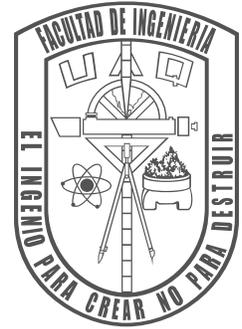


Universidad Autónoma de Querétaro

Facultad de Ingeniería



“Desarrollo de algoritmos para tareas cooperativas de robots manipuladores mediante observadores de fuerza”

TESIS

Que como parte de los requisitos para obtener el grado de:

Doctor en Mecatrónica

Presenta:

Ervin Galan Uribe

Dirigida por:

Dr. Luis Morales Velázquez

San Juan del Río, Querétaro, 5 de diciembre de 2023.



Dirección General de Bibliotecas y Servicios Digitales
de Información



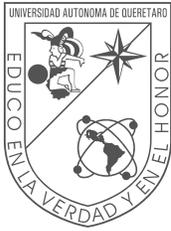
Desarrollo de algoritmos para tareas cooperativas de
robots manipuladores mediante observadores de
fuerza

por

Ervin Galán Uribe

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](#).

Clave RI: IGDCC-264094



Universidad Autónoma de Querétaro
Facultad de Ingeniería

“Desarrollo de algoritmos para tareas cooperativas de robots
manipuladores mediante observadores de fuerza”

TESIS

Que como parte de los requisitos para obtener el grado de:
Doctor en Mecatrónica

Presenta:

Ervin Galán Uribe

Asesor:

Dr. Luis Morales Velázquez

Dr. Luis Morales Velázquez

Presidente

Firma

Dr. Juan Primo Benítez Rangel

Secretario

Firma

Dr. Roque Alfredo Osornio Ríos

Vocal

Firma

Dr. Juan Pablo Amézquita Sánchez

Suplente

Firma

Dr. José de Jesús Rangel Magdaleno

Suplente

Firma

Dedicatoria

Para Ana con amor.

Agradecimientos

Quiero expresar mi sincero agradecimiento a mi madre y mis hermanos por su incondicional apoyo a lo largo de este camino.

A Ana, mi gratitud por su gran apoyo y cariño. Agradezco profundamente a mis profesores por su inmensa paciencia y por brindarme enseñanzas valiosas que han guiado mi trayectoria académica.

Al Dr. Roque Alfredo Osornio Ríos, mi gratitud por el constante apoyo y sabios consejos durante mi estancia en la universidad.

Al Dr. Pablo Amezcuita, agradezco su ayuda y su continuo entusiasmo para seguir trabajando.

Al Dr. Jose Magdaleno y al Dr. Juan Primo Benítez por compartir su valiosa experiencia y brindar recomendaciones clave que han enriquecido significativamente este proyecto.

Al Dr. Luis Morales Velazquez, mi más profundo agradecimiento por su apoyo desde la etapa de la maestría y aún más durante el doctorado; su confianza y amistad han sido fundamentales en mi desarrollo personal y académico. ¡Gracias!

Agradezco también a mis compañeros, Israel, Omar, Daniel, Jean, Darien, Salvador y Carlos Terrazas, por el tiempo compartido y la invaluable amistad.

A todos ustedes, mi más sincero agradecimiento por formar parte de este viaje y por contribuir de manera significativa a mi crecimiento personal y profesional. ¡Gracias!

RESUMEN

En este trabajo se abordan distintos factores cruciales en la interacción humano-robot (HRI): el espacio de trabajo del robot, su precisión en tareas y su interacción con el entorno mediante fuerzas de contacto. El espacio de trabajo del robot define su área física de operación e interacción con humanos, impactando significativamente en los sistemas HRI. Un espacio mal diseñado puede dificultar tareas del robot, limitando eficacia y aumentando riesgos de colisiones. Optimizar el espacio de trabajo es vital para una interacción segura y eficaz, mejorando accesibilidad, eficiencia y reduciendo colisiones. La precisión del robot es fundamental para HRI exitosa, especialmente en entornos cercanos a humanos. La confianza en la interacción depende de robots precisos, pero la precisión puede disminuir por factores ambientales o electromecánicos, requiriendo detección oportuna de degradación. La aplicación adecuada de fuerza por el robot al interactuar es crucial para la seguridad y eficacia en HRI. Medir fuerzas con sensores de fuerza montados en el brazo del robot es costoso. Alternativas como observadores de estado basados en algoritmos de inteligencia artificial como redes neuronales permiten estimar fuerzas basadas en el modelo dinámico del robot ofreciendo un enfoque de bajo costo. La HRI, un campo multidisciplinario, depende principalmente de la precisión y seguridad. El espacio de trabajo, la precisión del robot y la estimación de fuerza son factores esenciales y al explorar su relación, se mejora la eficacia, la eficiencia y la seguridad de la interacción humano-robot.

Palabras Clave: Robot, Degradación, Optimización, Aprendizaje Máquina.

ABSTRACT

This paper addresses several crucial factors in human-robot interaction (HRI): the robot's workspace, task accuracy, and interaction with the environment through contact forces. The robot's workspace defines its physical area of operation and interaction with humans, significantly impacting HRI systems. A poorly designed workspace can hinder robot tasks, limiting efficiency and increasing the risk of collisions. Optimizing the workspace is vital for safe and effective interaction, improving accessibility, efficiency and reducing collisions. Robot accuracy is critical for successful HRI, especially in environments near humans. Confidence in interaction depends on accurate robots, but accuracy can be diminished by environmental or electromechanical factors, requiring timely detection of degradation. Proper application of force by the robot when interacting is crucial for safety and efficiency in HRI. Measuring forces with force sensors mounted on the robot arm is costly. Alternatives such as state observers based on artificial intelligence algorithms like neural networks allow estimating forces based on the dynamic model of the robot offering a low-cost approach. HRI, a multidisciplinary field, depends primarily on accuracy and safety. Workspace, robot accuracy, and force estimation are essential factors and by exploring their relationship, the effectiveness, efficiency, and safety of human-robot interaction is improved.

Keywords: Robot, Degradation, Optimization, Machine Learning.

Índice General

Dedicatoria	I
Agradecimientos	II
RESUMEN	III
ABSTRACT	IV
Índice General	V
Índice de Tablas	VI
Índice de Figuras	VII
1. INTRODUCCIÓN	1
1.1. INTERACCIÓN HUMANO ROBOT	2
1.2. OPTIMIZACIÓN DEL ROBOT	5
1.3. DEGRADACIÓN DE LA POSICIÓN	7
1.4. ESTIMACIÓN DE FUERZAS CON REDES NEURONALES	8
1.5. DESCRIPCIÓN DEL PROBLEMA	9
1.6. JUSTIFICACIÓN	10
1.7. HIPÓTESIS Y OBJETIVOS	10
1.7.1. HIPÓTESIS	10
1.7.2. OBJETIVO GENERAL	10
1.7.3. OBJETIVOS ESPECÍFICOS	10
1.8. PLANTEAMIENTO GENERAL	11
2. FUNDAMENTACIÓN TEÓRICA	13
2.1. CINEMÁTICA DIRECTA	13
2.2. CINEMÁTICA INVERSA	14
2.3. CINEMÁTICA DIFERENCIAL	15
2.4. ÍNDICE DE CONDICIÓN GLOBAL	16
2.5. ÍNDICE DE LONGITUD ESTRUCTURAL	17
2.6. DINÁMICA DEL MANIPULADOR	17

2.6.1.	LAGRANGIANO	17
2.6.2.	MODELO DINÁMICO	18
2.7.	ALGORITMOS DE INTELIGENCIA COMPUTACIONAL	18
2.7.1.	INTELIGENCIA DE ENJAMBRE	19
2.7.2.	ALGORITMOS GENÉTICOS	19
2.7.3.	OPTIMIZADOR DEL LOBO GRIS	20
2.7.4.	OPTIMIZADOR DEL ÁGUILA HARRIS	22
2.7.5.	REDES NEURONALES ARTIFICIALES	23
2.7.6.	RED NEURONAL DE MEMORIA LARGA A CORTO PLAZO	24
2.8.	TRANSFORMADA WAVELET DISCRETA	25
2.9.	ANÁLISIS DE COMPONENTES PRINCIPALES	27
2.10.	INDICADORES NO LINEALES	28
2.10.1.	DIMENSIÓN DE FRACTALIDAD DE KATZ	28
2.10.2.	ÍNDICE DE ENERGÍA	29
3.	METODOLOGÍA	30
3.1.	OPTIMIZACIÓN DEL ROBOT	31
3.2.	DEGRADACION POSICIONAL	35
3.2.1.	DESCRIPCIÓN DE LA BASE DE DATOS	36
3.2.2.	PUESTA EXPERIMENTAL	36
3.3.	ESTIMACIÓN DE FUERZA	38
3.3.1.	DESCRIPCIÓN DE LA BASE DE DATOS	38
3.3.2.	IMPLEMENTACIÓN DE RED NEURONAL LSTM	39
4.	RESULTADOS Y DISCUSIÓN	44
4.1.	RESULTADOS DE LA OPTIMIZACIÓN CINEMÁTICA DEL ROBOT	44
4.2.	RESULTADOS DE LA DEGRADACIÓN POSICIONAL DEL ROBOT	50
4.3.	RESULTADOS DE LA ESTIMACIÓN DE FUERZA CON REDES NEURONALES	53
5.	CONCLUSIONES	56
5.1.	OPTIMIZACIÓN DEL ROBOT	56
5.2.	DEGRADACIÓN POSICIONAL DEL ROBOT	57
5.3.	ESTIMACIÓN DE FUERZAS CON REDES LSTM	58
5.4.	CONCLUSIONES GENERALES	59
	Referencias	60
	Apéndices	65
	Publicaciones	65

Índice de Tablas

2.1. Parámetros de Denavit-Hartenberg para un brazo robot de 6 grados de libertad.	14
3.1. RNA configuration parameters.	38
3.2. Parámetros de configuración de las capas de la red LSTM.	40
3.3. Parámetros de configuración de la red LSTM.	40
4.1. Resultados de la optimización para el SLI.	45
4.2. Recursos computacionales y rendimiento de los algoritmos de optimización para SLI.	46
4.3. Resultados de la optimización de la GCI.	47
4.4. Recursos informáticos y rendimiento de los algoritmos de optimización para la GCI.	48
4.5. Precisión de los clasificadores para los casos de velocidad media utilizando muestras de nivel de ruido combinado para la validación.	53
4.6. Precisión de los clasificadores para los casos de velocidad máxima utilizando muestras de nivel de ruido combinado para la validación.	53
4.7. Error RMS obtenido entre señales de torque de referencia y las estimadas por la red LSTM.	53

Índice de Figuras

1.1.	a) Robot guiado manualmente por el usuario (Universal Robots UR5), b) Brazo robot con plataforma móvil Kuka Mobile Robotics (KMR) Quantec.	1
1.2.	(a) Interacción humano-robot (KUKA LBR iiwa).(b) Interacción robot-robot, sistema multi-robot.	2
1.3.	Retos principales de HRI.	3
1.4.	(a) Espacios de trabajo en tareas cooperativas.	5
1.5.	Diagrama del planteamiento general de la tesis.	11
2.1.	Representación gráfica de los operadores de cruza y mutación en una red neuronal.	20
2.2.	Representación gráfica una manada de lobos rodeando a su presa.	21
2.3.	Representación gráfica de un halcón acechando y cazando a su presa.	23
2.4.	Diagrama general de una red neuronal artificial.	24
2.5.	Diagrama general de una red neuronal artificial.	25
2.6.	Diagrama de bloques de (a) Proceso de descomposición DWT de una señal $x[n]$. (b) Diagrama de reconstrucción DWT inversa de una señal $x'[n]$.	27
3.1.	Diagrama de las etapas de la metodología.	30
3.2.	a) Robot industrial Yaskawa Motoman, b) diagrama de alambre del robot, c) representación gráficas de los parámetros de DH.	31
3.3.	Metodología propuesta para la optimización del robot.	32
3.4.	Metodología propuesta para detección de degradación posicional del robot.	35
3.5.	Robot UR5 y trayectoria a seguir por el robot.	36
3.6.	Trayectoria propuesta para la estimación de torques.	39
3.7.	Señales simuladas de la articulación uno.	40
3.8.	Señales simuladas de la articulación dos.	41
3.9.	Señales simuladas de la articulación tres.	41
3.10.	Señales simuladas de la articulación cuatro.	42
3.11.	Señales simuladas de la articulación cinco.	42
3.12.	Señales simuladas de la articulación seis.	43
4.1.	Curvas de convergencia de los algoritmos de optimización del índice SLI.	45

4.2. Volumen del espacio de trabajo del brazo robótico optimizado mediante HHO y SLI.	46
4.3. Curvas de convergencia de los algoritmos de optimización del índice GCI.	47
4.4. Resultado de la prueba de 25 ejecuciones de los algoritmos de optimización para el índice SLI y GCI.	50
4.5. Señales actuales de la articulación uno con ruido de 5 dB: (a) señal para el estado de arranque en frío del robot; (b) señal para el estado de funcionamiento en caliente del robot.	51
4.6. Análisis tiempo frecuencia de la primera articulación del robot, con 5 dB ruido, para el caso de media velocidad: (a,b) aproximación y detalles para el nivel de descomposición seis para el CS señal; (c,d) aproximación y detalles para el nivel de descomposición seis para la señal HO.	52
4.7. Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones uno y dos.	54
4.8. Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones tres y cuatro.	55
4.9. Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones cinco y seis.	55

Capítulo 1

INTRODUCCIÓN

Acorde a la norma ISO 10218-1:2012 (*Robots for industrial environments Safety requirements Part1:Robot*) un robot industrial: es un manipulador controlado de forma automática, reprogramable y multifuncional, programable en tres o más ejes, puede ser móvil o estar fijo para uso en aplicaciones industriales automatizadas.

Además el término robot incluye: el manipulador con sus actuadores, el controlador incluyendo la unidad de programación y cualquier interfaz de comunicación tanto en hardware como software. Esta norma indica las medidas de operación requeridas para el uso de un robot industrial de forma segura para el operador.

Por otra parte, los robots colaborativos son definidos por la norma ISO 10218-2:2012 (*Robots and robotic devices Safety requirements for industrial robots Part 2: Robot systems and integration*) como: un robot diseñado para interactuar directamente con una persona dentro de un espacio de trabajo colaborativo, el cual es un espacio en el que, durante su funcionamiento el robot y el operario pueden desarrollar tareas de forma simultánea (Figura 1.2a).

El término robot colaborativo o cobot (collaborative robot) es un término que se refiere comúnmente a la interacción entre humanos y robots, sin embargo el término también es utilizado para hacer referencia a la interacción que existe entre un con-



Figura 1.1: a) Robot guiado manualmente por el usuario (Universal Robots UR5), b) Brazo robot con plataforma móvil Kuka Mobile Robotics (KMR) Quantec.



Figura 1.2: (a) Interacción humano-robot (KUKA LBR iiwa).(b) Interacción robot-robot, sistema multi-robot.

junto de robots (sistema multi-robots) que realizan una tarea de forma colaborativa (Figura 1.2b).

En este sentido la interacción entre el humano y el robot al realizar tareas cooperativas se ve influenciada por diferentes aspectos como: la seguridad, el entorno, el tipo de robot, el proceso a realizar, entre otros. La relación de los factores anteriormente mencionados y su influencia en las tareas humano-robot son abordadas en una línea de investigación denominada: Interacción Humano-Robot.

1.1. INTERACCIÓN HUMANO ROBOT

La interacción humano-robot, HRI del inglés "*Human-Robot Interaction*" es un campo multidisciplinario de rápido crecimiento que tiene como objetivo desarrollar robots que puedan interactuar con humanos en diversos entornos. Con la creciente demanda de automatización en industrias como la fabricación, la atención médica y la asistencia personal, el desarrollo de sistemas de HRI efectivos y eficientes es de suma importancia.

En tareas cooperativas, humanos y robots trabajan juntos para lograr un objetivo común. Algunos de los aspectos relevantes de HRI en tareas cooperativas son:

- **Diseño de tareas:** el diseño de tareas es un aspecto importante de HRI en tareas cooperativas. La tarea debe diseñarse para aprovechar las ventajas de los humanos y los robots. La tarea debe ser clara, bien definida y fácilmente comprensible tanto para humanos como para robots.
- **Diseño del robot:** El diseño del robot también es un aspecto crítico de HRI en tareas cooperativas. El robot debe estar diseñado para ser seguro, confiable y fácil de usar. Además el robot también debe estar diseñado para interactuar con humanos de una manera natural e intuitiva.
- **Comunicación:** La comunicación es un aspecto crucial de HRI en tareas cooperativas. Los humanos y los robots deben poder comprender las acciones y señales

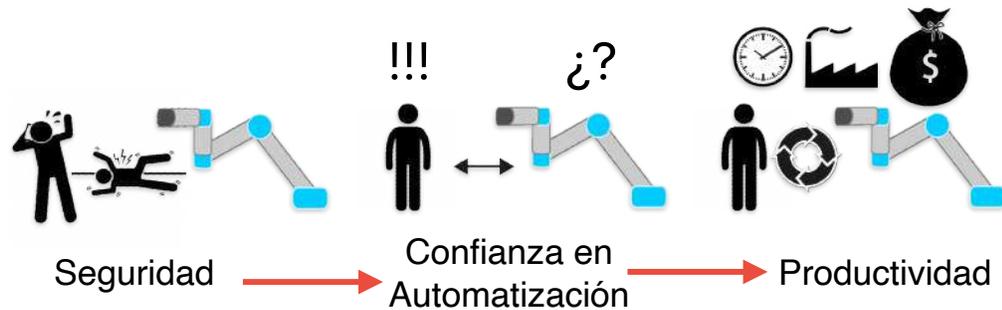


Figura 1.3: Retos principales de HRI.

de los demás para trabajar juntos de manera efectiva. La comunicación entre humanos y robots puede ser a través de medios visuales, auditivos o hápticos.

- **Colaboración:** Los humanos y los robots deben trabajar juntos para lograr el objetivo de la tarea. El robot debe programarse para operar dentro de parámetros seguros y evitar colisiones con humanos u otros objetos en el entorno.
- **Confianza:** Los humanos deben confiar en que el robot realizará sus tareas de manera segura y confiable.
- **Seguridad:** Deben implementarse las medidas de seguridad adecuadas para garantizar la seguridad del humano cuando trabaje en conjunto con un robot.

Como se muestra en la Figura 1.3, los principales retos de la HRI se basan en la seguridad pues de esta depende la confianza del operador con la máquina. Ya que la generación de confianza impacta en la productividad.

Las implicaciones de la seguridad física en HRI contemplan aspectos como:

- **Evaluación de riesgos:** antes de comenzar cualquier tarea cooperativa, se debe realizar una evaluación de riesgos para identificar los peligros potenciales y desarrollar las medidas de seguridad adecuadas. Esto incluye evaluar el entorno donde se llevará a cabo la tarea e identificar cualquier riesgo potencial para los humanos y los robots involucrados.
- **Medidas de seguridad:** una vez identificados los riesgos potenciales, se deben implementar medidas de seguridad para mitigarlos. Estos pueden incluir barreras físicas, sistemas de advertencia, botones de parada de emergencia o equipo de protección personal.
- **Diseño del robot:** los robots deben diseñarse para minimizar el daño físico a los humanos en caso de colisión u otro accidente. Algunas características a considerar

son el uso de materiales blandos para su fabricación, bordes redondeados en los eslabones, o sensores de detección de colisiones, entre otros.

- Comportamiento del robot: el robot debe de funcionar de forma precisa dentro de parámetros de operación seguros, evitar colisiones con humanos u otros objetos en el entorno y en caso de no poder evitarse, generar el menor daño posible. La precisión del robot, la capacidad de reaccionar a colisiones, la optimización tanto del espacio de trabajo como de las trayectorias a realizar así como esquemas de control de fuerza son aspectos a tomar en cuenta para este fin.
- Capacitación y comunicación: las personas que trabajan con robots en tareas cooperativas deben recibir capacitación adecuada sobre medidas de seguridad y cómo trabajar con el robot de manera segura. Sensores como micrófonos y cámaras son algunas de las estrategias utilizadas para la comunicación en HRI.

De los diferentes factores que intervienen en HRI en este trabajo se consideran el espacio de trabajo del robot mediante, la precisión del robot para realizar tareas y la interacción del robot con su entorno a través de las fuerzas de contacto.

En este sentido, el espacio de trabajo del robot determina el espacio físico en el que el robot puede operar e interactuar con los humanos como se observa en la Figura 1.4. El espacio de trabajo tiene un impacto significativo en los sistemas HRI. Un espacio de trabajo mal condicionado puede dificultar que el robot realice tareas con destreza y agilidad, lo que limita su eficacia y aumenta el riesgo de colisiones con humanos. La optimización del espacio de trabajo del robot es crucial para lograr una interacción efectiva y segura entre humanos y robots pues es posible mejorar la accesibilidad, aumentar la eficiencia de finalización de tareas y reducir el riesgo de colisiones entre humanos y robots.

Los robots precisos son esenciales para una HRI exitosa, ya que permiten una interacción segura y efectiva con los humanos. Esto es particularmente importante en situaciones en las que los robots y los humanos trabajan de manera próxima unos de otros. Además es más probable que los robots precisos generen confianza con los humanos en HRI. Sin embargo la precisión del robot puede degradarse debido a diferentes factores tanto ambientales como electromecánicos, por ello es importante detectar la degradación posicional del robot de manera oportuna.

Por otra parte, la fuerza aplicada por el robot al interactuar con el entorno tiene un impacto significativo en la seguridad y eficacia de los sistemas HRI. Los robots deben poder aplicar la cantidad adecuada de fuerza para realizar sus tareas sin causar daños a los objetos con los que interactúan sin representar un riesgo para los operadores humanos. Para ello la medición de fuerza en el robot es de suma importancia, para ello se utilizan sensores de torque y fuerza que se montan en el brazo robot. Estos sensores tienen un alto costo y solo se obtienen mediciones donde se encuentran instalados, por lo cual para poder obtener un sondeo de fuerzas y toques de un robot completo son necesarios numerosos sensores montados en diferentes posiciones. Algunas alternativas a los sensores de fuerza son los observadores de estado. Los observadores de estado

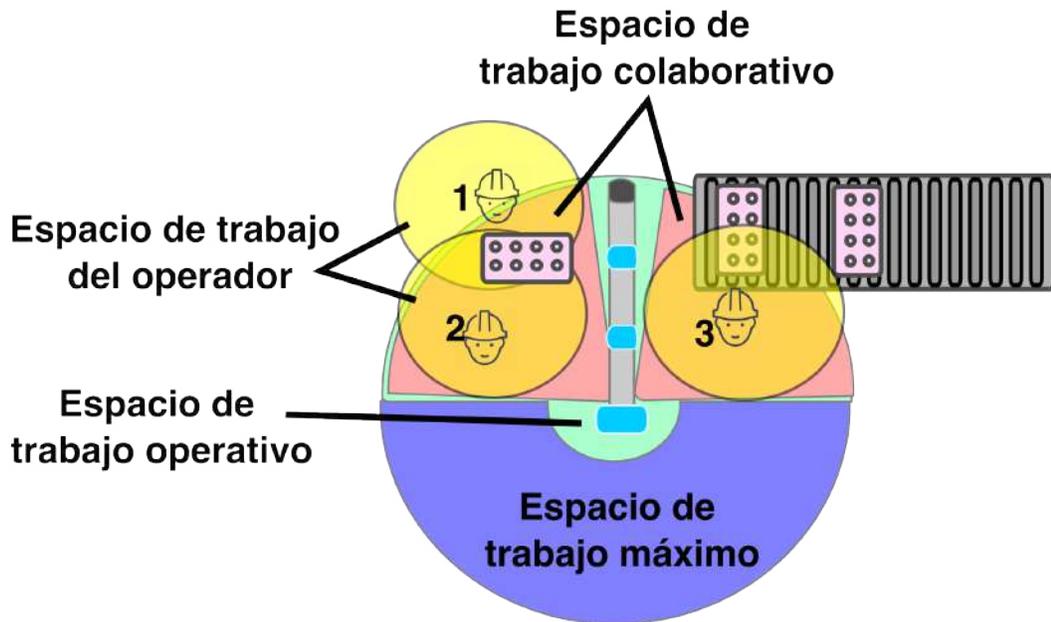


Figura 1.4: (a) Espacios de trabajo en tareas cooperativas.

permiten realizar la estimación de una variable dentro de un modelo matemático, en este caso la variable a observar es la fuerza y torque que se estiman a partir del modelo dinámico del robot. Un enfoque alternativo a los observadores de estados clásicos es la estimación del comportamiento de un sistema mediante algoritmos de inteligencia artificial como redes neuronales.

HRI es un campo complejo y multidisciplinario que depende en gran medida de la capacidad de los robots para trabajar con precisión y seguridad junto a los humanos. El espacio de trabajo óptimo del robot, la precisión del robot y la medición de fuerza sin sensores son factores críticos que afectan el rendimiento de los robots al trabajar en conjuntó con humanos. Al explorar la relación entre estos factores, podemos obtener una mejor comprensión de cómo mejorar la eficacia, la eficiencia y la seguridad de la interacción humano-robot.

1.2. OPTIMIZACIÓN DEL ROBOT

En tareas HRI, el operador y el robot comparten su espacio de trabajo. El espacio de trabajo del operador está conformado por el rango de movimiento necesario para llevar a cabo la tarea requerida, mientras que el espacio de trabajo del robot idealmente es una esfera con radio igual a la longitud del brazo robot extendido. Sin embargo el espacio de trabajo del robot puede ser no completamente regular, sino que debido a diferentes limitaciones mecánicas y eléctricas el alcance del robot se limita en algunas regiones del volumen. El espacio de trabajo total del robot es dividido en diferentes secciones dependiendo de la operación, generalmente se divide en espacio total y espacio

operativo; en tareas HRI se considera también la sección de espacio cooperativo. Dentro del espacio de trabajo operativo y cooperativo el robot ejecuta la tarea requerida, en estos espacios es posible generar un sub-espacio de trabajo donde el se mejore desempeño del robot. Para esto se emplean métodos de optimización los cuales generalmente están basados en lograr que los índices de rendimiento del robot obtengan sus mejores valores posibles. Estos índices de rendimiento consideran las características físicas del robot como la longitud del enlace, el espacio de trabajo, los grados de libertad, la masa, la rigidez y la inercia.

Las metodologías basadas en técnicas bio-inspiradas son ampliamente utilizadas para resolver problemas de optimización de robots.

Se han propuesto diferentes metodologías de diseño óptimo de robots utilizando la combinación de algoritmos de optimización e índices de rendimiento como funciones objetivo. Algunos de ellos reportan algoritmos como MultiStart para optimizar el índice de manipulabilidad (Filippeschi et al., 2021), minimización del índice de destreza (Zarkandi, 2021) y método exhaustivo para optimizar el volumen, el espacio de trabajo, la destreza y la eficiencia estática (Russo et al., 2018).

Se han informado en la literatura propuestas de diseño de robots basadas en algoritmos bioinspirados como la optimización de enjambre de partículas (PSO). Por ejemplo, optimizando la masa y la rigidez de una máquina cinemática paralela de cinco grados de libertad (5 GDL) se obtiene una mejora significativa en ambos parámetros (Sun y Lian, 2018). Además, la optimización de las estructuras cinemáticas de un robot híbrido paralelo utilizando el índice de destreza para obtener un mejor espacio de trabajo libre de singularidades en comparación con la plataforma general de Stewart se presenta en Kucuk (2018). En Zheng et al. (2022) presentan la optimización de la posición de instalación del cilindro hidráulico para robótica médica portátil que cumple con los requisitos del movimiento humano normal.

El trabajo reciente sobre el diseño óptimo de robots también ha informado el uso de algoritmos genéticos (AG) para optimizar diferentes parámetros, por ejemplo, los parámetros del manipulador de un robot serial utilizando el índice de longitud estructural (ILS), el índice de condicionamiento global (ICG) y el índice de acondicionamiento de dinámica modificada (IADM) que mejora el rendimiento cinemático y dinámico del robot son presentados en Hwang et al. (2017). En Meng et al. (2021) presentan la optimización del diseño estructural de un robot Delta paralelo utilizando el Índice de Evaluación de Rigidez (IER), mejorando la dinámica del robot. Además, en Amar y Nagase (2021) se optimizó con AG la geometría y el espacio de trabajo de un sistema robótico tipo árbol, y se comprueba el planteamiento con el diseño de la plataforma robótica y un sistema de agarre.

Actualmente, existen diferentes técnicas de optimización bioinspiradas ampliamente utilizadas en otras áreas de la ciencia, como el optimizador de lobo gris (GWO del inglés *Gray Wolf Optimizer*) y el algoritmo de optimización de águila Harris (HHO del inglés *Harris Hawk Optimizer*). Estos algoritmos tienen ventajas como la rápida convergencia, la carga computacional reducida y la facilidad de implementación. El algoritmo GWO se basa en el comportamiento de una manada de lobos (Mirjalili et al., 2014) y se

ha utilizado en el área de la robótica en distintas aplicaciones como planificación de trayectorias de robots (Chen et al., 2022), prevención de colisiones en robots móviles (Y. Wu et al., 2021), control (Roy y Ghoshal, 2020) y cálculo de cinemática inversa (Dereli, 2021). El HHO se basa en el comportamiento de caza del halcón de Harris (Heidari et al., 2019) y tiene un alto rendimiento en problemas de optimización. Las aplicaciones recientes de HHO son: diseño de vehículos (Issa y Samn, 2022), clasificación y optimización de funciones (Houssein et al., 2021), optimización de una red neuronal de memoria a corto plazo (Li et al., 2022), optimización de problemas de programación centrados en el consumo de energía (M. Wu et al., 2021) y control de aviones (Izci et al., 2020).

Los trabajos presentados muestran algunas de las aplicaciones recientes de los algoritmos bio-inspirados y su desempeño en diferentes problemas de optimización de robots de diseño y optimización de robots manipuladores.

1.3. DEGRADACIÓN DE LA POSICIÓN

Para garantizar la seguridad del operador en tareas de HRI, es necesario que el sistema robótico trabaje dentro de sus parámetros de operación óptimos. Parámetros como espacio de trabajo, velocidades y torques máximos están regulados mediante programación, esto evita que el robot alcance velocidades, posiciones, o ejerza torques que puedan comprometer su estructura y el correcto funcionamiento de los actuadores. Respecto a la posición, idealmente, una vez que el robot ha sido programado para realizar una tarea repetitiva, debe permanecer dentro de su trayectoria; sin embargo, en realidad, varios factores, como fallas en el ensamblaje del robot, defectos de fabricación en la estructura y contragolpe, pueden causar una degradación repentina o gradual de la precisión del robot (Qiao y Weiss, 2019).

En este sentido, se han realizado investigaciones como la presentada por Qiao et al. (2017), donde el enfoque es desarrollar metodologías para la evaluación rápida de la salud posicional del robot evaluando la precisión de la posición y orientación del robot. Qiao y Weiss (2019) realizaron un modelo de error avanzado, para caracterizar los errores dependientes presentes en el robot (deflexiones de la estructura y movimientos no ideales), utilizando polinomios de Chebyshev. La posición del robot se midió mediante un sistema de visión artificial, diseñado para tal fin y presentado por (Qiao y Weiss, 2018b). En Taha et al. (2021), se propuso una metodología basada en memoria profunda a largo plazo y ajuste de regresión multivariante, para modelar y predecir (respectivamente) el desplazamiento de un robot industrial, y estimar el error residual en el efector final. En Qiao y Garner (2020) se diseñó un objetivo inteligente para ser utilizado con el sistema de visión mencionado anteriormente; su viabilidad para realizar la evaluación de la precisión del robot se verificó en Qiao (2021). Qiao y Weiss (2018a), presentaron una metodología de análisis de sistema de medición y controlador externo, detección ambiental y adicional para el monitoreo y diagnóstico de precisión de la herramienta del robot. Y. Liu et al. (2020) se propuso un sistema de seguimiento óptico

para mejorar la precisión del robot y se propuso un método para calibrar el robot a través del análisis cinemático.

Cabe señalar que algunas de las propuestas para la evaluación de la salud posicional del robot utilizan un sistema (cámaras o láseres) externo al sistema robótico para medir el efector final. Aunque este sistema se puede utilizar en condiciones controladas y funciona correctamente, es complejo de implementar en un entorno industrial. Algunos factores que condicionan el uso de sistemas de medición externos en instalaciones industriales son las vibraciones, el ruido, los materiales reflectantes, la iluminación y la accesibilidad, además de la logística de analizar un gran grupo de robots. En este sentido, se propone utilizar metodologías en inteligencia artificial e información del robot obtenida a nivel de controlador, para obtener una metodología ágil para la evaluación rápida del estado posicional del robot.

1.4. ESTIMACIÓN DE FUERZAS CON REDES NEURONALES

La estimación precisa de fuerzas en robots manipuladores es fundamental para su desempeño seguro y preciso en tareas HRI. Sin embargo, el alto costo de los sensores de fuerza ha motivado a buscar diferentes formas para estimar las fuerzas de contacto en los robots. En particular, el uso de redes neuronales ha representado una poderosa herramienta para abordar este desafío, ofreciendo la capacidad de aprender y modelar relaciones complejas entre diferentes parámetros como posiciones, velocidades, aceleraciones o corrientes de los actuadores y las fuerzas ejercidas por el robot. A continuación se presentan algunos trabajos donde se han utilizado redes neuronales para estimar fuerzas actuando como observadores, estimadores de parámetros dinámicos o modeladores de fricción para estimar las fuerzas de contacto del robot.

En este sentido X. Liu et al. (2019), presentan un método para estimar las fuerzas externas en un robot flexible utilizando un sensor de torque articular y un modelo de redes neuronales estándar con una sola capa oculta para estimar un modelo de fricción. C. Yang et al. (2019) propone un esquema de control de fuerza sin sensores basado en redes neuronales enfocado en la interacción entre robots manipuladores y los brazos de operadores humanos en una colisión. El esquema de control utilizado es un control de admitancias en espacio articular apoyado de un observador que estima los toques externos aplicados por el operador del robot. Se utiliza una red neuronal de función de base radial para compensar las incertidumbres en el sistema.

De igual modo en (Lin et al., 2021), se utiliza un control de impedancia apoyado con redes neuronales y observador de disturbios para controlar un robot manipulador en operaciones de HRI. Se utilizan redes neuronales para estimar los parámetros dinámicos del robot y se utiliza el control de impedancias para ajustar la relación entre torque y posición.

J. Peng et al. (2019) presenta un esquema de control para robots que se ocupan de parámetros desconocidos y perturbaciones externas. Emplea un observador basado

en redes neuronales para estimar las velocidades de las articulaciones y desarrolla un controlador híbrido adaptativo de posición/fuerza basado en estas estimaciones.

De la misma forma Z. Yang et al. (2019) presenta un método de control de impedancias basado en una red neuronal adaptativa (función de base radial) para sistemas robóticos. Se utiliza un observador para estimar las velocidades de las articulaciones, lo que permite el desarrollo de un controlador de impedancias usando una red neuronal adaptativa para el seguimiento de las fuerzas de contacto deseadas del efector final y la trayectoria deseada del robot. Este controlador compensa las incertidumbres, mejorando la precisión de la posición de las articulaciones y el seguimiento de la fuerza.

Asimismo G. Peng et al., 2020 desarrolla un sistema de control basado en redes neuronales para manipuladores robóticos que operan en entornos desconocidos sin detección de fuerza, incluso en presencia de una zona muerta del actuador. El control de admisión permite respuestas compatibles ante fuerzas externas. Una estructura de red neuronal aplanada que utiliza funciones de base radial (RBF) y un algoritmo de aprendizaje incremental estima pares externos, lo que permite adaptabilidad sin un modelado completo del sistema.

Por otra parte, otros ejemplos de estimación de fuerzas utilizando redes neuronales son los de Abeywardena et al. (2019), que propone un algoritmo novedoso para estimar la interacción de la fuerza herramienta-tejido en cirugía mínimamente invasiva asistida por robot sin depender de sensores de fuerza externos. Este método aprovecha las corrientes del motor y las técnicas de redes neuronales para estimar la interacción de fuerzas. Se realizaron pruebas en línea y fuera de línea, lo que demuestra el potencial del algoritmo para la estimación en tiempo real de la fuerza herramienta-tejido. Esta innovación es prometedora para permitir la retroalimentación háptica en la cirugía robótica.

Finalmente Siu et al. (2021) utiliza redes neuronales de memoria larga a corto plazo para la estimación de torques de tobillo mediante información de electromiografía y acelerómetros.

1.5. DESCRIPCIÓN DEL PROBLEMA

En tareas cooperativas, humanos y robots trabajan juntos para lograr un objetivo común. Sin embargo para garantizar la seguridad del usuario y mejorar la confianza en la automatización es necesario que el robot trabaje dentro de los parámetros de operación establecidos para cada tarea específica. Un espacio de trabajo mal condicionado puede conducir a que el robot llegue a posiciones singulares y produzca daños a su entorno. Por otro lado, es necesario que el robot sea preciso al realizar las tareas cooperativas para evitar colisiones y daños al usuario o al propio robot. El alto costo de los sensores de fuerza y su intrusividad pueden condicionar la realización de tareas cooperativas además de que el sensor solo actúa en donde está montado por lo que son necesarios un mayor número de sensores para monitorizar completamente al robot.

1.6. JUSTIFICACIÓN

Aspectos importantes de HRI que repercuten en la productividad de las tareas cooperativas incluyen el diseño de robots, la comunicación, la colaboración, la confianza y la seguridad. El espacio de trabajo debe diseñarse para obtener el mejor desempeño mecánico del robot y así minimizar la carga de trabajo cognitiva y la tensión física del operador humano. Cuando los robots realizan tareas con precisión, es más probable que los humanos confíen en ellos y dependan de ellos para recibir asistencia. Por otro parte, los robots deben aplicar fuerzas que sean seguras y cómodas para que los operadores humanos interactúen, y el uso de la estimación de fuerza sin sensores puede ayudar reducir el peso del robot, reducir el costo monetario de la ejecución de tareas cooperativas y reducir el riesgo de daños a los objetos en el espacio de trabajo.

1.7. HIPÓTESIS Y OBJETIVOS

1.7.1. HIPÓTESIS

A través del modelado de los parámetros espacio de trabajo óptimo, salud posicional y estimación de fuerzas será posible determinar si el estado operativo del robot se encuentra dentro de los parámetros de operación necesarios para cumplir con tareas cooperativas.

1.7.2. OBJETIVO GENERAL

Desarrollar un conjunto de metodologías basadas en algoritmos inteligentes para determinar el estado operativo del robot en tareas cooperativas mediante un análisis de los parámetros de espacio de trabajo, precisión y estimación de fuerzas en un robot manipulador.

1.7.3. OBJETIVOS ESPECÍFICOS

- Obtener la descripción del movimiento del robot con y sin la intervención de fuerzas mediante el modelado cinemático y dinámico del robot para realizar simulaciones de su comportamiento ante diferentes condiciones de operación.
- Optimizar las longitudes de los eslabones del robot mediante algoritmos inteligentes para obtener un espacio de trabajo donde se obtenga el mejor desempeño en la ejecución de tareas de fuerza.
- Realizar una evaluación de la salud posicional del robot mediante el análisis de corrientes de los motores para determinar si pierde exactitud al realizar operaciones durante períodos prolongados de tiempo.

- Implementar un clasificador basado en algoritmos inteligentes para determinar si las articulaciones del robot tienen comportamientos anómalos que provoquen degradación posicional del robot mediante el análisis de corrientes de los motores.
- Implementar un observador de fuerza mediante algoritmos inteligentes que estimen las fuerzas internas del robot al realizar trayectorias de tareas cooperativas.
- Realizar la simulación de una tarea cooperativa considerando el observador de fuerzas y el monitoreo de la precisión para determinar si el robot trabaja dentro de los parámetros de operación requeridos.

1.8. PLANTEAMIENTO GENERAL

De acuerdo con los antecedentes presentados, el planteamiento del problema y la justificación del mismo es posible plantear una metodología para realizar la comprobación de la hipótesis. En la Figura 1.5 se muestran los tres tópicos de interés que intervienen en tareas HRI propuestos en este trabajo.

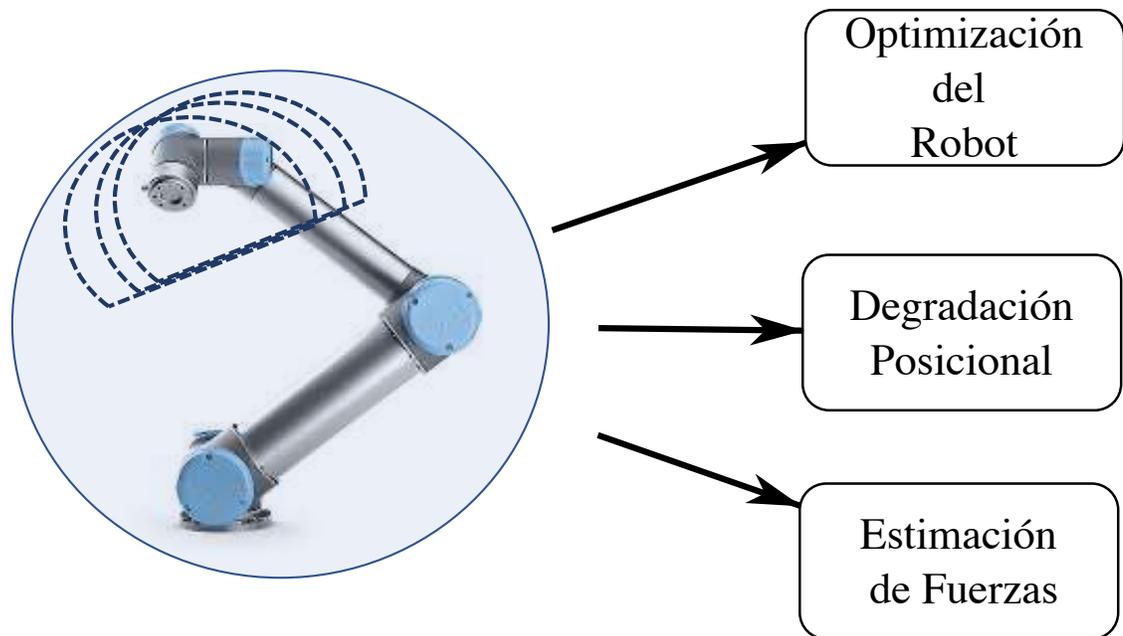


Figura 1.5: Diagrama del planteamiento general de la tesis.

La optimización del robot permite mejorar características de interés que pueden ser benéficas en operaciones HRI como son el espacio de trabajo, la longitud de los eslabones, capacidad de carga, delimitar regiones de trabajo de mejor rendimiento para el robot, etc. Por otra parte analizar la degradación de la posición del robot permite saber en qué parte del robot se genera una desviación que repercute en la precisión del efector

final del robot, esto a su vez posibilita la implementación de estrategias de mantenimiento de forma oportuna. Respecto a la estimación de fuerzas sin sensores, el uso de algoritmos inteligentes hace posible analizar todas las articulaciones del robot con un costo reducido. Cada uno de estos tópicos es desarrollado en la sección de metodología, de forma general en el Capítulo 1 se presenta la introducción a la interacción humano robot y a los tópicos de interés propuestos en este trabajo. En el Capítulo 2 se abordan los fundamentos teóricos de las técnicas utilizadas en este trabajo; mientras que en el Capítulo 3 se presenta la metodología utilizada en cada uno de los tópicos propuestos. Los resultados obtenidos y su discusión son presentados en el Capítulo 4, finalmente en el Capítulo 5 se exponen las conclusiones del trabajo realizado.

Capítulo 2

FUNDAMENTACIÓN TEÓRICA

Los fundamentos teóricos presentados a continuación son la base del desarrollo de este trabajo. Se incluyen conceptos como las bases del análisis cinemático y dinámico de manipuladores, además se presentan los conceptos base de diferentes algoritmos inteligentes, y técnicas de procesamiento de señales, entre otros.

2.1. CINEMÁTICA DIRECTA

La resolución del problema cinemático directo consiste en encontrar las relaciones que permiten conocer la localización del extremo del manipulador a partir de los valores de sus coordenadas articulares. La obtención de estas relaciones no es en general complicada siendo en ciertos casos fácil de encontrar mediante consideraciones geométricas. Para manipuladores de varios grados de libertad se propone un método basado en matrices de transformación homogénea. Debido a que a cada articulación del manipulador se le puede asignar un marco de referencia propio, se puede obtener la posición y orientación final del manipulador recorriendo los marcos de referencia utilizando las matrices de transformación homogénea.

La matriz que representa la posición y orientación entre dos sistemas de referencia consecutivos se denomina A , por ejemplo 0A_1 representa la transformación entre el sistema de referencia de la base y el sistema de referencia del primer eslabón.

La matriz de transformación que contempla todos los grados de libertad y marcos de referencia del manipulador se denomina T , para un manipulador de 3 grados de libertad la matriz T será (2.1):

$$T = {}^0A_1A_2A_3 \quad (2.1)$$

La forma más habitual de situar los marcos de referencias de cada articulación es la representación de Denavit-Hartenberg propuesta en 1955, de escoger de manera correcta los sistemas de referencia de cada eslabón, es posible pasar de un eslabón a otro usando las 4 transformaciones básicas que consisten en Tabla 2.1:

- Rotación en el eje z , un ángulo θ .

Tabla 2.1: Parámetros de Denavit-Hartenberg para un brazo robot de 6 grados de libertad.

Articulación	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	$\pi/2$
2	θ_2	0	a_2	0
3	θ_3	0	0	$\pi/2$
4	θ_4	d_4	0	$-\pi/2$
5	θ_5	0	0	$\pi/2$
6	θ_6	d_6	0	0

- Traslación a lo largo del eje z , una distancia d .
- Traslación a lo largo del eje x , una distancia α .
- Rotación a lo largo del eje x , un ángulo α .

Debido a que el producto matricial no es conmutativo, la multiplicación se debe realizar en el siguiente orden (2.2):

$${}^{i-1}A_i = T(z, \theta_i)T(0, 0, d_i)T(a_i, 0, 0)T(x, \alpha_i) \quad (2.2)$$

El producto de forma matricial (2.3) y (2.4):

$$A_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Donde C y S representan los senos y cosenos correspondientes a los ángulos θ_i y α_i (Barrientos et al., 2007).

2.2. CINEMÁTICA INVERSA

La cinemática inversa provee una relación funcional entre desplazamiento articular y la posición y orientación del efector final del manipulador, las ecuaciones cinemáticas deben de ser resueltas para los desplazamientos articulares, una vez resueltas estas ecuaciones los movimientos deseados del efector final pueden ser obtenidos moviendo

cada articulación al valor determinado. En el problema cinemático directo la posición del efector final es determinada por los desplazamientos articulares, en el caso de la cinemática inversa el problema es más complejo pues existen diversas configuraciones articulares que permiten llegar a la misma posición, además de que no siempre existen soluciones para un rango particular de posiciones del efector final debido a restricciones geométricas del mecanismo (Asada y Slotine, 1991). El problema de la cinemática inversa no cuenta con algoritmos estandarizados para resolverla pues la morfología de cada mecanismo es distinta y el problema en sí mismo es no lineal. Por ejemplo un mecanismo de seis grados de libertad contaría con una matriz de transformación con 12 ecuaciones y seis incógnitas, de estas 12 ecuaciones solo tres son independientes. En casos como este se recurre a una técnica llamada desacoplo cinemático la cual divide al mecanismo en 2 partes lo que simplifica el análisis. En general además de los métodos numéricos se utilizan dos métodos para resolver la cinemática inversa de un manipulador, el método algebraico que consiste básicamente en obtener la cinemática directa del manipulador mediante una matriz de transformación homogénea, formar un sistema de ecuaciones y resolverlo para obtener los valores articulares del manipulador. El método geométrico consiste en descomponer el manipulador en varios planos geométricos que representan problemas individuales más simples (Craig, 2006).

2.3. CINEMÁTICA DIFERENCIAL

La cinemática diferencial consiste en la obtención de la matriz Jacobiana y su análisis, la matriz Jacobiana es una forma de la derivada de una función de múltiples variables. Se supone por ejemplo que se tienen seis funciones, cada una con seis variables independientes (2.5):

$$\begin{aligned} y_1 &= f_1(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_2 &= f_2(x_1, x_2, x_3, x_4, x_5, x_6) \\ &\vdots \\ y_6 &= f_6(x_1, x_2, x_3, x_4, x_5, x_6) \end{aligned} \quad (2.5)$$

Se puede utilizar también la notación vectorial para escribir las ecuaciones como (2.6):

$$Y = F(X) \quad (2.6)$$

Donde F representa un arreglo de funciones con sus variables independientes (X). Si se calculan las derivadas parciales respecto a x , se tiene (2.7):

$$\begin{aligned} \partial y_1 &= \frac{\partial f_1}{\partial x_1} \delta x_1 + \frac{\partial f_1}{\partial x_2} \delta x_2 + \cdots + \frac{\partial f_1}{\partial x_6} \delta x_6 \\ \partial y_2 &= \frac{\partial f_2}{\partial x_1} \delta x_1 + \frac{\partial f_2}{\partial x_2} \delta x_2 + \cdots + \frac{\partial f_2}{\partial x_6} \delta x_6 \\ &\vdots \\ \partial y_6 &= \frac{\partial f_6}{\partial x_1} \delta x_1 + \frac{\partial f_6}{\partial x_2} \delta x_2 + \cdots + \frac{\partial f_6}{\partial x_6} \delta x_6 \end{aligned} \quad (2.7)$$

Que se puede escribir con notación vectorial como:

$$\delta Y = \frac{\partial F}{\partial X} \delta X \quad (2.8)$$

La matriz de 6×6 de las derivadas parciales se denomina Jacobiano y se representa con la letra J (2.8). En el campo de la robótica el Jacobiano se utiliza para relacionar las velocidades articulares a las velocidades cartesianas del efector final del manipulador (2.9).

$${}^0v = {}^0 J(\Theta) \dot{\Theta} \quad (2.9)$$

Donde Θ es el vector de ángulos articulares del manipulador y v es el vector de velocidades cartesianas el superíndice 0 indica el marco de referencia en cuestión. Se puede definir un Jacobiano de cualquier dimensión, el número de filas es igual al número de grados de libertad en el espacio cartesiano, el número de columnas en el Jacobiano es igual al número de articulaciones del manipulador (Craig, 2006).

2.4. ÍNDICE DE CONDICIÓN GLOBAL

El Índice de condición global (GCI) propuesto por Gosselin y Angeles (1991) es una adaptación del espacio de trabajo del índice de número de condición presentado por Salisbury y Craig (1982). El número de condición mide la independencia entre las columnas de la matriz jacobiana (Patel y Sobh, 2015). Para una matriz jacobiana de rango completo, el número de condición se define por la relación de los valores singulares máximo y mínimo de la matriz jacobiana. Es cierto que este cálculo no es sencillo y se recomienda utilizar la formulación de la norma de Frobenius, como se muestra en (2.10).

$$\kappa = \frac{1}{n} \sqrt{\text{tr}(JN J^T) \text{tr}(J^{-1} N J^{-T})} \in [1, \infty) \quad (2.10)$$

Donde n es la dimensión de la matriz del jacobiano J , tr es la traza y N está dada por (2.11), donde I es la matriz identidad.

$$N = \frac{1}{n} I_{n \times n} \quad (2.11)$$

El número de condición se puede interpretar de diferentes maneras según el autor. La distancia a una singularidad cinemática o la relación entre la fuerza de salida y la velocidad del espacio de trabajo a partir de los pares de entrada y las velocidades de las articulaciones se encuentra entre los más comunes (Patel y Sobh, 2015). Un valor de índice cercano a la unidad es el mejor valor posible para las condiciones antes mencionadas. Muestra que los valores singulares de la matriz jacobiana son iguales y se dice que el manipulador está en una configuración isotrópica. El número de condición no está completamente acotado (2.10); para evitar que el índice toma valores grandes se utiliza su recíproco, conocido como Índice de Condición Local (LCI) definido en (2.12).

$$LCI = \frac{1}{\kappa} \in [0, 1] \quad (2.12)$$

El LCI es un índice de desempeño local; depende de la configuración actual del robot en el momento del cálculo. La extensión de LCI a todo el espacio de trabajo se detalla en Gosselin y Angeles (1991), se propone un cálculo computacional más simple en Puglisi et al. (2012) y se da como:

$$GCI = \frac{1}{n_{WS}} \sum_{j=1}^{n_{WS}} \frac{1}{\kappa} \in (0, 1). \quad (2.13)$$

Donde n_{WS} es el número de nodos del espacio de trabajo. El GCI (2.13) está completamente acotado y estima la distribución de κ en todo el espacio de trabajo.

2.5. ÍNDICE DE LONGITUD ESTRUCTURAL

El índice de longitud estructural (SLI) es un índice que presenta la relación entre la longitud de los enlaces del robot y el volumen del espacio de trabajo W . Se define como (2.14):

$$SLI = \frac{L}{\sqrt[3]{W}}. \quad (2.14)$$

Desde (2.14) L es la suma de la longitud de los enlaces a_{i-1} y los desplazamientos d_{i-1} (2.15).

$$L = \sum_{i=1}^n a_{i-1} + d_{i-1} \quad (2.15)$$

Un diseño deseable tiene un SLI bajo a través de un gran volumen de espacio de trabajo y una longitud de enlace pequeña.

2.6. DINÁMICA DEL MANIPULADOR

2.6.1. LAGRANGIANO

Para obtener la relación entre el movimiento de un manipulador y el torque presente en sus articulaciones, se necesita realizar el análisis dinámico del mismo. Para ello se recurre a la formulación del Lagrangiano de un sistema mecánico (2.16), en la que el sistema puede ser definido como una función en un sistema de coordenadas generalizadas:

$$L = T - U \quad (2.16)$$

donde L es el Lagrangiano, T y U describen la energía cinética y potencial del sistema respectivamente. Entonces se pueden expresar las ecuaciones de Lagrange como derivadas parciales (2.17):

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (2.17)$$

para $i = 1, \dots, n$, L es el Lagrangiano, τ_i es la fuerza (torque) generalizada asociada a las coordenadas q_i , el subíndice i , enumera cada una de las articulaciones presentes. Esta fuerza generalizada se puede entender como la contribución de la fuerza en los actuadores de las articulaciones, la fuerza de fricción de las juntas y las fuerzas del efector final cuando está en contacto con el ambiente (Siciliano et al., 2010). En consecuencia, se puede entender el Lagrangiano como la relación entre fuerzas y desplazamiento de un sistema mecánico a partir de la determinación de las energías cinética y potencial.

2.6.2. MODELO DINÁMICO

El Lagrangiano puede expresarse como (2.18):

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - U(q) \quad (2.18)$$

Donde $M(q)$ es una matriz simétrica definida positiva denominada matriz de inercia y U es la energía potencial que depende del vector de variables articulares q . Con la forma del Lagrangiano (2.18) la ecuación de movimiento de Lagrange puede expresarse como (2.19):

$$\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \right] - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] + \frac{\partial U(q)}{\partial q} = \tau \quad (2.19)$$

O de forma compacta como (2.20):

$$\tau = M(q) \ddot{q} + C(q, \dot{q}) + g(q) \quad (2.20)$$

La ecuación (2.20) representa la dinámica para robots de n grados de libertad y es una ecuación diferencial vectorial no lineal. $C(q, \dot{q})$ es un vector de $n \times 1$ llamado vector de fuerzas centrífugas y de Coriolis, $g(q)$ es un vector llamado de fuerzas o pares gravitacionales y τ es un vector de fuerzas aplicadas a las articulaciones y fuerzas externas. Cada elemento del modelo en general es una función relativamente compleja de posiciones y velocidades de todas las articulaciones (Kelly, Santibánes et al., 2003).

2.7. ALGORITMOS DE INTELIGENCIA COMPUTACIONAL

En esta sección se abordan los algoritmos inteligentes utilizados en la elaboración de este trabajo, se abordan las redes neuronales artificiales así como los algoritmos de optimización del lobo gris y el águila Harris, además de los algoritmos genéticos y una breve introducción a los algoritmos de enjambre.

2.7.1. INTELIGENCIA DE ENJAMBRE

La inteligencia de enjambre es el comportamiento colectivo de sistemas descentralizados y auto-organizados, naturales o artificiales. Es una técnica heurística inspirada en el estudio de grupos de animales, más específicamente de insectos como abejas, hormigas y termitas entre otros. Este tipo de comunidades se caracterizan por la forma de trabajo en grupo por el mismo fin, que es para el bien de la colmena en general, son muy organizadas y comunicativas por lo que sí se ve a la colmena como un individuo, se suman las experiencias de cada uno de los elementos que pertenecen a ella y, por lo tanto le permite tomar decisiones más acertadas. Además de trabajar en grupo con el mismo objetivo se hacen más fáciles, rápidas y probables las tareas cotidianas. Por ejemplo encontrar mejores fuentes de comida para alimentar a la colmena, encontrando rutas óptimas y minimizando los esfuerzos realizados como por ejemplo encontrar el camino más corto a una fuente de comida cercana (Corne et al., 2012).

2.7.2. ALGORITMOS GENÉTICOS

Los algoritmos genéticos son utilizados en problemas de optimización con alto grado de complejidad o no linealidad, basan su funcionamiento en los procesos de la naturaleza que generan nuevos individuos a través de diferentes generaciones, como las mutaciones o el cruce de otros individuos buscando siempre el individuo más apto, los componentes principales de un algoritmo genético se describen a continuación (Rao, 2009).

- Población inicial: se compone de individuos generados de forma aleatoria, cada uno representa una posible solución al problema.
- Evaluación: se determina que tan bueno es un individuo para solucionar el problema mediante una función de aptitud, los mejores individuos tienen mejor aptitud.
- Selección: los mejores individuos pasan a la siguiente generación en base a su aptitud o de manera aleatoria, por lo general en parejas.
- Cruza: se emula el proceso de cruce en la naturaleza donde los genes de los padres se intercambian para generar nuevos individuos, por lo general con mejores características que sus predecesores (Figura 2.1).
- Mutación: La mutación se basa en un operador básico que brinda aleatoriedad a la población, esta mutación debe ser controlada mediante una muy baja probabilidad de acción sobre los descendientes de las nuevas generaciones obtenidos de la operación de cruce y debe producir individuos válidos. Se produce cambiando uno o más genes con distintos criterios o en modo aleatorio (Figura 2.1).

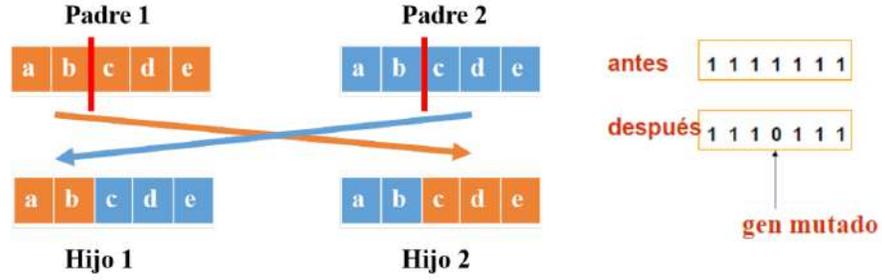


Figura 2.1: Representación gráfica de los operadores de cruce y mutación en una red neuronal.

2.7.3. OPTIMIZADOR DEL LOBO GRIS

El algoritmo optimizador del lobo gris GWO (del inglés *Grey Wolf Optimizer*) presentado en Mirjalili et al. (2014) se basa en el comportamiento de los lobos grises cuando cazan en manada, su organización jerárquica y su interacción social. En una manada de lobos, estos pueden clasificarse en función del nivel de la jerarquía; consideremos los tres lobos más esenciales: el líder de la manada es el lobo alfa (α), y el segundo y tercer lobos se denotan como beta (β) y delta (δ) respectivamente. Los lobos restantes se denominan omega (ω). Para reproducir el proceso de caza de la manada de lobos, hacemos las siguientes suposiciones:

- Supongamos una población (manada) de tamaño N candidatos a solución (lobos) para un problema de optimización.
- Al igual que los lobos, la población de soluciones se puede dividir en las categorías α, β, δ , y ω .
- Los lobos acechan a su presa hasta rodearla (Figura 2.2). Para ello, se mueven aleatoriamente alrededor de la presa, ayudados por las posiciones de los lobos de mayor jerarquía (α, β, δ).
- Cuando la presa se detiene, los lobos atacan.

El método matemático que imita el comportamiento de los lobos para rodear y cazar a sus presas viene dado por las siguientes ecuaciones:

$$D = |CX_p(k) - X_i(k)|, \quad (2.21)$$

$$X_i(k+1) = X_p(k) - AD, \quad (2.22)$$

$$A = 2ar_1 - a, \quad (2.23)$$

$$C = 2r_2. \quad (2.24)$$

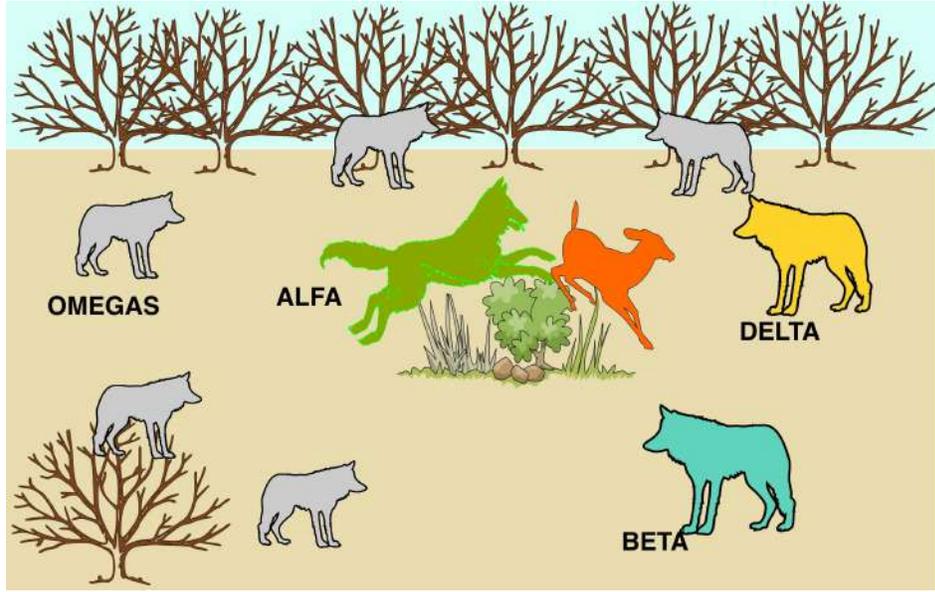


Figura 2.2: Representación gráfica una manada de lobos rodeando a su presa.

Donde k es el número de iteraciones, en (2.21) D representa la distancia entre la presa $X_p(k)$ y el lobo i $X_i(k)$, con $i = 1, \dots, N$. Para la siguiente posición del lobo, (2.22), A y C son vectores de coeficientes. El vector A condiciona la caza de la presa ($A > 1$) y su búsqueda ($A < 1$). El vector C imita los obstáculos para acercarse a la presa en la naturaleza. Finalmente (2.23) y (2.24), los términos r_1 y r_2 son valores aleatorios en $[0, 1]$, el término a disminuye linealmente de 2 a 0 a medida que transcurren las iteraciones.

Para modelar el método de caza del lobo, desconocemos la posición de la presa (solución del problema); suponemos que los lobos α , β y δ la conocen mejor y se toman como referencia para los lobos ω .

$$D_\alpha = |C_1 X_\alpha(k) - X_i(k)| \quad (2.25)$$

$$D_\beta = |C_2 X_\beta(k) - X_i(k)| \quad (2.26)$$

$$D_\delta = |C_3 X_\delta(k) - X_i(k)| \quad (2.27)$$

Con las distancias calculadas respecto a los lobos α (2.25), β (2.26), y δ (2.27), se calculan las nuevas posiciones de referencia.

$$X_1(k) = X_\alpha(k) - A_1 D_\alpha \quad (2.28)$$

$$X_2(k) = X_\beta(k) - A_2 D_\beta \quad (2.29)$$

$$X_3(k) = X_\delta(k) - A_3 D_\delta \quad (2.30)$$

Utilizando (2.28)-(2.30) se calculan los siguientes valores para las posiciones de los lobos ω .

$$X_i(k+1) = \frac{X_1 + X_2 + X_3}{3} \quad (2.31)$$

Con las ecuaciones descritas anteriormente, el algoritmo GWO se puede resumir de la siguiente manera:

1. Inicializar una población de lobos grises de tamaño N y los parámetros A , C .
2. Evaluar la aptitud de todos los lobos. A continuación, seleccione los tres mejores resultados para α , β , y δ lobos.
3. Ciclo hasta que el número máximo de iteraciones se alcanza.
 - Ciclo para actualizar la posición de cada lobo en la población utilizando (2.31).
 - A continuación, actualiza A y C .
 - Calcular la nueva aptitud para todos los lobos. A continuación, seleccione los tres mejores resultados y actualización α , β , y δ lobos.
 - Incrementar iteración.
4. Volver α lobo como la mejor solución.

2.7.4. OPTIMIZADOR DEL ÁGUILA HARRIS

La propuesta de optimización del Águila Harris HHO (del inglés *Harris Hawk Optimizer*) descrita por Heidari et al. (2019), toma su inspiración del comportamiento colaborativo observado en los halcones Harris durante la caza de presas en situaciones de escape. La estrategia de caza involucra una secuencia de embestidas sorpresa realizadas por múltiples halcones desde diversas direcciones. En congruencia con la naturaleza, este algoritmo también contempla la posibilidad de que la presa evada el ataque y sea perseguida hasta su agotamiento. De acuerdo a lo ilustrado en la Figura 2.3, el algoritmo HHO puede ser sintetizado de la siguiente manera: Para la etapa de exploración:

- Considerar la energía que tiene la presa para escapar como E .
- Cuando la presa tiene altos niveles de energía $E \geq 1$, la fase de exploración depende del factor q .
- Si $q \geq 0,5$, los halcones se posan en posiciones aleatorias. En caso contrario, se sitúan en función de la posición de otros halcones ($q < 0,5$).

Los halcones pasan de la fase de exploración a la de caza a medida que descende el nivel de energía de la presa. Para la fase de explotación:

- Supongamos que el término r es la probabilidad de escape de la presa. Considere $r \geq 0,5$ como un escape fallido. En caso contrario, $r < 0,5$ para un escape exitoso.
- El nivel de energía y la probabilidad de escape fallido se utilizan para determinar si los halcones asedian a la presa de forma suave ($r \geq 0,5$ y $E \geq 0,5$) o de forma dura ($r \geq 0,5$ y $E < 0,5$).

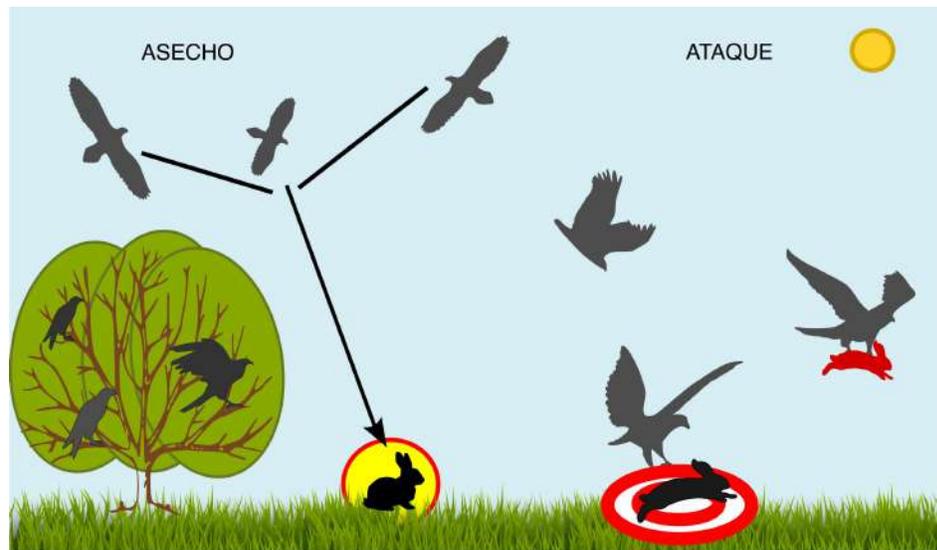


Figura 2.3: Representación gráfica de un halcón acechando y cazando a su presa.

- Si la probabilidad de escapar con éxito es alta ($E \geq 0,5$ y $r < 0,5$), los halcones realizan asedios suaves con inmersiones rápidas progresivas para corregir su posición y dirección a los movimientos engañosos realizados por la presa al escapar.
- Por último, cuando $E < 0,5$, la presa no tiene suficiente energía para escapar, los halcones preparan una serie de asedios duros con inmersiones rápidas progresivas antes de atrapar finalmente a la presa (encontrar la mejor solución).

Los modelos matemáticos de los conceptos anteriores pueden revisarse en profundidad en Heidari et al. (2019).

2.7.5. REDES NEURONALES ARTIFICIALES

Las redes neuronales tratan de emular el comportamiento del cerebro humano, caracterizado por el aprendizaje a través de la experiencia y la extracción de conocimiento genérico a partir de un conjunto de datos.

Estos sistemas imitan esquemáticamente la estructura neuronal del cerebro, bien mediante un programa de ordenador (simulación), bien mediante su modelado a través de estructuras de procesamiento con cierta capacidad de cálculo paralelo (emulación), o bien mediante la construcción física de sistemas cuya arquitectura se aproxima a la estructura de la red neuronal biológica (implementación en hardware). Las redes neuronales artificiales están formadas por una serie de procesadores elementales que constituyen dispositivos simples de cálculo que, bien a partir de un vector de entrada procedente del mundo exterior o bien a partir de estímulos recibidos de otras neuronas, proporcionan una respuesta única (salida).

Un perceptrón simple se refiere a una unidad matemática que simboliza una neurona en el contexto de redes neuronales. Esta definido como 2.32:

$$y = f\left(\sum_{i=1}^m w_i x_i + b\right). \quad (2.32)$$

Recibe un vector de entrada x con m elementos y realiza una secuencia de operaciones. Inicialmente, el vector de entrada es sometido a una multiplicación con un vector de pesos w , cuyos productos son luego sumados junto con un término de sesgo b . El valor resultante atraviesa una función de activación $f()$, la cual puede ser de tipo binario, lineal, sigmoidea o tangente hiperbólica. Esto conduce a la obtención de la salida y .

Una Red Neuronal Artificial (RNA) se compone de una colección de estos perceptrones básicos interconectados. Un tipo común es el perceptrón multicapa, que presenta una configuración donde todas las neuronas en las capas de entrada, ocultas y salida están conectadas entre sí. Esta disposición se ilustra en la Figura 2.4.

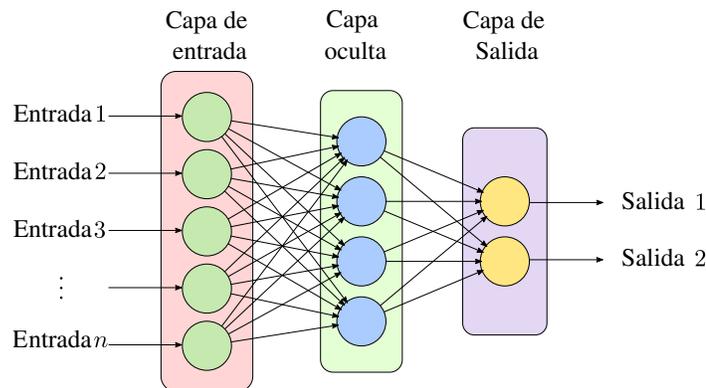


Figura 2.4: Diagrama general de una red neuronal artificial.

Las RNA tienen aplicaciones valiosas en tareas relacionadas con clasificación, aprendizaje y reconocimiento de patrones, siendo ampliamente utilizadas en diversas disciplinas científicas. Por ejemplo, usos recientes incluyen la detección temprana del cáncer de mama (Desai y Shah, 2021) y la combinación con algoritmos inspirados en la biología para mejorar el proceso de entrenamiento de las RNA (Heidari et al., 2020).

2.7.6. RED NEURONAL DE MEMORIA LARGA A CORTO PLAZO

Las redes neuronales de memoria a corto y largo plazo LSTM (del inglés *Long Short Term Memory*) son un tipo especializado de redes neuronales recurrentes RNN (del inglés *Recurrent Neural Network*) que tienen la capacidad de modelar y predecir secuencias de datos temporales. A diferencia de las RNN convencionales, las LSTM están diseñadas para no tener el problema del desvanecimiento del gradiente. La arquitectura

de las LSTM mostrada en la Figura 2.5 incorpora unidades de memoria, puertas de olvido y actualización, lo que les permite discernir qué información es crucial para retener y qué información descartar en cada paso de tiempo. Las variables c_i , h_i y x_i son los estados de la celda, salida y entrada actuales respectivamente, mientras que c_{i-1} , h_{i-1} son los estados pasados. Los estados presentes se encadenan con la siguiente celda LSTM y se convierten en estados pasados de la celda activa, de este modo el proceso se repite sucesivamente. La capacidad para capturar y retener secuencias temporales complejas las ha convertido en una herramienta importante en diferentes aplicaciones como reconocimiento de emociones del habla (Zhao et al., 2019), predicción de trayectorias en carretera (Alché y de La Fortelle, 2017), composición de música de forma automática (Choi et al., 2016) y clasificación de arritmias (Yildirim et al., 2019), entre otras.

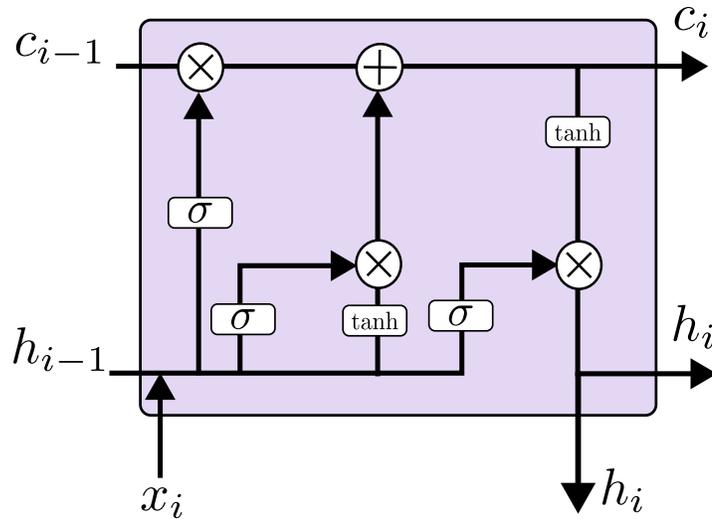


Figura 2.5: Diagrama general de una red neuronal artificial.

Para una revisión detallada del funcionamiento de las redes LSTM se recomienda consultar Staudemeyer y Morris (2019).

2.8. TRANSFORMADA WAVELET DISCRETA

La Transformada Wavelet Discreta DWT (del inglés *Discrete Wavelet Transform*) es un enfoque matemático que brinda información acerca de la relación tiempo-frecuencia en una señal (Sundararajan, 2016). En este método, la señal se divide en distintas bandas de frecuencia mediante un proceso de filtrado. Este proceso implica combinar la señal de interés con una función de transformación conocida como “ondícula madre”, que define los parámetros de los filtros de alta y baja frecuencia. En términos más simples, la DWT posibilita descomponer una señal en sus componentes de frecuencia, permitiendo así la detección de cambios en la frecuencia a lo largo del tiempo. El procedimiento completo consta de dos fases: primero, se calculan los coeficientes de

aproximación y detalle; luego, se reconstruye la señal al nivel de descomposición necesario mediante la utilización de los coeficientes de la etapa previa en una operación llamada DWT inversa. La ecuación 2.33 define la DWT de una señal discreta en el dominio temporal $x(k)$ con k muestras, mientras que ψ denota la ondícula madre discreta empleada para el nivel de descomposición i .

$$DWT_{ik} = \sum x(k)\psi_{i,k}(t). \quad (2.33)$$

La selección de la familia madre de wavelets depende de la aplicación; las familias más comunes son Meyer, Haar y Daubechies, entre otras. El algoritmo DWT consta de los siguientes pasos y se representa en la Figura 2.6a y b:

- Definir la función wavelet madre o base wavelet y el nivel de descomposición i a utilizar.
- La función wavelet madre proporciona los coeficientes del filtro $g[n]$ de paso alto y $h[m]$ de paso bajo.
- Realiza la operación de convolución entre la señal $x[n]$ y los filtros para separar la señal en bandas de alta y baja frecuencia.
- Disminuir la muestra en dos señales filtradas obtenidas en el paso anterior, manteniendo las muestras de índice par.
- Las salidas del filtrado y del downsampling de la señal proporcionan el vector de detalle d_i y el vector de aproximación a_i del filtrado pasa alto y pasa bajo, respectivamente, en cada nivel de descomposición i .
- Repetir el proceso de filtrado y reducción de muestreo utilizando la salida de aproximación anterior como entrada para el siguiente nivel de descomposición hasta completar todos los niveles.
- Para reconstruir las ramas de aproximación y detalle de la señal para un nivel específico i , sobremuestree los correspondientes vectores de coeficientes a_i y d_i por un factor de 2 (añadiendo ceros en las posiciones de las muestras eliminadas en el proceso de submuestreo, si es necesario).
- Aplicar los correspondientes coeficientes del filtro pasa-altas $g'[n]$ o pasa-bajas $h'[m]$ inversos en el tiempo.
- La reconstrucción de la señal $x'[n]$ se obtiene por el proceso inverso utilizando el último vector de aproximación a_i y todos los vectores de detalle d_i , $i = 1, 2, \dots, i$.

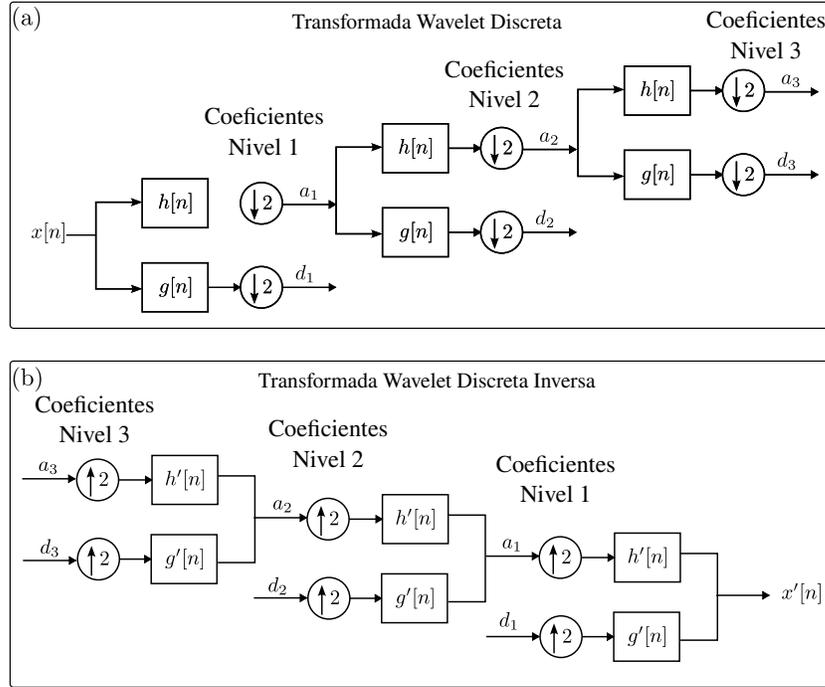


Figura 2.6: Diagrama de bloques de (a) Proceso de descomposición DWT de una señal $x[n]$. (b) Diagrama de reconstrucción DWT inversa de una señal $x'[n]$.

2.9. ANÁLISIS DE COMPONENTES PRINCIPALES

El método de Análisis de Componentes Principales PCA (del inglés *Principal Component Analysis*) es una estrategia empleada para disminuir la dimensionalidad de un conjunto de datos. Propuesto por Pearson en 1901 (Kherif y Latypova, 2020). En este enfoque, los datos originales se descomponen en magnitudes conocidas como valores propios y direcciones llamadas vectores propios. Estos elementos son utilizados para efectuar una proyección, generando así una representación nueva de la información. La proyección puede llevarse a cabo en diversos ejes, los cuales corresponden al número deseado de dimensiones y se conocen como componentes principales. En esta representación, la capacidad de representar los datos originales se organiza en orden descendente. Como resultado, las primeras componentes principales retienen la mayor cantidad de información significativa de los datos iniciales. Esta reducción en la información resulta en una simplificación tanto en la manipulación como en la visualización de los datos, al mismo tiempo que disminuye la cantidad de espacio que ocupan en la memoria.

El procedimiento del algoritmo PCA se puede resumir de la siguiente manera:

- Supongamos un conjunto de datos con vectores columna de la forma $X_n = [x_1, x_2, \dots, x_n]$.

- Calcular la media \bar{x}_i para cada uno de los vectores x_i y restarlos para cada elemento del mismo $\hat{x}_i = x_i - \bar{x}_i$. Definir $\hat{X}_n = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n]$.
- Generar la matriz de covarianza $B = \hat{X}^T \hat{X}$.
- Hallar los vectores y valores propios de B . Ordenar los valores propios en orden descendente, y aplicar el mismo orden a los vectores propios.
- Para reducir los datos originales a una dimensión menor k , seleccionar k eigenvectores y construir una matriz V_k .
- Finalmente, la nueva representación de los datos Z , en k -dimensiones, se obtiene mediante $Z = V_k^T B$.

Para una revisión completa del algoritmo, se recomienda consultar las siguientes referencias Abdi y Williams (2010) y Kherif y Latypova (2020).

2.10. INDICADORES NO LINEALES

En esta sección se presenta una breve introducción a los indicadores de energía y fractalidad de Katz utilizados en este trabajo.

2.10.1. DIMENSIÓN DE FRACTALIDAD DE KATZ

Katz introdujo este enfoque en Katz (1988), donde plantea la posibilidad de estimar la dimensión de fractalidad de una forma de onda (una curva en un plano) de tamaño N al examinar la distribución de sus puntos y las distancias entre ellos.

La expresión que define la dimensión fractal de Katz para una señal de tamaño N se encuentra en la Ecuación (2.34).

$$D_K = \frac{\log(n)}{\log\left(\frac{d}{L}\right) + \log(n)}, \quad (2.34)$$

En el contexto en el que n es igual a $N - 1$, el valor de d corresponde a la extensión plana de la curva, definida como el máximo de todas las distancias $dist(1, i)$. Aquí, la función $dist()$ calcula la distancia entre el punto inicial y los demás puntos en el caso de curvas que no se cruzan. Finalmente, el valor de L , que simboliza la longitud de la curva, se establece según lo indicado en la ecuación (2.35).

$$L = \sum_{i=1}^n dis(i, i + 1). \quad (2.35)$$

Para una serie temporal, la dimensión fractal proporciona información sobre la autosimilitud, es decir, cuánto se compone una serie de versiones más pequeñas de sí misma (Lima et al., 2021).

2.10.2. ÍNDICE DE ENERGÍA

La energía presente en una señal, denotada como E , se establece mediante la Ecuación (2.36), la cual se calcula como el espacio encerrado bajo la curva de la magnitud al cuadrado de la señal, según lo señalado por Kaiser (1990).

$$E = \sum_{k=1}^n |x(k)|^2, \quad (2.36)$$

El valor de n representa la cantidad de muestras de la señal.

Capítulo 3

METODOLOGÍA

En este capítulo se describe la metodología seguida para llevar a cabo en el presente proyecto de Tesis, el diagrama de la Figura 3.1 se muestran las etapas que se cubren en el desarrollo del sistema propuesto.

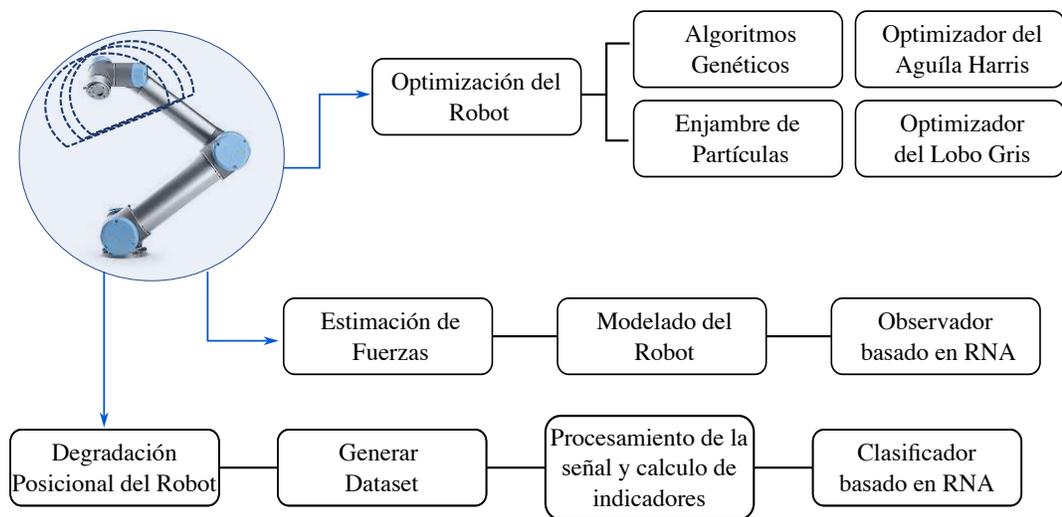


Figura 3.1: Diagrama de las etapas de la metodología.

La optimización del robot consiste en el mejoramiento del rendimiento del robot a través de la mejora de diferentes indicadores, para ello se utilizan cuatro algoritmos de optimización bio-inspirados. Para la etapa de fuerzas se usa de una red neuronal para realizar la estimación de fuerzas presentes en el robot, como una alternativa al costoso uso de sensores de fuerza. Respecto a la degradación de la posición del robot se implementa un método capaz de identificar que articulación presenta desviación, para ello se emplean método de procesamiento de señales y redes neuronales como clasificador entre los estados iniciales y finales del robot a realizar un periodo de trabajo prolongado.

3.1. OPTIMIZACIÓN DEL ROBOT

Con el fin de reducir la carga computacional de los algoritmos y el cálculo de funciones derivadas de la cinemática inversa, directa y diferencial se propone el uso de un manipulador de seis grados de libertad estándar Yaskawa Motoman en lugar de un robot colaborativo UR5. A partir del manipulador presentado en la Figura 3.2, se propone optimizar las dimensiones de los enlaces principales a_2 , d_4 y d_6 usando SLI y GCI como funciones objetivo, para ello se sigue el proceso mostrado en la Figura 3.3.

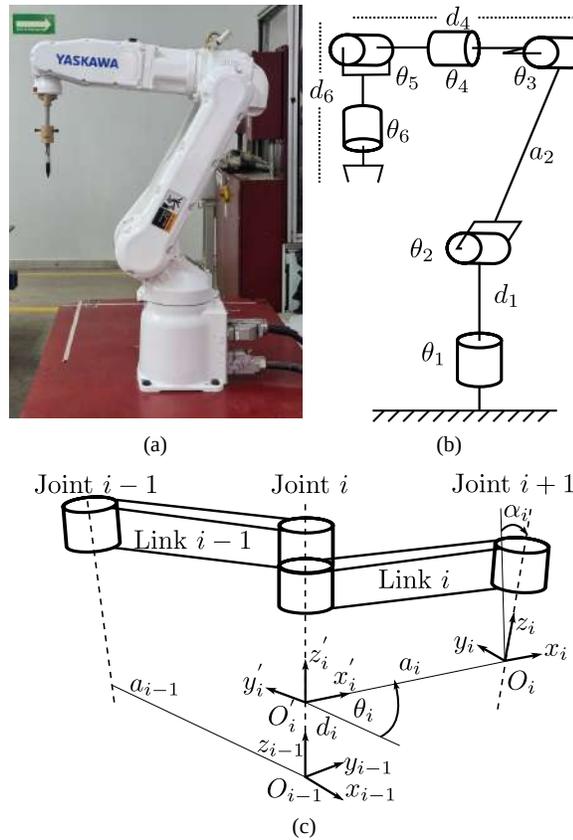


Figura 3.2: a) Robot industrial Yaskawa Motoman, b) diagrama de alambre del robot, c) representación gráfica de los parámetros de DH.

En primer lugar, se genera el vector de variables articulares q_i para cada articulación del robot con $i = 1, \dots, 6$. El tamaño del vector depende de la resolución deseada de los índices a evaluar (SLI y GCI). El vector q_i establece la posición y orientación del robot, y se genera utilizando el método de distribución beta propuesto en Cao et al. (2011). El objetivo de este método es obtener una mejor distribución de los valores articulares y mejorar el cálculo del espacio de trabajo del brazo robótico. El vector q_i se calcula una sola vez al comienzo de cada problema de optimización para que los cuatro algoritmos tengan los mismos valores conjuntos para sus respectivas k iteraciones.

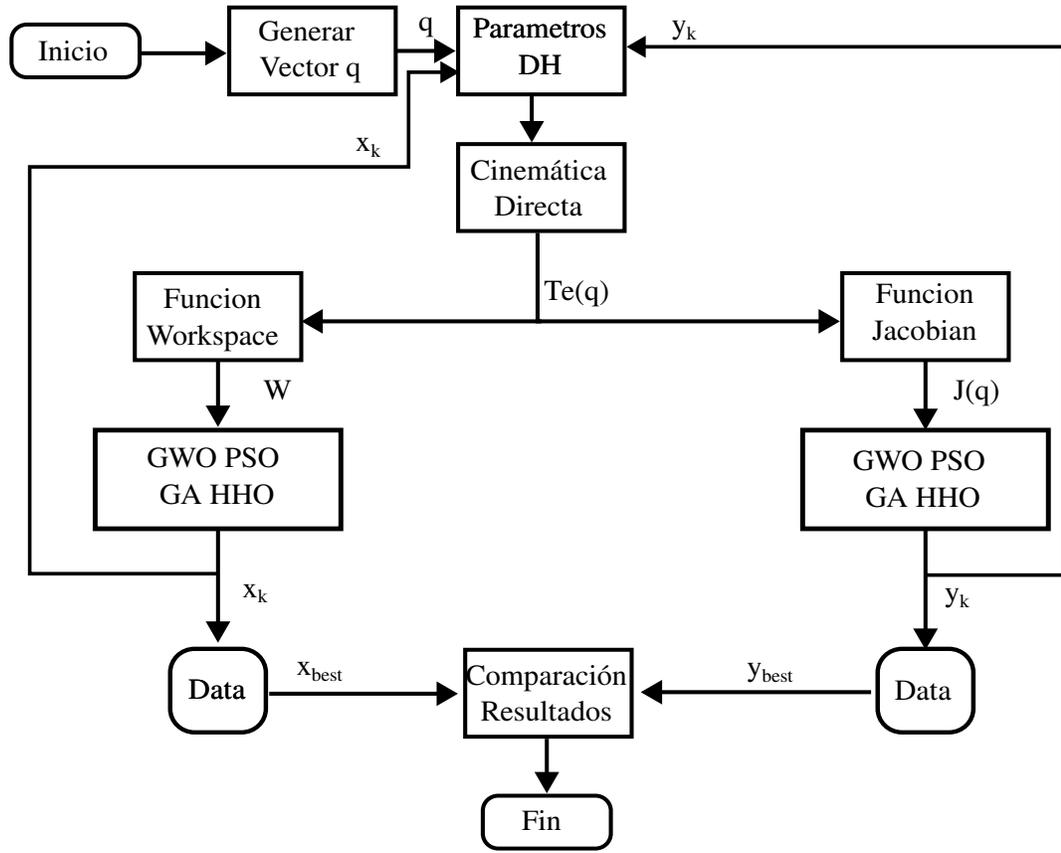


Figura 3.3: Metodología propuesta para la optimización del robot.

En segundo lugar, los parámetros DH (Tabla 2.1) se completan con el vector q_i que contiene los valores para θ_i y con los valores del vector x_k para SLI o y_k para GCI. Se supone que el vector x_k contiene los valores de solución generados por la técnica de optimización (GA, GWO, PSO o HHO). De esta forma, el vector x_k contiene los valores de las longitudes a_2 , d_4 y d_6 , las cuales terminaron de definir los parámetros del robot en todas las posiciones generadas previamente con el vector q_i .

En tercer lugar, la función de cinemática directa se calcula utilizando el método de matrices de transformación homogéneas (Siciliano et al., 2010) para todas las configuraciones generadas en el paso anterior. La función de cinemática directa genera una matriz $T_e(q)$ (3.1) compuesta por una matriz de rotación R_e (orientación del robot) y un vector de posición del robot P_e .

$$T_e(q) = \begin{bmatrix} R_e & P_e \\ 0^T & 1 \end{bmatrix} \quad (3.1)$$

Para el lado izquierdo del diagrama, con las matrices $T_e(q)$ generadas, se genera la función que calcula el espacio de trabajo del robot. Para ello se utiliza el método propuesto en Cao et al., 2011. Se basa en una nube de puntos y su posterior división en secciones horizontales y verticales a partir de las cuales se calcula el área y luego

se unen para obtener una aproximación del volumen de la nube de puntos. Para esta función W es necesario el valor del vector x_k , que contiene las longitudes del robot, y los valores de las matrices $T_e(q)$. Con la función Workspace lista, cada técnica de optimización la utiliza para calcular el SLI definido en (2.14). Este índice se utiliza como función objetivo para optimizar los algoritmos propuestos GA, PSO, GWO y HHO. Estos generan un vector de soluciones x_k con el que se repiten las operaciones de los bloques anteriores para cada iteración dentro del algoritmo de optimización hasta el número total de generaciones k .

Considere $x = [x_1, x_2, x_3]$ como el vector de solución para el problema de optimización (3.2).

$$\min_x \in [0,1, 1,0] \frac{L(x)}{\sqrt[3]{W(x)}} \quad (3.2)$$

Entonces el problema de minimización está sujeto a las siguientes restricciones no lineales (3.3) - (3.6):

$$g_1 = \frac{x_1}{x_2} \geq 1,1, \quad (3.3)$$

$$g_2 = \frac{x_1}{x_2} \leq 2,0, \quad (3.4)$$

$$g_3 = \frac{x_2}{x_3} \geq 1,1, \quad (3.5)$$

$$g_4 = \frac{x_2}{x_3} \leq 2,0. \quad (3.6)$$

Finalmente, cuando se alcanza el número de generaciones ($k = k_{max}$), los valores del vector x_k de la última iteración del algoritmo se guardan junto con los valores del tiempo y memoria ocupada durante la ejecución del algoritmo, esto se repite con cada una de las cuatro técnicas de optimización.

Para el lado derecho del diagrama, el proceso continúa a partir de la función Cinemática directa y las matrices de transformación $T_e(q)$, procediendo a realizar una función que calcula la cinemática diferencial del robot y obtiene la matriz jacobiana $J(q)$ para cada una de las configuraciones generadas en los procesos anteriores. Posteriormente, la matriz jacobiana se utiliza para operar el índice GCI definido en (2.13) y se utiliza como función objetivo para cada algoritmo de optimización. Para el lado derecho, se supone que cada técnica de optimización genera un vector de soluciones y_k , que contiene los valores de los enlaces del robot con los que se completan los parámetros DH, y se repiten los procesos mencionados anteriormente. Esta secuencia continúa hasta que se alcanza el número máximo de generaciones.

Defina $y = [y_1, y_2, y_3]$ como el vector solución para el siguiente problema de optimización (3.7):

$$\max_y \in [0,1, 1,0] \frac{1}{n_{WS}} \sum_{j=1}^{n_{WS}} \frac{1}{\kappa(y)}. \quad (3.7)$$

Al igual que el problema de minimización SLI (3.2), el problema de maximización GCI (3.7) está sujeto a restricciones no lineales equivalentes (3.8)-(3.11).

$$g_5 = \frac{y_1}{y_2} \geq 1,1, \quad (3.8)$$

$$g_6 = \frac{y_1}{y_2} \leq 2,0, \quad (3.9)$$

$$g_7 = \frac{y_2}{y_3} \geq 1,1, \quad (3.10)$$

$$g_8 = \frac{y_2}{y_3} \leq 2,0. \quad (3.11)$$

Las restricciones de desigualdad para x_k y y_k están representadas por g_i con $i = 1, \dots, 8$, los valores para estas restricciones se proponen en Kucuk y Bingul (2006). Se basan en la relación de longitud de enlace de los brazos robóticos manipuladores comerciales. La función (3.7) se basa en la ecuación (2.10), y se evalúa varias veces para cada individuo de la población. Debido al alto número de evaluaciones de una misma función, se ejecuta en paralelo, dividiendo el proceso entre los núcleos de la computadora. Luego, cada algoritmo de optimización evalúa la función GCI individualmente. El proceso del diagrama metodológico se puede resumir de la siguiente manera, para cada problema de optimización (3.2) y (3.7) en cada k -ésima iteración dentro de los algoritmos GA, PSO, GWO y HHO, una solución El vector (x_k o y_k) se genera con las longitudes de enlace optimizadas del brazo robótico a_2 , d_4 y d_6 . Cuando se alcanza el número de generaciones propuesto, cada algoritmo entrega el mejor valor de convergencia para el problema de optimización correspondiente (x_{best} y y_{best}), además del tiempo de cómputo (t_x y t_y) y la memoria (m_x y m_y) utilizada en la ejecución del algoritmo.

Finalmente, se comparan los resultados obtenidos de cada algoritmo de optimización para determinar su desempeño en el problema de diseño óptimo de brazos robóticos en serie. Los aspectos a evaluar en la comparación son el tiempo de cálculo, el número de iteraciones para obtener una solución convergente y los resultados de optimización para el robot propuesto.

Para la implementación de esta metodología se tomaron las siguientes consideraciones:

- Los cuatro algoritmos se implementan con sus valores estándar.
- A continuación se fija el valor de d_1 en 30cm.
- Se establece el rango de las variables conjuntas en radianes: q_1 y $q_5 = [-\pi/2, \pi/2]$, q_2 y $q_3 = [-\pi/3, 2\pi/3]$, finalmente para q_4 y $q_6 = [-\pi, \pi]$.
- Se utilizan un millón de puntos para calcular el volumen del espacio de trabajo del robot para el problema SLI.
- 5000 puntos se utilizan para calcular el problema GCI.

Posteriormente, los algoritmos utilizados se inicializan con los siguientes valores:

- GA: tamaño de la población: 50, generaciones: 100, tamaño del torneo: 5, tasas de mutación y cruce : 0,25, selección de torneo, cruce uniforme, mutación uniforme.
- PSO: tamaño de la población: 50, iteraciones: 100, c_1 y c_2 : 2, y w : 0,7.
- GWO: tamaño de la población: 50, y las iteraciones: 100.
- HHO: tamaño de la población: 50, iteraciones: 100, y factor de gravamen: 0,01.

La implementación de los índices de rendimiento se realiza en el lenguaje de programación Python versión 3.9, y los algoritmos de optimización se implementan desde el micro-framework de Python NiaPy para algoritmos bioinspirados (Vrbancić et al., 2018).

3.2. DEGRADACION POSICIONAL

La metodología propuesta para detectar la deterioración de la posición del robot mediante el análisis de la corriente del motor consta de siete fases, presentadas en la Figura 3.4.

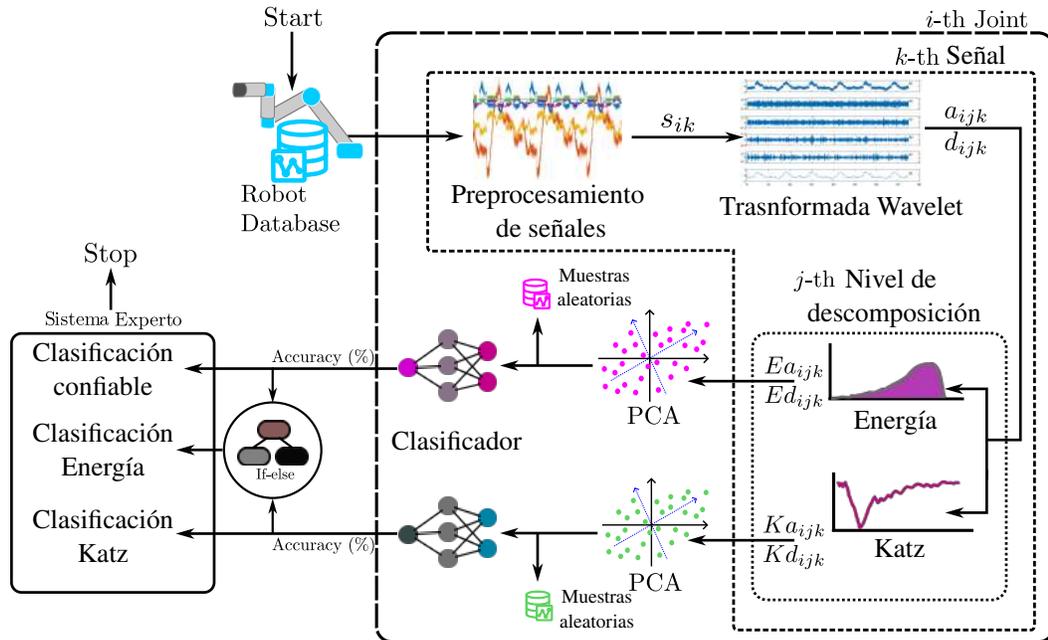


Figura 3.4: Metodología propuesta para detección de degradación posicional del robot.

La primera etapa consiste en una base de datos robótica de señales de corriente para cada articulación del motor. En la siguiente sección se ofrece una descripción de la base de datos.

3.2.1. DESCRIPCIÓN DE LA BASE DE DATOS

Para este trabajo, las señales tienen las siguientes características:

- El robot utilizado para obtener el conjunto de datos es un UR5, mostrado en la Figura 3.5a.
- Las señales de corriente de los actuadores del robot se obtuvieron a nivel de controlador, con una frecuencia de muestreo de 125 Hz.
- Las señales se captan cuando el robot realiza una trayectoria de repetición continua, que se muestra en la Figura 3.5b, con una masa de 4,5 libras en el efector final.
- El primer conjunto de datos de señales contiene tres señales cuando el robot arranca en frío (Cold Start CS) al 50% de la velocidad máxima y realiza la trayectoria de forma continua.
- El segundo conjunto de datos contiene tres señales de cuando el robot ha estado en funcionamiento continuo (Hot Operation HO), realizando la misma trayectoria durante aproximadamente 2 h.

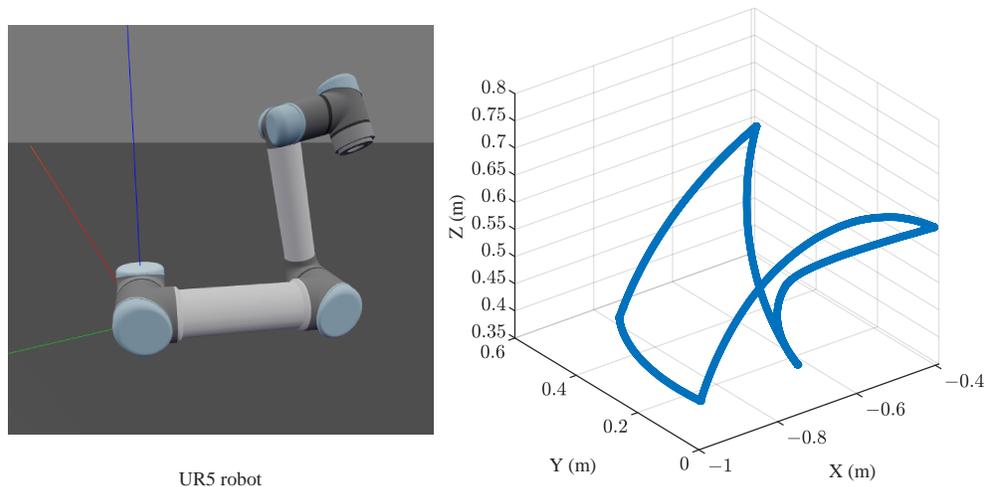


Figura 3.5: Robot UR5 y trayectoria a seguir por el robot.

3.2.2. PUESTA EXPERIMENTAL

Con la información de la base de datos lista, se propone continuar con el método de la siguiente manera:

- Las señales de la base de datos pasan por una etapa de preprocesamiento, donde se agregan diferentes niveles de ruido, para dar robustez a la metodología propuesta. Los niveles de ruido considerados son 5 dB, 10 dB, 15 dB y 20 dB, y se generan 50 señales para cada nivel de ruido. Este proceso se repite para las tres señales contenidas en cada conjunto de datos, para un total de 150 señales para cada condición.
- A continuación, cada señal s_{ik} , donde i representa el número de articulación, y k es el número de señal, se procesa con la transformada wavelet discreta, para separar la señal en diferentes bandas de frecuencia. Para las aproximaciones y detalles, denotados como $a_{i,jk}$ y $d_{i,jk}$, respectivamente, j indica el nivel de descomposición DWT utilizando Meyer discreta como la wavelet madre.
- Posteriormente, para cada nivel de aproximación y detalle de cada señal s_{ik} , se calculan las características de índice de energía ($Ea_{i,jk}$ y $Ed_{i,jk}$) y fractalidad de Katz ($Ka_{i,jk}$ y $Kd_{i,jk}$). Debido a que la DWT separa la señal original en diferentes bandas de frecuencia dependiendo del nivel de descomposición, el número de características aumenta con un nivel más alto, por ejemplo, para el nivel cuatro, se obtienen cuatro aproximaciones y cuatro detalles para un total de ocho características, a calcular para cada índice. Por esta razón, y con el fin de mantener la red neuronal lo más simple posible, cuando se calculan las aproximaciones y los detalles de cada índice para las señales $k = 150$, se ordenan en una matriz para cada característica, para reducir la dimensión de ambos indicadores utilizando el algoritmo PCA.
- En este trabajo se consideran n componentes principales, PC_n tales que la representabilidad del conjunto de datos original sea superior al 50%. Esto se consigue utilizando dos componentes principales. En esta etapa, para cada característica, se selecciona aleatoriamente un porcentaje de las muestras para cada nivel de ruido para la validación posterior de la RNA.
- A continuación, para cada índice, las matrices de componentes principales de todos los niveles de ruido se ordenan en una matriz que se utiliza como entrada para su RNA respectiva, para clasificar entre condiciones CS y HO. De este modo, se obtienen dos clasificadores RNA, y cada uno utiliza una característica distinta.
- La configuración de la red neuronal consta de una capa de entrada de dos neuronas. Se utiliza una capa oculta con cinco neuronas y una función de activación log-sigmoide. En la red neuronal se contemplan dos salidas (función de activación lineal), que indican a qué clase (CS u HO) pertenece la señal procesada.
- La selección de las muestras para las etapas de entrenamiento, validación y prueba de la red neuronal se realiza de forma aleatoria. A continuación, el conjunto de datos se divide del siguiente modo: para la etapa de entrenamiento, se utiliza el 70 % de las muestras, y para las etapas de prueba y validación, se utiliza el 15 %

para cada una. Un resumen de la configuración de la red neuronal se muestra en la Tabla 3.1.

Tabla 3.1: RNA configuration parameters.

Parameter	Value
Input layer	2
Neurons in hidden layer	5
Transfer function	log-sigmoid
Output layer	2
Data division	random
Train ratio	70 %
Validation ratio	15 %
Test ratio	15 %
Training	Levenberg–Marquardt
Epoch	5000

Finalmente, los resultados adquiridos mediante la utilización de la clasificación mediante redes neuronales se guardan para su posterior análisis. Los resultados provenientes de ambos clasificadores se combinan a través de una simple estructura condicional if-else, con el propósito de construir un sistema experto que conduzca a obtener un desenlace más confiable. Los sistemas expertos son sistemas que emulan el proceso de toma de decisiones de un especialista humano (Algehyne et al., 2022).

Las condiciones propuestas para el sistema experto son las siguientes: en caso de que ambas RNA arrojen resultados discrepantes al clasificar una misma señal, se interpreta que el análisis se encuentra en una situación incierta, lo que sugiere la repetición del análisis empleando un conjunto más amplio de señales. Por otro lado, si ambas RNA producen el mismo resultado, se procede a inspeccionar al robot en el momento en que se detecte una desviación en alguna de sus articulaciones.

3.3. ESTIMACIÓN DE FUERZA

En esta sección se describe el proceso llevado a cabo para la implementación de una red neuronal que pueda estimar los torques presentes en un robot manipulador con el fin de que en trabajos futuros pueda desarrollarse un modelo que sea capaz de estimar torques externos.

3.3.1. DESCRIPCIÓN DE LA BASE DE DATOS

Para la generación de la base de datos con la que se entrenará el modelo basado en redes neuronales LSTM se propone el uso de una trayectoria perimétrica descrita en la

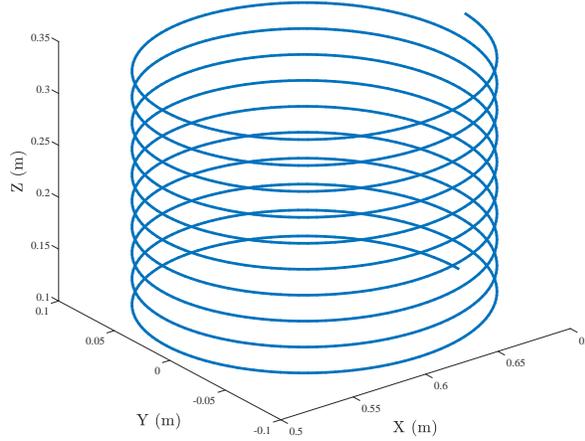


Figura 3.6: Trayectoria propuesta para la estimación de torques.

Ecuación 3.12 y mostrada en la Figura 3.6:

$$x = 0,6 + \frac{1}{10} \cos(2\pi t); y = \frac{1}{10} \sin(2\pi t); z = 0,125 \quad (3.12)$$

donde el tiempo t es un vector con 10 s de duración con una frecuencia de muestreo de 125 Hz. Con lo cual se obtienen 1250 muestras para cada uno de los ejes cartesianos, estos vectores de posición son derivados para obtener velocidad y aceleración.

Posteriormente se procede a realizar el cálculo de los torques que el robot utiliza para llevar a cabo la trayectoria propuesta, para ello se emplea la librería de (Corke y Haviland, 2021).

En las Figuras 3.7 a 3.12 se muestran las señales correspondientes para cada articulación. Estas son normalizadas para que su media sea cero y su varianza unitaria, con el fin de agilizar el entrenamiento. Finalmente se divide el tamaño de cada señal en dos partes iguales con un tamaño de 625 muestras, la primera mitad se utiliza para el entrenamiento y la segunda para la predicción.

3.3.2. IMPLEMENTACIÓN DE RED NEURONAL LSTM

La red neuronal LSTM utilizada para este trabajo tiene como entrada la posición, velocidad y aceleración de la articulación de interés y como salida se espera el torque de la misma. En este sentido se tienen seis redes neuronales, una para cada articulación. La configuración de las capas de la red se muestra en la Tabla 3.2, mientras que la configuración de la red se muestra en la Tabla 3.3. Los resultados obtenidos en la predicción, después del entrenamiento de la red son presentados en su correspondiente sección de resultados.

Tabla 3.2: Parámetros de configuración de las capas de la red LSTM.

Parámetro	Valor
sequence input layer	3
lstmLayer	100, OutputMode, sequence
fullyConnectedLayer	100
dropout layer	10 %
fullyConnectedLayer	1
Last layer	regressionLayer %

Tabla 3.3: Parámetros de configuración de la red LSTM.

Parámetro	Valor
Training	Adam
MaxEpochs	500
SequencePaddingDirection	left
GradientThreshold	1
InitialLearnRate	0.005
LearnRateSchedule	piecewise
LearnRateDropPeriod	125 %
LearnRateDropFactor	0.2 %

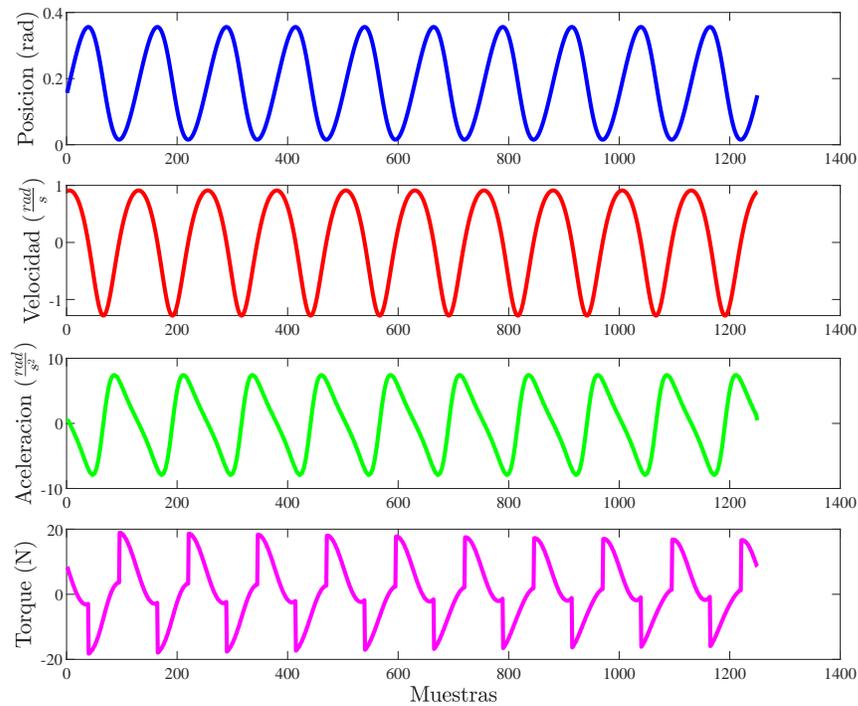


Figura 3.7: Señales simuladas de la articulación uno.

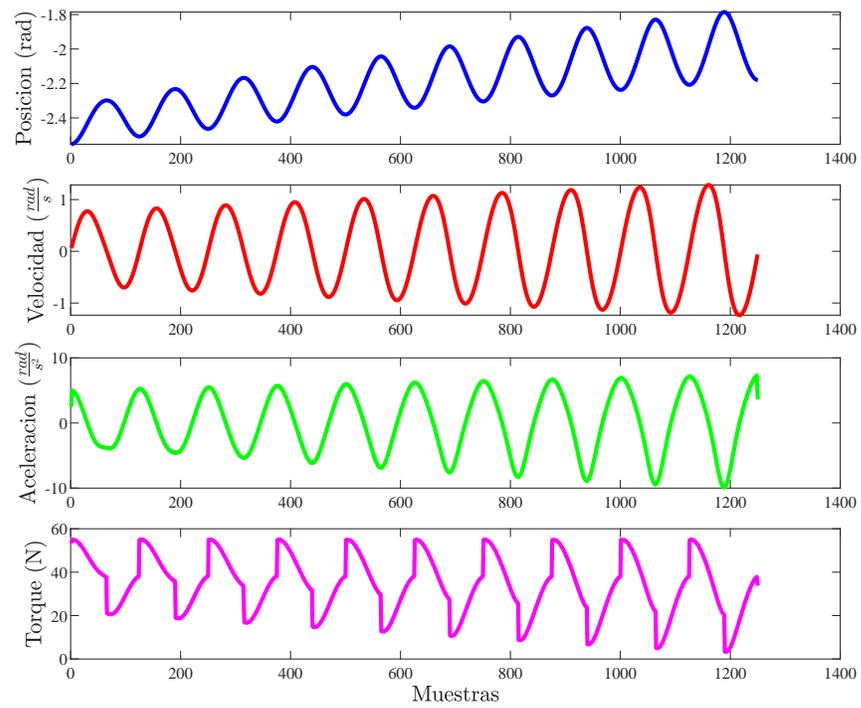


Figura 3.8: Señales simuladas de la articulación dos.

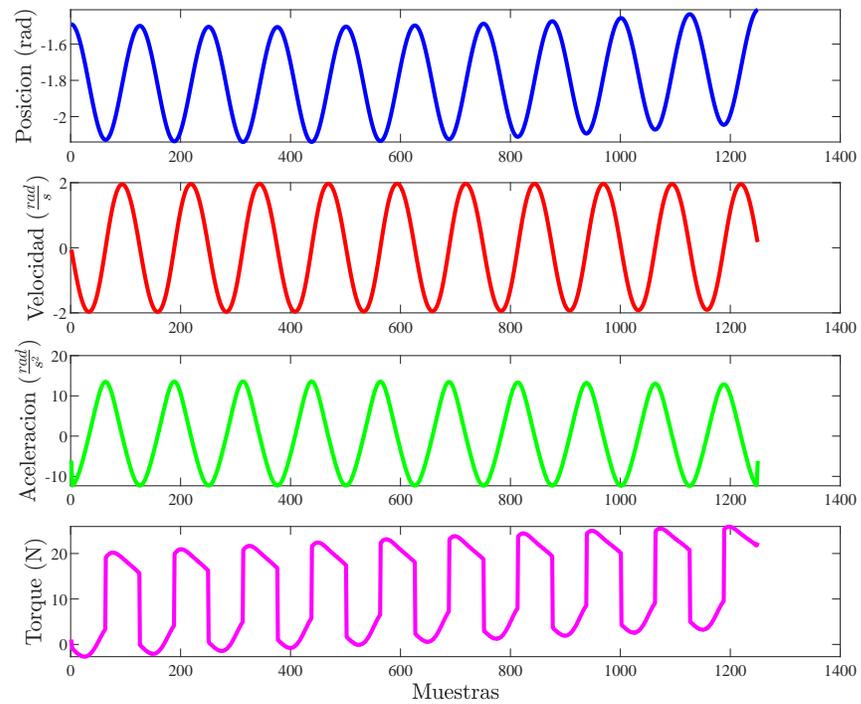


Figura 3.9: Señales simuladas de la articulación tres.

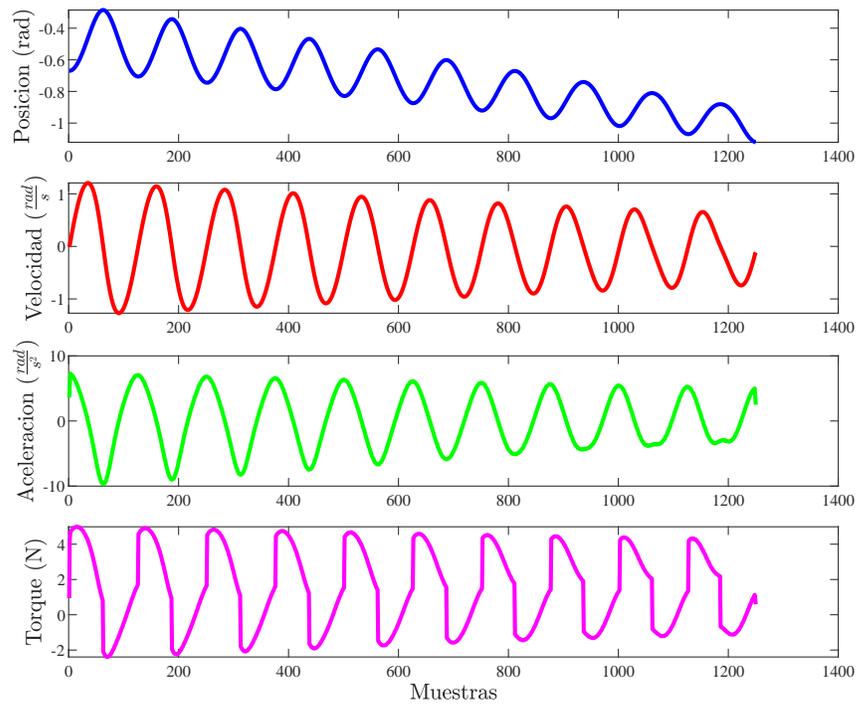


Figura 3.10: Señales simuladas de la articulación cuatro.

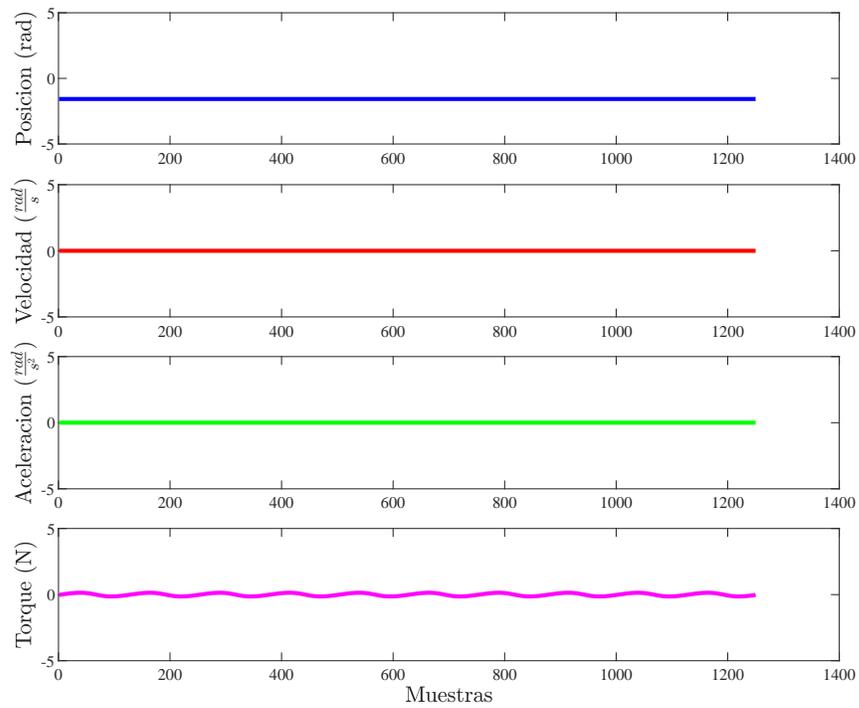


Figura 3.11: Señales simuladas de la articulación cinco.

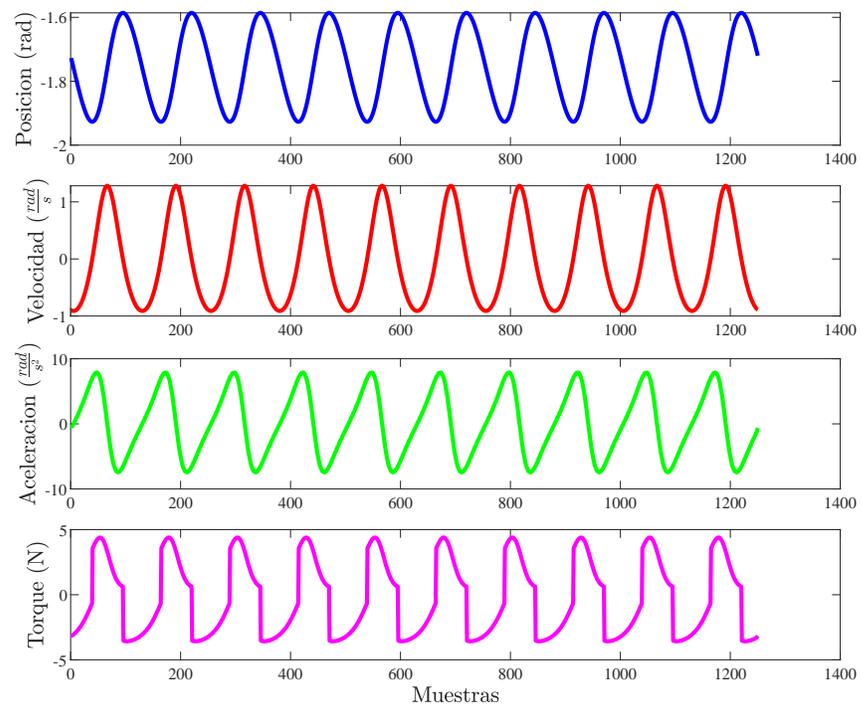


Figura 3.12: Señales simuladas de la articulación seis.

Capítulo 4

RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados y discusión de cada uno de los tópicos abordados en este proyecto.

4.1. RESULTADOS DE LA OPTIMIZACIÓN CINEMÁTICA DEL ROBOT

El objetivo de SLI es encontrar la mejor relación entre la longitud de los enlaces y un mayor espacio de trabajo del robot a través de un valor de índice bajo. En este sentido, es deseable un robot con el mayor volumen de espacio de trabajo.

Para el problema de minimización de SLI, los resultados obtenidos para las longitudes de los enlaces, el espacio de trabajo y el valor del índice se muestran en la tabla 4.1. La tabla 4.2 muestra la desviación estándar de convergencia de los algoritmos, el tiempo de cálculo y la memoria ocupada en el problema de implementación. Además, la Figura 4.1 muestra las curvas de convergencia de cada algoritmo de optimización.

De acuerdo con la Tabla 4.1, observamos que los algoritmos propuestos arrojan soluciones que tienden al límite superior del rango del vector solución. Además, observamos que todos los algoritmos convergen a valores similares. Sin embargo, el mejor resultado para el índice lo da el algoritmo HHO. El algoritmo HHO presenta una ventaja SLI sobre GA, PSO y GWO sobre 0,44 %, 0,09 % y 0,22 %, respectivamente.

En este contexto, aunque la diferencia entre los algoritmos es pequeña, los valores del espacio de trabajo tienen diferencias significativas.

El valor del espacio de trabajo más grande corresponde al robot generado con HHO (Figura 4.2); la ventaja entre los volúmenes generados con el resto de algoritmos es de un 3 % para GWO, un 2 % para PSO y un 8,6 % para GA, siendo este el volumen con menor valor. El algoritmo de convergencia más rápido es HHO, seguido de GWO, PSO y finalmente GA.

Los valores de la desviación estándar en la Tabla 4.2 ayudan a revisar el comportamiento de cada algoritmo con respecto a sus valores de convergencia. Un valor bajo muestra que la región de búsqueda del algoritmo es mejor porque el valor de la primera

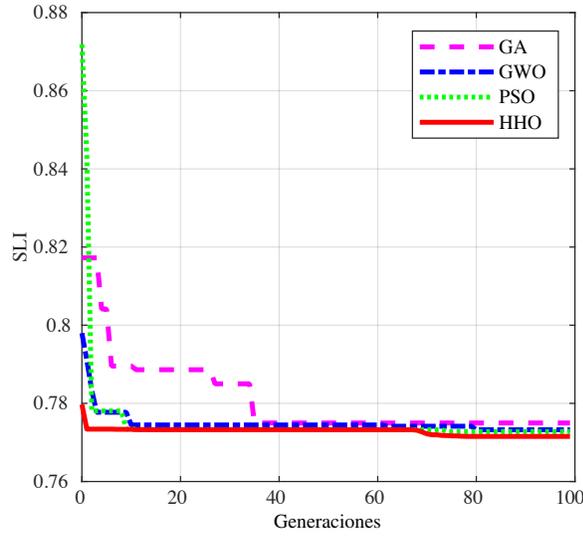


Figura 4.1: Curvas de convergencia de los algoritmos de optimización del índice SLI.

Tabla 4.1: Resultados de la optimización para el SLI.

	a_2 (cm)	d_4 (cm)	d_6 (cm)	W (cm ³)	SLI
GA	98.73	88.57	78.51	55.61×10^6	0.7749
PSO	100	90.85	81.08	59.64×10^6	0.7722
GWO	100	90.88	80.21	59.03×10^6	0.7732
HHO	100	90.90	82.64	60.89×10^6	0.7715

aproximación está cerca del valor óptimo final con menos fluctuaciones. Con base en la Tabla 4.2, observamos que la región de búsqueda del HHO es la mejor debido a que tiene la desviación estándar más baja. HHO alcanza un valor estable en la segunda aproximación, que se acerca a la mejor solución obtenida en la iteración 98. Para PSO, la desviación estándar es mayor debido al alto valor en su primera aproximación. Este algoritmo logra un valor estable en la iteración 10 y su mejor aproximación en la iteración 74. El GA es el algoritmo que converge primero, pero su asombroso comportamiento y el valor de su primera aproximación hacen que necesite 35 iteraciones para obtener un valor estable. El valor de convergencia de los otros algoritmos es mejor para la misma iteración. El algoritmo GWO alcanza un valor estable en la iteración 11 y converge en la iteración 80. Además, el valor de la convergencia a través del tiempo es entre PSO y GA. Finalmente, desde la primera iteración, observamos que HHO presenta un valor de SLI igual o mejor que el proporcionado por los demás algoritmos.

En cuanto al tiempo de cómputo, PSO presentó el mejor valor, pero no existen diferencias significativas entre este y GA o GWO. Sin embargo, entre PSO y HHO observamos un aumento de tiempo del 40%. Respecto a la memoria utilizada en cada implementación, no encontramos diferencias significativas.

Los resultados de optimización de GCI obtenidos para las longitudes de los enlaces,

Tabla 4.2: Recursos computacionales y rendimiento de los algoritmos de optimización para SLI.

	SLI_{σ}	Time (s)	Memory (MB)
GA	0.010100	18024	304
PSO	0.012077	17984	296
GWO	0.003169	18016	296
HHO	0.001030	25190	297

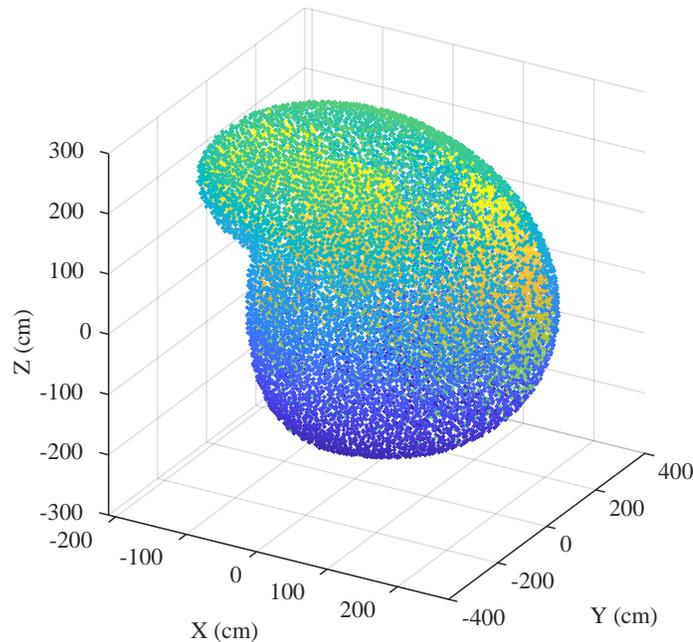


Figura 4.2: Volumen del espacio de trabajo del brazo robótico optimizado mediante HHO y SLI.

el espacio de trabajo y el valor del índice se muestran en la tabla 4.3. En la Figura 4.3, se muestran las curvas de convergencia para cada algoritmo. La tabla 4.4 muestra la desviación estándar de la convergencia de los algoritmos, el tiempo de cálculo y la memoria ocupada en el problema de maximización.

La optimización del GCI consiste en encontrar las longitudes para los eslabones del robot que entreguen el índice con un valor más cercano a la unidad, representando que el robot está mejor acondicionado. Tomando el brazo robótico de Yaskawa que se muestra en la Figura. 3.2(a) como referencia, y reemplazando los valores de DH de la tabla 2.1: $d_1 = 33$ cm, $d_4 = 40,5$ cm, $d_6 = 8$ cm, $a_1 = 8,8$ cm, $a_2 = 40$ cm, $a_3 = 4$ cm, y acotando el vector q_i en el rango de $[-\pi, \pi]$ para usar el método de distribución beta con $q_1 = [-17\pi/18, 17\pi/18]$, $q_2 = [-13\pi/36, 5\pi/6]$, $q_3 = [-32\pi/45, 5\pi/4]$, $q_5 = [-3\pi/4, 3\pi/4]$, y $q_4, q_6 = [-\pi, \pi]$.

Al implementar la misma metodología para el cálculo de GCI, obtenemos un valor

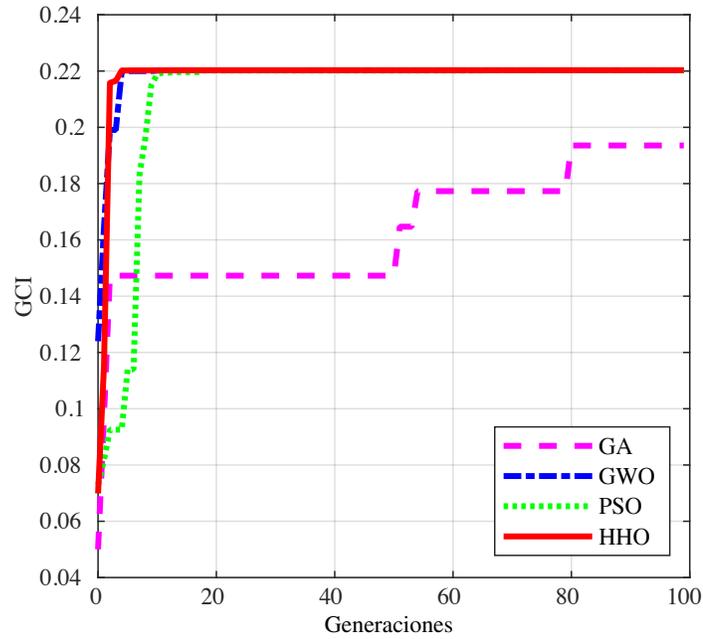


Figura 4.3: Curvas de convergencia de los algoritmos de optimización del índice GCI.

Tabla 4.3: Resultados de la optimización de la GCI.

	a_2 (cm)	d_4 (cm)	d_6 (cm)	W (cm ³)	GCI	LCI
GA	17.56	15.41	11.36	257.16×10^3	0.19354	0.8111
PSO	16.68	13.53	10	190.39×10^3	0.22026	0.9658
GWO	16.51	13.23	10	184.06×10^3	0.22028	0.9689
HHO	15.57	13.32	10	186.02×10^3	0.22029	0.9682

de 0.18258, con un valor máximo para LCI de 0.5826. De acuerdo con los valores que se muestran en la Tabla 4.4, observamos que las longitudes de los enlaces del robot tienden hacia el límite inferior del rango del vector solución. La optimización del índice implica la reducción de los enlaces de longitud y, en consecuencia, se reduce el volumen del espacio de trabajo del robot. Además, el valor promedio para GCI es 0.21359 y 0.9285 para LCI. Por tanto, observamos una mejora del 14% para GCI y del 37% para LCI respecto al brazo robótico mencionado anteriormente.

Centrándose en el rendimiento de los algoritmos de optimización, GA obtiene el valor de convergencia más alejado del valor deseado. Mientras que los algoritmos PSO, GWO y HHO convergen prácticamente al mismo valor, el mejor resultado para el índice GCI corresponde al HHO.

El robot resultante de la optimización con HHO muestra una mejora en GCI del 0,012% respecto a PSO, del 0,002% respecto a GWO y del 12% respecto a GA. A pesar de que HHO tiene el mejor valor para GCI, el mejor valor de LCI se obtiene con GWO. La razón es que mientras que para una configuración conjunta, las longitudes

Tabla 4.4: Recursos informáticos y rendimiento de los algoritmos de optimización para la GCI.

	GCI_{σ}	Time (s)	Memory (MB)
GA	0.0231	32576	95
PSO	0.0325	33374	95
GWO	0.0114	32022	94
HHO	0.0182	57025	95

de los robots pueden mejorar el valor del índice, para otras configuraciones, el índice puede verse afectado.

Tomando como referencia el volumen del espacio de trabajo obtenido con HHO, muestra una disminución del 2,3 % para PSO y del 38,2 % para GA. Sin embargo, presenta una mejora en el volumen del 1 % respecto a GWO. En este sentido, observamos que la diferencia de volumen entre GA y HHO se debe al tamaño de enlace optimizado entregado por cada algoritmo.

En cuanto a la velocidad de convergencia, al revisar la Figura 4.3, observamos que el algoritmo más lento es GA, seguido de PSO y GWO, mientras que HHO es el algoritmo de convergencia más rápido. Al revisar la desviación estándar de la Tabla 4.4, observamos que el valor mínimo corresponde al GWO, por lo que tiene la mejor región de búsqueda. Dado que alcanza un valor estable en la iteración 5, su mejor valor se obtiene en la iteración 36 y permanece constante hasta el final de la ejecución. Si bien observamos en la gráfica una rápida convergencia de HHO, su desviación estándar no es la mejor debido al bajo valor de su primera aproximación y a que hace sutiles ajustes a partir de la iteración 5, donde alcanza un valor convergente hasta llegar a su mejor resultado en la iteración 94. La desviación estándar de PSO es la más alta porque se necesitan diez iteraciones para alcanzar el valor de referencia y se ajusta con el tiempo hasta que alcanza su valor final en la iteración 97. Finalmente, el GA presenta un valor de desviación estándar entre PSO y GWO debido a una convergencia lenta y escalonada con su mejor valor en la iteración 80.

En cuanto al tiempo de cómputo, la implementación más rápida corresponde a GWO, con una ventaja del 1,7 % sobre GA, del 4,2 % para PSO y del 78 % sobre HHO. Al igual que en el problema SLI anterior, no existen diferencias significativas en la memoria al implementar técnicas GCI.

Se realiza una prueba adicional para estudiar la robustez de los algoritmos de optimización. Esta prueba consiste en realizar una evaluación de ejecución múltiple para las funciones GCI y SLI. Para las pruebas se realizaron las siguientes configuraciones de los algoritmos.

- Para reducir el tiempo de las simulaciones y probar la robustez de los algoritmos, se realiza una reducción de la población e iteraciones de los algoritmos de optimización.

- El tamaño de la población se redujo de 50 a 20 individuos (40 %) para todos los algoritmos tanto para SLI como para GCI.
- Se redujo el número de generaciones de 100 a 20 (20 %) para SLI y GCI.
- Workspace reducido de 1 millón a 300.000 posiciones (30 %) para SLI y de 5.000 a 1.500 posiciones (30 %) para GCI.
- Se realizaron 25 pruebas para cada algoritmo.

Los resultados de la prueba para SLI se muestran en la Figura 4.4(a). Observamos en el diagrama de caja la robustez de los algoritmos para 25 ejecuciones. El diagrama de caja muestra la información más relevante sobre un conjunto de datos y proporciona una descripción general de la distribución de datos y su simetría. Con base en lo anterior, se puede observar que el algoritmo PSO muestra el mejor comportamiento. Sin embargo, el mejor valor para el índice se obtiene con HHO aunque esta técnica tiene un comportamiento menos consistente que PSO. El segundo mejor rendimiento es GWO, que tiene menos variabilidad de datos que HHO y GA. En cuanto a GA, se muestra un comportamiento más disperso de los resultados, y su valor mínimo dista mucho de los obtenidos por GWO, PSO y HHO.

Los resultados obtenidos para la prueba de 25 corridas para GCI se muestran en la Figura 4.4(b). En el boxplot observamos que el mejor comportamiento de los algoritmos de optimización se obtiene con HHO, obteniendo el mejor valor para el índice GCI entre todas las pruebas seguido del valor máximo de PSO. El segundo mejor comportamiento lo obtiene GA, mientras que PSO y GWO tienen comportamientos dispersos ya que su rango de valores es más amplio que HHO. En el caso de GA y GWO, una distribución casi simétrica

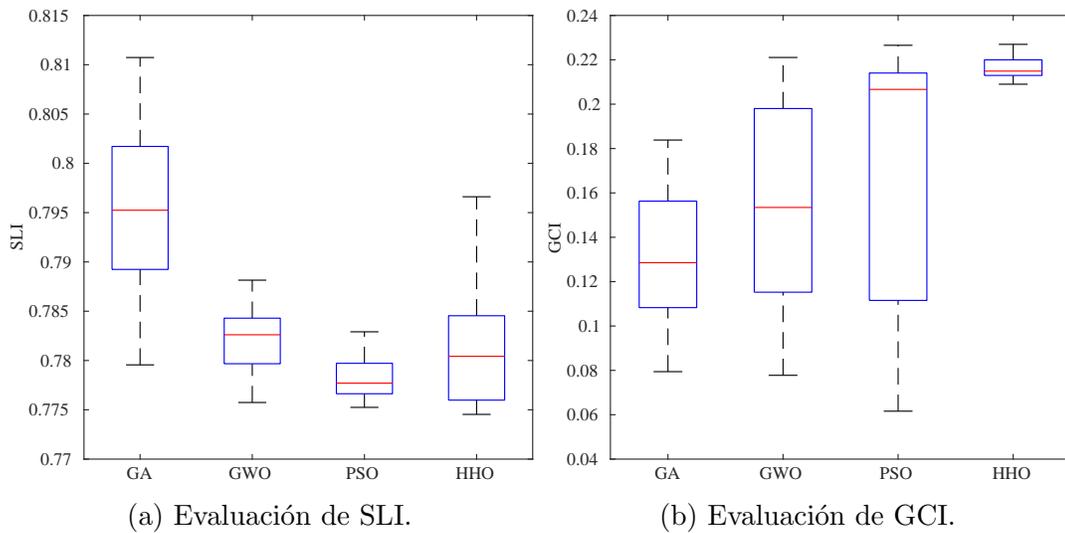


Figura 4.4: Resultado de la prueba de 25 ejecuciones de los algoritmos de optimización para el índice SLI y GCI.

4.2. RESULTADOS DE LA DEGRADACIÓN POSICIONAL DEL ROBOT

A continuación se presentan los resultados obtenidos con la metodología propuesta para los casos de media velocidad y velocidad máxima.

En la Figura 4.5 se muestran las gráficas de las señales de corriente para la articulación uno con un nivel de ruido de 5 dB. La Figura 4.5a muestra la señal CS, y la Figura 4.5b muestra la señal tras 2 h de funcionamiento continuo del robot HO. Comparando ambos gráficos, no se encuentran diferencias significativas a primera vista. Por este motivo, se realiza un análisis tiempo-frecuencia utilizando DWT, debido a la capacidad de conservar las características de la señal de las diferentes bandas de frecuencia.

El análisis se realiza de los niveles uno a seis, pero los dos últimos niveles no han ofrecido información significativa sobre el fenómeno, debido a que las bandas de frecuencia son muy estrechas, además la amplitud de estas bandas en comparación con los otros niveles es mínima, lo que indica que la mayor parte de la información se encuentra en los cuatro primeros niveles de descomposición.

Se observa que los rangos de frecuencia son bajos; para las aproximaciones, los rangos son de 0 a 0,95 Hz para el nivel seis, y de 0 a 1,95 Hz para el nivel cinco. En el caso de los detalles, los rangos son de 0,97 a 1,95 Hz para el nivel seis y de 1,95 a 3,92 Hz para el nivel cinco. Por estos motivos, y debido a la frecuencia de muestreo de 125 Hz de las señales, en este trabajo se utiliza un nivel de descomposición $j = 4$.

La Figura 4.6 muestra los resultados de aplicar el nivel cuatro de la DWT a las señales actuales presentadas en la Figura 4.5. La Figura 4.6a,b muestra los cuatro niveles

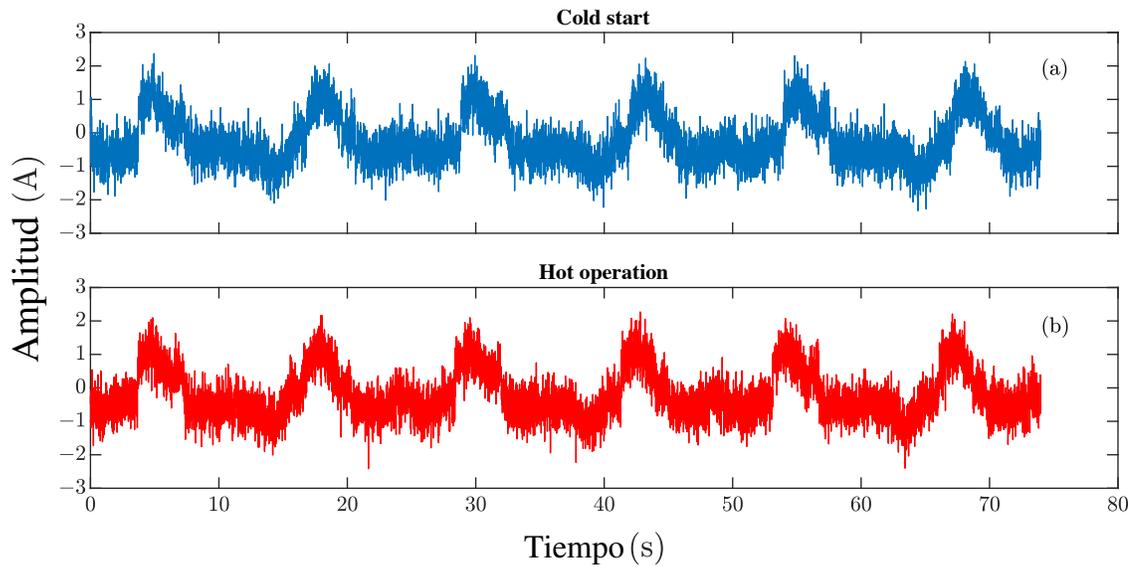


Figura 4.5: Señales actuales de la articulación uno con ruido de 5 dB: (a) señal para el estado de arranque en frío del robot; (b) señal para el estado de funcionamiento en caliente del robot.

de aproximación y detalle para la señal CS, mientras que la Figura 4.6c,d muestra los niveles de descomposición para la señal HO.

En este sentido, las aproximaciones y detalles para CS y HO no muestran un patrón visual que pueda asociarse a fenómenos de degradación de posición.

Por este motivo, se analizan todas las bandas de frecuencia mediante diferentes indicadores no lineales con el fin de encontrar patrones, posteriormente la información de cada indicador es reducida con PCA. El resultado de la reducción es utilizado para entrenar un clasificador basado en RNA para cada indicador, los resultados de los clasificadores son presentados a continuación.

Las Tablas 4.5 y 4.6 muestran la exactitud obtenida por los clasificadores para cada índice cuando las señales con todos los niveles de ruido propuestos son usadas para entrenar los clasificadores, se observa que el clasificador correspondientes a las articulaciones 5 y 6, tiene una precisión del 100 %, tanto con el indicador de energía como con el indicador de fractalidad de Katz. En cuanto a las articulaciones una a cuatro, se observa que en el caso de la velocidad máxima con el índice de energía, la clasificación obtiene resultados superiores al 97 %. En cambio, con el índice de fractalidad de Katz, se obtiene una precisión del 100 %. Por último, en el caso de media velocidad, se obtiene una precisión superior al 99 % para ambos indicadores.

La validación con señales de nivel de ruido específico (cada nivel evaluado de forma independiente), para los casos de media velocidad y máxima velocidad mostraron una precisión del 100 % en la clasificación, utilizando únicamente señales de ruido específico en ambas condiciones de velocidad. Por último, para obtener un resultado más fiable, se

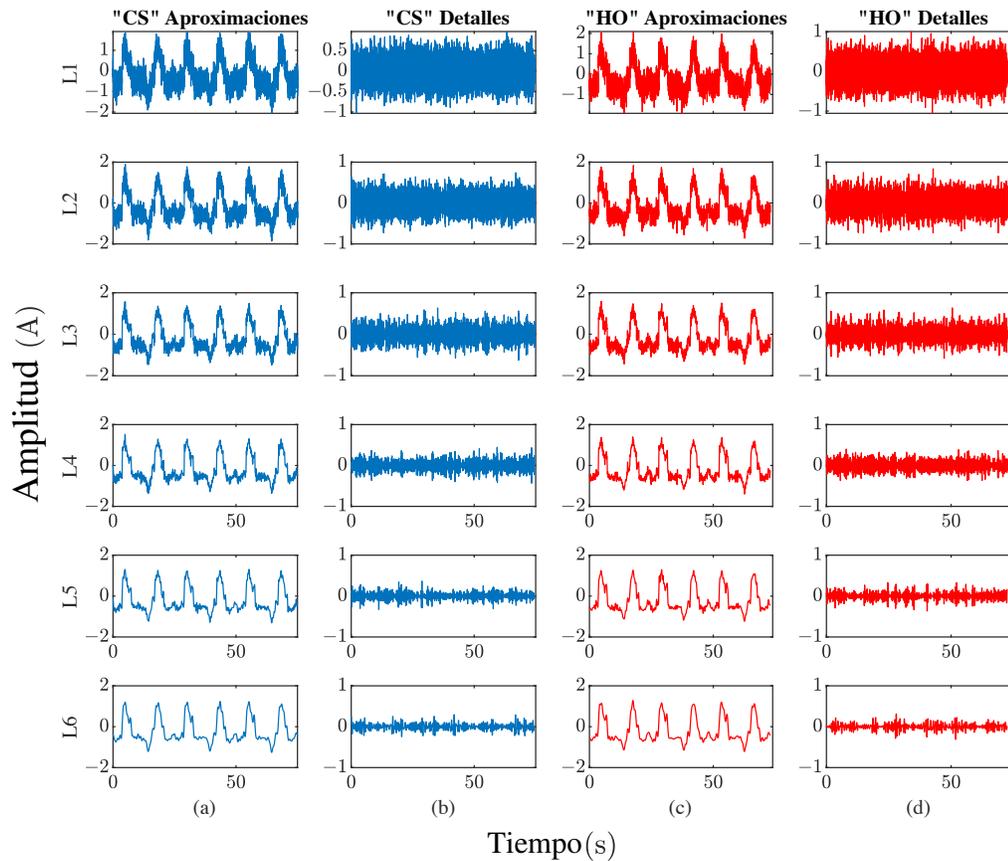


Figura 4.6: Análisis tiempo frecuencia de la primera articulación del robot, con 5 dB ruido, para el caso de media velocidad: (a,b) aproximación y detalles para el nivel de descomposición seis para el CS señal; (c,d) aproximación y detalles para el nivel de descomposición seis para la señal HO.

combinan los resultados estimados de los clasificadores FFNN, utilizando características de Katz y de energía para señales específicas de nivel de ruido, para construir un sistema experto utilizando las condiciones if-else propuestas anteriormente. Debido a que los resultados obtenidos para ambos clasificadores presentan una precisión del 100%, el sistema experto propuesto tiene una precisión del 100% para cada articulación, en diferentes condiciones de ruido y velocidad.

Tabla 4.5: Precisión de los clasificadores para los casos de velocidad media utilizando muestras de nivel de ruido combinado para la validación.

Indice	q_1	q_2	q_3	q_4	q_5	q_6
Energia	100 %	99.5 %	100 %	100 %	100 %	100 %
Katz	100 %	100 %	100 %	100 %	100 %	100 %

Tabla 4.6: Precisión de los clasificadores para los casos de velocidad máxima utilizando muestras de nivel de ruido combinado para la validación.

Indice	q_1	q_2	q_3	q_4	q_5	q_6
Energia	97.8 %	100 %	97 %	99.4 %	100 %	100 %
Katz	100 %	100 %	100 %	100 %	100 %	100 %

4.3. RESULTADOS DE LA ESTIMACIÓN DE FUERZA CON REDES NEURONALES

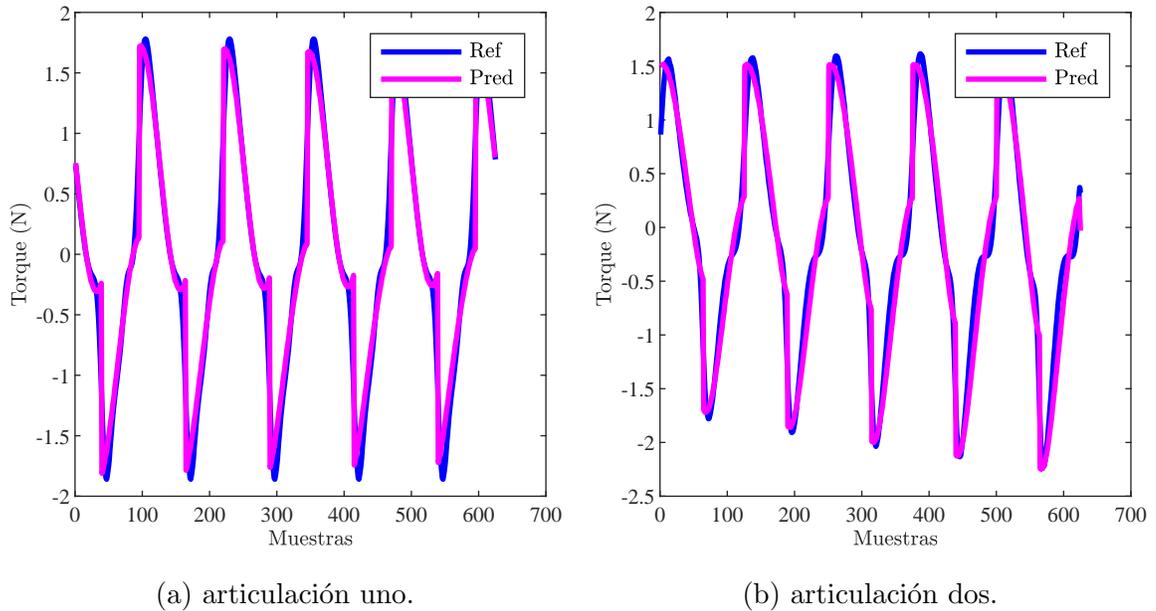
En esta sección se presentan los resultados obtenidos para la estimación de fuerzas con redes neuronales LSTM.

La implementación de la red neuronal LSTM permite estimar o aproximar sucesiones de datos de una serie de tiempo a partir de la información anterior, en este sentido el objetivo es estimar el torque para la completar la segunda mitad de la trayectoria presentada en la Figura 3.6. Para ello se entrenó una red neuronal para cada articulación del robot, y los resultados de las predicciones se observan en las Figuras 4.7 a 4.9. Para las articulaciones uno y dos presentadas en la Figura 4.7(a) y (b), se observa que los torques estimados son muy parecidos a los torques de referencia que el robot necesita para cumplir la trayectoria, en la Tabla 4.7 se presentan los valores de error RMS de ambas articulaciones donde de acuerdo a estos valores la articulación dos presenta una mejor estimación de torque.

Tabla 4.7: Error RMS obtenido entre señales de torque de referencia y las estimadas por la red LSTM.

Error	q_1	q_2	q_3	q_4	q_5	q_6
RMS	0.029	0.0102	0.5035	0.0292	0	0.0008

Por otra parte, la Figura 4.8(a) y (b), muestra las estimaciones para las articulaciones tres y cuatro respectivamente. En estas, se observa a simple vista que la articulación tres tiene una diferencia notable entre la referencia y la estimación de torque, se aprecia que la estimación sigue la forma general de la señal de referencia pero pierde los detalles en las crestas y los valles de la señal. En cuanto a la articulación cuatro, se aprecia una buena estimación del torque con cierta imprecisión al final de la trayectoria. Acorde a



(a) articulación uno.

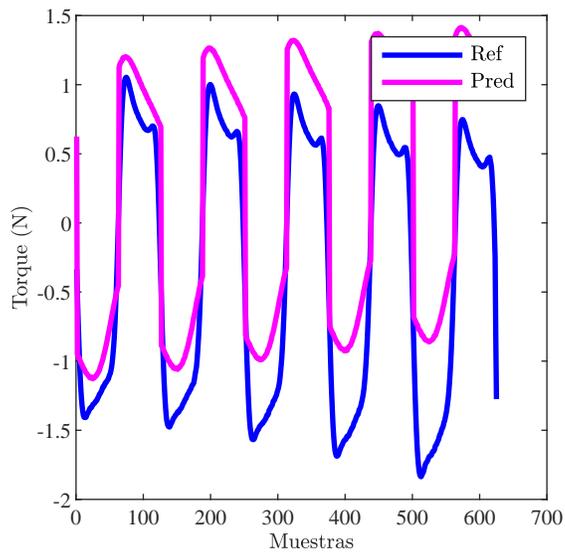
(b) articulación dos.

Figura 4.7: Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones uno y dos.

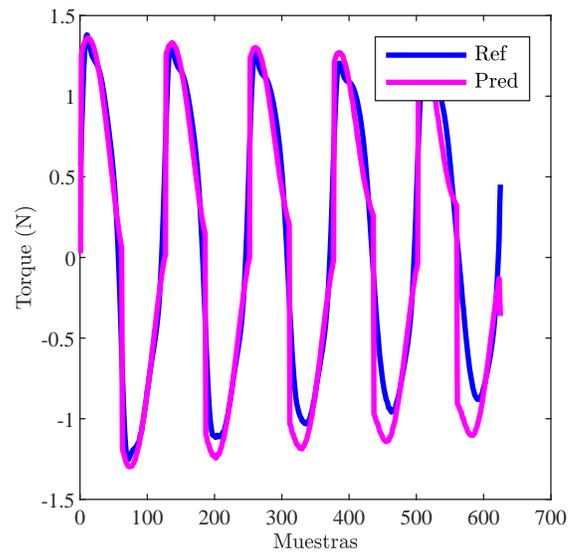
los datos de la Tabla 4.7 la articulación tres es la que presenta el valor de error más desfavorable, mientras que la articulación cuatro tiene un valor parecido a la articulación uno.

En cuanto a las articulaciones cinco y seis mostradas en la Figura 4.9(a) y (b) respectivamente, se observa que para la articulación cinco la estimación de torque es por parte de la red LSTM es prácticamente igual a la referencia. En este sentido hay que puntualizar que las señales utilizadas para el entrenamiento y validación de la red están normalizadas por lo cual los resultados de la estimación son indiferentes a la escala de la señal, en este caso la señales para la articulación cinco tienen amplitudes cercanas a al valor de cero, pero debido a que no son exactamente cero y el cálculo del modelo dinámico del robot aún estima torque para esta articulación. Finalmente para la articulación seis se observa una buena estimación de torques por parte de la red neuronal LSTM, soportada por el error RMS el cual es el menor de entre todas las articulaciones a excepción de la articulación cinco.

Los resultados obtenidos muestran las capacidades de las redes neuronales LSTM para predecir series de tiempo, en este caso se aplican a la estimación de torques de un robot manipulador, sin embargo el modelo implementado tiene algunas restricciones debido a que se entrena con tres señales a la entrada y una señal a la salida, la estimación de muestras en lazo abierto (estimar con referencia previa) y lazo cerrado (con retroalimentación de estimaciones) no han sido exploradas.

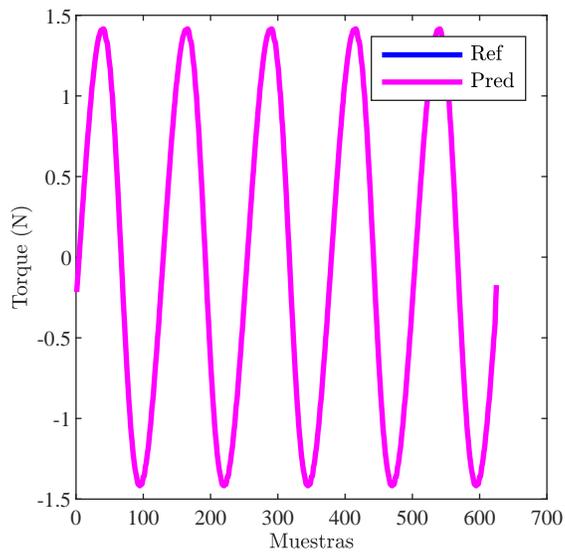


(a) articulación tres.

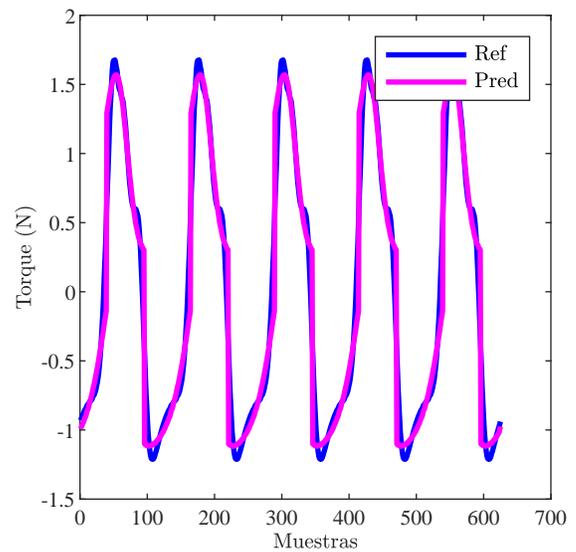


(b) articulación cuatro.

Figura 4.8: Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones tres y cuatro.



(a) articulación cinco.



(b) articulación seis.

Figura 4.9: Comparación entre los torques de referencia y los torques estimados por la red LSTM de las articulaciones cinco y seis.

Capítulo 5

CONCLUSIONES

En esta sección se presentan las conclusiones de cada uno de los tópicos abordados en este trabajo, así como una conclusión general del proyecto.

5.1. OPTIMIZACIÓN DEL ROBOT

En este estudio se proponen los algoritmos bioinspirados HHO y GWO como soluciones para el problema de diseño óptimo de un brazo robótico serial de 6 grados de libertad, utilizando los índices SLI y GCI como funciones objetivo. Se compara el rendimiento de estos algoritmos con GA y PSO.

Para el problema SLI, el uso de HHO resulta en un robot con un amplio espacio de trabajo debido a las longitudes de eslabones más grandes. Por otro lado, en el problema GCI, el robot obtenido mejora el índice del robot Yaskawa y se acerca a una configuración isotrópica, aunque su espacio de trabajo es menor que el original del robot Yaskawa Motoman-MH5LS II.

La comparación muestra que HHO, PSO y GWO obtienen resultados similares en los problemas GCI y SLI. Sin embargo, HHO destaca con una alta velocidad de convergencia, a pesar de su mayor tiempo computacional. GWO se acerca a PSO en rendimiento y velocidad de convergencia, pero supera a ambos en términos de búsqueda efectiva. GA tiene la convergencia más lenta y resultados comparables a las otras técnicas en el problema SLI, mientras que es menos convergente en GCI.

En una prueba de 25 ejecuciones para minimizar SLI, se nota que HHO tiene margen de mejora, siendo PSO el algoritmo con menor variabilidad. GWO también muestra un comportamiento aceptable comparado con HHO y GA. En el caso de maximizar GCI, HHO se destaca, seguido por GA, GWO y PSO, aunque su rango de valores es más amplio, lo que resulta en un comportamiento más disperso.

Los resultados podrían verse influenciados por la sensibilidad al tamaño de población y el número limitado de generaciones. Además, la resolución de las funciones evaluadas y la generación aleatoria de los valores del robot también pueden afectar los resultados.

Dado el rápido acercamiento al resultado óptimo y la mejor convergencia, HHO

se sugiere como la técnica preferida para el diseño óptimo de brazos robóticos. Esta comparación puede ser útil para acelerar el proceso de diseño en diferentes aplicaciones de fabricación.

En futuros trabajos, se considera implementar los algoritmos en problemas de optimización multiobjetivo y explorar otras variantes bioinspiradas para abordar el diseño óptimo de robots.

5.2. DEGRADACIÓN POSICIONAL DEL ROBOT

Los sistemas robóticos son elementos esenciales en la fabricación moderna, ya que permiten realizar diversas tareas, por ejemplo, recoger y colocar, soldar, mecanizar y pintar, etc. La exactitud del robot a la hora de posicionarse y orientarse es esencial para la precisión y la calidad; sin embargo, su exactitud en el proceso puede verse degradada por diversos factores, como las condiciones ambientales y el desgaste de los componentes mecánicos y eléctricos. En este sentido, en este trabajo se introduce una metodología de evaluación rápida de la salud posicional robótica, para determinar si existe degradación de la posición en las articulaciones de un robot industrial tras dos horas de funcionamiento continuo.

El método propuesto puede identificar, de forma automatizada, en qué parte del robot se encuentra la desviación, no requiere modelado del robot ni sensores externos para realizar las medidas, y el método es aplicable a robots de diferentes grados de libertad. La identificación temprana de la articulación desviada permite la implementación de metodologías de calibración del robot y la generación de estrategias de diagnóstico para el mantenimiento preventivo del robot, reduciendo tiempos y costes de mantenimiento, ambos recursos valiosos en el ámbito industrial. Cabe destacar que los resultados obtenidos deben considerarse preliminares, ya que la propuesta se evalúa utilizando un número reducido de ensayos de la base de datos. Por este motivo, en futuros trabajos, es importante validar la metodología propuesta con una base de datos más amplia, que además incluya información sobre diferentes fallos que afectan a la precisión posicional del robot. Además, es fundamental explorar otras técnicas de machine-learning y deep-learning y su aplicación en la generación de metodologías rápidas de evaluación de la salud posicional del robot.

5.3. ESTIMACIÓN DE FUERZAS CON REDES LSTM

La interacción del robot con su entorno y con el operador debe garantizar la seguridad del operador, en tareas cooperativas este factor es fundamental para generar confianza en la relación humano-máquina y con esto mejorar la eficiencia y el desempeño del trabajador. En este sentido, para lograr una interacción segura en tareas HRI es necesaria la estimación de fuerzas, para ello se utilizan sensores de fuerza y torque que tienen un alto costo y solo captaron señales en el lugar donde son posicionados. Para afrontar esta limitación se han propuesto esquemas de estimación de fuerza sin sensores que permiten estimar torques y momentos en todas las articulaciones del robot.

En este trabajo se propone el uso de redes neuronales LSTM para estimar las fuerzas necesarias para que el robot lleve a cabo una trayectoria simulando que se realiza una tarea cooperativa. Los resultados demuestran que las redes LSTM pueden estimar las fuerzas correctamente cuando estas tienen un perfil simple (curvas suaves sin cambios bruscos en amplitud). Los resultados obtenidos en las pruebas realizadas, deben ser considerados preliminares pues solo se contemplan las fuerzas internas del robot, además debido a la configuración de la red LSTM implementada, no es posible realizar una estimación utilizando las estimaciones pasadas como entrada (lazo cerrado), por lo cual es necesario utilizar otras magnitudes como entrada para la red, en este caso se recomienda corriente debido a su estrecha relación con el torque. Sin embargo para esto es necesario tener un modelo fiel de fricción del robot para obtener mejores resultados, este modelo puede ser estimado utilizando los algoritmos bio-inspirados presentados en este trabajo o bien redes neuronales, por lo cual este tema se explorará en próximos trabajos.

Además, en trabajos futuros se pretende considerar la opción de utilizar la red LSTM para estimar un modelo de fricción y completar la dinámica de un robot real en un esquema de control de fuerza, para realizar una predicción de torques externos usando observadores de estados basados en diferentes técnicas como teoría de control y redes neuronales.

5.4. CONCLUSIONES GENERALES

En este trabajo se abordan tres aspectos importantes que intervienen en la interacción humano-robot: optimización del robot, precisión del robot y la estimación de fuerzas. Cada uno de estos aspectos impactan directamente en la seguridad de la ejecución de tareas cooperativas. En este sentido se presenta una metodología para explorar cada uno de ellos:

- Optimización del robot: se realiza la optimización del espacio de trabajo de un robot industrial utilizando algoritmos bioinspirados, entre estos dos (GWO y HHO) que no habían sido utilizados para este fin. Los resultados demuestran que los algoritmos nuevos tienen mejor desempeño que algoritmos tradicionales como GA o PSO.
- Precisión del robot: se propone una metodología para detectar cual articulación del robot presenta desviación con base a un análisis de corrientes de un robot UR5. Se utilizan métodos de machine learning para conformar un sistema experto compuesto de dos análisis con diferentes indicadores no lineales. Esta metodología puede utilizarse para saber cuando un robot se ha descalibrado y programar operaciones de mantenimiento preventivo.
- Estimación de fuerzas: se propone el uso de redes LSTM para la estimación de fuerzas internas del robot UR5, con el fin de estudiar su viabilidad en futuras aplicaciones que contemplan la estimación de fuerzas externas en operaciones HRI. Los resultados obtenidos muestran que la estimación es acertada cuando los perfiles de fuerza son suaves.

Aunque se han tenido resultados satisfactorios en los desarrollos propuestos en este trabajo, hay que señalar que la mayoría de estos han sido llevados a cabo con simulaciones o con una cantidad de datos limitada, por lo que la validación con una mayor cantidad de señales obtenidas de robots y sensores es necesaria y considerada para realizarse en un futuro. Además debido a la importancia de la interacción humano-robot se propone abordar otras problemáticas como la comunicación humano-robot mediante reconocimiento de gestos y lenguaje, zonas seguras, seguimiento de objetos, etc.

Bibliografía

- Abdi, H., y Williams, L. J. (2010). Principal component analysis: Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), 433-459. <https://doi.org/10.1002/wics.101>
- Abeywardena, S., Yuan, Q., Tzemanaki, A., Psomopoulou, E., Droukas, L., Melhuish, C., y Dogramadzi, S. (2019). Estimation of Tool-Tissue Forces in Robot-Assisted Minimally Invasive Surgery Using Neural Networks. *Frontiers in Robotics and AI*, 6. <https://doi.org/10.3389/frobt.2019.00056>
- Algehyne, E. A., Jibril, M. L., Algehainy, N. A., Alamri, O. A., y Alzahrani, A. K. (2022). Fuzzy neural network expert system with an improved Gini index random forest-based feature importance measure algorithm for early diagnosis of breast cancer in Saudi Arabia. *Big Data and Cognitive Computing*, 6(1), 13.
- Altché, F., y de La Fortelle, A. (2017). An LSTM network for highway trajectory prediction. *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, 353-359.
- Amar, J., y Nagase, K. (2021). Genetic-algorithm-based global design optimization of tree-type robotic systems involving exponential coordinates. *Mechanical Systems and Signal Processing*, 156, 107461. <https://doi.org/10.1016/j.ymssp.2020.107461>
- Asada, H., y Slotine, J.-J. (1991). *Robot analysis and control*. John Wiley & Sons.
- Barrientos, A., et al. (2007). Fundamentos de robótica.
- Cao, Y., Lu, K., Li, X., y Zang, Y. (2011). Accurate numerical methods for computing 2d and 3d robot workspace. *International Journal of Advanced Robotic Systems*, 8(6), 76. <https://doi.org/10.5772/45686>
- Chen, L., Zhang, Y., Xue, Y., y Chen, Y. (2022). Robot Path Planning Based on Improved Particle Swarm Optimization. *2022 Power System and Green Energy Conference (PSGEC)*, 507-511. <https://doi.org/10.1109/PSGEC54663.2022.9881021>
- Choi, K., Fazekas, G., y Sandler, M. (2016). Text-based LSTM networks for automatic music composition. *arXiv preprint arXiv:1604.05358*.
- Corke, P., y Haviland, J. (2021). Not your grandmother's toolbox—the Robotics Toolbox reinvented for Python. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11357-11363.

- Corne, D. W., Reynolds, A. P., y Bonabeau, E. (2012). Swarm Intelligence. *Handbook of natural computing, 2017*(6), 1599-1622.
- Craig, J. J. (2006). *Introduction to robotics*. Pearson Educacion.
- Dereli, S. (2021). A new modified grey wolf optimization algorithm proposal for a fundamental engineering problem in robotics. *Neural Computing and Applications, 33*(21), 14119-14131. <https://doi.org/10.1007/s00521-021-06050-2>
- Desai, M., y Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clinical eHealth, 4*, 1-11.
- Filippeschi, A., Griffa, P., y Avizzano, C. A. (2021). Kinematic optimization for the design of a collaborative robot end-effector for tele-echography. *Robotics, 10*(1), 8. <https://doi.org/10.3390/robotics10010008>
- Gosselin, C., y Angeles, J. (1991). A Global Performance Index for the Kinematic Optimization of Robotic Manipulators. *Journal of Mechanical Design, 113*(3), 220-226. <https://doi.org/10.1115/1.2912772>
- Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., y Mafarja, M. (2020). Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications, 23-46*.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., y Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems, 97*, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- Houssein, E. H., Ahmad, M., Hosney, M. E., y Mazzara, M. (2021). Classification Approach for COVID-19 Gene Based on Harris Hawks Optimization. En D. Oliva, S. A. Hassan y A. Mohamed (Eds.), *Artificial Intelligence for COVID-19* (pp. 575-594). Springer International Publishing. https://doi.org/10.1007/978-3-030-69744-0_32
- Hwang, S., Kim, H., Choi, Y., Shin, K., y Han, C. (2017). Design optimization method for 7 DOF robot manipulator using performance indices. *International Journal of Precision Engineering and Manufacturing, 18*(3), 293-299. <https://doi.org/10.1007/s12541-017-0037-0>
- Issa, M., y Samn, A. (2022). Passive vehicle suspension system optimization using Harris Hawk Optimization algorithm. *Mathematics and Computers in Simulation, 191*, 328-345. <https://doi.org/10.1016/j.matcom.2021.08.016>
- Izci, D., Ekinici, S., Demirören, A., y Hedley, J. (2020). HHO algorithm based PID controller design for aircraft pitch angle control system, 1-6. <https://doi.org/10.1109/HORA49412.2020.9152897>
- Kaiser, J. (1990). On a simple algorithm to calculate the 'energy' of a signal. *International Conference on Acoustics, Speech, and Signal Processing, 381-384* vol.1.
- Katz, M. J. (1988). Fractals and the analysis of waveforms. *Computers in biology and medicine, 18*(3), 145-156.
- Kelly, R., Santibánes, V., et al. (2003). Control de movimiento de robots manipuladores.

- Kherif, F., y Latypova, A. (2020). Principal component analysis. En *Machine Learning* (pp. 209-225). Elsevier.
- Kucuk, S. (2018). Dexterous Workspace Optimization for a New Hybrid Parallel Robot Manipulator [064503]. *Journal of Mechanisms and Robotics*, 10(6). <https://doi.org/10.1115/1.4041334>
- Kucuk, S., y Bingul, Z. (2006). Comparative study of performance indices for fundamental robot manipulators. *Robotics and Autonomous Systems*, 54(7), 567-573. <https://doi.org/10.1016/j.robot.2006.04.002>
- Li, B., Bian, Z., y Wang, Z. (2022). Research on the Electric Life Prediction of Relay Contact Based on Harris Hawk Optimized. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-10. <https://doi.org/10.1109/TIM.2021.3135550>
- Lima, T. L. d. V., Filho, A. C. L., Belo, F. A., Souto, F. V., Silva, T. C. B., Mishina, K. V., y Rodrigues, M. C. (2021). Noninvasive Methods for Fault Detection and Isolation in Internal Combustion Engines Based on Chaos Analysis. *Sensors*, 21(20). <https://doi.org/10.3390/s21206925>
- Lin, G., Shan, B., Ma, Y., Tian, X., y Yu, J. (2021). Adaptive neural network command filtered backstepping impedance control for uncertain robotic manipulators with disturbance observer. *Transactions of the Institute of Measurement and Control*, 44, 799-808. <https://doi.org/10.1177/01423312211009376>
- Liu, X., Zhao, F., Ge, S., Wu, Y., y Mei, X. (2019). End-Effector Force Estimation for Flexible-Joint Robots With Global Friction Approximation Using Neural Networks. *IEEE Transactions on Industrial Informatics*, 15, 1730-1741. <https://doi.org/10.1109/TII.2018.2876724>
- Liu, Y., Li, Y., Zhuang, Z., y Song, T. (2020). Improvement of Robot Accuracy with an Optical Tracking System. *Sensors*, 20(21), 6341. <https://doi.org/10.3390/s20216341>
- Meng, Q., Li, J., Shen, H., Deng, J., y Wu, G. (2021). Kinetostatic design and development of a non-fully symmetric parallel Delta robot with one structural simplified kinematic linkage. *Mechanics Based Design of Structures and Machines*, 1-21. <https://doi.org/10.1080/15397734.2021.1937213>
- Mirjalili, S., Mirjalili, S. M., y Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Patel, S., y Sobh, T. (2015). Manipulator performance measures-a comprehensive literature survey. *Journal of Intelligent & Robotic Systems*, 77(3), 547-570. <https://doi.org/10.1007/s10846-014-0024-y>
- Peng, G., Chen, C. L. P., He, W., y Yang, C. (2020). Neural-Learning-Based Force Sensorless Admittance Control for Robots With Input Deadzone. *IEEE Transactions on Industrial Electronics*, 68, 5184-5196. <https://doi.org/10.1109/TIE.2020.2991929>
- Peng, J., Ding, S., Yang, Z., y Zhang, F. (2019). Neural Network-Based Hybrid Position/Force Tracking Control for Robotic Systems Without Velocity Measurement.

- Neural Processing Letters*, 51, 1125-1144. <https://doi.org/10.1007/s11063-019-10138-1>
- Puglisi, L. J., Saltaren, R. J., Moreno, H. A., Cárdenas, P. F., Garcia, C., y Aracil, R. (2012). Dimensional synthesis of a spherical parallel manipulator based on the evaluation of global performance indexes. *Robotics and Autonomous Systems*, 60(8), 1037-1045. <https://doi.org/https://doi.org/10.1016/j.robot.2012.05.013>
- Qiao, G. (2021). Advanced Sensing Development to Support Robot Accuracy Assessment and Improvement. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 917-922. <https://doi.org/10.1109/ICRA48506.2021.9561242>
- Qiao, G., y Garner, J. (2020). Advanced Sensing Development to Support Accuracy Assessment for Industrial Robot Systems. *Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*. <https://doi.org/10.1115/MSEC2020-8281>
- Qiao, G., Schlenoff, C., y Weiss, B. A. (2017). Quick positional health assessment for industrial robot prognostics and health management (PHM). *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1815-1820. <https://doi.org/10.1109/ICRA.2017.7989214>
- Qiao, G., y Weiss, B. A. (2018a). Monitoring, Diagnostics, and Prognostics for Robot Tool Center Accuracy Degradation. *Volume 3: Manufacturing Equipment and Systems*, V003T02A029. <https://doi.org/10.1115/MSEC2018-6603>
- Qiao, G., y Weiss, B. A. (2018b). Quick health assessment for industrial robot health degradation and the supporting advanced sensing development. *Journal of Manufacturing Systems*, 48, 51-59. <https://doi.org/10.1016/j.jmsy.2018.04.004>
- Qiao, G., y Weiss, B. A. (2019). Industrial Robot Accuracy Degradation Monitoring and Quick Health Assessment. *Journal of Manufacturing Science and Engineering*, 141(7), 071006. <https://doi.org/10.1115/1.4043649>
- Rao, S. S. (2009). Modern Methods of Optimization. En *Engineering Optimization* (pp. 693-736). John Wiley & Sons, Ltd. <https://doi.org/https://doi.org/10.1002/9780470549124.ch13>
- Roy, R. G., y Ghoshal, D. (2020). Grey wolf optimization-based second order sliding mode control for inchworm robot. *Robotica*, 38(9), 1539-1557. <https://doi.org/10.1017/S0263574719001620>
- Russo, M., Herrero, S., Altuzarra, O., y Ceccarelli, M. (2018). Kinematic analysis and multi-objective optimization of a 3-UPR parallel mechanism for a robotic leg. *Mechanism and Machine Theory*, 120, 192-202. <https://doi.org/10.1016/j.mechmachtheory.2017.10.004>
- Salisbury, J. K., y Craig, J. J. (1982). Articulated hands: Force control and kinematic issues. *The International journal of Robotics research*, 1(1), 4-17. <https://doi.org/10.1177/027836498200100102>
- Siciliano, B., Sciavicco, L., Villani, L., y Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. Springer London.

- Siu, H., Sloboda, J., McKindles, R., y Stirling, L. (2021). A Neural Network Estimation of Ankle Torques From Electromyography and Accelerometry. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, 1624-1633. <https://doi.org/10.1109/TNSRE.2021.3104761>
- Staudemeyer, R. C., y Morris, E. R. (2019). Understanding LSTM—a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.
- Sun, T., y Lian, B. (2018). Stiffness and mass optimization of parallel kinematic machine. *Mechanism and Machine Theory*, 120, 73-88. <https://doi.org/10.1016/j.mechmachtheory.2017.09.014>
- Sundararajan, D. (2016). *Discrete wavelet transform: a signal processing approach*. John Wiley & Sons.
- Taha, H. A., Yacout, S., y Birglen, L. (2021). Detection and monitoring for anomalies and degradation of a robotic arm using machine learning. En *Advances in automotive production technology—theory and application* (pp. 230-237). Springer.
- Vrbančič, G., Brezočnik, L., Mlakar, U., Fister, D., y Fister Jr., I. (2018). NiaPy: Python microframework for building nature-inspired algorithms. *Journal of Open Source Software*, 3. <https://doi.org/10.21105/joss.00613>
- Wu, M., Yang, D., y Liu, T. (2021). Harris Hawks Optimization for Solving Flexible Jobshop Scheduling Problem Considering Energy Consumption. *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*, 388-392. <https://doi.org/10.1109/ICRAE53653.2021.9657813>
- Wu, Y., Ren, B., y Chen, H. (2021). Application of Grey Wolf Optimization in Anti-Collision Vehicle. *2021 The 5th International Conference on Machine Learning and Soft Computing*, 135-140. <https://doi.org/10.1145/3453800.3453825>
- Yang, C., Peng, G., Cheng, L., Na, J., y Li, Z. (2019). Force Sensorless Admittance Control for Teleoperation of Uncertain Robot Manipulator Using Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51, 3282-3292. <https://doi.org/10.1109/TSMC.2019.2920870>
- Yang, Z., Peng, J., y Liu, Y. (2019). Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer. *Neurocomputing*, 331, 263-280. <https://doi.org/10.1016/j.neucom.2018.11.068>
- Yildirim, O., Baloglu, U. B., Tan, R.-S., Ciaccio, E. J., y Acharya, U. R. (2019). A new approach for arrhythmia classification using deep coded features and LSTM networks. *Computer methods and programs in biomedicine*, 176, 121-133.
- Zarkandi, S. (2021). Kinematic analysis and workspace optimization of a novel 4R P SP+ PS parallel manipulator. *Mechanics Based Design of Structures and Machines*, 49(1), 131-153. <https://doi.org/10.1080/15397734.2020.1725564>
- Zhao, J., Mao, X., y Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical signal processing and control*, 47, 312-323.
- Zheng, Y., Wang, Y., y Liu, J. (2022). Research on structure optimization and motion characteristics of wearable medical robotics based on Improved Particle Swarm Optimization Algorithm. *Future Generation Computer Systems*, 129, 187-198. <https://doi.org/10.1016/j.future.2021.11.021>

Publicaciones



Received 21 September 2022, accepted 8 October 2022, date of publication 14 October 2022, date of current version 21 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3214850



Kinematic Optimization of 6DOF Serial Robot Arms by Bio-Inspired Algorithms

ERVIN GALAN-URIBE¹ AND LUIS MORALES-VELAZQUEZ¹, (Member, IEEE)

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, San Juan del Río, Santiago de Querétaro 76807, Mexico

Corresponding author: Luis Morales-Velazquez (luis.moralesv@uaq.mx)

This work was supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT) under Grant 783320 and Grant FONDEC-UAQ-2021-LMV-6829.

ABSTRACT Robotic systems are essential to technological development in the industrial, medical, and aerospace sectors. Nevertheless, their use in different applications requires that the robot have the best possible execution efficiency. For this, a robot with specific optimized characteristics is necessary to impact the performance of the specific task. Different techniques have been used in robot optimization, the most widely used being genetic algorithms (GA) and particle swarm optimization (PSO). However, there are optimization algorithms with high convergence speeds inspired by animal behavior whose application in robot optimization has not been reported. In this work, bio-inspired algorithms Harris hawks optimization (HHO) and Grey wolf optimizer (GWO) are applied to a six-degree-of-freedom (6DOF) robot arm design through kinematic optimization. The lengths of the main robot links are optimized to improve the workspace volume and obtain a better-conditioned robot using the structural length index (SLI) and global condition index (GCI) as objective functions. A comparison is made between the proposed algorithms and GA and PSO regarding convergence speed, computational load, and optimality. Similar behavior has been found for HHO, GWO, and PSO compared to GA for both indexes. For the GCI problem, an average improvement of 14% was found when optimizing an industrial robot arm. Furthermore, multiple runs experiment is performed to test the robustness of the algorithms. The results show that HHO is the best technique to obtain an optimal robot design because it needs less than ten iterations to provide a better result despite its computational load.

INDEX TERMS Evolutionary computation, genetic algorithms, grey wolf optimizer, Harris hawk optimizer, particle swarm optimization, robot kinematics.

I. INTRODUCTION

Robotic systems can be described as the combination of three central systems: a mechanical, an electronic, and a computational system working together to perform the required tasks [1]. However, the mechanical system interacts directly with the environment and is subject to physical restrictions that determine the design of the robot. These limitations are resolved by optimizing robot performance indices based on physical characteristics, such as link length, workspace, degrees of freedom, mass, stiffness, and inertia. Optimization methodologies based on bio-inspired techniques are widely used to solve robot mechanical design problems.

The associate editor coordinating the review of this manuscript and approving it for publication was Shaoyong Zheng².

Different robot design methodologies have been proposed using the combination of optimization algorithms and performance indices as objective functions. Some of them report algorithms such as MultiStart to optimize manipulability index [2], minimization of dexterity index [3], and exhaustive method to optimize volume, workspace, dexterity, and static efficiency [4]. Robot design proposals based on bio-inspired algorithms like particle swarm optimization (PSO) have been reported in the literature. For example, optimizing the mass and stiffness of a five-degree-of-freedom (5DOF) parallel kinematic machine obtains a significant improvement in both parameters [5]. Also, the optimization of the kinematic structures of a hybrid parallel robot using the dexterity index to obtain a better singularity-free workspace compared to the general Stewart platform is presented in [6]. In [7] is



Article

Supervised Machine-Learning Methodology for Industrial Robot Positional Health Using Artificial Neural Networks, Discrete Wavelet Transform, and Nonlinear Indicators

Ervin Galan-Uribe , Juan P. Amezcua-Sanchez and Luis Morales-Velazquez *

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, Col. San Cayetano, San Juan del Río 76807, QRO, Mexico

* Correspondence: luis.moralesv@uaq.mx

Abstract: Robotic systems are a fundamental part of modern industrial development. In this regard, they are required for long periods, in repetitive processes that must comply with strict tolerance ranges. Hence, the positional accuracy of the robots is critical, since degradation of this can represent a considerable loss of resources. In recent years, prognosis and health management (PHM) methodologies, based on machine and deep learning, have been applied to robots, in order to diagnose and detect faults and identify the degradation of robot positional accuracy, using external measurement systems, such as lasers and cameras; however, their implementation is complex in industrial environments. In this respect, this paper proposes a method based on discrete wavelet transform, nonlinear indices, principal component analysis, and artificial neural networks, in order to detect a positional deviation in robot joints, by analyzing the currents of the actuators. The results show that the proposed methodology allows classification of the robot positional degradation with an accuracy of 100%, using its current signals. The early detection of robot positional degradation, allows the implementation of PHM strategies on time, and prevents losses in manufacturing processes.

Keywords: robot health; accuracy degradation; fault prognosis; principal component analysis; Katz fractal dimension; discrete wavelet transform



Citation: Galan-Uribe, E.; Amezcua-Sanchez, J.P.; Morales-Velazquez, L. Supervised Machine-Learning Methodology for Industrial Robot Positional Health Using Artificial Neural Networks, Discrete Wavelet Transform, and Nonlinear Indicators. *Sensors* **2023**, *23*, 3213. <https://doi.org/10.3390/s23063213>

Academic Editors: Yong Oh Lee and Mayank JHA

Received: 16 February 2023

Revised: 8 March 2023

Accepted: 14 March 2023

Published: 17 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robotic systems are an essential part of modern industrial revolution 4.0 manufacturing. Among the most common tasks for industrial robots, are pick-and-place, welding, machining parts, and painting [1]. A critical requirement, to perform these tasks with a high level of precision and quality, is the accuracy of the robot in positioning and orienting itself in the required positions, and repeating the operation continuously for long periods, with the least possible error. Ideally, once the robot has been programmed to perform a repetitive task, it should stay within its trajectory; however, in reality, various factors such as robot assembly failures, manufacturing defects in the structure, and backlash, can cause sudden or gradual degradation of the accuracy of the robot [2].

Prognostics and health management (PHM), is an engineering research area whose purpose is to reduce maintenance time to a minimum, through diagnostic monitoring and detection of anomalies and failures [3]. PHM is focused on analysis at the level of components present in a more complex system. In the case of a robotic system, the components to analyze would be the mechanical components present in the mechanical transmission stage and the electrical components. On the other hand, PHM centers on a system-level analysis, that considers factors such as the performance and operation of the system itself, and the environment in which it is located [4]. Different PHM strategies applied to robot manipulators have been developed at the component level. For example, in [5], a methodology was proposed to diagnose failures in the ballscrew of a robot, using time-frequency



Article

FPGA-Based Methodology for Detecting Positional Accuracy Degradation in Industrial Robots

Ervin Galan-Uribe , Luis Morales-Velazquez  and Roque Alfredo Osoño-Ríos 

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, Col. San Cayetano, San Juan del Río 76807, Mexico; egalan18@alumnos.uaq.mx (E.G.-U.); luis.moralesv@uaq.mx (L.M.-V.)

* Correspondence: raosonio@hspdigital.org

Abstract: Industrial processes involving manipulator robots require accurate positioning and orienting for high-quality results. Any decrease in positional accuracy can result in resource wastage. Machine learning methodologies have been proposed to analyze failures and wear in electronic and mechanical components, affecting positional accuracy. These methods are typically implemented in software for offline analysis. In this regard, this work proposes a methodology for detecting a positional deviation in the robot's joints and its implementation in a digital system of proprietary design based on a field-programmable gate array (FPGA) equipped with several developed intellectual property cores (IPcores). The method implemented in FPGA consists of the analysis of current signals from a UR5 robot using discrete wavelet transform (DWT), statistical indicators, and a neural network classifier. IPcores are developed and tested with synthetic current signals, and their effectiveness is validated using a real robot dataset. The results show that the system can classify the synthetic robot signals for joints two and three with 97% accuracy and the real robot signals for joints five and six with 100% accuracy. This system aims to be a high-speed reconfigurable tool to help detect robot precision degradation and implement timely maintenance strategies.

Keywords: FPGA; positional accuracy; discrete wavelet transform; industrial robot; degradation; neural networks



Citation: Galan-Uribe, E.; Morales-Velazquez, L.; Osoño-Ríos, R.A. FPGA-Based Methodology for Detecting Positional Accuracy Degradation in Industrial Robots. *Appl. Sci.* **2023**, *13*, 8493. <https://doi.org/10.3390/app13148493>

Academic Editor: Alessandro Di Nuovo

Received: 4 July 2023

Revised: 21 July 2023

Accepted: 21 July 2023

Published: 23 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robotic systems have revolutionized the manufacturing industry by providing unprecedented accuracy, precision, and speed. Accuracy is critical to ensure the robot can position and orient itself to the required locations and repeat the task over a long time with minimal error. One way to enhance the positioning accuracy of the robot is by focusing on the design stage. For instance, Kelaiaia et al. [1], present a comprehensive review of optimal design aspects for parallel manipulators and propose a methodology for achieving the optimal design of such manipulators. Additionally, improving robot performance indices can contribute to better accuracy. Brahmia et al. [2], propose a new dimensionless sensitivity index for Parallel Kinematic Manipulators (PKM), based on the definition of the local sensitivity index (LSI). This index aids in identifying the contribution of error sources to the positioning error of the manipulator for a given task. By reducing geometric errors in PKMs, this index can significantly enhance the positioning accuracy of the robot.

Despite their advanced capabilities, robots can still experience accuracy degradation due to several factors such as environmental conditions, assembly errors, manufacturing defects in the structure, backlash, and component wear [3]. Therefore, it is essential to monitor and maintain the accuracy of robotic systems to ensure optimal performance over time. Different factors involved in the positional degradation of industrial robots and failures caused by wear in electrical and mechanical components are frequently studied. In this regard, methods based on machine learning techniques have been proposed to identify failures, such as in [4] where the wavelet packet transform (WPT) and the hidden Markov