lagrange se obtiene

$$D_j\ddot{q} + V_j\dot{q} + C_j = Q_j + \Gamma_j^k$$

para $j = 1, 2, ..., 9$. La estructura de las matrices D_j, V_jy C_j no se modifican ya que no dependen de la fase del ciclo de caminado en la que se encuentre el bípedo, sin embargo, las fuerzas de restricción holonómicas si dependerán del ciclo de marcha del bípedo, por lo tanto, la ecuación dinámica general que describe el ciclo de caminado del bípedo se describe de la siguiente expresión.

$$D_j\ddot{q} + V_j\dot{q} + C_j = A_\tau\tau + \left(\frac{\partial\Phi_k^0}{\partial q}\right)^T \lambda_k^0$$

con restricciones

$$\Phi_k^0 = 0$$

para $t \in T_k$, donde $k = 1, 2, 3$ representa las fases del ciclo de marcha

Se observa que diseñar un controlador para los modelos dinámicos descritos en este capítulo, es complejo con cualquier enfoque de control clásico. Sin embargo, en el siguiente capítulo se describirá el diseño de un controlador que no dependa de los modelos.

Capítulo 5

Control basado en DRL para plataforma experimental

En el presente capítulo se describen a detalle la implementación del enfoque de aprendizaje automático basado en datos para el bípedo con ciclo de caminado semi-pasivo. Se muestra la implementación en el software de Matlab aprovechando el toolbox ya implementado en Simulink del enfoque de Reinforcement Learning. Se detalla todos los elementos de diseño en Deep Reinforcement Learning (observaciones, acciones, funciones de recompensa, redes neuronales y entrenamiento). Se presenta la gráfica obtenida de la convergencia del entrenamiento y las gráficas que describen el comportamiento del robot bípedo durante el ciclo de caminado.

Para el bípedo con ciclo de caminado semi-pasivo se retoma los conceptos plasmados en el capítulo 2 del marco teórico; en la sección 5.1, se describe de forma secuencial, la metodología de diseño para Deep Reinforcement Learnign, que será una guía de diseño para controladores con este enfoque. En la sección 5.2 se describe de forma general el objetivo de control, la elección de observaciones, acciones a realizar y condiciones de paro de emergencia, además, de los datos requeridos que se envían desde el sistema embebido al software de Matlab. En la sección 5.3, se detalla la estructura para las funciones de recompensa. Para la sección 5.4, se muestra la estructura y funciones de activación de las redes neuronales para el crítico y actor que requerirá cada agente computacional. En la sección 5.5, se explica los agentes utilizados y sus parámetros de entrenamiento; por último, para la sección 5.6, se muestran los resultados de la evolución en la función de recompensa por episodio, y las gráficas que describen el comportamiento del ciclo de caminado del robot bípedo (gráfica error vs tiempo y la gráfica de la acción de voltaje vs tiempo).

5.1. Metodología general para Deep Reinforcement Learning

La metodología general que fue utilizado para el diseño del enfoque de Reinforcement Learning fue el siguiente:

1. Descripción de entorno.
    - *a*) Selección del número de agentes.
    - *b*) Descripción de variables de interés que provienen de los sensores implementados en la plataforma experimental.
    - *c*) Definición del tiempo de muestreo Ts, tiempo final por episodio Tf y número máximo de episodios en el entrenamiento.
2. Selección de las variables de interés.

   a) Variables de observación en el contexto de Reinforcement Learning.
   b) Acciones como salida del agente.
   c) Condiciones de paro de emergencia.

3. Diseño de funciones de recompensa.

4. Diseño de parámetros de los agentes computacionales.

   a) Selección de tipo de agentes computacionales en Deep Reinforcement Learning.
   b) Diseño de estructura de las redes neuronales actor-crítico
      1) Elección del número de capas.
      2) Selección del número de neuronas en cada capa de la red.
      3) Elección de las funciones de activación.

5. Configuración de parámetros de entrenamiento.

De la metodología anterior, se describe a detalle el enfoque de Deep Reinforcement Learning aplicado al robot bípedo que logré complete el ciclo de caminado semi-pasivo.

5.2. Entorno

Para la solución del robot bípedo, se propuso una solución de entrenamiento multiagente que actúan de forma colaborativa para resolver el problema de regulación de posiciones deseadas para completar el ciclo de caminado. El objetivo principal es hacer que el bípedo se desplace en linea recta, realizando trabajo conmutado entre el actuador de las piernas y el actuador del péndulo. Este último genera el movimiento transitorio necesario en el plano frontal, permitiendo que la actuación de la pierna sea sin obstáculo por rozamiento con el suelo.

Se propone dos agentes computacionales, que de forma independiente, enviarán acciones de bajo nivel (voltajes) a su respectivo actuador; dichos valores de voltaje serán mapeados por el microcontrolador transformándolos en valores de ancho de pulso (PWM). Para el ciclo de marcha, se proporciona la conmutación de cada fase de caminado utilizando una máquina de estados, esta envía la información de referencia de posición angular, sentido de giro y condiciones para las transiciones entre estados. La razón principal del uso de dos agentes computacionales es porque, scuando la máquina de estados envía la referencia deseada, al subdividir la tarea de un agente a dos, permite facilitar la estimación de las acciones para lograr el caminado semipasivo en el robot bípedo. Sin embargo, la solución pudo haberse obtenido con el uso de un agente computacional, la complejidad de este enfoque es que la función de recompensa y el vector de observación se vuelven más compleja en su diseño, ya que depende de mayor número de variables independientes que permitan evaluar el desempeño del sistema. La complejidad de tener un enfoque multiagente, es la elección del vector de observación que permite al agente estimar acciones que al reproducirse en su actuador, tiene repercusión en todo el sistema.

Las variables que provienen de los sensores son las siguientes:

θ_1 posición angular proveniente del encoder del actuador en la pierna

θ_2 posición angular proveniente del encoder del actuador del péndulo transversal

a_{θ_x} rotación angular del acelerómetro en el eje x [deg]

a_{θ_y} rotación angular del acelerómetro en el eje y [deg]

a_{θ_z} rotación angular del acelerómetro en el eje z [deg]

V_1 voltaje en el actuador de la pierna [V]

V_2 voltaje en el actuador del péndulo transversal [V]

Se define como parámetros iniciales los siguientes valores: el tiempo de muestreo de $Ts = 10[ms]$, tiempo final o máximo tiempo de duración de cada episodio $Tf = 10[s]$ y número máximo de episodios durante el entrenamiento $Numero_episodios = 1000$.

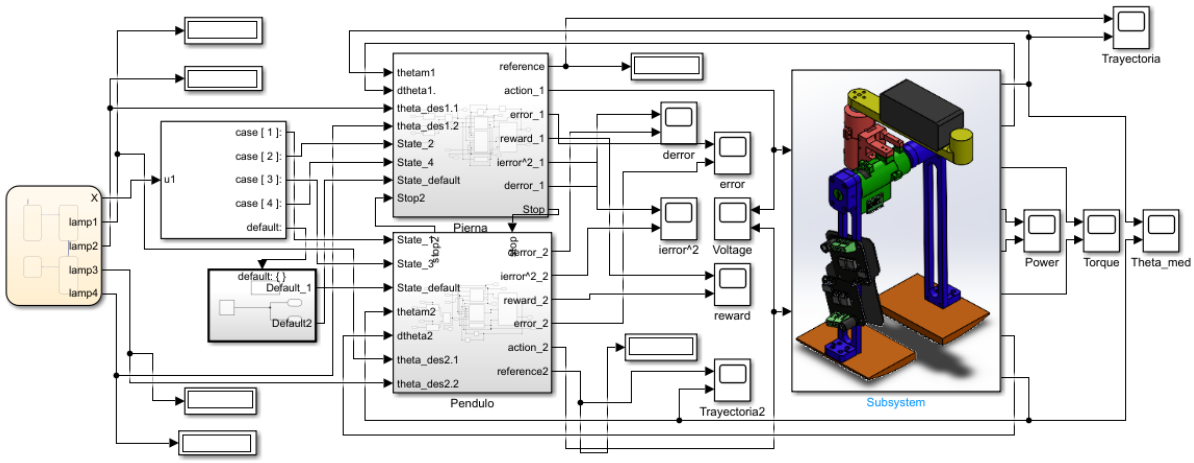


Figura 5.1: Diagrama de bloques general del entorno de Simulink

5.3. Observaciones, acciones y condiciones de paro de emergencia

5.3.1. Observaciones

En el capitulo 2, se definió como observaciones aquellas variables de interés que permitirán al algoritmo de Reinforcement Learning entender la repercusión de las acciones que envíe al bípedo en cada instante de tiempo.

Las observaciones definidas son:

$$Observador\_1 = \left[\cos(e(\theta_1)) \quad \sin(e(\theta_1)) \quad e(\dot{\theta_1}) \quad \int e(\theta_1)\,dt \quad a_{\theta_x} \quad a_{\theta_y} \quad a_{\theta_z} \quad \dot{\theta}_1 \right]$$

$$Observador\_2 = \left[\cos(e(\theta_2)) \quad \sin(e(\theta_2)) \quad e(\dot{\theta_2}) \quad \int e(\theta_2)\,dt \quad a_{\theta_x} \quad a_{\theta_y} \quad a_{\theta_z} \quad \dot{\theta}_2 \right]$$

donde

$e(\theta_1)$ error de posición para actuador en la pierna [deg]

$e(\dot{\theta_1})$ derivada del error para actuador en la pierna

$\int e(\theta_1)\,dt$ integral del error para actuador en la pierna

$\dot{\theta}_1$ velocidad angular en el actuador de la pierna $\left[\frac{\text{deg}}{s}\right]$

$e(\theta_2)$ error de posición para actuador en el péndulo [deg]

$e(\dot{\theta_2})$ derivada del error para actuador en el péndulo

$\int e(\theta_2)\,dt$ integral del error para actuador en el péndulo

$\dot{\theta}_2$ velocidad angular en el actuador de la pierna $\left[\frac{\text{deg}}{s}\right]$

a_{θ_x} rotación angular del centro de masa respecto al eje x [deg]

a_{θ_y} rotación angular del centro de masa respecto al eje y [deg]

a_{θ_z} rotación angular del centro de masa respecto al eje z [deg]

La observación 1 y la observación 2 en conjunto, permiten al algoritmo interpretar las consecuencias de las acciones de forma independiente y en conjunto respecto al objetivo de control para el sistema, la observación general se define con el vector de la siguiente forma:

$$Observador = \left[\begin{array}{cc} Observador\_1 & Observador\_2 \end{array} \right]$$

5.3.2. Acciones

El agente genera de señales de voltaje para las articulaciones revolutas de la pierna izquierda y el péndulo transversal del bípedo. Los límites globales de voltaje de cada articulación son de +/- 12 V.

5.3.3. Condiciones de paro de emergencia

Las condiciones de paro de emergencia, interrupciones de software y tienen como objetivo salvaguardar al robot bípedo durante todo el entrenamiento. Estas condiciones una vez que son activadas, dentro del microcontrolador, se ejecutarán ya que tienen la máxima prioridad en las tareas.

Las condiciones de paro programadas son las siguientes:

1. *STOP1.-* Paro de emergencia que es activado por sobrepasar el limite angular permitido en la acción de la pierna del robot bípedo. El rango permitido es $0[deg] \leq \theta_f \leq 30[deg]$. Si sobrepasa dicho límite, el robot bípedo tendrá una abertura de paso muy grande que desplazará al centro de masa fuera de la región del polígono de estabilidad.

2. *STOP2.-* Paro de emergencia que es activado por sobrepasar el limite angular permitido en la acción del péndulo transversal del robot bípedo. El rango permitido es $60[deg] \leq \theta_p \leq 130[deg]$. Si sobrepasa dicho límite, el robot caerá ya que el centro de masa se desplazó fuera de la región del polígono de estabilidad.

3. *STOP3.-* Este paro tiende dos maneras de activarse; la primera será cuando el acelerómetro, a través de sus mediciones de ángulo de giro en sus 'x' y 'y', detecta un cambio abrupto en la medición mayor de 30[deg]. Este cambio indicará si el robot bípedo a caído o estuvo cerca de caerse. La segunda forma de activación, será activado por el usuario por medio de un control RF.

Si alguno de estas condiciones de paro de emergencia son activados, corta la transferencia bilateral de datos, dando aviso al agente que ha finalizado el episodio.

5.4. Función de recompensa

Del capitulo 2, se comentó que la función de recompensa es la ecuación que evalúa el desempeño del agente de acuerdo a las acciones propuestas por el agente en el sistema. Reinforcement Learning es un algoritmo de optimización ya que busca maximizar la función de recompensa; cuando más cercano a su máximo global este dicha función, se puede asegurar que cumple el objetivo de control por el que se ha diseñado.

Los agentes, al tener estructuras similares en el vector de observación y de acción, se propone una estructura general, dicha función se observa en la ecuación 5.1

$$r_{t_k} = f\left(\begin{array}{ccccccccc} e(\theta_k)^2 & e(\dot{\theta}_k)^2 & \int e(\theta_k)^2 dT_s & V_{k-1} & V_{k-2} & \dot{\theta}_k^2 & Stop1 & Stop2 & Stop3 \end{array} \right) \quad (5.1)$$

para $k = 1, 2$, donde

$e(\theta_k)^2$ error cuadratico medio de posición angular en el actuador k

$e(\dot{\theta}_k)^2$ derivada del error cuadratico medio de posición angular en el actuador k

$\int e(\theta_k)^2 dT_s$ integral del error cuadrático medio de posición angular en el actuador k

V_{k-1} acción de voltaje con un paso de tiempo anterior

V_{k-2} acción de voltaje con dos pasos de tiempo anteriores

$\dot{\theta}_k^2$ velocidad angular cuadrática del actuador k

$Stop1$ Paro de emergencia referente al actuador de la pierna del bípedo

$Stop2$ Paro de emergencia referente al actuador del péndulo transversal

$Stop3$ Paro de emergencia referente al desplazamiento del centro de masa fuera del polígono de estabilidad o paro de emergencia activada por el usuario.

En Reinforcement Learning, es necesario cuantificar la función de recompensa en cada paso de tiempo. Para el diseño de la función, se propone de forma heurística coeficientes que multipliquen a las variables de interés en el robot bípedo. Los coeficientes se asignan de forma subjetiva, de acuerdo a su importancia en objetivo de control. También, se eligen restricciones de penalización que son deseables que no se activen en el ciclo de marcha, si se activan no detiene el episodio.

Las función de recompensa y sus restricciones de penalización implementadas en el robot bípedo están dadas por las siguientes expresiones

$$\begin{aligned} r_{t_k} &= r_{t_{k1}} + r_{t_{k2}} + r_{t_{k3}+25*\frac{Ts}{Tf}} \\ r_{t_{k1}} &= -\left(a * e(\theta_k)^2 + b * \left(e(\dot{\theta}_k)^2 + \dot{\theta}_k^2\right) + c * V_k + d * \int e(\theta_k)^2 dT_s\right) \\ r_{t_{k2}} &= -(e * Stop1 + f * Stop2 + g * Stop3) \\ r_{t_{k3}} &= -(h * restriccion\_1 + i * restriccion\_2) \\ V_k &= (V_{k-1} - V_{k-2})^2 \end{aligned}$$

para $k = 1, 2$ y $a, b, c, d, e, f, g, h, i \in R^+$. Donde:

$r_{t_{k1}}$ término que evalúa el desempeño del caminado y las acciones propuestas del agente.

$r_{t_{k2}}$ término que evalúa la activación de paros de emergencia

$r_{t_{k3}}$ término que evalúa la activación de las restricciones de penalización

Con las siguientes restricciones de penalización:

$$\begin{aligned} e(\theta_k)^2 &\leq 1.2[\text{deg}] \\ V_k &\leq 0.5[V] \end{aligned}$$

En la función de recompensa se premia con $25\frac{Ts}{Tf}$ para propiciar al agente computacional que durante el entrenamiento, evite la terminación anticipada. La terminación anticipada se presenta si se activa algún para de emergencia ocasionando que el episodio tenga una duración menor al tiempo final propuesto ($Tf = 10[s]$). Evitar la activación de los paros de emergencia, será la primera tarea que el agente resolverá con el aprendizaje que adquiera en la etapa de exploración de cada episodio. Está acción indica que el agente es capaz de proponer acciones que no activen las condiciones de paro, por ende, la tendencia del comportamiento de la función de recompensa será ascendente. Después de que el agente pueda estimar acciones que no activen las condiciones de paro, con el paso de episodios, el agente estimará aquellas acciones que no activen las

restricciones de penalización, logrando la tarea a realizar.

Las constantes de diseño elegidas para la función de recompensa en cada actuador se muestran en la tabla 5.1. Estas constantes son propuestas de forma heurística.

	Actuador pierna	**Actuador péndulo**
a	7	5.5
b	0.05	0.5
c	0.06	0.5
d	0.055	0.07
e	100000	7000
f	70000	10000
g	120000	120000
h	17.5	18
h	20	20

Tabla 5.1: Constantes de diseño para función de recompensa en cada actuador

5.5. Agentes y redes neuronales

5.5.1. Agente para actuador de Pierna

En la elección del tipo de agente, se consideraron las siguientes características:

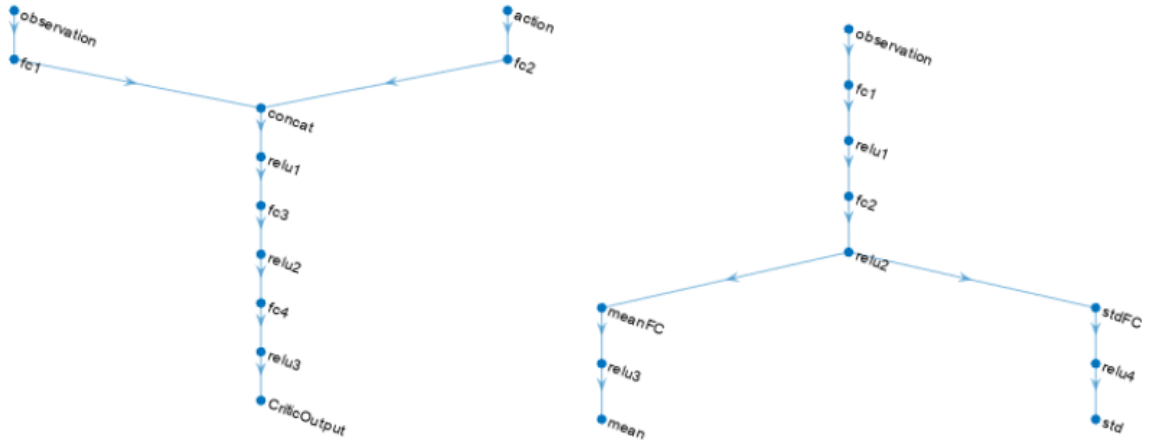
1. el rango de la posición angular es $-3[deg] \leq \theta_p \leq 11[deg]$

2. la velocidad angular no debe rebasar $\dot{\theta_p} = 2[rev/s]$.

De las características anteriores, al requerir un rango de posición angular pequeño y baja velocidad, se eligió el agente **SAC**. El agente computacional SAC, en la literatura, se utiliza para acciones suaves y precisas; es un agente que acepta redes neuronales profundas del tipo perceptrón. Si durante el entrenamiento, la posición angular del pie sobrepasa el valor de $11[rad/s]$ o la velocidad angular es $\dot{\theta_p} = 2[rev/s]$, se generará mayor error de posición respecto a la referencia del pie, provocando una caída.

La figura 5.2 muestra la arquitectura de la red neuronal profunda diseñada de forma heurística en la sección del crítico y del actor. En la tabla 5.2 se muestran los parámetros de la arquitectura de cada red neuronal. Del lado izquierdo se muestra el número de capas, número de neuronas y funciones de activación para la sección del crítico; del lado derecho se muestra los mismos parámetros para la sección del actor.

Los parámetros de la red neuronal como el número de capas, número de neuronas y funciones de activación para la red del actor y la red del crítico, se proponen de forma heurística.



(a) arquitectura de red neuronal crítico (b) arquitectura de red neuronal actor

Figura 5.2: Arquitectura red neuronal actor-crítico para agente SAC

Capa	Neuronas	F. Activación	Capa	Neuronas	F. Activación
1.1	128	Concatenación	*1*	128	Relu
1.2	128	Concatenación	*2*	64	Relu
2	2	Relu	*3.1*	32	Relu
3	64	Relu	*3.2*	1	Relu
4	32	Relu			

Tabla 5.2: Parámetros arquitectura de las redes neuronales para el agente SAC

5.5.2. Agente para actuador de péndulo

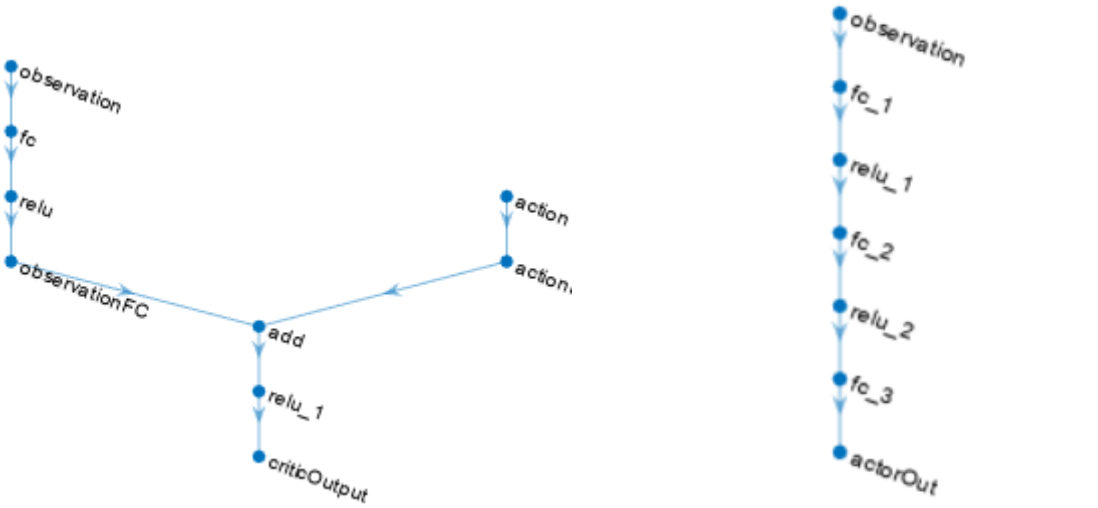
En la elección del tipo de agente, se consideraron las siguientes características:

1. el rango de la posición angular θ_p entre $70[deg] \leq \theta_p \leq 120[deg]$
2. péndulo con masa en el efector final de aproximadamente 200g, a una distancia del eje de giro.
3. la velocidad angular no debe rebasar $\dot{\theta_p} = 2[rev/s]$.

De las características anteriores, al requerir un rango de posición angular mayor al actuador de pierna, baja velocidad por la masa que se ubica a una distancia cercana del efector final, generando un momento por el brazo de palanca, se eligió un agente **DDPG.** El agente computacional DDPG , en la literatura, es el más utilizado en Deep Reinforcement Learning, se utiliza cuando se desean acciones más agresivas. Si durante el entrenamiento, la posición angular del péndulo sobrepasa el valor de $70[deg] \leq \theta_p \leq 120[deg]$ o la velocidad angular es $\dot{\theta_p} = 2[rev/s]$, se generará mayor desplazamiento del centro de masa y en algún instante puede salir del polígono de estabilidad provocando una caída.

La figura 5.3 muestra la arquitectura de la red neuronal profunda para el actor y para el crítico. La tabla 5.3 muestra los parámetros de la arquitectura de cada red neuronal. Del lado izquierdo se muestra el número de capas, número de neuronas y funciones de activación para la sección del

crítico; del lado derecho se muestra los mismos parámetros para la red neuronal para la sección del actor.



(a) arquitectura de red neuronal del crítico (b) arquitectura de red neuronal del actor

Figura 5.3: Redes neuronales actor y crítico para agente DDPG

Capa	Neuronas	F. Activación	Capa	Neuronas	F. Activación
1.1	250	Relu	*1*	220	Relu
1.2	300	Relu	*2*	150	Relu
2	250	Concatenación			
3	128	Relu			

Tabla 5.3: Parámetros arquitectura de las redes neuronales para el agente DDPG

Los parámetros de la red neuronal como el número de capas, número de neuronas y funciones de activación para la red del actor y la red del crítico, se proponen de forma heurística.

5.6. Entrenamiento y análisis de resultados

Para el entrenamiento del robot bípedo, se estableció como límite máximo 1000 episodios para estimar la dinámica y los agentes propongan acciones que permita realizar la tarea de caminado. Los parámetros de entrenamiento de la tabla , en conjunto con el vector de observación, funciones de recompensa y redes neuronales modificaron en cada tiempo de muestreo la estimación de la acción para cada agente. Los algoritmos SAC y DDPG se detallan en el apéndice C.

Párametro	**Valor**	**Descripción [34]**
Tamaño de buffer de Experiencia	$1e^6$	Durante el entrenamiento, el agente calcula actualizaciones adquiridos del buffer aleatoriamente.
Factor de descuento	0.99	Se aplica a recompensas futuras en el entrenamiento.
Factor de Suavización	$1e^{-3}$	Factor de cambio para la actualización de las redes neuronales crítico-actor.
Opciones de ruido	0.15	Ruido Ornstein-Uhlebeck que modifica la desviación estándar de las acciones que puede proponer el agente.
Actualización de Target	1	Número de pasos entre actualizaciones del actor y crítico de las redes neuronales.
Número de futuras recompensas	1	Número de futuras recompensas usadas en la estimación del valor de la política.
Algoritmo de optimización	ADAM	Estimación adaptativa que especifica tasas de decaimiento de la media del gradiente.
Rango de aprendizaje	0.01	Tasa de aprendizaje que durante el entrenamiento del actor y aproximador de la función crítica.
Umbral del gradiente	1	Límite de cuánto puede cambiar los parámetros de la red neuronal en cada tiempo de muestreo de cada episodio.
Tamaño de minibach	128	Se muestrean aleatoriamente experiencias del buffer y calcula los gradientes actualizando las propiedades críticas.
Tamaño de la puntuación promedio	250	Longitud del promedio de la función de recompensa y número de pasos para cada agente

Tabla 5.4: Parámetros de configuración de entrenamiento de los agentes DDPG y SAC

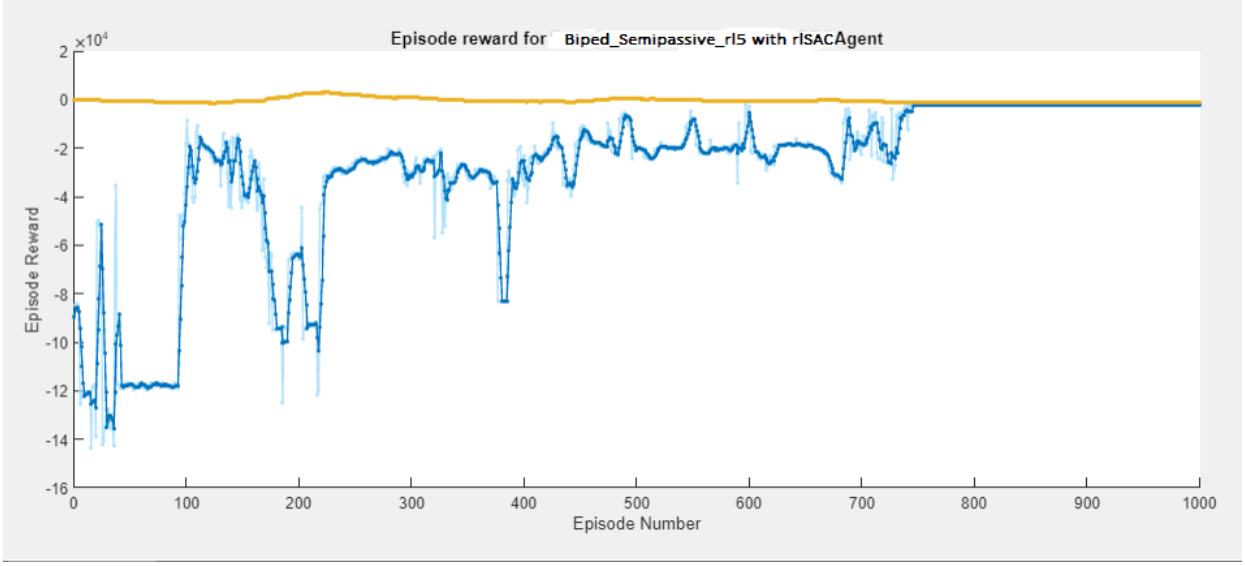


Figura 5.4: Gráfica de evolución de la función de recompensa respecto al número de episodios en el actuador de la pierna

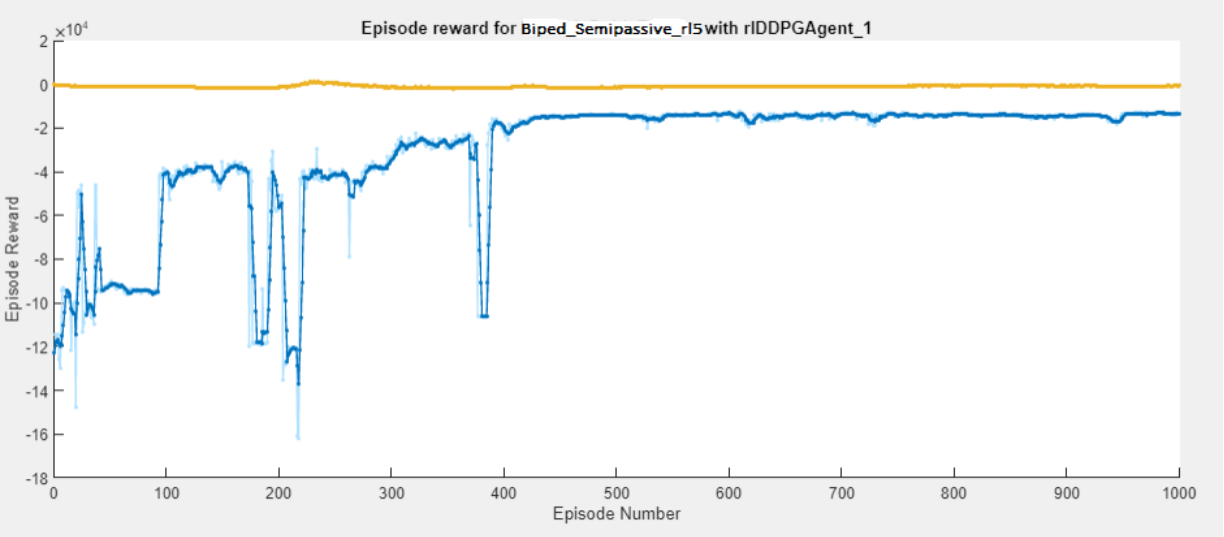


Figura 5.5: Gráfica de evolución de la función de recompensa respecto al número de episodios en el actuador del péndulo

Las figuras 5.4 y 5.5 muestran la evolución de la función de recompensa durante los 1000 episodios en el entrenamiento. La línea amarilla representa a la estimación de las recompensas futuras, la línea azul muestra los valores numéricos de la función de recompensa de cada episodio. Si durante el entrenamiento, la función de recompensa (línea azul) logra igualar o superar sus estimaciones futuras (línea amarilla), se concluye que el agente fue capaz de estimar la dinámica de su actuador y proponer acciones para cumplir la tarea de control. Para la gráfica 5.4, se observa que el agente, aproximadamente en el episodio 750, pudo estimar la dinámica del actuador de la pierna en el ciclo de caminado. Para la gráfica de 5.5, se observa que el agente DDPG del péndulo transversal, se acerca a la estimación de recompensas futuras en el episodio 400, pero no pudo estimar de forma completa la dinámica de actuador del péndulo, únicamente lo hizo de forma parcial. Para mejorar dicho resultado, se puede tomar las siguientes acciones:

1. En la función de recompensa, modificar la constante a, b y c de la Tabla 5.1
2. Para el entrenamiento, aumentar el número máximo de episodios.
3. Cambiar de tipo de agente.
   a) Usar el agente computacional TD3 (versión mejorada y más compleja del agente computacional DDPG).
   b) Usar el agente computacional PPO (agente computacional con actualizaciones de política más estables, pero requiere mayor número de episodios para lograr convergencia).
4. Modificar la estructura de las redes neuronales.

A continuación, se presentan las gráficas obtenidas de forma experimental durante el ciclo de caminado del robot bípedo en el episodio 1000 del entrenamiento con Deep Reinforcement Learning.

La figura 5.6 muestra los datos adquiridos del ángulo yaw del robot bípedo provenientes del sensor del acelerómetro MPU6050 respecto al tiempo de duración del episodio. La interpretación de las características de la gráfica obtenida se muestra a continuación:

1. Se interpreta que la función que describe el movimiento de rotación angular del centro de masa en el eje z, cada mínimo o máximo indica que se ha realizado una fase del ciclo de marcha.

2. Se observa que la frecuencia de la función varía ligeramente, esto indica que la velocidad de caminado fue variando durante el episodio.

3. Entre los 3[s] a 8[s], el ángulo de inclinación fue menor respecto al inicio o final del entrenamiento. Se interpreta que el ángulo requerido en el actuador de la cadera realicé su movimiento sin rozamiento entre el suelo y la planta del pie, debe ser mayor aproximadamente de $\mp 2[^{o}]$ según sea la fase del ciclo de caminado.

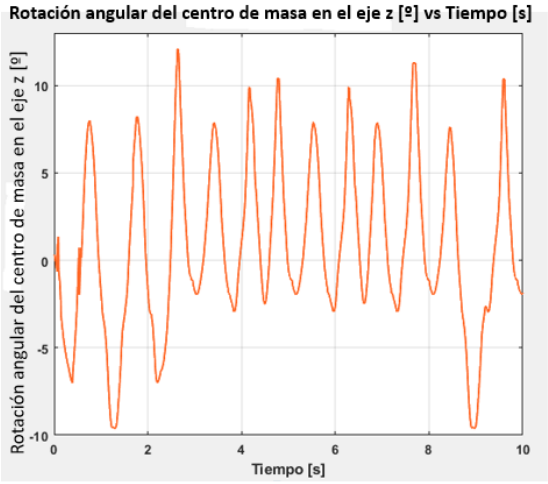
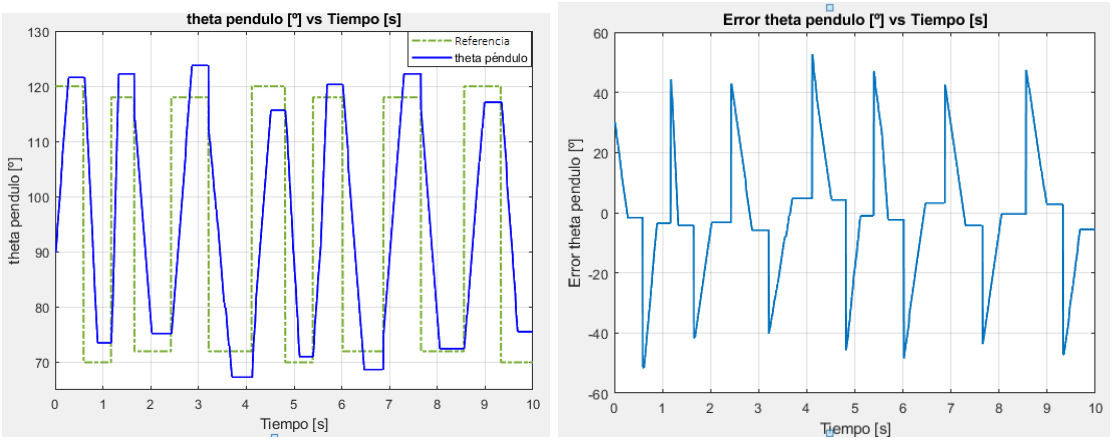


Figura 5.6: Rotación angular del centro de masa en el eje z del robot bípedo en función del tiempo

Del lado izquierdo de la figura 5.6, se muestra la gráfica que describe la referencia de posiciones finales respecto al tiempo de entrenamiento (10[s]), la referencia dependerá del estado que se encuentre la máquina de estados; se observa que el actuador no llegó en ningún momento a la referencia de manera precisa. Del lado derecho de la misma figura, se observa la gráfica del error de posición angular, al realizar movimiento, el error de posición angular $e\theta_p$ se encuentra aproximadamente en $\pm 5[^{o}]$. Esto se debe a la condición de transición de estado, esta se realizará cuando $e\theta_p < 6$ y $-3 \leq \theta_z \leq 4$, donde θ_z es el angulo de rotación del centro de masa medido por el acelerómetro en el eje z. Las constantes de las condiciones de cambio de transición fueron propuestas de forma heurística.



(a) Posiciones angulares de referencia de del péndulo respecto a la respuesta obtenida de θ_p

(b) Error de posición angular θ_p del péndulo respecto y su referencia

Figura 5.7: Error de posición angular θ_p del péndulo respecto a su referencia

La figura 5.8 muestra el voltaje medido en el actuador del péndulo. Se observa que el voltaje que el agente estima son funciones escalón de aproximadamente 6.3[V] y -6.3[V] de forma conmutada. Los pulsos de voltaje tienen diferente duración de tiempo, aproximadamente de $100[ms] \leq t \leq 350[ms]$. Está variación indica que él movimiento del péndulo transversal depende de la velocidad de giro y los errores de posición angular de la fase del ciclo de caminado anterior.

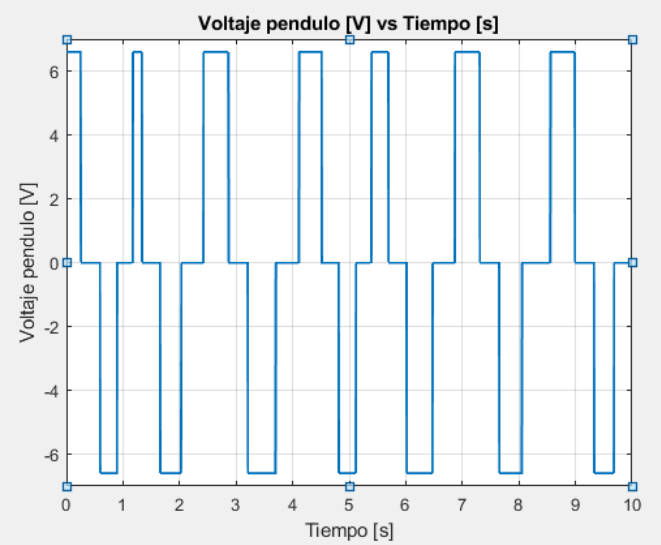
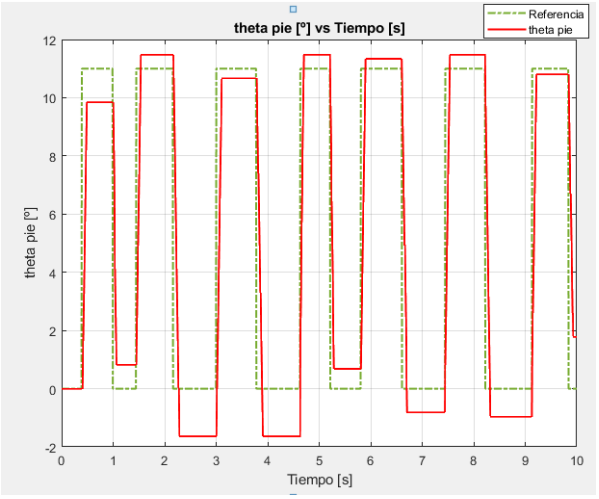
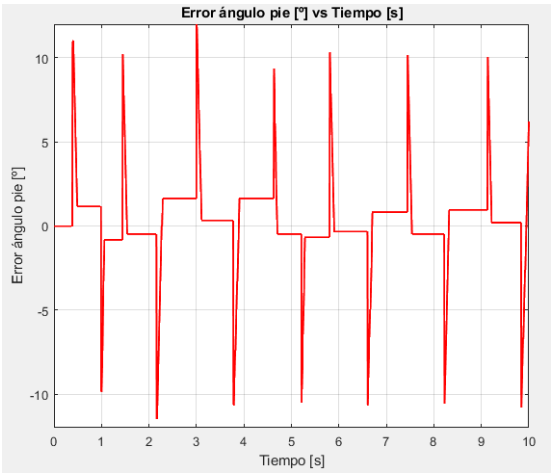


Figura 5.8: Voltaje en función del tiempo del actuador del péndulo

Del lado izquierdo de la figura 5.6, se muestra la gráfica que describe la referencia de posiciones finales respecto al tiempo de entrenamiento, de igual forma que el actuador del péndulo, la referencia dependerá del estado que se encuentre la máquina de estados; se observa que el actuador no llegó en ningún momento a la referencia de manera precisa. Del lado derecho de la misma figura, se observa la gráfica del error de posición angular, al realizar movimiento, el error de posición angular $e\theta_f$ se encuentra aproximadamente en $\pm 2[^{o}]$. Las condiciones que permite la transición de estado será cuando se cumplan $e\theta_p < 2.5$ y $-3 \leq \theta_z \leq 4$, donde θ_z es el angulo de rotación del centro de masa proveniente del acelerómetro en el eje z. Las constantes de las condiciones de cambio de transición fueron propuestas de forma heurística.



(a) Posición angular de referencia de del péndulo respecto a la respuesta obtenida de θ_f



(b) Error de posición angular θ_f del péndulo respecto y su referencia

Figura 5.9: Error de posición angular θ_p del péndulo respecto a su referencia

La figura 5.8 muestra el voltaje medido en el actuador del péndulo. Se observa que el voltaje que el agente estima son funciones escalón de aproximadamente 8.5[V] y -8.5[V] de forma conmutada. Los pulsos de voltaje tienen diferente duración de tiempo, aproximadamente de $70[ms] \leq t \leq 100[ms]$. El tiempo que recepción de la señales de voltaje provenientes de la salida del agente computacional al actuador, y la integral de error de posición es menor, comparado con el actuador del péndulo.

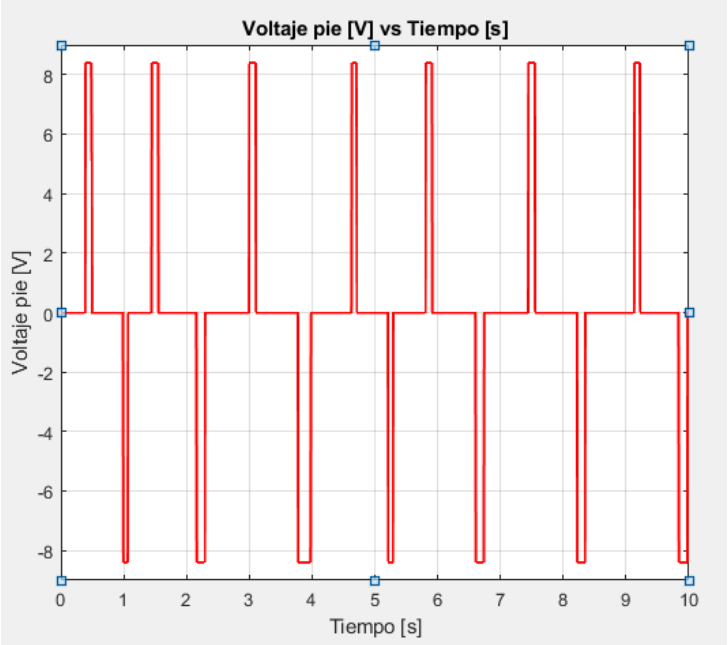


Figura 5.10: Voltaje en función del tiempo del actuador de la pierna

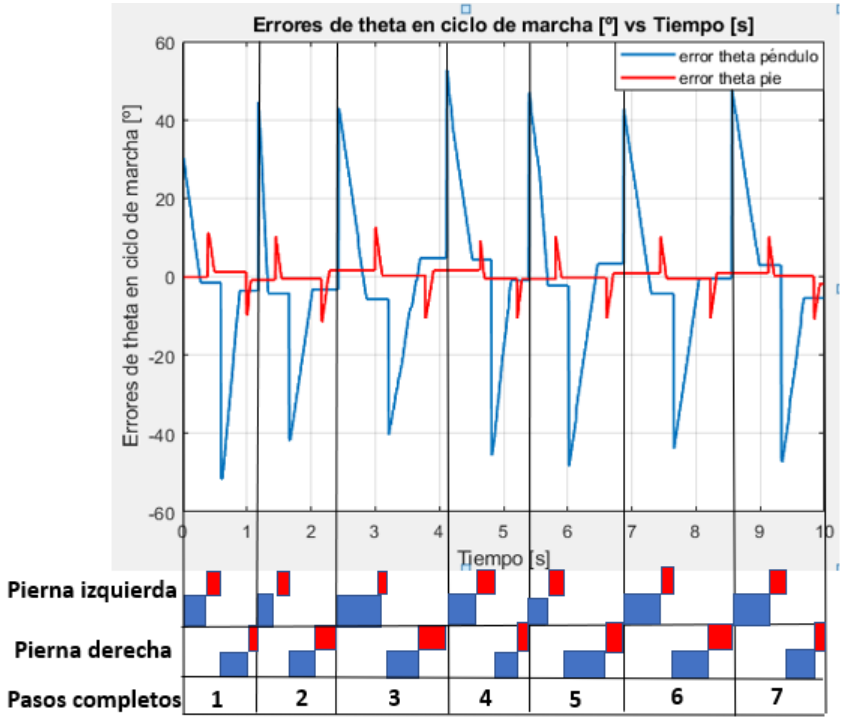


Figura 5.11: Representación del ciclo de caminado y conmutación en la gráfica de error de posición de cada actuador

La figura 5.11 muestra a detalle la conmutación de cada actuador en la gráfica de error de posición respecto al tiempo. En la parte inferior de la imagen, los cuadros azules representan el movimiento del péndulo transversal, además indica que pierna puede utilizarse sin que está tenga contacto con el suelo; los cuadros de color rojo representan la pierna que realizó el paso. El robot bípedo fue capaz de realizar 7 ciclos de marcha completos en 10[s].

El proceso de diseño del controlador para el robot bípedo e intentos fallidos, fue razón para elaborar una guía que pueda ser utilizada como base para el trabajo futuro o implementación en diferentes sistemas mecatrónicos.

5.7. Guía general para Deep Reinforcement Learning

La siguiente guía, tiene como objetivo proporcionar herramientas bases para el diseño de controladores por Deep Reinforcement, compartiendo la experiencia adquirida en las pruebas y soluciones fallidas.

Tomando la metodología descrita en la sección 5.1 de este trabajo de tesis, se hacen los siguientes comentarios:

1. **Anotaciones referente al entorno.**

   a) El número de agentes dependerá de los objetivos de control que se tienen en el sistema. Sin embargo, si las unidades de las variables de control tienen las mismas unidades, se pueden simplificar en un único agente. La reducción a un agente será indistinto del rango de valores que la acción pueda tomar.

   b) Elegir el tiempo de muestreo más pequeño que sea posible. Si el sistema tiene un tiempo de muestreo mayor a 20[ms][1], probablemente el sistema se vuelva no controlable y el enfoque de Reinforcement Learning no sea capaz de estimar la dinámica del sistema.

   c) Si el sistema cuenta con un sistema de comunicación bidireccional con retardos considerablemente grandes para cada cada tiempo de muestreo, el enfoque de Deep Reinforcement Learning no tendrá convergencia a un solución. Se recomienda cambiar el enfoque a Reinforcement Larning con otro algoritmo que permita al agente proponer acciones que compensen cualquier retardo.

2. **Anotaciones referente a los vectores de observación y de acción.**

   a) Dependiendo del sistema a controlar, el vector de observación puede definirse de las dos formas siguientes:

      1) considerar el error, derivada del error e integral del error de la variable que deseamos controlar, en su mayoría, son variables cinemáticas. Para el robot bípedo se siguió este enfoque.
      2) considerar variables cinemáticas (posición, velocidad y aceleraciones) que puedan obtenerse con sensores o con métodos de estimación. Pueden ser lineales o angulares. Si en el robot, se hubiera seguido este enfoque, el vector de observación hubiera quedado de la siguiente forma:

$$Observador\_k = \begin{bmatrix} \theta_k & \dot{\theta_k} & a_{\theta_x} & a_{\theta_y} & a_{\theta_z} \end{bmatrix} \text{ para } k = 1, 2$$

[1] valor probado en simulación como ejercicio de entendimiento de Reinforcement Learning, se deseaba resolver el problema de regulación de posición angular deseada en un péndulo doble

b) Las acciones pueden ser de dos formas:

1) *Acciones normalizadas.* El agente normaliza las acciones entre valores ±1 respecto a sus límites inferior y superior. Se recomienda utilizar para acciones de control con variables que no contengan unidades adimensionales. Al disminuir es espacio de acción, se podría reducir el tiempo de entrenamiento, sin embargo, el algoritmo de Reinforcement Learning tiene menor estabilidad.
2) *Acciones no normalizadas.* Funciona para variables con unidades adimensionales y no adimensionales. Requerirá mayor tiempo para lograr convergencia, pero el algoritmo de Reinforcement Learning será más estable.

3. **Anotaciones referente al tipo de controlador del sistema mecatrónico** Se identifican dos tipos de controladores de sistemas mecatrónicos.

a) Para sistemas mecatrónicos comerciales, donde tienen una caja negra y este no permita el control a bajo nivel, se recomienda utilizar un vector de observación como se describe en la sección 2.a.2 de está guía y acciones no normalizadas como se describe en la sección 2.b.2. Esta acción permite aumentar la rapidez de convergencia del sistema.

b) Para sistemas mecatrónicos, que se pueda acceder a bajo nivel al controlador controlador, se recomienda utilizar un vector de observación como se describe en la sección 2.a.1. El vector de acción que se describe en la sección 2.b queda a elección del diseñador.

4. **Anataciones referente a la función de recompensa**

En la literatura, no hay una estructura definida que ayude al diseño este parámetro e incluso, muchos de los documentos no se comparte, sin embargo, se propone lo siguiente:

a) La estructura general de la función de recompensa será la suma ponderada de los siguientes elementos:

$$r_{t_k} = -(r_{t_{k1}} + r_{t_{k2}} + r_{t_{k3}})$$

donde

$r_{t_{k1}}$ término que evalúa el desempeño del sistema.
$r_{t_{k2}}$ término que evalúa la activación de paros de emergencia
$r_{t_{k3}}$ término que evalúa la activación de las restricciones de penalización

El signo negativo tiene que ver con el objetivo del algoritmo de Reinforcement Learning. Recordar que el método busca a largo plazo maximizar la función de recompensa. Al proponerlo como negativo, se sabe que el máximo valor que podría obtener dicha función tenderá a 0. De está forma, es fácil interpretar la gráfica de entrenamiento (como la figura 5.4 del capítulo 5), ya que entre más cercano se encuentre el valor numérico de la función de recompensa en 0, es decir, se está logrando convergencia en el entrenamiento.

5. **Anotaciones referente al tipo de agentes a utilizar.**

La elección del agente dependerá del espacio del vector de observación, del espacio del vector de acción y la dificultad computacional. Las figuras 5.12, 5.13 y 5.14 muestran la clasificación comprendida de acuerdo a los parámetros ya mencionados. Si se desea mayor detalle de cada agente, ver las tablas 2.2, 2.7 y 2.8 del presente trabajo de tesis.

a) Espacio de observación discreta - Espacio de acción discreta.

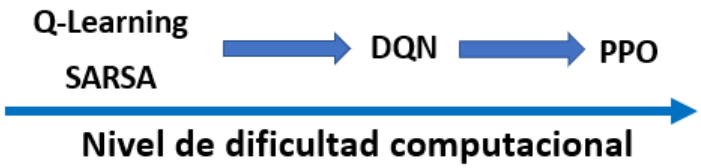


Figura 5.12: Nivel de dificultad computacional para agentes con observaciones discreta y acciones discretas

b) Espacio de observación continua - Espacio de acción discreta.

Figura 5.13: Nivel de dificultad computacional para agentes con observaciones continuas y acciones discretas

c) Espacio de observación continua - Espacio de acción continua.

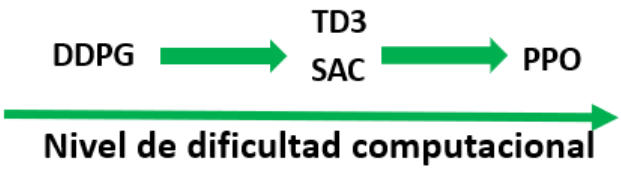


Figura 5.14: Nivel de dificultad computacional para agentes con observaciones continuas y acciones continuas

Comenzar con agentes de menor dificultad computacional como se muestra en las figuras 5.12, 5.13 y 5.14.

En el contexto de multiagentes, se recomienda usar agentes de baja dificultad en conjunto con los de mediana o alta dificultad. Suelen ser enfoques de mayor estabilidad y requiere menor número de recursos computacionales.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones Generales

Reinforcement Learning es un enfoque basado en datos del aprendizaje automático con ventajas importantes como el no depender de un modelo matemático que represente la dinámica del sistema, si bien, el enfoque aún tiene algunas desventajas como la aleatoriedad y heurística del método y la complejidad de diseño de algunos parámetros, poco a poco, con algoritmos evolutivos o diferentes técnicas de software se mejora la eficiencia en los siguientes rubros: minimizar la etapa de exploración que requiere el algoritmo, aumentar la rapidez de convergencia del método sin que requiera un alto número de episodios, solución de sistemas altamente no lineales y reducir la heurística del método.

Se seleccionó un sistema mecatrónico que se pudiera manufacturar e implementar con Reinforcement Learning los tres tópicos principales de control: regulación, seguimiento de trayectorias y planeación de ruta. En el trabajo de tesis que se desarrollo, se logró resolver la tarea de regulación para el ciclo de caminado semi-pasivo de un robot bípedo, con el enfoque de Reinforcement Learning, multiagentes computacionales estimaron aquellas acciones a bajo nivel que requieren realizar los actuadores, de acuerdo a la morfología del robot bípedo propuesta. El trabajo pone las bases para la solución de los tópicos de control restantes que se pueden retormar como trabajo futuro.

El robot bípedo con ciclo de caminado semi-pasivo construido, es un prototipo funcional de bajo costo comparado con los bípedos comerciales. El conocimiento que se adquiera a través de la experimentación que se realicé con él, puede extrapolarse a bípedos con mayor número de grados de libertad o a sistemas mecatrónicos de trabajo conmutado. La propuesta de la morforlogía del robot bípedo, permitió que el caminado semipasivo no dependiera del suelo inclinado para la generación del movimiento transitorio en el plano frontal. Al actuar el péndulo transversal, se pudo generar dicho movimiento transitorio permitiendo el ciclo de caminado. Este péndulo transversal, puede funcionar para modificar la dirección de caminado si se modifica el rango de actuación, frecuencia de conmutación y velocidad angular.

A pesar de que Reinforcement Learning permite estimar la dinámica del sistema sin necesidad de su modelo, conocer algunas representaciones sencillas, ayuda a proponer paros de emergencia o restricciones de penalización para las funciones de recompensa que evalúan el desempeño del sistema en cada tiempo de muestreo. Para el robot bípedo, conocer su modelo permitió tomar en cuenta aquellas variables que pudieran intervenir para la evaluación del desempeño por medio de la función de recompensa.

Reinforcement Learning es un enfoque sensible a cambios en los entrenamientos. En el robot

bípedo, se hicieron cambios en constantes de la función de recompensa, parámetros en las configuración del entrenamiento y redes neuronales y tipos de agente utilizados, todos estos cambios modificaron el aprendizaje. La guía propuesta ayuda a que los cambios sean en menor cantidad y permite al diseñador, tener una idea clara de que cambios realizar para lograr el objetivo del sistema.

En la literatura, se ha documentado pocos trabajos experimentales, en un alto porcentaje de investigación se realiza en simulación, esto es, porque en la etapa de exploración, existe un alto riesgo de que los sistemas puedan dañarse. La mayor parte de simulaciones se realizan con el lenguaje de programación de Python en conjunto con la librería creada de Gym, sin embargo a pesar de que facilita la implementación de software tiene la límitante de que no se simula con gemelos digitales, por lo que la aplicación y explotación de los enfoques que se proponen en muchos artículos suelen ser ideales y funcionales bajo ciertas condiciones. La gran ventaja de tener gemelos digitales es reducir la fatiga de los componentes y el aprovechamiento del tiempo, ya que de la forma experimental para el ciclo de caminado del bípedo, realizar 1000 episodios se llevó cerca de 31 horas de con pequeños lapsos de descanso, el proceso más tardado es colocar al sistema en la condición inicial y establezca comunicación.

6.2. Trabajo futuro

1. Cambiar el protocolo de comunicación de UART a una opción inalambrica como WIFI, TCP/IP o Bluetooth.

2. Resolver el ciclo de caminado incrementando el número de agentes computacionales, con la finalidad de que se pueda estimar el orden y condiciones de transición de la máquina de estados, además de estimar las referencias para cada fase de conmutación.

3. Resolver las tareas de control de seguimiento de trayectorias para el péndulo transversal, de esa forma, permitirá controlar la dirección de caminado del robot bípedo.

4. Una vez que se haya resuelto el seguimiento de trayectoria en el robot bípedo, resolver el tópico de planeación de ruta, de esta manera se podrá controlar la dirección de caminado semipasivo.

5. Obtener solución del ciclo de marcha con el enfoque basado en en datos de los operadores infinitesimales de Koopman y realizar una comparación [5].

6. Proponer arquitecturas de redes neuronales óptimas, donde se pueda definir el número de capas, número de neuronas y las funciones de activación para el actor y crítico.

7. Implementar el robot bípedo con Safe Reinforcement Learning y evaluar su desempeño y beneficio de forma experimental.

8. Evaluar el desempeño de entrenamientos existentes en la literatura donde se realiza sinergia entre Reinforcement Learning y algoritmos evolutivos para la reducción de episodios.

9. Desarrollar e implementar sistemas hápticos, de teleoperación o sistemas que se encuentren en la industria 4.0 o IoT, dichos desarrollos se resolverían en el contexto de Deep Reinforcement Learning.

Apéndice A

Apéndice A

A.1. Dimensiones Generales

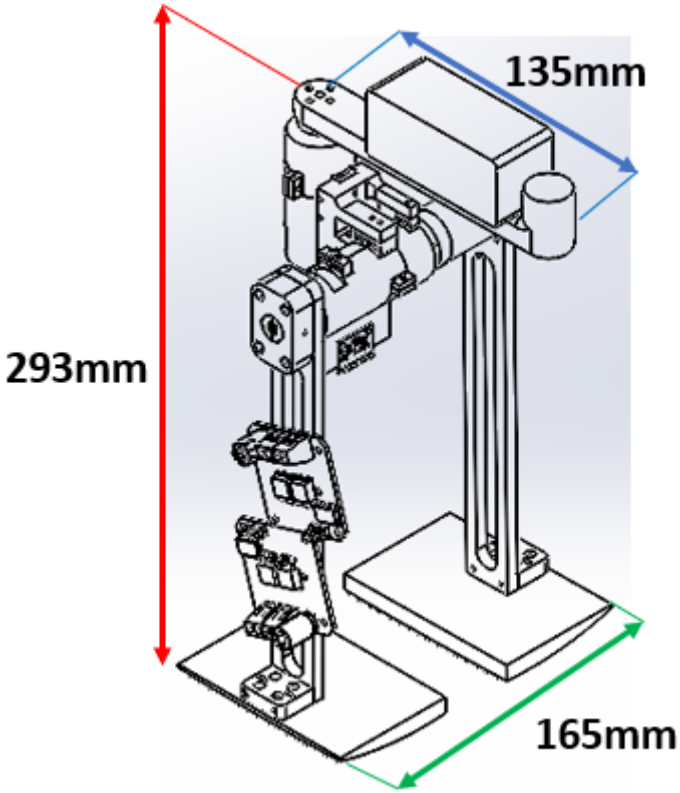


Figura A.1: Dimensiones Generales de robot bípedo

A.2. Diseño del pie

Para el robot bípedo con caminado semi-pasivo, se retomó el diseño de [3]. Para el diseño curvilíneo de la planta del pie, se ha utilizado la siguiente metodología.

Del plano frontal del bípedo, se tiene la siguiente figura:

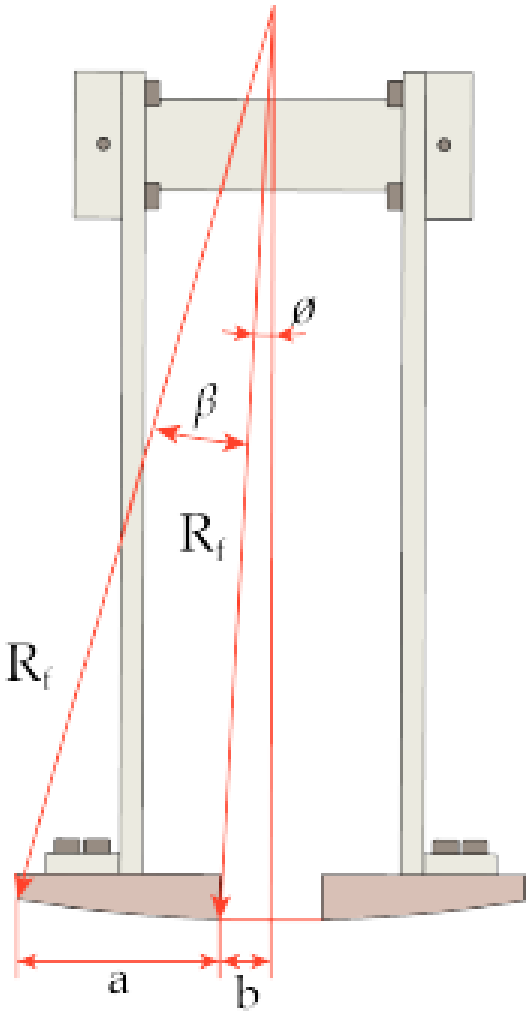


Figura A.2: Vista en el plano frontal [3]

a es el ancho del pie
b es la distancia de separación de los pies
R_f es el radio de curvatura del pie en el plano frontal
ϕ es el ángulo de abertura de las piernas en el plano frontal
β desplazamiento angular máximo

Del plano frontal, se obtienen los parámetros angulares β y ϕ

$$\begin{aligned} \phi &= \arcsin\left(\frac{b}{R_f}\right) \\ \beta &= \arcsin\left(\frac{a+b}{R_f}\right) - \phi \end{aligned}$$

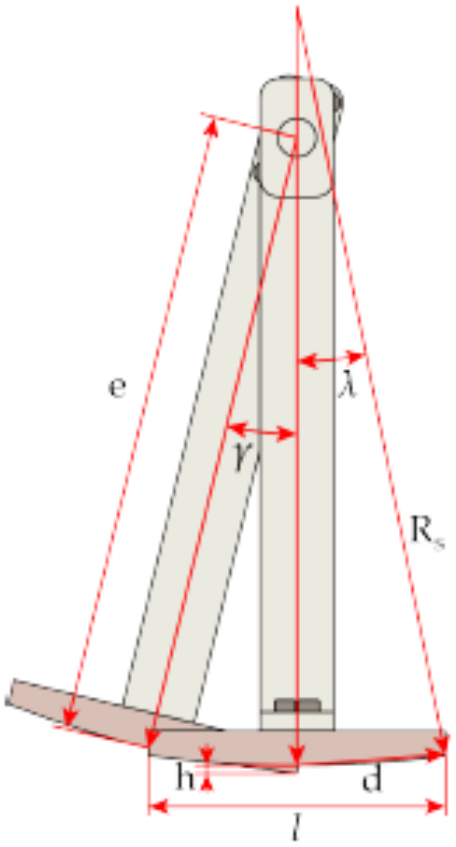


Figura A.3: Vista en el plano sagital [3]

De la figura A.3 se tiene el valor de que indica el centro de la planta del pie para calculo de la diferencia de alturas. Auxiliandonos de A.4 que es una vista de los triángulos formados en el plano sagital.

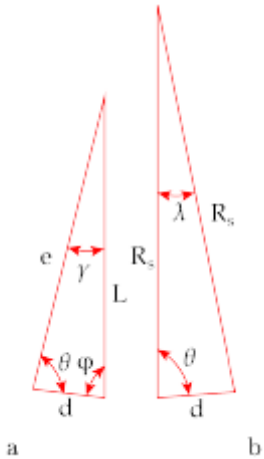


Figura A.4: Triángulos formados por los parámetros en el plano sagital [3]

donde

$$L = e + h$$

calculando los ángulos de los triángulos isósceles, se tienen las siguientes ecuaciones:

$$\begin{aligned} \lambda &= \arcsin\left(\frac{l}{2R_s}\right) \\ \theta &= \frac{(\pi-\lambda)}{2} \\ d &= \frac{\sin(\lambda)}{\sin(\theta)}R_s \end{aligned}$$

donde:

R_s es el radio de curvatura del pie en el plano sagital
l es el largo del pie

Utilizando la ley de senos y cosenos se calculan las incógnitas L y γ

$$\begin{aligned} L &= \sqrt{e^2+d^2-2de\cos(\theta)} \\ \gamma &= \arcsin\left(\frac{d\sin(\theta)}{L}\right) \end{aligned}$$

donde :

e la distancia del punto de giro de la cadera hasta el radio de curvatura en el plano
h diferencia de alturas entre las piernas

Para calcular la diferencia de alturas, se utiliza la siguiente expresión:

$$h-L-e$$

Conociendo el valor de h, el calculo necesario para que no exista rozamiento o contacto con el suelo, de la vista frontal se observa dos casos:

$$\begin{aligned} \phi &\geq \psi \\ \phi &\leq \psi \end{aligned}$$

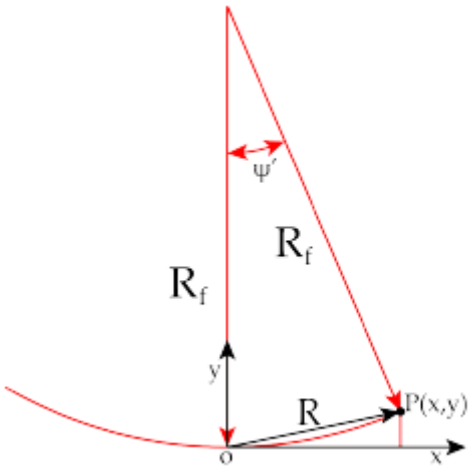


Figura A.5: Figura auxiliar para obtener el ángulo complementario [3]

De la figura A.5 se calculan todos los ángulos restantes:

$$\begin{aligned} r &= \sqrt{4b^2 + h^2} \\ y &= r\sin(\psi - \phi) \\ x &= \sqrt{R_f^2 - (y - R_f)^2} \\ R &= \sqrt{x^2 + y^2} \\ \psi' &= \arccos\left(1 - \frac{R^2}{2R_f^2}\right) \end{aligned}$$

una vez que se tienen los ángulos, se concluye que $\psi_T > \beta$, el robot bípedo no podría caminar porque no ha liberado o ha realizado el triangulo.

Los valores de la dimension de pie se muestran en la tabla A.2

	Nomenclatura	**Valor**
Largo de pierna	e	212 mm
Radio Frontal	R_f	420 mm
Radio Sagital	R_s	370 mm
Separación de los pies	b	4 mm
Ancho del pie	a	65 mm
Largo del pie	l	100 mm

Tabla A.1: Constantes propuestas para dimensiones del pie

los valores obtenidos son:

$$\begin{aligned} \phi &= 0.5456 \\ \beta &= 8.91 \\ \lambda &= 7.7664 \\ \gamma &= 13.4766 \end{aligned}$$

Apéndice B

Apéndice B

B.1. Modelo dinámico en el plano frontal

El modelo dinámico que representa el movimiento transitorio del bípedo se basó del trabajo de [2].

De la figura B.1 se consideran dos posibles casos en el movimiento transitorio que requiere el bípedo por sus plantas del pie curvas. Los detalles del cálculo basado en el trabajo de [2], se encuentra en el apéndice B.

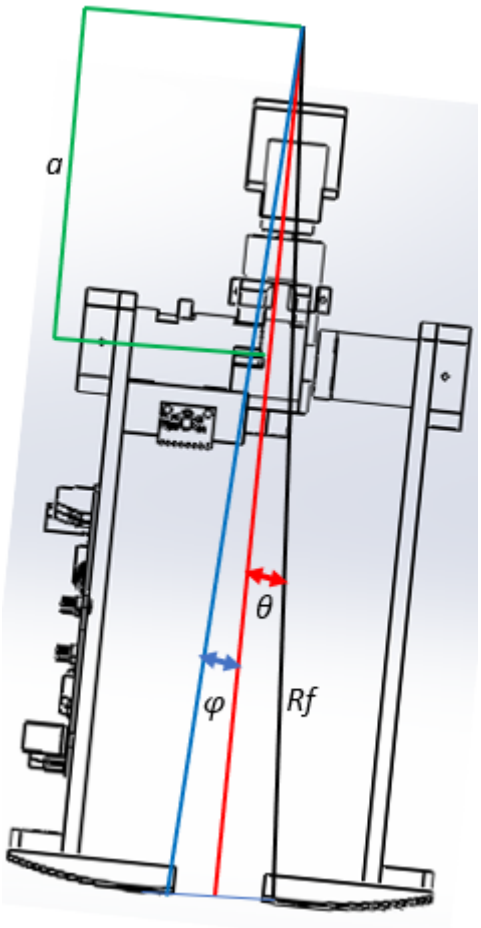


Figura B.1: Sistemas de coordenadas con la postura de referencia

donde en [3]:

a es la distancia del centro del radio al centro de masa
R_f Radio del pie en el plano frontal
θ Ángulo de giro
φ Ángulo que se forma de la línea media del robot y R_f

1. **Caso 1** $|\theta| > \phi$

El vector de posición de centro de masa es:

$$^0P_{c1} = \begin{bmatrix} R_f\theta - a\sin(\theta) \\ R_f - a\cos(\theta) \\ 0 \end{bmatrix}$$

velocidad del centro de masa es

$$^0V_{c1} = \begin{bmatrix} R_f\theta - a\sin(\theta) \\ R_f - a\cos(\theta) \\ 0 \end{bmatrix} \dot{\theta}$$

una vez que se tiene la velocidad del centro de masa, se obtiene la energía cinética por la ecuación siguiente

$$K_i = \frac{1}{2} m_i^0 V_{c1}^T {}^0V_{c1} + \frac{1}{2}{}^0\omega_i^T I^0\omega_i$$

donde

I es el tensor de inercia del sistema
$^0\omega_i$ es la velocidad angular medida desde el marco de referencia base

Considerando el plano frontal, se sabe que la velocidad es

$$^0\omega_i = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}$$

Desarrollando en termino de las velocidades lineales, se tiene

$$^0V_{c1}^T {}^0V_{c1} = \left[(R_f - a\cos\theta)^2 + (a\sin\theta)^2\right]\dot{\theta}^2 = \left[R_f^2 + a^2 - 2R_f a\cos(\theta)\right]\dot{\theta}^2$$

La energía potencial es

$$V = mgh = mg\left(R_f - a\cos(\theta)\right)$$

donde:

h es la altura de centro de masa a partir de un marco de referencia

El Lagrangiano es:

$$L = \frac{1}{2}\left[m\left[R_f^2 + a^2 - 2R_f a\cos(\theta)\right] + I\right]\dot{\theta}^2 - mg\left(R_f - a\cos(\theta)\right)$$

La ecuación dinámica general de Euler-Lagrange es

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}$$

obteniendo todos los términos, se tiene que la ecuación dinámica es:

$$\tau = \left[m\left(R_f^2 + a^2 - 2R_f a\cos(\theta)\right) + I\right]\ddot{\theta} + mR_f a\sin(\theta)\dot{\theta}^2 + mg\left(a\sin(\theta)\right)$$

obteniendo la representación matricial se tiene.

$$\begin{aligned} M(\theta) &= m\left(R_f^2 + a^2 - 2R_f a\cos(\theta)\right) + I \\ C\left(\theta,\dot{\theta}\right) &= mR_f a\sin(\theta)\dot{\theta} \\ G(\theta) &= mg\left(a\sin(\theta)\right) \end{aligned}$$

donde

m masa total del bípedo
I Tensor de inercia total del robot

2. **Caso 2** $|\theta| \leq \phi$

El vector de posición de centro de masa es:

$${}^0P_{c1} = \begin{bmatrix} -R_f\sin(\phi-\theta) - a\sin(\theta) \\ R_f\cos(\phi-\theta) - a\cos(\theta) \\ 0 \end{bmatrix}$$

velocidad del centro de masa es:

$${}^0V_{c1} = \begin{bmatrix} R_f(\phi-\theta) - a\cos(\theta) \\ R_f\sin(\phi-\theta) - a\sin(\theta) \\ 0 \end{bmatrix}\dot{\theta}$$

Desarrollando en termino de las velocidades lineales, se tiene:

$$\begin{aligned} {}^0V_{c1}^T{}^0V_{c1} &= \left[\left(R_f\cos(\phi-\theta) - a\cos\theta\right)^2 + \left(R_f\sin(\phi-\theta) - a\sin\theta\right)^2\right]\dot{\theta}^2 \\ &= \left[R_f^2 + a^2 - 2R_f a\cos(\theta)\right]\dot{\theta}^2 \end{aligned}$$

La velocidad angular esta dada por

$${}^0\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}$$

La energía cinética es:

$$K = \frac{1}{2}\left[m\left(R_f^2 + a^2 - 2R_f a\cos(\theta) + I\right)\right]\dot{\theta}^2$$

La energía potencial es:

$$V = mgh = mg\left(R_f\cos(\phi-\theta) - a\cos\theta\right)$$

donde

h es la altura de centro de masa a partir de un marco de referencia

El Lagrangiano es:

$$L = \frac{1}{2}\left[m\left[R_f^2 + a^2 - 2R_f a\cos(\theta)\right] + I\right]\dot{\theta}^2 - mg\left(R_f\cos(\phi-\theta) - a\cos\theta\right)$$

La ecuación dinámica general de Euler-Lagrange es

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}$$

obteniendo todos los términos se tiene que la ecuación dinámica es

$$\tau = \left[m\left(R_f^2 + a^2 - 2R_f a\cos\left(\theta\right)\right) + I\right]\ddot{\theta} - mg\left(\left(R_f \sin\left(\phi - \theta\right) - a\sin\theta\right)\right)$$

obteniendo la representación matricial se tiene. Se tiene que la ecuación de movimiento general de Lagrange

$$\begin{aligned} M\left(\theta\right) &= m\left(R_f^2 + a^2 - 2R_f a\cos\left(\varphi\right)\right) + I \\ C\left(\theta, \dot{\theta}\right) &= 0 \\ G\left(\theta\right) &= mg\left(R_f \sin\left(\theta - \varphi\right) - a\sin\left(\theta\right)\right) \end{aligned}$$

Se considera que el signo φ cambia $\theta < 0$, al tener dicho comportamiento se considera que los elementos de la ecuación de movimiento general de Lagrange son:

$$\begin{aligned} M\left(\theta\right) &= m\left(R_f^2 + a^2 - 2R_f a\cos\left(\varphi\right)\right) + I \\ C\left(\theta, \dot{\theta}\right) &= 0 \\ G\left(\theta\right) &= mg\left(R_f \sin\left(\theta - sign\left(\theta\right)\varphi\right) - a\sin\left(\theta\right)\right) \end{aligned}$$

La transferencia de velocidades en el momento de impacto al dar un paso se rige por la ecuación

$$\dot{\theta}^{+} = \dot{\theta}^{-}\cos\left(2\alpha\right)$$

donde 2α se muestra en la figura **??**

$$2\alpha = 2\tan^{-1}\left(\frac{R_f \sin\left(\varphi\right)}{Rf\cos\left(\varphi\right) - a}\right)$$

B.1.1. Modelo dinámico ciclo de caminado

Cálculo de velocidades del centro de masa respecto al tiempo

Derivando los vectores b_B^0, b_P^0, b_i^0 y b_d^0 con respecto al tiempo se tiene:

$$\begin{aligned} \dot{b}_B^0 &= v_B^0 + \omega_B^0 \times r_{GB}^0 \\ \dot{b}_P^0 &= v_B^0 + v_p^0 + \omega_P^0 \times r_{GP}^0 \\ \dot{b}_i^0 &= v_B^0 + v_i^0 + \omega_i^0 \times r_{Gi}^0 \\ \dot{b}_d^0 &= v_B^0 + v_d^0 + \omega_d^0 \times r_{Gd}^0 \end{aligned}$$

desarrollando cada vector de velocidad, se tiene lo siguiente:

Para $\dot{b}_B^0$

$$\begin{aligned} \dot{b}_B^0 &= v_B^0 + \omega_B^0 \times r_{GB}^0 \\ \dot{b}_B^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \left(\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0\right) \times r_{GB}^0 \\ \dot{b}_B^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \left(\dot{\theta}_B i_{\theta B}^0 \times r_{GB}^0\right) + \left(\dot{\phi}_B j_{\phi B}^0 \times r_{GB}^0\right) + \left(\dot{\psi}_B k_{\psi B}^0 \times r_{GB}^0\right) \\ \dot{b}_B^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta B}^B + \dot{\phi}_B e_{\phi B}^B + \dot{\psi}_B e_{\psi B}^B \end{aligned}$$

Para $\dot{b}_P^0$

$$\begin{aligned}
\dot{b}_P^0 &= v_B^0 + v_P^0 + \omega_P^0 \times r_{GP}^0 \\
\dot{b}_P^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \omega_B^0 \times r_P^0 + \omega_P^0 \times r_{GP}^0 \\
\dot{b}_P^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \left(\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0\right) \times r_P^0 + (\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0 + \\
&\quad + \dot{\theta}_P k_{0P}^0) \times r_{GP}^0 \\
\dot{b}_P^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta P}^B + \dot{\phi}_B e_{\phi P}^B + \dot{\psi}_B e_{\psi P}^B + \dot{\theta}_i e_{\theta P}^0
\end{aligned}$$

Para $\dot{b}_i^0$

$$\begin{aligned}
\dot{b}_i^0 &= v_B^0 + v_i^0 + \omega_i^0 \times r_{Gi}^0 \\
\dot{b}_i^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \omega_B^0 \times r_i^0 + \omega_i^0 \times r_{Gi}^0 \\
\dot{b}_i^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \left(\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0\right) \times r_i^0 + (\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0 + \\
&\quad + \dot{\theta}_i k_{0i}^0) \times r_{Gi}^0 \\
\dot{b}_i^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta i}^B + \dot{\phi}_B e_{\phi i}^B + \dot{\psi}_B e_{\psi i}^B + \dot{\theta}_i e_{\theta i}^0
\end{aligned}$$

Para $\dot{b}_d^0$

$$\begin{aligned}
\dot{b}_d^0 &= v_B^0 + v_d^0 + \omega_d^0 \times r_{Gd}^0 \\
\dot{b}_d^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \omega_B^0 \times r_d^0 + \omega_d^0 \times r_{Gd}^0 \\
\dot{b}_d^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \left(\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0\right) \times r_d^0 + (\dot{\theta}_B i_{\theta B}^0 + \dot{\phi}_B j_{\phi B}^0 + \dot{\psi}_B k_{\psi B}^0 + \\
&\quad \dot{\theta}_i k_{0i}^0) \times r_{Gd}^0 \\
\dot{b}_d^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta d}^B + \dot{\phi}_B e_{\phi d}^B + \dot{\psi}_B e_{\psi d}^B + \dot{\theta}_i e_{\theta d}^0
\end{aligned}$$

donde

$$\begin{array}{ll}
e_{\theta B}^B = i_{\theta B}^0 \times r_{GB}^0 & e_{\theta i}^B = i_{\theta B}^0 \times \left(r_i^0 + r_{Gi}^0\right) \\
e_{\phi B}^B = j_{\phi B}^0 \times r_{GB}^0 & e_{\phi i}^B = j_{\phi B}^0 \times \left(r_i^0 + r_{Gi}^0\right) \\
e_{\psi B}^B = k_{\psi B}^0 \times r_{GB}^0 & e_{\psi i}^B = k_{\psi B}^0 \times \left(r_i^0 + r_{Gi}^0\right) \\
e_{\theta P}^B = i_{\theta P}^0 \times \left(r_P^0 + r_{GP}^0\right) & e_{\theta d}^B = i_{\theta B}^0 \times \left(r_d^0 + r_{Gd}^0\right) \\
e_{\phi P}^B = j_{\phi P}^0 \times \left(r_P^0 + r_{GP}^0\right) & e_{\phi d}^B = j_{\phi B}^0 \times \left(r_d^0 + r_{Gd}^0\right) \\
e_{\psi P}^B = k_{\psi P}^0 \times \left(r_P^0 + r_{GP}^0\right) & e_{\psi d}^B = k_{\psi B}^0 \times \left(r_d^0 + r_{Gd}^0\right) \\
e_{\theta P}^0 = \left(k_{\theta P}^0 \times r_{GP}^0\right) & \\
e_{\theta d}^0 = \left(k_{\theta d}^0 \times r_{Gd}^0\right) & e_{\theta i}^0 = \left(k_{\theta i}^0 \times r_{Gd}^0\right)
\end{array}$$

Las velocidades de los centros de gravedad se reescriben como:

$$\begin{aligned}
\dot{b}_B^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta B}^B + \dot{\phi}_B e_{\phi B}^B + \dot{\psi}_B e_{\psi B}^B \\
\dot{b}_P^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta P}^B + \dot{\phi}_B e_{\phi P}^B + \dot{\psi}_B e_{\psi P}^B + \dot{\theta}_P e_{\theta P}^0 \\
\dot{b}_i^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta i}^B + \dot{\phi}_B e_{\phi i}^B + \dot{\psi}_B e_{\psi i}^B + \dot{\theta}_i e_{\theta i}^0 \\
\dot{b}_d^0 &= \dot{x_B} i_0 + \dot{y_B} j_0 + \dot{z_B} k_0 + \dot{\theta}_B e_{\theta d}^B + \dot{\phi}_B e_{\phi d}^B + \dot{\psi}_B e_{\psi d}^B + \dot{\theta}_d e_{\theta d}^0
\end{aligned}$$

Reuniendo las coordenadas generalizadas

$$q = \begin{bmatrix} x_B \\ y_B \\ z_B \\ \theta_B \\ \phi_B \\ \psi_B \\ \theta_P \\ \theta_i \\ \theta_d \end{bmatrix} \quad \dot{q} = \begin{bmatrix} \dot{x_B} \\ \dot{y_B} \\ \dot{z_B} \\ \dot{\theta_B} \\ \dot{\phi_B} \\ \dot{\psi_B} \\ \dot{\theta_P} \\ \dot{\theta_i} \\ \dot{\theta_d} \end{bmatrix}$$

Reescribiendo los vectores de velocidad de forma matricial, se tiene

$$\begin{aligned}
\dot{b}_B^0 &= \begin{bmatrix} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & 0 & 0 \end{bmatrix} \dot{q} = J_B \dot{q} \\
\dot{b}_P^0 &= \begin{bmatrix} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & e_{\theta P}^0 & 0 & 0 \end{bmatrix} \dot{q} = J_P \dot{q} \\
\dot{b}_i^0 &= \begin{bmatrix} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & e_{\theta i}^0 & 0 \end{bmatrix} \dot{q} = J_i \dot{q} \\
\dot{b}_d^0 &= \begin{bmatrix} i_0 & j_0 & k_0 & e_{\theta B}^B & e_{\phi B}^B & e_{\psi B}^B & 0 & 0 & e_{\theta d}^0 \end{bmatrix} \dot{q} = J_d \dot{q}
\end{aligned}$$

Términos de ecuación de Euler-Lagrange

Las ecuaciones de movimiento de Euler-Lagrange es

$$\frac{d}{dt}\left(\frac{\partial L\left(q,\dot{q}\right)}{\partial \dot{q}}\right) - \frac{\partial L\left(q,\dot{q}\right)}{\partial q} = Q - \Gamma_j^k$$

Encontrando los términos del lado izquierdo de la ecuación de movimiento

$$\begin{aligned}
\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}} &= \frac{\partial}{\partial \dot{q}}\left(\frac{1}{2}\dot{q}^T N_B \dot{q} + m_B g^T b_B^0\right) + \left(\frac{1}{2}\dot{q}^T N \dot{q} + m_P g^T b_P^0\right) \\
&\quad + \left(\frac{1}{2}\dot{q}^T N_i \dot{q} + m_i g^T b_i^0\right) + \left(\frac{1}{2}\dot{q}^T N_d \dot{q} + m_B g^T b_d^0\right) \\
\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}} &= \frac{1}{2}\frac{\partial}{\partial \dot{q}}\left(\dot{q}^T N_B \dot{q} + \dot{q}^T N_P \dot{q} + \dot{q}^T N_i \dot{q} + \dot{q}^T N_d \dot{q}\right) \\
\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}} &= \frac{1}{2}(\frac{\partial \dot{q}^T}{\partial \dot{q}} N_B \dot{q} + \dot{q}^T N_B \frac{\partial \dot{q}}{\partial \dot{q}} + \frac{\partial \dot{q}^T}{\partial \dot{q}} N_P \dot{q} + \dot{q}^T N_P \frac{\partial \dot{q}}{\partial \dot{q}} + \\
&\quad + \frac{\partial \dot{q}^T}{\partial \dot{q}} N_i \dot{q} + \dot{q}^T N_i \frac{\partial \dot{q}}{\partial \dot{q}} + \frac{\partial \dot{q}^T}{\partial \dot{q}} N_d \dot{q} + \dot{q}^T N_d \frac{\partial \dot{q}}{\partial \dot{q}})
\end{aligned}$$

Derivando con respecto al tiempo la expresión anterior se tiene:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}}\right) &= \frac{1}{2}\frac{d}{dt}(\frac{\partial \dot{q}^T}{\partial \dot{q}}N_B\dot{q}+\dot{q}^TN_B\frac{\partial \dot{q}}{\partial \dot{q}}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_P\dot{q}+\dot{q}^TN_P\frac{\partial \dot{q}}{\partial \dot{q}}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_i\dot{q}+\dot{q}^TN_i\frac{\partial \dot{q}}{\partial \dot{q}}+\\
&\quad +\frac{\partial \dot{q}^T}{\partial \dot{q}}N_d\dot{q}+\dot{q}^TN_d\frac{\partial \dot{q}}{\partial \dot{q}})\\
\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}}\right) &= \frac{1}{2}(\frac{d}{dt}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\right)N_B\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_B\dot{q}+N_B\ddot{q}\right)+\left(\ddot{q}^TN_B+\dot{q}^T\dot{N}_B\right)\frac{\partial \dot{q}}{\partial \dot{q}}+\\
&\quad \dot{q}^TN_B\frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial \dot{q}}\right))+\frac{1}{2}(\frac{d}{dt}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\right)N_P\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_P\dot{q}+N_P\ddot{q}\right)+\\
&\quad \left(\ddot{q}^TN_P+\dot{q}^T\dot{N}_P\right)\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^TN_P\frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial \dot{q}}\right))+\frac{1}{2}(\frac{d}{dt}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\right)N_i\dot{q}+\\
&\quad \frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_i\dot{q}+N_i\ddot{q}\right)+\left(\ddot{q}^TN_i+\dot{q}^T\dot{N}_i\right)\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^TN_i\frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial \dot{q}}\right))+\\
&\quad \frac{1}{2}(\frac{d}{dt}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\right)N_d\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_d\dot{q}+N_d\ddot{q}\right)+\left(\ddot{q}^TN_d+\dot{q}^T\dot{N}_d\right)\frac{\partial \dot{q}}{\partial \dot{q}}+\\
&\quad \dot{q}^TN_d\frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial \dot{q}}\right))
\end{aligned}$$

Si $\frac{d}{dt}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\right)=0^T$ y $\frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial \dot{q}}\right)=0$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}}\right) &= \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_B\dot{q}+N_B\ddot{q}\right)+\left(\ddot{q}^TN_B+\dot{q}^T\dot{N}_B\right)\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_P\dot{q}+N_P\ddot{q}\right)+\left(\ddot{q}^TN_P+\dot{q}^T\dot{N}_P\right)\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_i\dot{q}+N_i\ddot{q}\right)+\left(\ddot{q}^TN_i+\dot{q}^T\dot{N}_i\right)\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\left(\dot{N}_d\dot{q}+N_d\ddot{q}\right)+\left(\ddot{q}^TN_d+\dot{q}^T\dot{N}_d\right)\frac{\partial \dot{q}}{\partial \dot{q}}\right)\\
\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}}\right) &= \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_B\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_B\ddot{q}+\ddot{q}^TN_B\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^T\dot{N}_B\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_P\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_P\ddot{q}+\ddot{q}^TN_P\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^T\dot{N}_P\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_i\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_i\ddot{q}+\ddot{q}^TN_i\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^T\dot{N}_i\frac{\partial \dot{q}}{\partial \dot{q}}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_d\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_d\ddot{q}+\ddot{q}^TN_d\frac{\partial \dot{q}}{\partial \dot{q}}+\dot{q}^T\dot{N}_d\frac{\partial \dot{q}}{\partial \dot{q}}\right)
\end{aligned}$$

Utilizando la propiedad $(ABC)^T=C^TB^TA^T$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial \dot{q}}\right) &= \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_B\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_B\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_B^T\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_B^T\dot{q}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_P\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_P\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_P^T\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_P^T\dot{q}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_i\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_i\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_i^T\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_i^T\dot{q}\right)+\\
&\quad \frac{1}{2}\left(\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_d\dot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_d\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}N_d^T\ddot{q}+\frac{\partial \dot{q}^T}{\partial \dot{q}}\dot{N}_d^T\dot{q}\right)
\end{aligned}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial\dot{q}}\right) &= \frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(N_B+N_B^T\right)\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(\dot{N}_B+\dot{N}_B^T\right)\dot{q}\right)+\\ &\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(N_P+N_P^T\right)\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(\dot{N}_P+\dot{N}_P^T\right)\dot{q}\right)+\\ &\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(N_i+N_i^T\right)\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(\dot{N}_i+\dot{N}_i^T\right)\dot{q}\right)+\\ &\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(N_d+N_d^T\right)\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\left(\dot{N}_d+\dot{N}_d^T\right)\dot{q}\right)+\end{aligned}$$

Aprovechando la misma propiedad para la matriz de N_B

$$\begin{aligned}N_B^T &= \left(m_BJ_B^TJ_B+W_B^TI_B^0W_B\right)^T\\ N_B^T &= m_BJ_B^TJ_B+\left(W_B^TI_B^0W_B\right)^T=m_BJ_B^TJ_B+W_B^T\left(I_B^0\right)^TW_B\end{aligned}$$

donde

$$\left(I_B^0\right)^T=\left(R_B^0I_B^B\left(R_B^0\right)^T\right)^T=R_B^0\left(I_B^B\right)^T\left(R_B^0\right)^T$$

en tanto que las matrices de inercia en su base local son simétricas, por lo tanto se deduce:

$$N_B^T=m_BJ_B^TJ_B+W_B^T\left(I_B^0\right)^TW_B=N_B$$

de forma análoga, las matrices N_P, N_i y N_d son simétricas. Por lo tanto, la derivada con respecto al tiempo se reduce en la siguiente expresión:

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial\dot{q}}\right) &= \frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}2N_B\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}2\dot{N}_B\dot{q}\right)+\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}2N_P\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}2\dot{N}_P\dot{q}\right)+\\ &\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}2N_i\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}2\dot{N}_i\dot{q}\right)+\frac{1}{2}\left(\frac{\partial\dot{q}^T}{\partial\dot{q}}2N_d\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}2\dot{N}_d\dot{q}\right)\\ \frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial\dot{q}}\right) &= \frac{\partial\dot{q}^T}{\partial\dot{q}}N_B\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_B\dot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}N_P\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_P\dot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}N_i\ddot{q}+\\ &\frac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_i\dot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}N_d\ddot{q}+\frac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_d\dot{q}\\ \frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial\dot{q}}\right) &= D_B\ddot{q}+V_B\dot{q}+D_P\ddot{q}+V_P\dot{q}+D_i\ddot{q}+V_i\dot{q}+D_d\ddot{q}+V_d\dot{q}\\ \frac{d}{dt}\left(\frac{\partial L\left(\dot{q},q\right)}{\partial\dot{q}}\right) &= \left(D_B+D_P+D_i+D_d\right)\ddot{q}+\left(V_B+V_P+V_i+V_d\right)\dot{q}\end{aligned}$$

donde

$$\begin{aligned}D_B&=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}N_B \quad V_B=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_B\\ D_P&=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}N_P \quad V_P=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_P\\ D_i&=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}N_i \quad V_i=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_i\\ D_d&=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}N_d \quad V_d=\tfrac{\partial\dot{q}^T}{\partial\dot{q}}\dot{N}_d\end{aligned}$$

desarrollando el termino $\frac{\partial L}{\partial q}$ respecto a las coordenadas generalizadas, se obtiene la siguiente expresión:

$$\begin{aligned}
\frac{\partial L}{\partial q} &= \frac{\partial}{\partial q}(\left(\frac{1}{2}\dot{q}^T N_B \dot{q} + m_B g^T b_B^0\right) + \left(\frac{1}{2}\dot{q}^T N \dot{q} + m_P g^T b_P^0\right) + \\
&\quad + \left(\frac{1}{2}\dot{q}^T N_i \dot{q} + m_i g^T b_i^0\right) + \left(\frac{1}{2}\dot{q}^T N_d \dot{q} + m_B g^T b_d^0\right)) \\
&= \frac{1}{2}\dot{q}^T \frac{\partial N_B}{\partial q}\dot{q} + m_B g^T \frac{\partial b_B^0}{\partial q} + \frac{1}{2}\dot{q}^T \frac{\partial N_P}{\partial q}\dot{q} + m_P g^T \frac{\partial b_P^0}{\partial q} + \\
&\quad \frac{1}{2}\dot{q}^T \frac{\partial N_i}{\partial q}\dot{q} + m_i g^T \frac{\partial b_i^0}{\partial q} + \frac{1}{2}\dot{q}^T \frac{\partial N_d}{\partial q}\dot{q} + m_d g^T \frac{\partial b_d^0}{\partial q} \\
&= \acute{V}_B \dot{q} + \acute{C}_B + \acute{V}_P \dot{q} + \acute{C}_P + \acute{V}_i \dot{q} + \acute{C}_i + \acute{V}_d \dot{q} + \acute{C}_d \\
&= \left(\acute{V}_B + \acute{V}_P + \acute{V}_i + \acute{V}_d\right)\dot{q} + \left(\acute{C}_B + \acute{C}_P + \acute{C}_i + \acute{C}_d\right)
\end{aligned}$$

donde

$$\begin{aligned}
\acute{V}_B &= \tfrac{1}{2}\dot{q}^T \tfrac{\partial N_B}{\partial q} \quad \acute{C}_B = m_B g^T \tfrac{\partial b_B^0}{\partial q} \\
\acute{V}_P &= \tfrac{1}{2}\dot{q}^T \tfrac{\partial N_P}{\partial q} \quad \acute{C}_P = m_P g^T \tfrac{\partial b_P^0}{\partial q} \\
\acute{V}_i &= \tfrac{1}{2}\dot{q}^T \tfrac{\partial N_i}{\partial q} \quad \acute{C}_i = m_i g^T \tfrac{\partial b_i^0}{\partial q} \\
\acute{V}_d &= \tfrac{1}{2}\dot{q}^T \tfrac{\partial N_d}{\partial q} \quad \acute{C}_d = m_d g^T \tfrac{\partial b_d^0}{\partial q}
\end{aligned}$$

Con las expresiones anteriores se tiene que el termino de la ecuación de movimiento de Euler-Lagrange es

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = D_j \ddot{q} + V_j \dot{q} + C_j$$

donde

$$\begin{aligned}
D_j &= D_B + D_P + D_i + D_d \\
V_j &= V_B + V_P + V_i + V_d - \left(\acute{V}_B + \acute{V}_P + \acute{V}_i + \acute{V}_d\right) \\
C_j &= -\left(\acute{C}_B + \acute{C}_P + \acute{C}_i + \acute{C}_d\right)
\end{aligned}$$

Bibliografía

[1] H. D. M. Hagan and M. Beale, *Neuronal Network Desing.* PWS Publishing Company, 2002.

[2] M. F. R. Tedrake, T.W. Zhang and H. Seung, "Actuating a simple 3d passive dynamic walker." IEEE, 2004.

[3] R. G. Saldívar, *Robot bipedo semi-pasivo actuado con un mínimo de energía.* IPN, 2012.

[4] M-Rudra and V. Kumar, "Review of machine learning models for credit scoring analysis." Annamacharya Institute of Technology and Sciences, 2020.

[5] S. Brunton and J. Kutz, *Data Driven Science and Enginering. Machine Learning, Dynamical Systems and Control.* University Printing House, Cambridge CB2 8BS, United Kingdom: Cambridge University Press, 2019.

[6] X. W. Q. Zhang, D. Meng and W. Lu, "Learning to control space robots with flexible appendages using model-based polity search," in *International Conference on Robotics and Biomimetics.* IEEE, 2017.

[7] R. W. R. Ma, Y. Wang and S. Wang, "Omnidirectional drift control of an underwater biomimetic vehicle-manipulator system via reinforcement learning," in *10th Data-driven control and learning systems conference.* IEEE, 2021.

[8] Y. G. N. Wang and X. Zhang, "Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle," in *Transactions on neural networks and learning systems.* IEEE, 2021.

[9] J. P. J. M. R. B. A. Amini, I. Gilitschenski and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," in *Robotics and automation letters.* IEEE, 2020.

[10] K. Krishnachalitha and C. Priya, "Wireless sensor network-based hybrid instrusion detection system on feature extraction deep reinforcement learning and reinforcement learning techniques." Springer, 2020.

[11] J. P. R. Mitchell, J. Fletch and A. Prorok, "Multi-vehicle mixed-reality reinforcement learning autonomous multi-lane driving," 2020.

[12] I. K. D. Rastogi and J. Kober, "Sample-efficient reinforcement learning via diferrence model," 2018.

[13] P. Klose and R. Mester, "Benchmarketing model-free and model-based optimal control," in *Robotics and Automation Letters.* IEEE, 2017.

[14] C. S. K. C. Blad and S. Bogh, "Control of hvac-systems using reinforcement learning with hysteresis and tolerance control," in *International Symposium on System Integration Honolulu.* IEEE-SICE, 2020.

[15] N. C. X. Sun, M. Shi and J. Gu, "Integral sliding mode control for partially unknown t-s fuzzy sistems based on reonforcement learning method," in *11th Data Driven Control and Learning Systems Conference.* IEEE, 2022.

[16] H. V. R. B. I. Koryakovskiy, M. Kudruss and W. Caarls, "Model-plant mismatch compensation using reinforcement learnig," in *Robotics and Automation Letters.* IEEE, 2018.

[17] W. M. B. S. J. T. Y. Xie, X. Tang and S. Q. Zhang, "Iterative data-driven fractional model reference control of industrial robot for repetitive precise speed tracking," in *Transactions on mechatronics.* IEEE-ASME, 2019.

[18] R. V. R. Toro, T. Klassen and S. Mellraith, "Using reward machines for high-level task specification and descomposition in reinforcement learning," in *35th International Conference on Machine Learning in Stockholm, Sweden.* PMLR, 2018.

[19] F. Pagnozzi and M. Birattari, "Off-policy evaluation of the performance of a robot swarm: importance sampling to assess potential modifications to the finite-state machine that controls the robots," in *Frontier in Robotics and AI.* Brief Research Report, 2021.

[20] P. Klose and R. Mester, "Simulates autonomous driving in a realistic driving enviroment usin deep reinforcement learning and deterministic finite state machine," in *Transactions on Intelligent transportation systems.* IEEE, 2022.

[21] J. K.L. Seulbin Hwang and K. Dongsuk, "Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine," in *Transactions on Intelligent transportation systems.* IEEE, 2022.

[22] I. Park and M. Teng-Sheng, "Multi-agent deep reinforcement learning for walker systems," in *20th International Conference on Machine Learning and Applications (ICMLA).* IEEE, 2021.

[23] L. L. B. Chen, M. Xu and D. Zhao, "Delay-aware model-based reinforcement learning for continuous control," in *Neurocomputing 450.* Elsevier, 2021.

[24] Z. C. B. X. Z. Wang, W. Bai, "Multiple-pilot collaboration for advanced remote intervention using reinforcement learning," 2021.

[25] R. B. I. Koryakovskiy, H. Vallery and C. Wouter, "Evaluation of physical damage associated with action selection strategies in reinforcement learning," in *Paper Conference IFAC*, 2017.

[26] Y. R. M. L. Z. Zhou, O.S. Oguz and M. Buss, "Data generation method for learning a low-dimensional safe region in safe reinforcement learning," 2021.

[27] H. Huang and Y. Gong, "Constrastive learning: an alternative surrogate for offline data-driven evolucionary combination," in *Transactions on Evolucionary Computation.* IEEE-SICE, 2021.

[28] T. Blum and K. Yoshida, "Ppmc rl training algorith: Rough terrain intelligent robots through reinforcement learning," 2020.

[29] S. S. J. L. J. M. G. W. Y. T. T. E. Z. W. S. A. E. M. R. N. Heess, D. TB and D. Silver, "Emergence of locomotion behaviours in rich enviroments." DeepMind, 2017.

[30] I. Park and M. Teng-Sheng, "Data-driven multiobjetive controller optimization for a magnetically levitated nanopositioning system," in *Transactions on Mechatronics.* IEEE-ASME, 2020.

[31] Y. G. C. He, Y. Wan and F. Lewis, "Integral reiforcement learning based approximate minimum time-energy path planning in an unknown environment," in *Int J Robust Nonlinear Control*, 2021, pp. 1905–1922.

[32] D. C. S. J.T. Yu Yu, X.Z. Jiayi Huang and K. Wong, "Multi-objetive optimization for uav-assisted wireless powered iot networks based on extended ddpg algorith," in *Transactions on Communications*. IEEE-ASME, 2021.

[33] Z. W. Z. Li and Y. Fu, *Active/passive walking strategy for a biped robot using CPG with sensory interaction*. IEEE, 2014.

[34] *RL Toolbox User Guide*, Mathwotks.Inc, 1 Apple Hill Drive Natick, MA 01760-2098, 2019-2020.

[35] R. Sutton and A. Barto, *Reinforcement Learning. An introduction*. Cambridge, Massachusetts: The MIT Press, 2018.

[36] Z. D. H. Dong and S.Zhang, *Deep Reinforcement Learning Fundamentals, Research and Applications*. University Printing House, Cambridge CB2 8BS, United Kingdom: Springer, 2019.

[37] D. Anderson and G. McNeill, *Artificial Neuronal Network Technology*. Kaman Sciences Corporation, 2002.

[38] S. O. I. Farkas, P. Masaulli and S. Wermten, *Artificial Neural Networks and Machine Learning - ICANN 2021*. PWS Publishing Company, 2021.

[39] Y. Yu-He, *Deep Reinforcement Learning for wireless network*. Sringer, 2021.

[40] D. Marin, *Modelado dinámico en 3d y diseño de observador para un caminador bípedo pasivo*. Universidad de los Andes, Bogota, 2017.

[41] Y. Z. S. Iqbal, X. Zang and J. Zhao, *Bifurcations and chaos in passive dynamic walking: a review*. ELSEVIER, 2014.

[42] J. H. L. H.Susuki and S. Okamoto, *Development of semi-passive biped walking embedded with CPG-based locomotion control*. 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2017.

[43] T. Yumamoto and T. Suglhara, *Foot-guided control of a biped robot throught ZMP manipulation*. Advanced Robotics, 2020.

[44] J. G.-R. J.S. Pantoja-García, M.G. Villarreal-Cervantes and G. S. Cervantes, "Síntesis óptima de un mecanismo para la marcha bípeda utilizando evolución diferencial." Revista Internacional de Métodos Númericos para el Cálculo y Diseño en Ingeniería, 2017.

[45] M. T. T.Narukawa and K. Yoshida, "Biped locomotion on level fround by torso and swing-leg control based on passive-dynamic walking," 2022.

[46] A. D. C. Vasileiou, A. Smyrli and E. Papadopoulos, "Development of a passive biped robot digital twin using analisys experiments, and a multibody simulation environment." ELSEVIER, 2021.

[47] R. Ricci and S. Wilmert, *Sistemas Digitales: Principios y aplicaciones*, 2020.

[48] C. A. M. Dominguez, *Obtención del modelo dinámico inverso de un robot bípedo de 12 grados de libertad mediante el algoritmo recursovo de Euler-Newton*. IPN, 2017.

[49] S. Saha and J. Dutt, "Dynamics of three-type robotic systems." Springer, 2003.

[50] S. H. Mark W. Spong and M. Vidyasagar, *Robot Dinamics and Control*, 2006.

[51] O. Narvaez, *Modelo dinámico de un robot bípedo de 12 GDL internos.* UNAM, 2012.

[52] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms.* Springer, 2007.

[53] G. A. C. Chevallereau, G. Bessonet and Y. Aostin, *Bipedal Robots: Modeling, desing and walking synthesis.* John Wiley and Sons, 2009.