

**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



TESIS:

**“CONTROL DIFUSO PI DE UN EXOESQUELETO ROBÓTICO FLEXIBLE DE
RODILLAS PARA LA REHABILITACIÓN DE UN MANIQUÍ DE PIERNAS DE BEBÉ”**

**PARA OBTENER EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS EN
INGENIERÍA ELECTRÓNICA CON MENCIÓN EN AUTOMÁTICA E
INSTRUMENTACIÓN**

ELABORADO POR:

GUSTAVO ADOLFO GONZALES LÓPEZ

ASESOR:

Dr. EMIR VELA SAAVEDRA

LIMA – PERÚ

2024

DEDICATORIA

Este trabajo está dedicado a mi familia por su apoyo incondicional en todos los momentos de mi vida.

AGRADECIMIENTOS

Agradezco al CONCYTEC/PROCIENCIA por la financiación obtenida para la elaboración de esta tesis a través del contrato No. 105-2021 FONDECYT Proyectos de Investigación Aplicada y Desarrollo Tecnológico. Agradezco el apoyo brindado a mi asesor de tesis, Dr. Emir Vela por facilitar las instalaciones del Laboratorio de Micro-Biorrobótica de la Universidad de Ingeniería y Tecnología para el desarrollo de esta investigación. A mis compañeros del Laboratorio de Micro-Biorrobótica y a mis compañeros de la maestría FIEE UNI por apoyarme durante la etapa de recolección de datos y desarrollo de esta tesis.

ÍNDICE DE CONTENIDOS

RESUMEN.....	X
ABSTRACT	XI
INTRODUCCIÓN.....	1
CAPITULO I	
ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA	3
1.1. Antecedentes bibliográficos	3
1.2. Descripción de la realidad problemática.....	10
1.3. Formulación del problema.....	14
1.4. Justificación e importancia de la investigación	14
1.5. Objetivos.....	15
1.5.1. Objetivo general	15
1.5.2. Objetivos específicos	15
1.6. Hipótesis	15
1.6.1. Hipótesis principal.....	15
1.7. Variables e indicadores.....	16
1.7.1. Variables independientes e indicadores.	16
1.7.2. Variables dependientes e indicadores.....	16
1.8. Unidad de análisis.....	16
1.9. Tipo y nivel de investigación	16
1.9.1. Tipo de investigación	16
1.9.2. Nivel de investigación	17
1.10. Periodo de análisis.....	18
1.11. Fuentes de información e instrumentos utilizados	18
1.12. Técnicas de recolección y procesamiento de datos	18
CAPITULO II	
MARCO TEÓRICO Y MARCO CONCEPTUAL.....	19
2.1. El marco teórico.	19
2.1.1. Conjuntos difusos.....	19
2.1.2. Funciones de membresía.....	19
2.1.3. Variables lingüísticas.....	20
2.1.4. Control difuso.....	21

2.1.5. Módulo de fusificación.....	22
2.1.6. Módulo de defusificación.....	22
2.1.7. Base de conocimiento.....	23
2.1.8. Máquina de inferencia.....	24
2.1.9. Control PI difuso.....	26
2.1.10. Identificación de sistemas dinámicos no lineales.....	27
2.1.10.1. Redes neuronales de memoria a corto - largo plazo	27
2.1.10.2. Modelado de sistemas no lineales usando redes neuronales LSTM	28
2.1.10.3. Identificación para control.....	30
2.2. El marco conceptual.....	30
2.2.1. Tratamientos para rehabilitación robótica.....	30
2.2.2. Exoesqueletos para rehabilitación.....	31
CAPITULO III	
DESARROLLO DEL TRABAJO DE TESIS	33
3.1. Metodología de la tesis.	33
3.2. Diseño, integración e implementación del hardware y software de un exoesqueleto flexible de rodilla	34
3.2.1. Descripción del exoesqueleto flexible de rodilla	34
3.2.2. Hardware y software del exoesqueleto flexible de rodilla	37
3.3. Implementación de un control difuso PI para la ejecución de la rehabilitación de ambas piernas en un maniquí de 6 meses utilizando un exoesqueleto flexible de rodilla.....	38
3.3.1. Implementación de un controlador PI para el exoesqueleto flexible de rodilla	38
3.3.1.1. Especificaciones de control del sistema robótico.....	38
3.3.1.2. Metodología de diseño de un controlador difuso PI.....	39
3.3.1.3. Diseño preliminar del controlador	41
3.3.1.4. Identificación y ajuste de los parámetros estructurales.....	42
3.3.1.5. Cálculos de las acciones de control	43
3.3.1.6. Desempeño del controlador difuso	44
3.3.1.7. Ajuste del controlador basado en experiencia	44
3.4. Estabilidad y robustez del sistema frente a diferentes perturbaciones.....	44
3.5. Comparación del controlador difuso PI con respecto a los requisitos médicos y un controlador PI.	45
CAPITULO IV	
ANÁLISIS Y RESULTADOS	46
4.1. Resultados experimentales usando un controlador PI clásico	46
4.2. Resultados del diseño del controlador difuso	47

4.2.1. Resultados de la identificación y ajuste de los parámetros estructurales.....	47
4.2.2. Resultados del cálculo de las acciones de control del controlador difuso PI.....	51
4.2.3. Resultados del desempeño del controlador difuso PI.....	51
4.2.4. Resultados del ajuste basado en experiencia del controlador difuso PI	52
4.3. Resultados de la estabilidad y robustez frente a diferentes perturbaciones del controlador difuso PI.	54
4.4. Análisis y discusión de los resultados	56
4.5. Contrastación de la hipótesis	57
CONCLUSIONES	58
RECOMENDACIONES.....	59
BIBLIOGRAFÍA.....	60
ANEXOS.....	73
A.1 Pseudocódigo controlador difuso PI	73
A.2 Código fuente para la identificación en MATLAB.....	74
A.3 Código fuente para la adquisición, control y visualización en Python.....	77

ÍNDICE DE TABLAS

Tabla 1.1. Estado del arte de exoesqueletos de rehabilitación.....	6
Tabla 1.2. Estado del arte de exoesqueletos de rehabilitación (continuación).	7
Tabla 1.3. Estado del arte de exoesqueletos de rehabilitación (continuación).	8
Tabla 1.4. Estado del arte de exoesqueletos de rehabilitación (continuación).	9
Tabla 1.5. Variable independiente control PI difuso.	17
Tabla 1.6. Variable dependiente error del seguimiento de la trayectoria.	18
Tabla 1.7. Variable dependiente robustez del sistema.	18
Tabla 3.1. Requerimientos de la rehabilitación.....	39
Tabla 3.2. Tabla de reglas difusas.	41
Tabla 4.1. Rangos de la señal de error para el controlador PI clásico.....	47
Tabla 4.2. Señal de error para el controlador difuso PI.	54
Tabla 4.3. Presión de control para el controlador difuso PI.	54
Tabla 4.4. Valores de NRMSE para el controlador difuso PI y PI clásico.	57

ÍNDICE DE ILUSTRACIONES

Figura. 1.1 Médula espinal en un bebé con espina bífida y uno saludable.....	10
Figura. 1.2 Protuberancia ocasionada por la patología de mielomeningocele	11
Figura.1.3 Gastos médicos anuales en euros: personas con y sin espina bífida.....	12
Figura. 1.4 Exoesqueleto flexible montado en una maniquí de bebé de 6 meses.	12
Figura. 1.5 Músculo artificial neumático de vacío. (A)	13
Figura. 2.1 Funciones de membresía trapezoidal y triangular.	20
Figura. 2.2 Proceso de fusificación.	22
Figura. 2.3 Estructura de un controlador difuso clásico.....	23
Figura. 2.4 Trayectorias de fase representado sobre las reglas de control.	24
Figura. 2.5 Proceso de inferencia de Mamdani.	25
Figura. 2.6 Estructura interna de una red neuronal LSTM.....	28
Figura. 2.7 Red neuronal LSTM para regresión.	30
Figura. 2.8 Representación de un exoesqueleto rígido para rehabilitación de brazo.....	32
Figura. 2.9 Exoesqueleto Blando para rehabilitación de codo.....	32
Figura. 3.1 Metodología de la tesis.	34
Figura. 3.2 Músculo artificial neumático de presión negativa.	35
Figura. 3.3 Exoesqueleto flexible de rodilla implementado en un maniquí de 6 meses. ...	35
Figura. 3.4 Hardware implementado en el maniquí de 6 meses.....	36
Figura. 3.5 Hardware implementado en el maniquí de 6 meses.....	37
Figura. 3.6 Trayectoria de rehabilitación recomendada por el especialista.	39
Figura. 3.7 Trayectorias para la rehabilitación intensiva.....	40
Figura. 3.8 Diseño de los parámetros óptimos de un controlador difuso PI.....	40
Figura. 3.9 Configuración A del controlador difuso PI para ambas piernas.	42
Figura. 3.11 Controlador PI para ambas piernas.....	43
Figura. 3.12 Datos de posición angular y presión (entrada/salida) del sistema.	44
Figura. 3.13 Maniquí de bebé con discos metálicos.....	45
Figura. 4.1 Seguimiento de la posición angular del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.	46
Figura. 4.2 Presión de control del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.	47

Figura. 4.3 Error de la posición angular del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando diferentes masas de prueba.	48
Figura. 4.4 Proceso de entrenamiento de la red neuronal LSTM.	48
Figura. 4.5 Comportamiento de la salida real y simulada del sistema.	49
Figura. 4.6 Comportamiento de la Presión versus Posición Angular.	49
Figura. 4.7 Comportamiento del error del modelo identificado.	49
Figura. 4.8 Programa en SIMULINK del controlador difuso PI.	50
Figura. 4.9 Simulación de la trayectoria del controlador difuso PI.	50
Figura. 4.10 Simulación de la señal de control usando un control difuso PI.	51
Figura. 4.11 Superficie de control durante la simulación.	52
Figura. 4.12 Señal de error del seguimiento de la trayectoria durante la simulación.	52
Figura. 4.13 Conjuntos difusos de la derivada de la presión.	53
Figura. 4.14 Conjuntos difusos del error.	53
Figura. 4.15 Conjuntos difusos de la derivada del error.	53
Figura. 4.16 Seguimiento de la posición angular del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.	55
Figura. 4.17 Presión de control del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.	55
Figura. 4.18 Error de la posición angular del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando diferentes masas de prueba.	56

RESUMEN

Espina bífida es una malformación congénita grave que afecta a muchos infantes anualmente, causando la pérdida de la movilidad en miembros inferiores en el peor de los casos. La aplicación de exoesqueletos blandos para la asistencia y rehabilitación ha estado en apogeo en los últimos años, pero se han dejado de lado las soluciones para infantes y recién nacidos. Nuevos estudios indican que los exoesqueletos de rehabilitación blandos pueden ser efectivos en la terapia de infantes con esta condición, sobre todo si se realiza en los primeros meses de vida.

Esta tesis se ha enfocado en el diseño e implementación de un controlador PI y un controlador difuso PI que permita controlar el seguimiento de la trayectoria de rehabilitación de ambas piernas de un maniquí, que simula un bebé 6 meses en posición decúbito prono, haciendo uso de un exoesqueleto blando de rodilla. El exoesqueleto blando está confeccionado con tela de algodón y es accionado a través de músculos artificiales neumáticos de vacío, fabricados con materiales de bajo costo. Se lleva a cabo una metodología iterativa para el diseño del controlador difuso PI, que parte de la identificación de un modelo de caja negra, usando los datos de un controlador PI previamente sintonizado, y culmina con una simulación en MATLAB & SIMULINK donde obtenemos los parámetros del controlador difuso PI.

El control PI y el control difuso PI fueron evaluados experimentalmente, tomando en cuenta una trayectoria de rehabilitación de 5° a 55° para ambas piernas. Además, para evaluar la robustez del controlador frente a los cambios del sistema, se realizaron múltiples experimentos agregando masas de 40, 60, 80 y 100 gramos en cada pierna. Los resultados indican que el controlador difuso PI presenta mayor robustez frente a los cambios de masa y mejor rendimiento en el seguimiento de la trayectoria, sin embargo, cuenta con mayores oscilaciones durante la fase de extensión. En contraste el controlador PI cuenta con menor oscilación durante la fase de extensión, pero menor robustez frente a los cambios de masa del maniquí, produciendo errores de mayor magnitud.

ABSTRACT

Spina bifida is a serious congenital malformation that affects many infants each year and, in the worst cases, leads to loss of mobility in the lower limbs. The use of soft exoskeletons for support and rehabilitation has boomed in recent years, but solutions for infants and newborns are left out. New studies suggest that soft rehabilitation exoskeletons may be effective in treating infants with this condition, especially when performed in the first months of life.

This work focused on the design and implementation of a PI controller and a Fuzzy-PI controller that allow to control the rehabilitation trajectory of both legs of a dummy, simulating a 6-month-old baby in prone position, using soft knee exoskeleton. The soft exoskeleton is made of cotton fabric and powered by vacuum-pneumatic artificial muscles made of low-cost materials. For the design of Fuzzy-PI controller, an iterative methodology is carried out, starting with the identification of a black box model using data from a previously tuned PI controller and culminating with a simulation in MATLAB & SIMULINK where we obtain the parameters of Fuzzy-PI controller.

PI controller and Fuzzy-PI controller were experimentally evaluated considering a rehabilitation trajectory of 5° to 55° for both legs. Additionally, to evaluate the robustness of the controller to system changes, several experiments were conducted in which masses of 40, 60, 80, and 100 grams were added to each leg. The results suggest that Fuzzy-PI controller has greater robustness to mass changes and better trajectory tracking performance, but has larger oscillations during the extension phase. In contrast, PI controller has fewer oscillations during the extension phase but is less robust to changes in the mass of the dummy, leading to larger errors.

INTRODUCCIÓN

En la actualidad los trastornos del neurodesarrollo se sitúan alrededor del 15 % en todo el mundo [73], y a menudo vienen asociados a alteraciones y disfunciones motoras graves [84]. Dos de las patologías discapacitantes más recurrentes en infantes son conocidas como parálisis cerebral y espina bífida [25]. Espina bífida se ocasiona debido a una malformación congénita en el tubo neural que ocurre al final del primer mes del desarrollo embrionario [22]. Espina bífida ocurre en varios niveles de severidad que se incluyen dentro de dos grupos, espina bífida abierta y espina bífida oculta. Dependiendo de la gravedad del daño se desencadenan una serie de discapacidades que van desde síntomas leves hasta síntomas neurológicos complejos como discapacidades motoras, discapacidades sensoriales, disfunciones en órganos internos, escoliosis, contracturas, entre otras [75]. La atrofia muscular, la reducción de la fuerza y la limitación de los movimientos de los miembros inferiores son algunas de las principales consecuencias que se desencadenan durante la vida del paciente [68].

El entrenamiento físico y la terapia de rehabilitación genera progresos en la recuperación motora de pacientes con lesión cerebral, en gran medida debido a la plasticidad neuronal [7]. Además, es conocido que el desarrollo psicomotor del ser humano inicia desde los primeros meses de vida, a través de la generación de diversos movimientos involuntarios en las extremidades [33]. Estos beneficios se han utilizado para lograr una mejora significativa en el desarrollo psicomotor de bebés con mielomeningocele (el tipo más grave de Espina Bífida), mediante de la estimulación de la caminata a través de ejercicios controlados de bajo riesgo [46]. Por ende, la rehabilitación temprana mejora significativamente la restauración de la movilidad en niños con lesiones cerebrales congénitas [24, 63], lo que es extrapolable a infantes y recién nacidos.

Los exoesqueletos de rehabilitación han sido utilizados con mucho éxito en una variedad de aplicaciones médicas alrededor del mundo [31, 64, 88]. Por muchos años el enfoque de diseño de los exoesqueletos estuvo dominado por la rigidez y el sobrepeso, dejando de lado la comodidad y la ligereza que permiten mayor libertad al paciente. Estas

capacidades han sido exploradas con la introducción de los exoesqueletos blandos o Soft Exoskeletons [14]. Los exoesqueletos blandos integran nuevos materiales en la fabricación de actuadores, mecanismos, y estructuras para dar mayor flexibilidad al conjunto [67]. En los últimos años se ha incrementado el número de publicaciones relacionadas a exoesqueletos flexibles de miembros inferiores [42], debido a que cuentan con cualidades de comodidad y seguridad que los hacen singulares frente a los exoesqueletos rígidos.

Los exoesqueletos blandos transmiten fuerzas motoras a las articulaciones del usuario a través de diferentes mecanismos de actuación, como actuadores blandos impulsados por fluidos [57], transmisiones tendón-vaina [91], cables [19], actuadores de aleación con memoria de forma [38], series de actuadores elásticos [4], siendo los músculos artificiales impulsados por vacío uno de los sistemas más prometedores [52]. Los VPAM se contraen en lugar de expandirse cuando se activan porque su presión operativa es menor que la presión atmosférica. Debido a las presiones operativas reducidas, los VPAM exhiben un mayor nivel de seguridad en la interacción entre humanos y robots, lo que los hace adecuados para aplicaciones médicas en bebés y recién nacidos.

El desarrollo de dispositivos de rehabilitación y asistencia de extremidades se ha incrementado en los últimos años, enfocándose ampliamente en jóvenes y adultos [32, 69, 71], así mismo se ha tenido una menor consideración en infantes [21], sobre todo durante el primer año de vida. Por el momento la mayor parte de los desarrollos dirigidos al área pediátrica se han enfocado en exoesqueletos rígidos [16, 29, 41] dejando de lado las ventajas otorgadas por los exoesqueletos flexibles.

La rehabilitación robótica de miembros inferiores en infantes representa un reto debido a que hasta la fecha no se han desarrollado dispositivos que estén al nivel de los requerimientos médicos de la aplicación. A pesar de ello, una hipótesis es que los exoesqueletos robóticos blandos reúnen las principales características de seguridad y robustez para ser aplicados en la rehabilitación física de infantes [26]. Un estudio previo muestra las ventajas de aplicar un Controlador Difuso PI durante la flexión/extensión de una pierna, en dos maniquís que emulan a bebés de 0 y 6 meses [8], aunque se requiere comprobar si se mantiene el mismo comportamiento al ejecutar la rehabilitación cíclica de ambas piernas del maniquí. En la presente tesis de maestría se continua con la investigación y se implementa un Controlador PI y un Controlador Difuso PI para realizar la terapia de rehabilitación pasiva de flexión/extensión de ambas piernas de un maniquí de bebé de 6 meses haciendo uso de un exoesqueleto robótico blando.

Se ha desarrollado una metodología de diseño que considera la identificación de un modelo experimental basado en redes neuronales LSTM, la simulación y el ajuste de los parámetros del controlador difuso PI, la implementación del controlador difuso PI en una tarjeta Raspberry Pi, y la comprobación de la robustez frente a diferentes cambios en las masas de ambas piernas. Este trabajo está dirigido fundamentalmente a comprobar si es posible utilizar controladores PI o difusos PI en la rehabilitación de bebés afectados con la patología de espina bífida.

CAPITULO I

ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA

En este capítulo se llevan a cabo los antecedentes bibliográficos, descripción de la realidad problemática, la justificación e importancia de la investigación, la formulación del problema, las hipótesis, las variables e indicadores, las unidades de análisis, el tipo y nivel de la investigación, el periodo de análisis, las fuentes de información e instrumentos utilizados, y finalmente las técnicas de recolección y procesamiento de datos.

1.1. Antecedentes bibliográficos

El exoesqueleto que proponemos en la presente tesis de maestría es de tipo blando, debido a que los materiales de fabricación y los sistemas de actuación son flexibles, además es un exoesqueleto de miembros inferiores ya que el foco de aplicación es la rodilla. El exoesqueleto está orientado a la rehabilitación pasiva de las rodillas en bebés de 6 meses con diagnóstico de espina bífida o parálisis cerebral.

Los exoesqueletos de rehabilitación están dirigidos a la asistencia durante la marcha, el apoyo en las actividades de la vida cotidiana, o la ejecución de terapias de rehabilitación pasivas y activas [104]. Existen muchas maneras de agrupar a los exoesqueletos. En la literatura se han considerado características como los grados de libertad (articulación simple o múltiple), el tipo de actuador (eléctrico, neumático, hidráulico, etc.), la articulación involucrada (rodilla, cadera, codo, etc.), o el tipo de aplicación (médica, militar, asistencia, etc.).

En [31], los autores han clasificado a los exoesqueletos de miembros superiores en dos categorías, los exoesqueletos de asistencia y los exoesqueletos de rehabilitación médica. Esta clasificación se aplica de forma similar en exoesqueletos de miembros inferiores. Los exoesqueletos de asistencia están dirigidos a la amplificación o incremento de alguna capacidad motora durante actividades físicamente agotadoras [98], lo que conlleva el monitoreo de la intención de movimiento del usuario y la ejecución de una fuerza equivalente a la intención medida. Por otro lado, los exoesqueletos de rehabilitación médica son utilizados para la ejecución de terapias de rehabilitación pasivas, activas o la

combinación de ambas [62]. En una terapia de rehabilitación activa, el grado de asistencia del dispositivo robótico depende del nivel de daño que haya alcanzado el paciente.

Muchas de las aplicaciones han sido orientadas a la asistencia activa durante la marcha utilizando exoesqueletos rígidos impulsados con motores de DC. Debido a las no linealidades e incertidumbres durante las diferentes fases de la marcha, algunos autores han considerado la aplicación de técnicas de control por modo deslizante que introducen un observador de estado extendido como forma de compensar las perturbaciones globales del sistema [9, 11, 49]. Otros autores han considerado combinar las capacidades de aproximación de las redes neuronales de función de base radial (RBFNN) con las cualidades de rechazo de perturbaciones del control por modo deslizante clásico [101]. A pesar de que RBFNN permitió la integración de un esquema de aprendizaje adaptativo, eliminando la necesidad de un modelo matemático basado en parámetros, se notó una gran fluctuación en el momento inicial, por ende, el autor recomendó mayor estudio y perfeccionamiento a futuro.

En los casos más graves, los daños del sistema nervioso ocasionan una pérdida considerable de la fuerza del paciente, por ende, se precisa una asistencia absoluta durante las etapas más tempranas de la recuperación (rehabilitación pasiva). La rehabilitación robótica pasiva se desarrolla en ambientes con condiciones controladas y establecidas por un especialista, lo que significa que se realizan movimientos que siguen trayectorias rígidas. La rehabilitación pasiva precisa de un controlador de posición, que resulta ser del tipo PID en la mayoría de los casos. Esto sucede generalmente porque no se requiere un posicionamiento preciso de la articulación, debido a que la estructura flexible del exoesqueleto agrega esa ligera incertidumbre de igual forma [5]. A pesar de ello, se ha extendido el uso de técnicas no lineales para el control de posición en exoesqueletos de rehabilitación [51, 55, 103]. Este tipo de controladores son efectivos para combatir las perturbaciones e incertezas de un entorno complejo, pero uno de los inconvenientes encontrados parte de la necesidad de conocer el modelo matemático del sistema, lo que muchas veces resulta impráctico, sobre todo cuando estamos frente a dinámicas no lineales como las observadas en algunos músculos artificiales neumáticos y exoesqueletos flexibles.

Algunas publicaciones resaltan la utilidad de RBFNN para mejorar la adaptabilidad del sistema ante perturbaciones e incertezas encontradas durante la operación [18, 35, 82], sin considerar que las redes neuronales requieren el ajuste de un gran número de parámetros. Además, la estructura y el tamaño de la red toman mucha relevancia en la convergencia, todo esto lleva a realizar procedimientos de entrenamiento bastante

extensos y costosos computacionalmente, que muchas veces solo permite que el controlador opere en una banda limitada [86].

Una amplia variedad de controladores difusos ha sido aplicada de manera efectiva en exoesqueletos robóticos rígidos [12, 47, 83, 85]. Un tipo de controlador difuso que emula la estructura lineal de un controlador PID ha mostrado un buen rendimiento cuando se aplicó durante el seguimiento de las trayectorias de un robot manipulador [13]. El controlador difuso PID ha mostrado mejor rendimiento que el clásico controlador PID cuando se aplica en exoesqueletos de miembros inferiores, sobre todo si se complementa con técnicas de optimización global de parámetros [2, 66, 77].

En las últimas décadas, ha tomado lugar un extenso desarrollo de investigaciones relacionadas a exoesqueletos blandos con el objetivo de solucionar los problemas de incomodidad y peso encontrados en los exoesqueletos rígidos. Muchos esfuerzos se han dirigido al desarrollo de diferentes mecanismos blandos y sus respectivas estrategias de control [6, 19, 45, 58, 94]. Investigaciones han hecho énfasis en la regulación de la impedancia, torque, posición, admitancia o presión de exoesqueletos blandos, a través de estrategias de control basadas en el control PID [36, 54, 92, 93].

En [92], se propone una estrategia de control pasiva y activa para un exoesqueleto blando de rodilla. Un control PID difuso fue utilizado en el control de la posición durante la rehabilitación pasiva. Por otro lado, las capacidades adaptativas de las redes neuronales se aplicaron al control de la impedancia durante la rehabilitación activa del paciente. Los resultados indican que el control PID difuso tiene una respuesta más rápida, un menor error en las trayectorias y una mayor precisión en el control en comparación con el control PID clásico. En [96], se propone el diseño y control de un exoesqueleto blando para asistir en la flexión de la cadera y el plantar del tobillo de ambas piernas durante la marcha. La estrategia aplicada se basa en un control iterativo PD cuyos parámetros fueron ajustados experimentalmente. El dispositivo fue probado en adultos tomando en cuenta los niveles de fatiga encontrados en los diferentes músculos empleados en la caminata. Los niveles de fatiga encontrados en los músculos del paciente indican que el exoesqueleto reduce en un 14.69%, 6.66%, 8.15% y 17.77% el esfuerzo durante la caminata. En [100], los autores han diseñado un actuador rotatorio flexible accionado por vacío para impulsar un exoesqueleto blando de rodilla, con el objetivo de generar una asistencia activa de torque durante la caminata y reducir el gasto metabólico del usuario. El actuador propuesto presenta un comportamiento lineal y tiene la capacidad de operar bajo presión negativa y positiva, logrando generar movimientos de flexión y extensión rotacionales. Se utiliza un control PID para regular la presión de operación. Los resultados muestran que el exoesqueleto es capaz de reducir el costo metabólico durante la marcha.

Tabla 1.1. Estado del arte de exoesqueletos de rehabilitación.

AÑO	Exoesqueleto	TIPO	ESTRATEGIA	ACTUADOR	CONTEXTO
[49] – 2022	Cadera y rodilla	Rígido	Extended State Observer Based Nonlinear Terminal Sliding Mode Control with Feedforward Compensation	Motor CD	Marcha en Adultos
[11] – 2019	Cadera y rodilla	Rígido	Extended State Observer Based Sliding Mode Control	Músculos neumáticos artificiales de presión positiva	Marcha en Adultos
[9] – 2019	Miembros inferiores con 7 grados de libertad	Rígido	Active Disturbance Rejection with Fast Terminal Sliding Mode Control	Motor CD	Marcha en Adultos
[95] – 2019	Miembros inferiores	Rígido	Neural Network Control	-	Adultos
[89] – 2019	Codo	Blando	Neural network enhanced torque estimation control	Actuador elástico basado en servomotores	Adultos
[102] – 2018	Miembros inferiores de 10 grados de libertad	Rígido	Model free based neural network control with time delay estimation	Motor CD	-

Tabla 1.2. Estado del arte de exoesqueletos de rehabilitación (continuación).

AÑO	EXOESQUELETO	TIPO	ESTRATEGIA	ACTUADOR	CONTEXTO
[70] – 2019	Miembros inferiores de 7 grados de libertad	Rígido	Adaptive Neural Network Fast Fractional Sliding Mode Control	Motor CD	Adultos
[87] – 2021	Miembros inferiores	Rígido	PD Controller Based on Sliding Mode RBFNN	-	Adultos
[99] – 2021	Miembro superior	Rígido	Sliding Mode Control Algorithm Based on RBFNN	-	Pacientes con accidente cerebrovascular
[34] – 2018	Miembro inferior de 12 grados de libertad	Rígido	Adaptive computed torque control based on RBFNN	Motor CD	Marcha en adultos
[101] – 2020	Cadera y rodilla	Rígido	RBFNN Based Adaptive Sliding Mode Control	Servomotores	Adultos con parálisis cerebral discinética
[40] – 2015	Miembros inferiores con 10 grados de libertad	Rígido	Virtual Torque Control using RBF Network Compensation	Motor CD	Marcha en Adultos
[82] – 2020	Miembros superiores de tres grados de libertad	Rígido	Nonsingular Fast Terminal Sliding Mode Control And RBFNN	Motor CD	-

Tabla 1.3. Estado del arte de exoesqueletos de rehabilitación (continuación).

AÑO	EXOESQUELETO	TIPO	ESTRATEGIA	ACTUADOR	CONTEXTO
[18] – 2015	Rodilla	Rígido	A Radial Basis Function Neural Network Adaptive Controller	Motor CD sin escobillas	Rehabilitación de rodilla en adultos
[35] – 2022	Miembros inferiores de 7 grados de libertad	Rígido	Radial Basis Function Based Controller	-	Rehabilitación de cadera, rodilla y tobillo
[93] – 2019	Codo	Blando	PID Based Admittance Control	Mecanismo sobreactuado basado en un motor CD	Asistencia en la flexión/extensión del codo aplicado a adultos
[36] – 2019	Codo	Blando	PID control	Actuador de tendón basado en un motor CD	Asistencia en la flexión/extensión del codo aplicado a adultos
[54] – 2022	Extremidades superiores	Blando	PID control	Actuadores neumáticos blandos	Asistencia de las extremidades superiores aplicado a niños

Tabla 1.4. Estado del arte de exoesqueletos de rehabilitación (continuación).

AÑO	DISPOSITIVO	TIPO	ESTRATEGIA	ACTUADOR	CONTEXTO
[92] – 2021	Rodilla	Blando	Adaptive Impedance Control Based on Back Propagation Neural Network and Fuzzy PID Control	Actuador elástico basado en servomotores	Rehabilitación de rodilla en adultos
[96] – 2021	Cadera y tobillo	Blando	PD iterative control	Motor CD sin escobillas	Rehabilitación de cadera y tobillo en adultos
[100] – 2020	Rodilla	Blando	PID control	Actuador rotatorio flexible neumático de vacío	Marcha en adultos
Presente proyecto	Rodilla	Blando	Fuzzy PI control	Músculo neumático blando de presión negativa	Rehabilitación de la rodilla en bebés

El control PID tiene la ventaja de no requerir un modelo del sistema para operar, por lo tanto, se reduce el tiempo de diseño. Por otro lado, muchos autores resaltan la dificultad para ajustar los parámetros del controlador cuando se presentan dinámicas no lineales y variantes en el tiempo debido a que las técnicas de sintonía clásica no se aplican en forma eficiente. Es posible que el controlador PID no proporcione el control deseado y no pueda manejar las perturbaciones internas y externas, para un sistema más complejo, no lineal y variable en el tiempo, como un exoesqueleto [61]. En este contexto múltiples investigaciones se han desarrollado en el área de control difuso [1, 3, 61, 72]. El controlador difuso tiene una estructura no lineal por ende tiende a ser más eficiente que el control PID clásico.

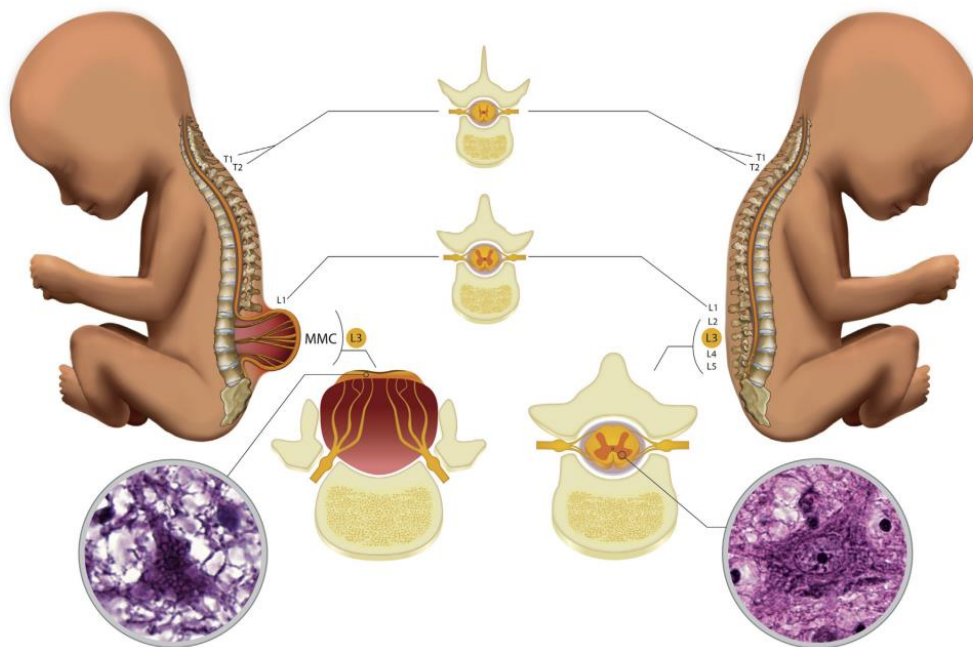


Figura. 1.1 Médula espinal en un bebé con espina bífida y uno saludable.

(Fuente: Referencia [53])

1.2. Descripción de la realidad problemática

Espina bífida es una malformación congénita donde la columna vertebral se divide producto de un defecto en el desarrollo del tubo neural embrionario [17]. Este defecto ocasiona el daño de la médula espinal, lo que conlleva la manifestación de múltiples problemas en el desarrollo psicomotor del infante, incluyendo la pérdida de movilidad en miembros inferiores (véase Figura. 1.1).

La forma más grave de espina bífida se denomina mielomeningocele, también denominada espina bífida abierta. Mielomeningocele ocasiona una abertura en el canal espinal a través de la cual las meninges y la médula están expuestas [74], generando una protuberancia que sobresale en la espalda del bebé (véase Figura. 1.2). Por más que se han producido esfuerzos para limitar la preponderancia de esta patología, aún se mantiene alrededor de 1 por cada 1000 recién nacidos a nivel global [53]. Por otro lado, parálisis cerebral describe un conjunto de trastornos del movimiento que afectan de 2 a 3 bebés por cada 1000 recién nacidos anualmente [39]. Dependiendo de la gravedad de la lesión cerebral, se desencadenan importantes déficits en el control postural y trastornos del movimiento que afectan el correcto desarrollo de las actividades diarias del individuo [81]. Los niños con parálisis cerebral comúnmente desarrollan síndromes espásticos, discinéticos o atáxicos en la primera infancia, lo que conlleva la pérdida parcial o total del movimiento en miembros inferiores [30].



Figura. 1.2 Protuberancia ocasionada por la patología de mielomeningocele
(Fuente: Referencia [37])

La aplicación de terapia física en bebés con parálisis cerebral y espina bífida representa un reto para las familias con bajos recursos y acceso restringido a servicios de salud, asimismo se debe considerar el costo económico añadido de contratar fisioterapeutas especializados en el trastorno. En estados unidos, un paciente con espina bífida incurre en gastos médicos elevados durante toda su vida (véase Figura. 1.3), además de los costos relacionados con sillas de ruedas y aparatos ortopédicos o protésicos [97].

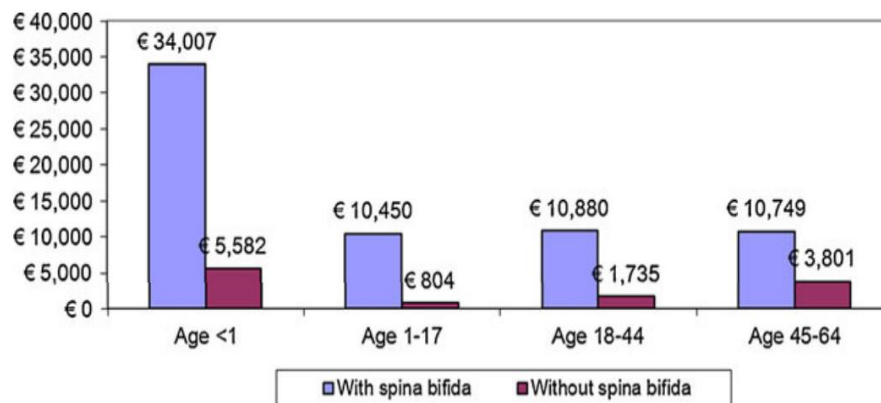


Figura.1.3 Gastos médicos anuales en euros: personas con y sin espina bífida.
(Fuente: Referencia [97])

En este contexto, el grupo de investigación en Micro-Biorobótica de la Universidad de Ingeniería y Tecnología desarrolló un músculo artificial neumático de presión negativa dirigido específicamente para la rehabilitación de miembros inferiores en bebés de 6 meses [26]. El músculo artificial cuenta con un buen desempeño torque/velocidad y tiene un costo de fabricación bastante accesible (véase Figura. 1.5). A pesar de estos beneficios, es importante considerar los cambios de presión generados dentro de la cámara del músculo artificial, aparecen comportamientos dinámicos no lineales que deben ser mitigados durante la operación del sistema.



Figura. 1.4 Exoesqueleto flexible montado en una maniquí de bebé de 6 meses.

Como segunda fase del proyecto se ha diseñado un exoesqueleto flexible para la rehabilitación de rodilla en bebés de 6 meses, utilizando los actuadores neumáticos anteriormente descritos (Figura. 1.4). Como fase final del proyecto se ha establecido diseñar una técnica de control automático que permita ejecutar los movimientos de flexión – extensión de las rodillas que requieren los bebés con discapacidad motora. También es importante considerar que el bebé produce movimientos involuntarios, ocasionando perturbaciones en la trayectoria deseada que deben ser minimizadas por el control. En conclusión, es importante diseñar una estrategia de control que permita la rehabilitación pasiva de miembros inferiores de los bebés de manera precisa y robusta.

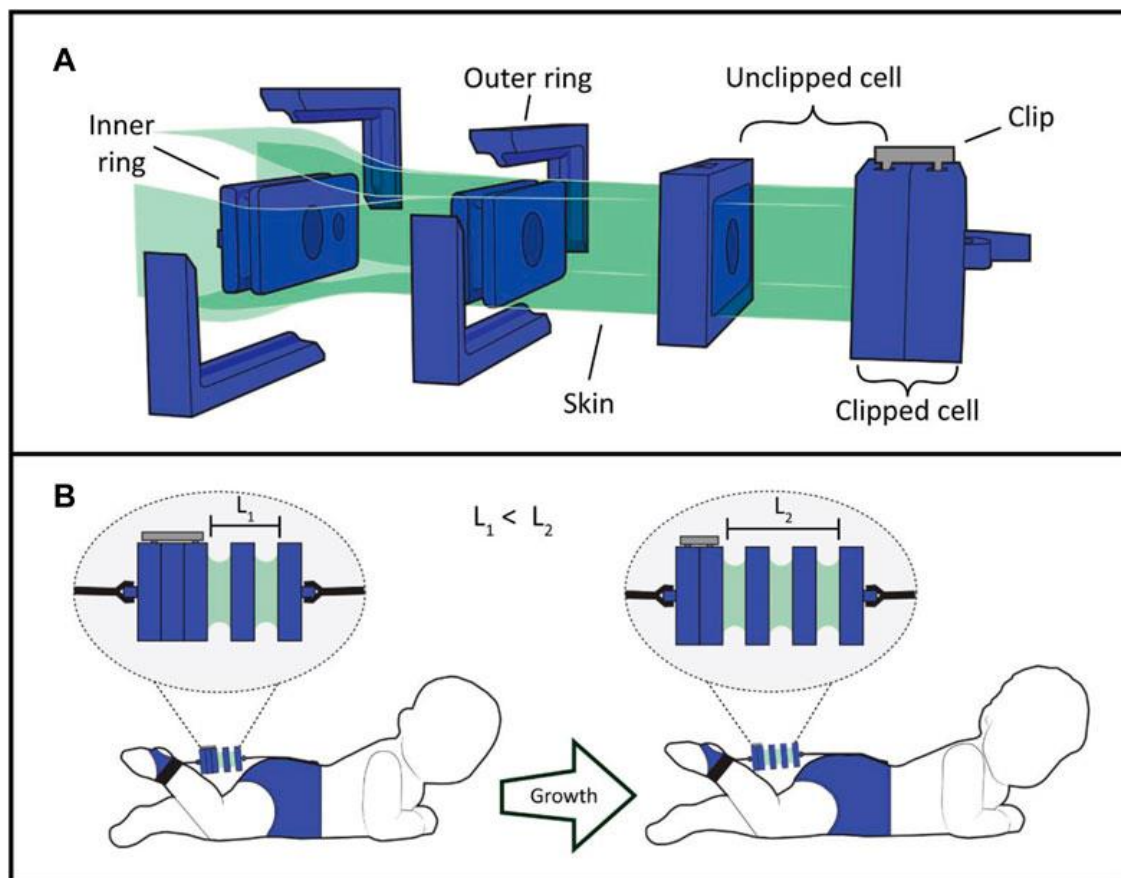


Figura. 1.5 Músculo artificial neumático de vacío. (A) Ensamble de los componentes del músculo artificial de vacío (B) Ubicación del actuador durante la rehabilitación del bebé.

(Fuente: Referencia [26])

Existen dos estrategias para controlar las trayectorias en la rehabilitación pasiva de miembros inferiores. La primera engloba las técnicas basadas en modelos del sistema, y la segunda considera las técnicas libres de modelos del sistema. Las técnicas de control basadas en modelos requieren la construcción de un modelo paramétrico preciso del

sistema, por el contrario, los controles libres de modelo no necesitan un modelo matemático exacto, sino que se aproximan las funciones desconocidas del sistema usando estrategias que se encuentran dentro del marco de control inteligente. El control PID es un controlador que no requiere el modelo del sistema para operar. El control PID requiere que sus parámetros se adecuen a la dinámica de la planta, por ende, es necesario aplicar estrategias de afinamiento en línea o fuera de línea. El control PID es limitado cuando hace frente a sistema no lineales y variantes en el tiempo, por ende, es necesario expandir sus potencialidades a través del control inteligente. Los controladores difusos son más robustos que los controladores PID, ya que cubren rangos más amplios de condiciones de operación que los controladores PID [44].

1.3. Formulación del problema

En la presente tesis se busca dar respuesta a la pregunta: ¿Cuál es el nivel de eficiencia del control PI difuso en la reducción del error del seguimiento de la trayectoria deseada de la flexión – extensión de las rodillas y el incremento de la robustez del sistema aplicado a un maniquí bebé de 6 meses de edad?

1.4. Justificación e importancia de la investigación

La justificación e importancia de la presente tesis de maestría ha sido desarrollada en 4 partes. Se ha considerado el punto de vista científico, el punto de vista técnico, el punto de vista social, y finalmente el punto de vista práctico. Tomando en cuenta el punto de vista científico, el desarrollo de estrategias de control para exoesqueletos flexibles dirigidos a bebés es un tema novedoso que ha llamado la atención de la comunidad científica en los últimos años. En la presente investigación se consideran bebés con diagnóstico de Espina Bífida, problema que no ha sido tocado antes en la literatura. La terapia de rehabilitación de rodillas dirigida a bebés con esta patología debe considerar temas de seguridad y robustez del dispositivo, puntos que no se consideran al diseñar exoesqueletos rígidos. Por lo tanto, nuevas estrategias y compensaciones son necesarias para obtener un resultado satisfactorio para aplicaciones médicas. Desde el punto de vista técnico, la presente investigación cuenta con financiamiento de PROCENCIA y cubre todos los costos de fabricación y evaluación. Se cuenta con el prototipo de bebé de 6 meses para probar las rehabilitaciones, así como también con el hardware y software necesarios para el diseño y la implementación del controlador. Se ha tratado de obtener la mejor

configuración para el sistema, con la finalidad de reducir el uso de dispositivos extra y en consecuencia incrementar los costos. Desde el punto de vista social, el dispositivo está dirigido a niños con diagnóstico de espina bífida los cuales actualmente no son tratados desde edades tempranas. El uso de este dispositivo permitirá una rehabilitación más temprana de los bebés (desde el primer mes de nacimiento), generando una mayor capacidad de recuperación de la movilidad de las extremidades inferiores. La novedad del dispositivo es la reducción de los costos de rehabilitación de los pacientes, que actualmente son aplicados por terapeutas especialistas en bebés. Para personas de bajos recursos es un reto adquirir estos servicios, por lo tanto, el dispositivo posee un impacto significativo para estos sectores vulnerables. Finalmente, desde el punto de vista práctico, el controlador propuesto otorga buen rendimiento y mejora las capacidades de adaptación ante las incertidumbres.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar e implementar un control difuso PI para controlar las trayectorias de flexión/extensión de las rodillas, realizadas por un exoesqueleto robótico flexible en un maniquí de bebé de 6 meses, con una robustez y un error superior a un control PI.

1.5.2. Objetivos específicos

- Seleccionar e implementar el hardware y software para el controlador.
- Diseñar e implementar un control PI para controlar las trayectorias de flexión/extensión de las rodillas.
- Validar el control difuso PI frente a un control PI utilizando las especificaciones de robustez y error de cada controlador.

1.6. Hipótesis

1.6.1. Hipótesis principal

“Si se aplica un control difuso PI, entonces se obtendrá una respuesta con un error de seguimiento de la trayectoria deseada de flexión/extensión de las rodillas y una robustez superior a un control PI”.

1.7. Variables e indicadores

A continuación, se describen las variables dependientes e independientes identificadas en la estructura de la formulación del problema realizado previamente. Cada una de las variables consideradas cuenta con cualidades como definición conceptual, dimensiones e indicadores.

1.7.1. Variables independientes e indicadores.

En la presente tesis de maestría la variable independiente se ha identificado como “control PI difuso” y sus características fundamentales (definición conceptual, dimensiones e indicadores) se describen en la Tabla 1.5.

1.7.2. Variables dependientes e indicadores.

En la presente tesis de maestría las variables dependientes se han identificado como “Error en el seguimiento de la trayectoria” y “Robustez del sistema” y sus características fundamentales (definición conceptual, dimensiones e indicadores) se describen en la Tabla 1.6. y Tabla 1.7 respectivamente.

1.8. Unidad de análisis

Prototipo de maniquí que simula un bebé de 6 meses.

1.9. Tipo y nivel de investigación

1.9.1. Tipo de investigación

La investigación es de tipo aplicada ya que se plantea la solución de una problemática que se aterriza en la implementación práctica de un sistema de control que satisface las condiciones de rehabilitación de las rodillas de un bebé de 6 meses de edad. Además, la investigación es experimental ya que se plantea la manipulación de variables independientes con el objetivo de mejorar u optimizar las variables dependientes.

1.9.2. Nivel de investigación

El desarrollo de exoesqueletos blandos es un tema bastante vigente en los últimos años dentro del ámbito académico, debido a sus múltiples ventajas de comodidad y portabilidad frente a los exoesqueletos rígidos que son pesados e incómodos. Muchas de las aplicaciones militares y médicas encontradas en la literatura han sido enfocadas a adultos y jóvenes, sin tomar en cuenta a los infantes, que también presentan patologías incapacitantes tratadas con los exoesqueletos de rehabilitación blandos. En este contexto la presente investigación ha tocado una problemática novedosa y única que ha sido vista pocas veces en la literatura. La metodología de diseño no es la misma que la vista en exoesqueletos rígidos convencionales, ya que nuevos problemas y retos se presentan debido a la flexibilidad e incertidumbres de los componentes que conforman el traje. Debido al enfoque aplicado de la investigación se han considerado temas de robustez y precisión de la rehabilitación, elementos pocos vistos en el estado del arte de los exoesqueletos revisados hasta hoy.

Tabla 1.5. Variable independiente control PI difuso.

VARIABLE INDEPENDIENTE	CONTROL PI DIFUSO	
DEFINICIÓN	DIMENSIÓN	INDICADOR
Sistema de control que se basa en reglas de inferencias difusas considerando la estructura del controlador PI.	Parámetros PI.	Error en la trayectoria durante la variación de la ganancia k_c .
	Número de funciones de membresía en las entradas y salidas.	Error en la trayectoria durante la variación del número de funciones de membresía en las entradas y salidas.
	Forma de las funciones de membresía.	Error en la trayectoria durante la variación de la forma de las funciones de membresía.
	Reglas difusas.	Error en la trayectoria durante la variación de las reglas difusas.

Tabla 1.6. Variable dependiente error del seguimiento de la trayectoria.

VARIABLE DEPENDIENTE	ERROR DEL SEGUIMIENTO DE LA TRAYECTORIA	
DEFINICIÓN	DIMENSIÓN	INDICADOR
Error encontrado al comparar la señal deseada de entrada al sistema de control con respecto a la señal medida en el sistema real.	Error del seguimiento en la trayectoria deseada	Variación del NRMSE (Error cuadrático medio de la raíz normalizado) de 0 a 100% durante una terapia.

Tabla 1.7. Variable dependiente robustez del sistema.

VARIABLE DEPENDIENTE	ROBUSTEZ DEL SISTEMA	
DEFINICIÓN	DIMENSIÓN	INDICADOR
Capacidad del controlador para soportar cambios bruscos en los parámetros del sistema.	Error del seguimiento en la trayectoria deseada frente a cambios en la masa de cada pierna.	Variación del NRMSE (Error cuadrático medio de la raíz normalizado) de 0 a 100% durante una terapia.

1.10. Periodo de análisis

Los datos son tomados en cualquier periodo de tiempo sin afectar los resultados del experimento.

1.11. Fuentes de información e instrumentos utilizados

Se utilizaron artículos científicos de las editoriales ELSEVIER, IEEE, WILEY y SPRINGER, que están disponibles en las bases de datos como SCOPUS, IEEE XPLORE, WILEY ONLINE LIBRARY, Y SPRINGERLINK.

1.12. Técnicas de recolección y procesamiento de datos

Los datos de la posición angular son medidos a través de una unidad de medida inercial (IMU). El algoritmo de control y visualización de datos se ha desarrollado en Python, y están embebidos en una tarjeta Raspberry Pi. Los datos son adquiridos usando una tarjeta Raspberry Pi y se almacenan en un archivo en formato CSV.

CAPITULO II

MARCO TEÓRICO Y MARCO CONCEPTUAL

En este capítulo se presentan las teorías y conceptos más relevantes sobre los controladores difusos y los exoesqueletos de rehabilitación. Se inicia el marco teórico con las definiciones y conceptos más importantes relacionados a la lógica difusa (conjuntos difusos, funciones de membresía, variables lingüísticas), posteriormente se introduce la teoría de los controladores difusos y su variante PI. El marco conceptual desarrolla los conceptos relacionados a los tipos de tratamientos para rehabilitación robótica, los tipos de exoesqueletos de acuerdo con la rigidez de sus estructuras (rígidos y blandos), y finalmente la clasificación de los exoesqueletos de acuerdo con su ubicación en el cuerpo (exoesqueletos de miembros superiores e inferiores).

2.1. El marco teórico.

2.1.1. Conjuntos difusos

Un conjunto borroso A en un universo de discurso U se caracteriza por una función de pertenencia (MF) $\mu_A(x)$ que toma los valores dentro del intervalo $[0,1]$ y se define según la Ecuación (2.1).

$$A = \{(x, \mu_A(x)), x \in U\} \quad (2.1)$$

Para U discreta, A se escribe como (2.2).

$$A = \sum_U \frac{\mu_A(x)}{x} \quad (2.2)$$

El signo de sumatoria no representa una suma aritmética sino denota la colección de todos los puntos $x \in U$ con MF asociado $\mu_A(x)$.

2.1.2. Funciones de membresía

Un subconjunto borroso A de un conjunto X es una función $A: X \rightarrow [0, 1]$. El valor de $A(x)$ se considera como el grado de pertenencia de x en A , y se denomina función de pertenencia de A . Otra notación de uso común para la función de pertenencia del subconjunto difuso A de X incluye $\mu_A: X \rightarrow [0, 1]$.

Una de las funciones de membresía más utilizadas son las lineales, ya que a menudo las representaciones gráficas y las operaciones con estos conjuntos difusos son muy simples, además se construyen fácilmente sobre la base de poca información [60].

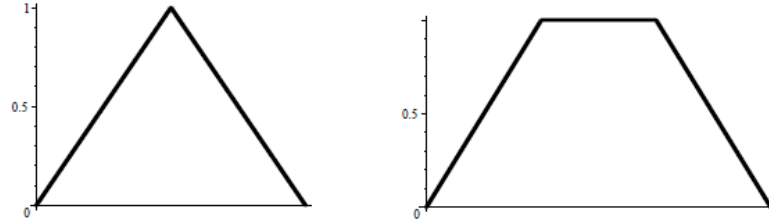


Figura. 2.1 Funciones de membresía trapezoidal y triangular.
(Fuente: Referencia [60])

La función triangular A (véase Figura 2.1) con extremos $(a, 0)$ y $(b, 0)$, y punto alto (c, α) está definida por (2.3).

$$A(x) = \begin{cases} \frac{\alpha(x-a)}{c-a}, & \text{si } a \leq x \leq c \\ \frac{\alpha(x-b)}{c-b}, & \text{si } c \leq x \leq b \\ 0, & \text{de lo contrario} \end{cases} \quad (2.3)$$

La función trapezoidal B (véase Figura 2.1) con extremos $(a, 0)$ y $(b, 0)$, y puntos altos (c, α) y (d, α) está definida por (2.4).

$$B(x) = \begin{cases} \frac{\alpha(x-a)}{c-a}, & \text{si } a \leq x \leq c \\ \alpha, & \text{si } c \leq x \leq d \\ \frac{\alpha(x-b)}{d-b}, & \text{si } d \leq x \leq b \\ 0, & \text{caso contrario} \end{cases} \quad (2.4)$$

2.1.3. Variables lingüísticas

Se conoce como variable lingüística a una variable X que toma como sus valores palabras u oraciones en lenguajes naturales. Las variables lingüísticas sirven como medio de caracterización aproximada de sistemas que no son descritos con precisión mediante valores numéricos u otros términos cuantitativos tradicionales [78]. Una variable lingüística se define como (2.5).

$$\{X, T, U, G, M\} \quad (2.5)$$

Donde X es el nombre de la variable, T denota el conjunto de términos de X , es decir, el conjunto de nombres de etiquetas lingüísticas de X sobre un universo de discurso U , G es la regla sintáctica o gramática para generar nombres, y M es la regla semántica para asociar a cada X . El conjunto de términos de la variable lingüística velocidad se expresa como (2.6).

$$T(\text{velocidad}) = \{\text{lento}, \text{medio}, \text{rápido}\} \quad (2.6)$$

Supongamos que la entrada de un controlador de lógica difusa toma valores de 0 a 500. Entonces, una posible definición de una entrada de controlador difuso como variable lingüística podría ser:

$$X: \text{entrada}, T: \{\text{grande}, \text{mediana}, \text{pequeña}, \text{cero}\}, U: [0, 500], M: X \rightarrow T$$

Definido como:

$$\text{grande} = \{(x, \mu_g(x)), x \in U\}$$

$$\text{mediana} = \{(x, \mu_m(x)), x \in U\}$$

$$\text{pequeño} = \{(x, \mu_p(x)), x \in U\}$$

$$\text{cero} = \{(x, \mu_c(x)), x \in U\}$$

2.1.4. Control difuso.

Las estrategias de control inteligentes surgieron de la necesidad de superar los problemas encontrados en los métodos de control clásicos, que consideran el desarrollo de modelos matemáticos del sistema y del controlador. El modelado tradicional no siempre describe un comportamiento dinámico de los procesos con exactitud, ya que las condiciones reales del sistema difieren de las asunciones hechas durante el desarrollo del modelo matemático. En este contexto, gracias al desarrollo de la lógica difusa, la aplicación del conocimiento y experiencia humana han tomado relevancia en el diseño de las estrategias de control.

La lógica difusa aplica un enfoque lingüístico borroso que traduce la experiencia humana a sistema de reglas de inferencia que permiten modelar incertidumbres y no linealidades complejas [79]. La arquitectura de un controlador difuso consta de un módulo de fusificación, un módulo de defusificación, una máquina de inferencia y una base de conocimiento (véase Figura 2.3).

2.1.5. Módulo de fusificación

- **Normalización.** Es una etapa de modificación de la escala, donde se mapean las variables del proceso actuales en un universo de discurso normalizado [20].
- **Fusificación.** En sistemas de control los datos observados casi siempre son nítidos (lectura de sensores), por lo tanto, es necesario integrar una etapa que descomponga las variables nítidas de entrada o salida del proceso en uno o más conjuntos difusos [50]. Por ejemplo, la lectura de un sensor es definida a través de la variable $x_0 \in R$ y sigue el proceso de fusificación en un universo de discurso que cuenta con dos funciones de membresía μ_{A_i} y $\mu_{A_{i+1}}$ triangulares. A través del proceso de fusificación se determinan los grados de pertenencia $\mu_{A_i}(x_0)$ y $\mu_{A_{i+1}}(x_0)$ de la variable x_0 con respecto a los conjuntos difusos A_i y A_{i+1} (véase Figura 2.2).

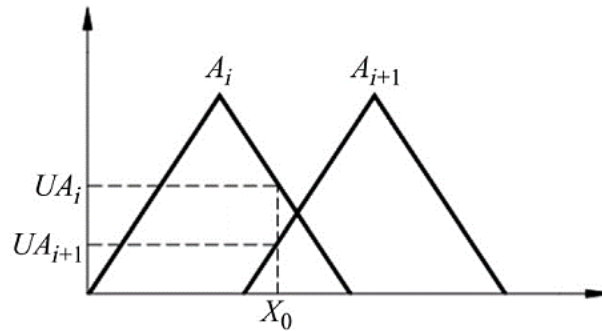


Figura. 2.2 Proceso de fusificación.

(Fuente: Referencia [50])

2.1.6. Módulo de defusificación

- **Denormalización.** Permite generar la salida física del sistema a partir del valor normalizado de la variable de control [20].
- **Defusificación.** Es el proceso opuesto a fusificación. Genera una transformación del conjunto de variables de salidas difusas en una salida nítida. La defusificación también se denomina decodificación de las salidas, lo que produce una acción de control no difusa [50]. El método más popular es el centro de área o centro de gravedad (2.7).

$$output = \frac{\sum_{i=0}^n d_i \mu_A(d_i)}{\sum_{i=0}^n \mu_A(d_i)} \quad (2.7)$$

Donde d_i es el i – ésimo valor del dominio y $\mu_A(d_i)$ es el valor de pertenencia de verdad para el punto de dominio.

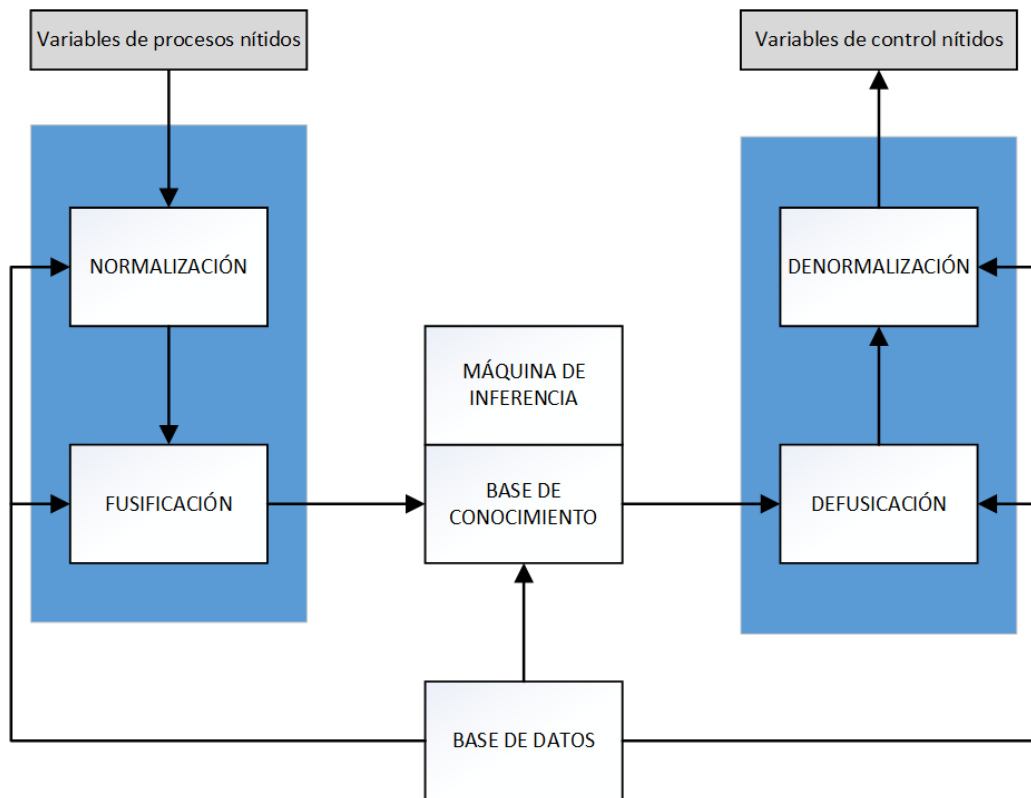


Figura. 2.3 Estructura de un controlador difuso clásico.
(Fuente: Referencia [20])

2.1.7. Base de conocimiento

La base de conocimiento consta de una base de datos, que provee la información necesaria para el funcionamiento de todo el conjunto, y una base de reglas difusas que representa la política de control de un experto con experiencia en el proceso. La base de datos engloba los conjuntos difusos, los dominios físicos con sus contrapartes normalizadas y denormalizadas, por otro lado, su diseño requiere la elección de las funciones de membresía y los factores de escala.

La base de reglas difusas es un componente central del controlador difuso y representa la "inteligencia" en cualquier algoritmo de control difuso, además es el lugar donde el conocimiento y la experiencia del diseñador deben interpretarse correctamente y organizarse en un conjunto de reglas adecuado [44]. El diseño de las reglas difusas requiere la elección de las variables de proceso y de control del sistema, elección de los contenidos de los antecedentes y consecuentes de las reglas difusas, y además la elección de los dominios de los valores lingüísticos.

La forma más efectiva de mostrar las reglas difusas es utilizando una tabla. En la Figura 2.4, se muestra una tabla de reglas difusas de un controlador, con 5 conjuntos difusos triangulares definidos por las entradas " e " y " Δyf ", y las salidas " u " como se muestra a continuación: negativo grande (NL), negativo pequeño (NS), cero (Z), positivo pequeño (PS), positivo grande (PL). Sobre la tabla de reglas difusas, se representan las trayectorias de fase (1) y (2) del sistema.

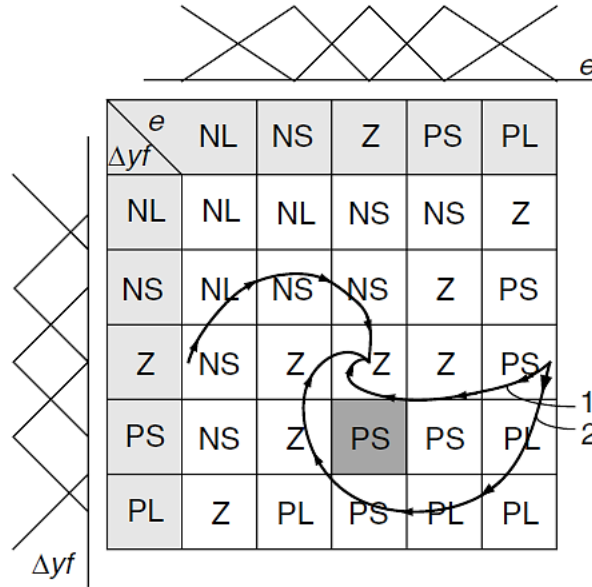


Figura. 2.4 Trayectorias de fase representado sobre las reglas de control.

(1) y (2) son las trayectorias de convergencia del plano fase del sistema.

(Fuente: Referencia [44])

2.1.8. Máquina de inferencia

Permite calcular la salida de control a partir de los aportes de cada regla de la base de conocimiento [20], o en otras palabras extraer la conclusión a partir de las premisas simbólicas generadas por las reglas difusas de la base de conocimiento. Uno de los métodos de inferencia más utilizados es el método de inferencia de Mamdani. El método de Mamdani (véase Figura 2.5) se describe a continuación, tomando en cuenta [50]. Para las variables lingüísticas A , B y C , que contienen un cierto número de términos descritos por conjuntos borrosos, A_i , B_j y C_k .

$$A = A_1, \dots, A_i, A_{i+1}, \dots, A_n$$

$$B = B_1, \dots, B_j, B_{j+1}, \dots, B_m$$

$$C = C_1, \dots, C_k, C_{k+1}, \dots, C_l$$

Donde las entradas A_i , B_j y la salida C_k se definen como:

$$A_i = \{(x, \mu_{A_i}(x))\}, i = 1, \dots, n$$

$$B_j = \{(x, \mu_{B_j}(x))\}, j = 1, \dots, m$$

$$C_k = \{(x, \mu_{C_k}(x))\}, k = 1, \dots, l$$

La base de reglas difusas se construye utilizando las siguientes premisas:

Si x es A_i y y es B_j entonces z es C_k

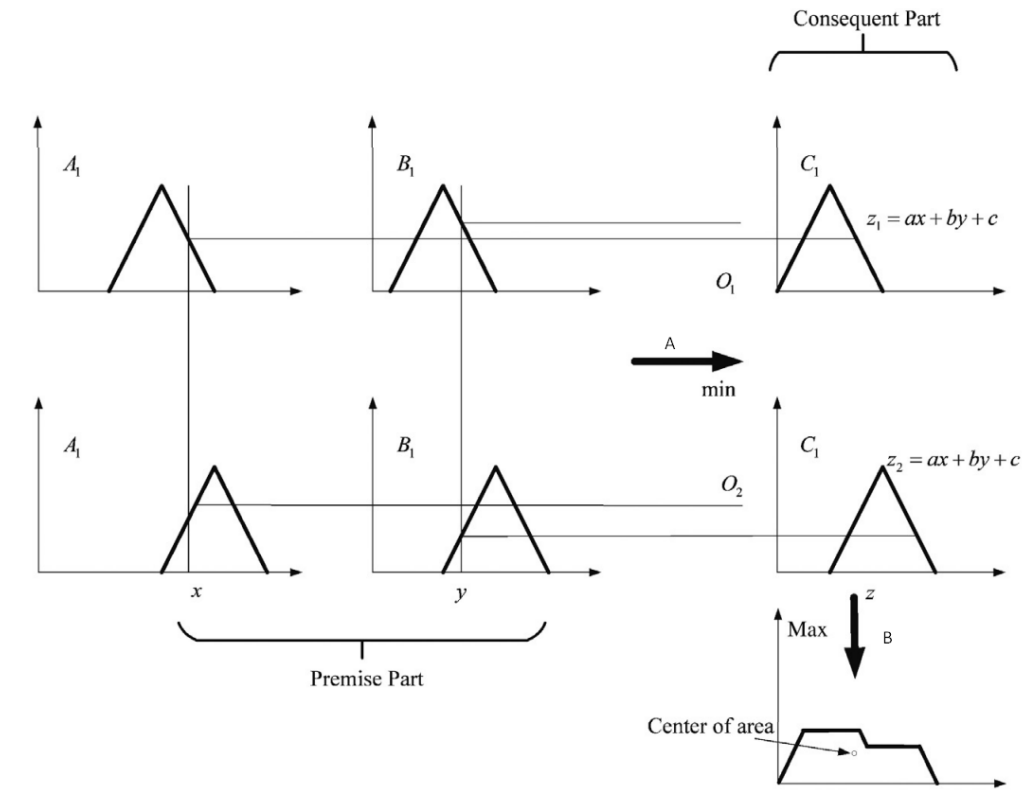


Figura. 2.5 Proceso de inferencia de Mamdani.

(A) Proceso de composición de A_i y B_j (Min), (B) Proceso de agregación de C_k (Max).

(Fuente: Referencia [50])

De acuerdo con la conjunción de composición, es fácil obtener la regla de inferencia como se muestra en (2.8).

$$Salida\ Borrosa_v = \min(\mu_{A_i}(x), \mu_{B_j}(y), \mu_{C_k}(z)) \quad (2.8)$$

Donde $v \in LV$, y LV es una variable lingüística. La salida agregada se expresa como se muestra en (2.9).

$$\mu_G(z) = \max (Salida\ Borrosa_v) \quad (2.9)$$

2.1.9. Control PI difuso.

Los controladores PI son ampliamente utilizados en la industria debido a su flexibilidad, sencillez de diseño, y bajo costo de recursos computacionales. A pesar de las ventajas mencionadas, no siempre un sistema se controla de manera eficiente utilizando un control PID, sobre todo cuando las no linealidades e incertidumbre son elevadas. Los controladores difusos son de naturaleza no lineal por ende son efectivos frente a dinámicas complicadas e inciertas. La integración de ambos conceptos mejora significativamente la eficiencia del sistema de control. Por ello, un controlador difuso permite generar acciones de control utilizando los mecanismos básicos del controlador PID [80]. Es importante tomar como punto de partida un controlador PID lineal para el diseño del controlador difuso, ya que deseamos utilizar todo el potencial del controlador no lineal que se origina del original PID. Un controlador proporcional integral (PI) convencional se describe como;

$$u = K_p e + K_I \int e \, dt \quad (2.10)$$

Donde K_p y K_i son los coeficientes de ganancia proporcional e integral. Tomando la derivada con respecto al tiempo de (2.11);

$$\dot{u} = K_p \dot{e} + K_I e \quad (2.11)$$

Los términos derivados de la expresión (2.11) se reescriben según:

$$\Delta u(k) = K_p \Delta e(k) + K_I e(k) \quad (2.12)$$

Esto produce una ecuación de controlador incremental similar a PI. En consecuencia, la base de reglas del controlador PI consta de reglas de la forma:

$$\text{Si } e \text{ es } A_i \text{ y } \Delta e \text{ es } B_j \text{ entonces } \Delta u \text{ es } C_k \quad (2.13)$$

En este caso, para obtener el valor de la salida de control $u(k)$, el cambio de salida de control $\Delta u(k)$ se suma a $u(k - 1)$ tal que;

$$u(k) = \Delta u(k) + u(k - 1) \quad (2.14)$$

2.1.10. Identificación de sistemas dinámicos no lineales

2.1.10.1. Redes neuronales de memoria a corto - largo plazo

En 1997, Hochreiter y Schmidhuber desarrollaron las redes LSTM para resolver algunos problemas encontrados en las redes neuronales artificiales recurrentes clásicas. En lugar de utilizar una arquitectura recurrente tradicional, con neuronas conectadas de forma recurrente, LSTM introduce una capa de bloques de memoria, que se encarga de almacenar el estado de la red [76]. Esta característica lo hace factible para aplicaciones de identificación de sistemas dinámicos no lineales complejos [48]. Un bloque LSTM general es una memoria compuesta por tres estructuras principales, denominadas puertas, utilizadas para construir el estado y la salida de la memoria. La estructura de una red neuronal artificial LSTM se observa en la Figura 2.6, donde x_k representa la entrada de la red, donde h_k representa la salida del bloque LSTM; L el número de bloques LSTM; M el número de entradas; sigmoide logística ($\sigma(\cdot)$) y tangente hiperbólica ($\tanh(\cdot)$) son funciones de activación no lineales; $[]$ siendo la concatenación de dos vectores. En las ecuaciones 2.15 – 2.21, W_f , W_c , W_i , W_o representan pesos entrenables con dimensión $R^{L \times (M+L)}$ para cada capa de puerta; b_f , b_c , b_i , b_o representa el sesgo asociado con cada capa de puerta.

$$f_k = \sigma(W_f \cdot [h_{k-1}, x_k] + b_f) \quad (2.15)$$

$$i_k = \sigma(W_i \cdot h_{k-1} + b_i) \quad (2.16)$$

$$\hat{c}_k = \tanh(W_c \cdot [h_{k-1}, x_k] + b_c) \quad (2.17)$$

$$g_k = i_k \cdot \hat{c}_k \quad (2.18)$$

Con el estado de celda actualizado definido como;

$$c_k = f_k * c_{k-1} + g_k \quad (2.19)$$

La última es la puerta de salida, que define la salida del bloque LSTM. La capa o_k define la cantidad de estado de celda que se generará, con h_k construido como;

$$o_k = \sigma(W_o \cdot [h_{k-1}, x_k] + b_o) \quad (2.20)$$

$$h_k = o_k * \tanh(c_k) \quad (2.21)$$

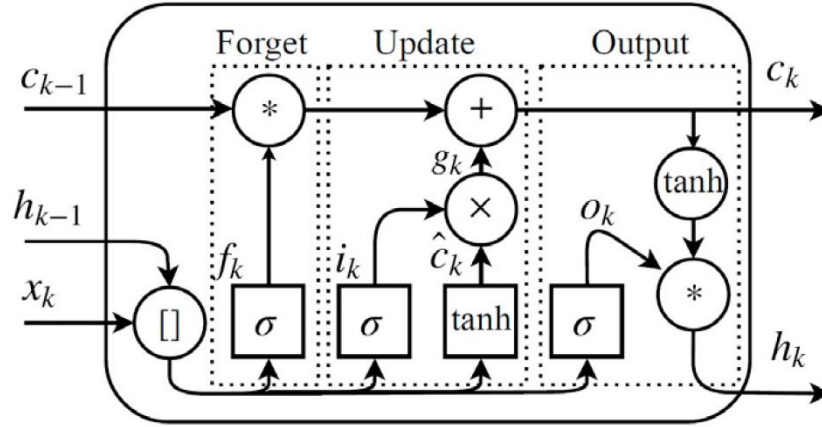


Figura. 2.6 Estructura interna de una red neuronal LSTM.

(Fuente: Referencia [76])

2.1.10.2. Modelado de sistemas no lineales usando redes neuronales LSTM

Un sistema dinámico no lineal tiene la siguiente forma;

$$y(k) = \varphi(y(k-1), \dots, y(k-n_y); u(k), \dots, u(k-n_u)) \quad (2.22)$$

Donde $\varphi(\cdot)$ es una ecuación no lineal desconocida que representa la dinámica de la planta. $u(k)$ y $y(k)$ son entradas y salidas escalares medibles. Nótese que aquí la estructura del sistema es una planta SISO, que se extiende al caso multivariable. La salida $y(k)$ está relacionada de forma no lineal con sus señales de entrada y salida pasadas y actuales $u(k-i)$ donde $i \in \{1, \dots, n_u\}$ y $y(k-j)$ donde $j \in \{1, \dots, n_y\}$.

Un sistema en tiempo discreto de la forma (2.22) es identificado usando los siguientes dos tipos de modelos [56];

- **Modelo de simulación**

$$\hat{y}(k) = N(u(k), \dots, u(k - n)) \quad (2.23)$$

Donde $\hat{y}(k)$ es la salida del modelo, $N[\cdot]$ es la estructura del modelo, que en esta ocasión es una red neuronal, n es el orden de la regresión para la entrada $u(k)$, n_u es desconocido, $n \neq n_u$. En la forma general (2.23) se escribe como:

$$\hat{y}(k) = N(\hat{y}(k - 1), \dots, \hat{y}(k - m); u(k), \dots, u(k - n)) \quad (2.24)$$

- **Modelo de predicción**

$$\hat{y}(k) = N(y(k - 1), \dots, y(k - m); u(k), \dots, u(k - n)) \quad (2.25)$$

Donde m es el orden de la regresión para la salida $y(k)$, $m \neq n_y$.

En aplicaciones donde se requieren predicciones de varios pasos, las salidas pasadas $y(k - i)$ no están disponibles y el modelo de predicción no es usado, además un modelo de simulación (2.23) no representa las relaciones dinámicas entre $u(k)$ y $y(k)$, en estos casos es usual usar redes neuronales recurrentes como LSTM [28].

El proceso de identificación incluye construir un modelo de identificación apropiado para estimar el sistema real, que está definido por la ecuación (2.22). El objeto del modelado de sistemas no lineales mediante LSTM es actualizar los pesos W_f , W_c , W_i , W_o de modo que la salida de la red neuronal converge a la salida del sistema;

$$\arg \min_{W_f, W_c, W_i, W_o} [\hat{y}(k) - y(k)]^2 \quad (2.26)$$

La arquitectura más simple de red neuronal usada para regresión consta de un bloque LSTM en serie con un bloque *Fully Connected* (véase Figura 2.7). Esta arquitectura se expande en profundidad agregando más capas con bloques LSTM ocultas a la red.

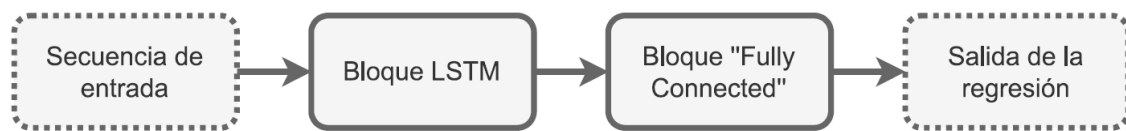


Figura. 2.7 Red neuronal LSTM para regresión.

2.1.10.3. Identificación para control

La identificación para control está orientada a la identificación de modelos que sirvan para el diseño de un controlador, lo que significa que, el objetivo no es un modelo muy preciso sino un buen desempeño del control. Se demuestra bajo algunas condiciones que el mejor modelo para el diseño de control se identifica en lazo cerrado con un método de identificación indirecta [59]. Esto nos lleva a pensar en los llamados esquemas iterativos de identificación y control [23]. En [59], describen el proceso de la siguiente manera;

- Las mediciones de lazo cerrado se recopilan con un controlador estabilizador simple cuyo diseño posiblemente se base en heurística y no requiera un modelo explícito.
- Se identifica un modelo con estos datos.
- Se diseña un nuevo controlador con este modelo.
- Luego, todos los pasos se repiten con el nuevo controlador hasta que el modelo y, por lo tanto, el controlador haya convergido.

2.2. El marco conceptual

2.2.1. Tratamientos para rehabilitación robótica.

- **Terapia pasiva.** Algunos trastornos del movimiento como parálisis cerebral o espina bífida generan la pérdida parcial o total de la movilidad de diferentes partes del cuerpo. En el peor de los casos los pacientes no cuentan con la fuerza necesaria para ejercer movimiento sobre la extremidad afectada. La terapia pasiva se lleva a cabo en los casos donde el paciente no ejerce ningún esfuerzo significativo por sí mismo. En estos casos el robot ejecuta una trayectoria de rehabilitación preconcebida durante el tiempo que dure una sesión [69].

- **Terapia activa.** La terapia activa se lleva a cabo en pacientes que tienen una pérdida parcial de la movilidad de alguna de sus extremidades. Existen dos modos para realizar una terapia activa, la primera se denomina terapia activa de asistencia que involucra la aplicación de una fuerza externa regulada que permite suplir parte de la fuerza que el paciente requiere para realizar los movimientos, por el contrario, la segunda denominada tratamiento de resistencia activa requiere la aplicación de una fuerza opuesta al movimiento del paciente [69].
- **Terapia bilateral.** La terapia bilateral permite la ejecución de movimientos de la extremidad perjudicada a través del seguimiento de la extremidad funcional [69]. Este seguimiento es posible gracias a la utilización de diversos sistemas de censado.

2.2.2. Exoesqueletos para rehabilitación.

Tomando como referencia al cuerpo humano, se clasifican en tres tipos. Los exoesqueletos ubicados a lo largo del cuerpo cuentan con múltiples grados de libertad y están destinados a la asistencia de diferentes partes del cuerpo, por otro lado, los exoesqueletos de miembros superiores son usados para rehabilitar partes del torso o los brazos. Finalmente, los exoesqueletos de miembros inferiores permiten rehabilitar cadera, rodilla, tobillo o una combinación de ellos [65].

Otros autores muestran énfasis en los tipos de materiales usados en la fabricación, llegando a dividir a los exoesqueletos en rígidos y blandos [15]. Los exoesqueletos rígidos (véase Figura. 2.8) cuentan con estructuras y articulaciones consistentes, aunque no se descarta que alguna de las partes sea flexible, como se muestra en [14]. En el extremo opuesto se encuentran los exoesqueletos blandos (véase Figura. 2.9) que introducen uniones y estructuras fabricadas con telas o elastómeros que permiten mayor flexibilidad, seguridad y portabilidad para el paciente [10, 90].

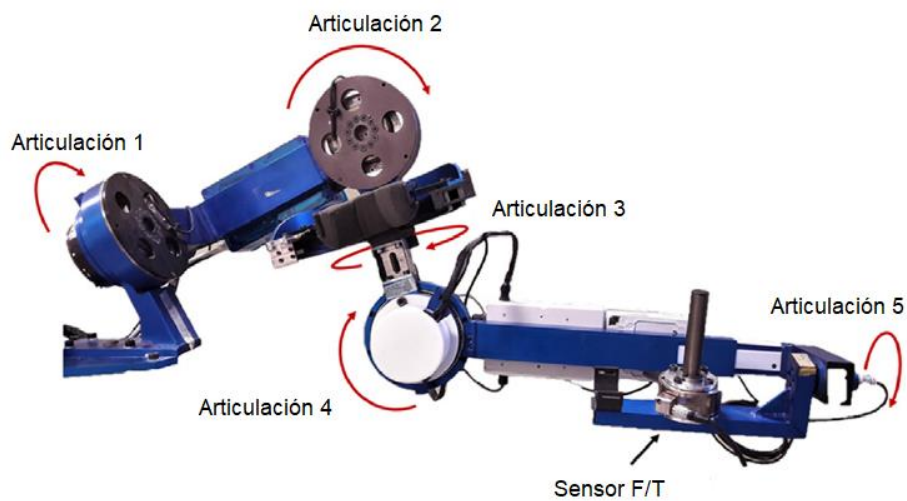


Figura. 2.8 Representación de un exoesqueleto rígido para rehabilitación de brazo.
(Fuente: Referencia [14])

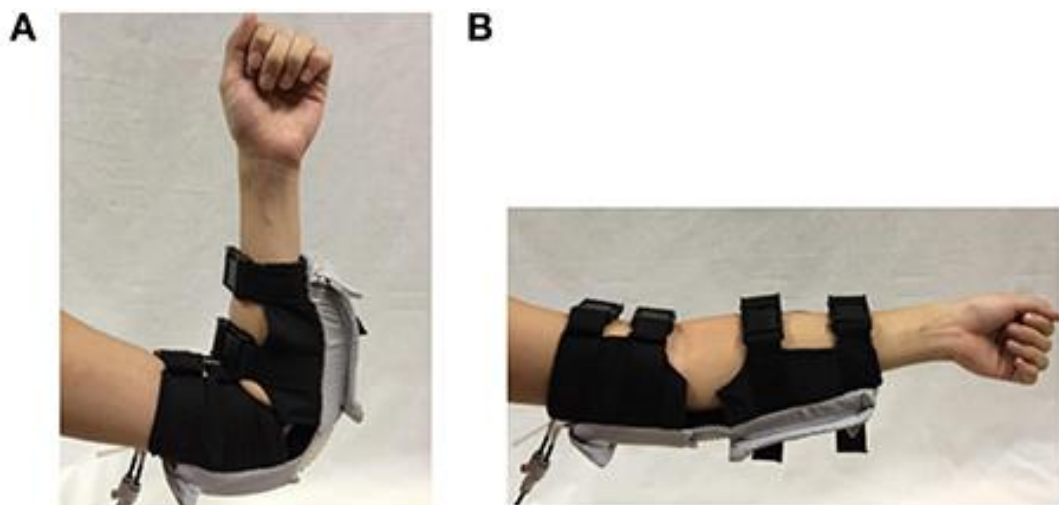


Figura. 2.9 Exoesqueleto Blando para rehabilitación de codo.
(A) Configuración de flexión, (B) Configuración de extensión.
(Fuente: Referencia [43])

CAPITULO III

DESARROLLO DEL TRABAJO DE TESIS

En el presente capítulo se desarrollan los procedimientos de diseño, implementación y validación del sistema mecatrónico utilizado en la presente tesis. Se inicia describiendo el proceso metodológico aplicado en la tesis, posteriormente se describe el diseño e implementación del hardware, software y control del exoesqueleto flexible de rodilla que permitirá ejecutar la flexión – extensión de las piernas de un maniquí de 6 meses en posición decúbito prono.

3.1. Metodología de la tesis.

La metodología general de la tesis se ha dividido en 4 etapas (véase Figura. 3.1). El procedimiento se inicia con el diseño e implementación del hardware y software del exoesqueleto flexible de rodillas, para posteriormente equiparlo en el maniquí de pruebas (simula los miembros inferiores de un bebé de 6 meses de edad) y poder ejecutar la flexión/extensión cíclica de ambas piernas en posición decúbito prono. La segunda etapa consiste en el diseño e implementación de un controlador difuso PI para el control de las trayectorias de rehabilitación en ambas piernas del maniquí mediante un método de diseño iterativo que inicia con el modelado y simulación en MATLAB/SIMULINK, y culmina con el afinamiento de los parámetros del controlador usando la experiencia del diseñador. En la tercera etapa, se evalúa la estabilidad y robustez del sistema controlado durante la ejecución de la terapia de rehabilitación frente a diferentes masas. Por ende, se sujetan discos metálicos en ambas piernas (encima el talón), ejecutando pruebas con masas que van de 0 a 100 gramos aproximadamente. Finalmente se validan los resultados de la integración del controlador en el exoesqueleto flexible de rodilla comparándolo en rendimiento con un controlador PI.

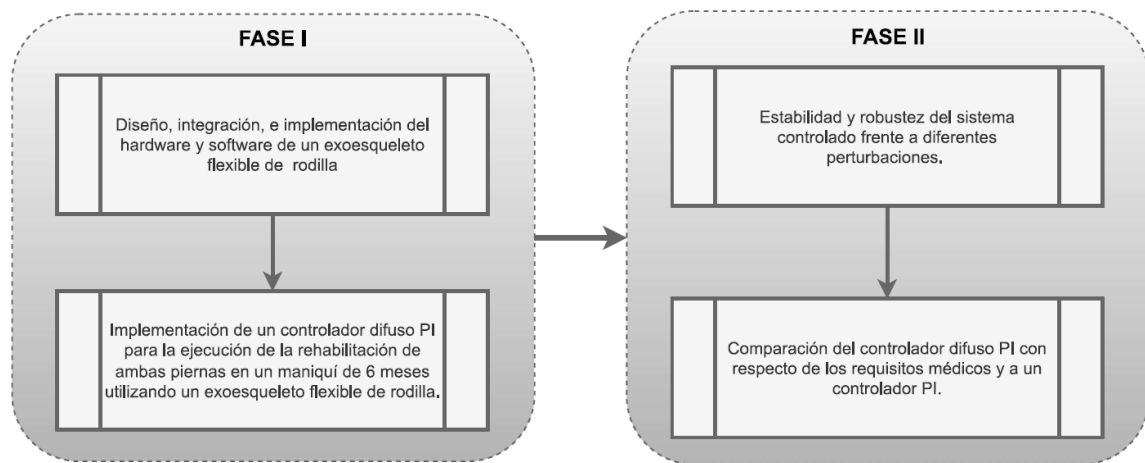


Figura. 3.1 Metodología de la tesis.

3.2. Diseño, integración e implementación del hardware y software de un exoesqueleto flexible de rodilla

3.2.1. Descripción del exoesqueleto flexible de rodilla

El exoesqueleto flexible desarrollado para asistir en la flexión – extensión de las rodillas en bebés, es un dispositivo que integra actuadores ligeros y modulares, un traje ergonómico, sensores de posición angular, y equipos electrónicos de control diseñados e implementados bajo recomendaciones de un especialista en neonatología e ingeniería biomédica. La ejecución de los movimientos se lleva a cabo con un músculo artificial impulsado por vacío (MAIV) que se comprime en rangos de presión negativa (extracción de aire interno). El MAIV cuenta con 09 celdas rectangulares (cada una conformada de anillos internos y externos) elaboradas con ácido poliláctico (PLA) en una impresora 3D (PRUSA MK3S), además se agrega un filme de polietileno que opera como sostén y cuerpo del actuador (véase Figura 3.2). La utilización de 09 celdas permitió un rango óptimo de flexión de la rodilla ($5^{\circ} - 55^{\circ}$) para ejecutar la terapia de rehabilitación en posición decúbito prono. Los actuadores se instalaron en un maniquí antropométrico fabricado de silicona y PLA que simula los miembros inferiores de un bebé de 6 meses de edad. El MAIV cuenta con hebillas de bloqueo en las celdas que se sitúan en los extremos, con la finalidad de facilitar la sujeción del actuador al traje. El traje está conformado de dos conjuntos, un short y dos tobilleras fabricados con tela de algodón que cuenta con ajuste de velcro. Las correas de polipropileno permiten un acoplamiento sencillo, ya que se adaptan perfectamente a las hebillas de bloqueo del actuador. Las tobilleras incluyen un soporte de caucho para colocar los sensores BNO055 (véase Figura 3.3).

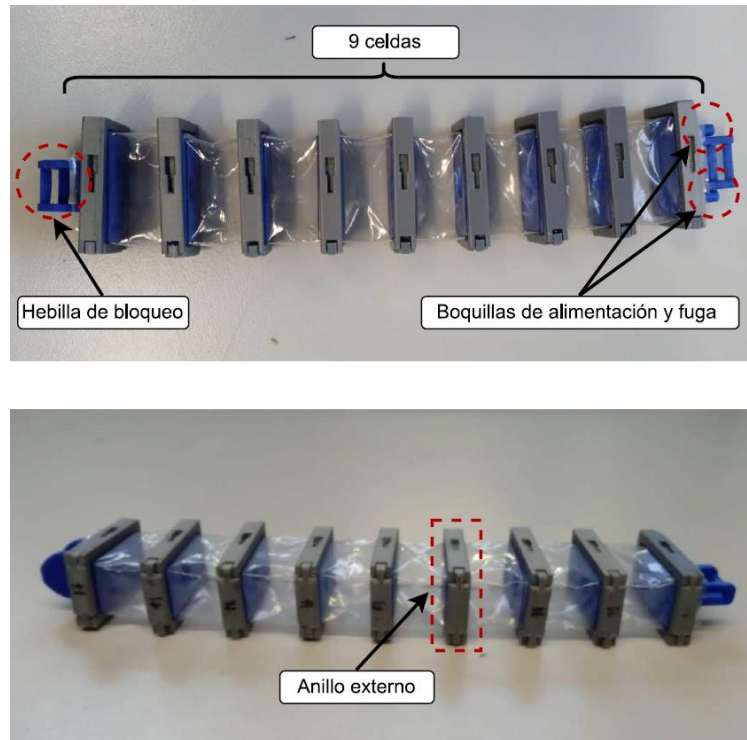


Figura. 3.2 Músculo artificial neumático de presión negativa.

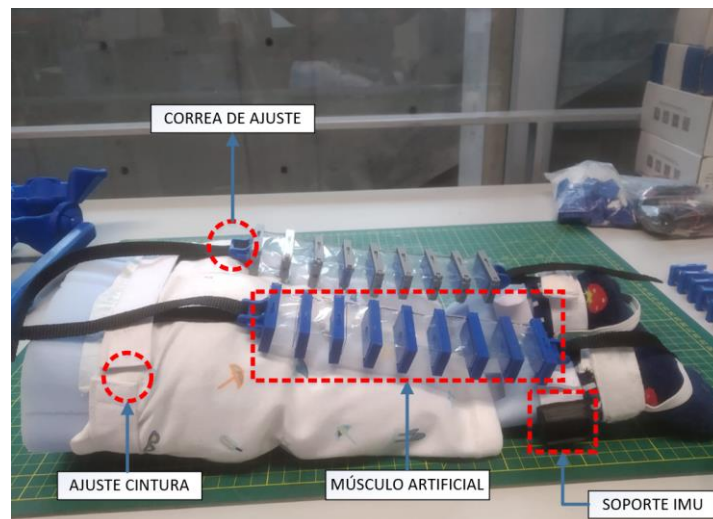


Figura. 3.3 Exoesqueleto flexible implementado en un maniquí de 6 meses.

Después de múltiples experimentos, se puede observar que el comportamiento del MAIV en un esquema de control PI ejecutado en el maniquí, tenía un menor error y una mayor estabilidad en el seguimiento de la referencia si se dejaba ingresar un pequeño flujo de aire constante al actuador durante los ciclos de compresión/extensión. Por lo tanto, con

la finalizar de linealizar el sistema se agregó una boquilla extra en uno de los extremos del MAIV, y así generar una fuga artificial que mejore las condiciones de control. Las boquillas permiten la salida de aire durante la flexión de la pierna y la entrada de aire durante la extensión de la pierna respectivamente. Es conocido en la literatura que un músculo artificial neumático de vacío con similar configuración que la propuesta en la presente tesis [27], sin el efecto de la gravedad, presenta un comportamiento diferente durante la compresión y extensión, lo que lo hace un sistema de naturaleza no lineal. Si se incluye un músculo artificial neumático en un esquema geométrico donde sufre los efectos de la gravedad, que es el caso de la pierna de un bebé cuando se encuentra en posición decúbito prono, esto acelera la pierna durante la fase de extensión. Las aceleraciones generadas no son contenidas por el músculo artificial, ya que el regulador electrónico solo tiene una salida a la atmosfera durante la fase de extensión, lo que no permite la entrada de aire a una velocidad que compense las altas velocidades generadas por la gravedad. Esto ocasionó muchos problemas al controlar la fase de extensión de ambas piernas. Es observable que este efecto se ha mitigado a partir de la inclusión de una fuga artificial en el actuador (véase Figura 3.4), aunque no se sepa medir cuantitativamente cual es el tamaño de esta fuga. Esta fuga permite un mayor flujo de aire hacia el actuador durante la fase de extensión de la pierna, lo que permite compensar mejor los cambios de la velocidad debido a los efectos gravitatorios.

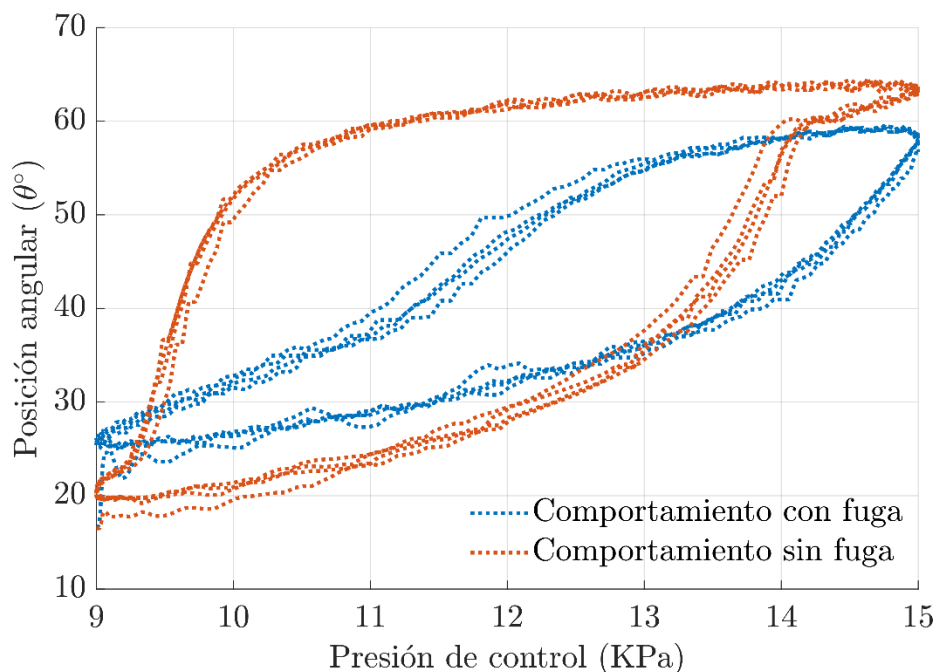


Figura. 3.4 Hardware implementado en el maniquí de 6 meses.

3.2.2. Hardware y software del exoesqueleto flexible de rodilla

El dispositivo de actuación del sistema es neumático, por ende, los equipos de control cuentan con componentes neumáticos y electrónicos (véase Figura 3.5). La neumática que acciona el MAIV incluye un regulador electroneumático SMC ITV2091-21N2BS5, una bomba de vacío RS-2, un tanque de reserva de vacío, conexiones neumáticas con ajuste a presión, y tubos neumáticos. El regulador electroneumático se excita con señales eléctricas de voltaje (0 a 10VDC) o corriente (0 a 20mA) y su salida genera una presión de vacío proporcional (-1.3 a -80KPa) que efectúa la compresión del MAIV. Para nuestra aplicación se estableció un rango de voltaje de 0 a 5VDC como punto de operación óptimo, debido a que generaba presiones de trabajo adecuadas para el MAIV. Los comandos de presión se procesan en el microcomputador Raspberry Pi modelo 4B, donde luego de ser normalizados en términos de voltaje, son generados por un convertidor digital/análogo (DAC) MCP4725 de 12 bits (soporta comunicación I2C). La señal de voltaje generada excita al regulador electroneumático de manera que se produce a su salida la presión de vacío necesaria para que el MAIV se comprima/extienda cíclicamente y pueda ejecutar la trayectoria de rehabilitación de manera precisa. La ejecución de la flexión/extensión de ambas piernas del maniquí requiere de 02 reguladores electroneumáticos, 02 DAC, 02 IMU, 01 tarjeta de adquisición/control, 01 pantalla de visualización, 01 tanque de reserva de vacío, 01 bomba de vacío, conexiones y tubos neumáticos.

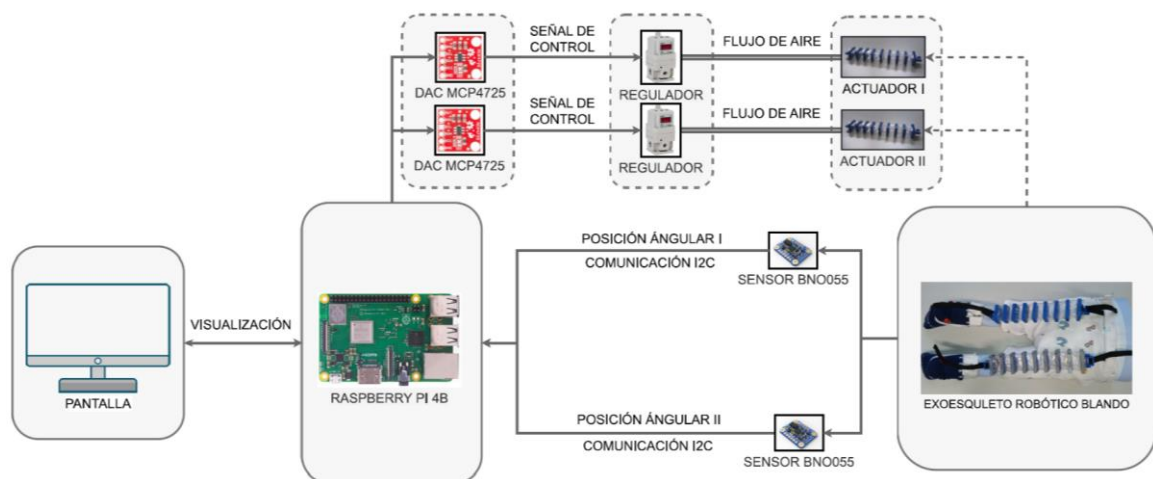


Figura. 3.5 Hardware implementado en el maniquí de 6 meses.

Se utiliza un sensor de medida inercial BNO055 (IMU) para monitorear la posición angular de ambas piernas. Los datos de los sensores fueron adquiridos por un microcomputador Raspberry Pi modelo 3B, usando una comunicación I2C. La tarjeta

Raspberry Pi se alimenta a través de su entrada de alimentación USB-C, por ende, por el momento se requiere la conexión de una fuente de alimentación 5V/3A. El sistema de adquisición de datos y control del exoesqueleto flexible de rodilla se ha embebido en una tarjeta de desarrollo Raspberry Pi que recibe los datos de los módulos de sensado, y a través de una interfaz permite visualizar las trayectorias de posición angular que son relevantes para medir la eficiencia de la rehabilitación. La interfaz se desarrolló utilizando el lenguaje de programación Python3, al igual que los algoritmos de adquisición de datos y control, y además todo ha sido integrado en un solo programa.

3.3. Implementación de un control difuso PI para la ejecución de la rehabilitación de ambas piernas en un maniquí de 6 meses utilizando un exoesqueleto flexible de rodilla

3.3.1. Implementación de un controlador PI para el exoesqueleto flexible de rodilla

3.3.1.1. Especificaciones de control del sistema robótico

Los requerimientos de operación del dispositivo de rehabilitación pasiva fueron establecidos con ayuda de un especialista del Instituto Nacional de Salud del Niño San Borja que ha trabajado con casos de bebés con espina bífida y mielomeningocele (véase Tabla 3.1). La trayectoria de rehabilitación que se ajusta a estas recomendaciones se muestra en la Figura 3.5. Se han establecido una rehabilitación intensiva que considera la ejecución de medio ciclo de rehabilitación de la primera pierna, antes de que se ejecute medio ciclo de la segunda pierna (véase Figura 3.6).

El controlador debe ejecutar la rehabilitación de manera suave y lineal durante la flexión/extensión de las piernas. Durante la extensión existe la posibilidad de que la pierna pueda golpear la superficie, por ende, es necesario tomar cuidado y extender el tiempo de ejecución a 5 segundos. La posición angular inicial de ambas piernas es 5° cuando el maniquí reposa sobre una superficie plana, por otro lado, se recomienda que la flexión no supere 55° (posición angular de 60°), por ende, se selecciona un rango de operación que va desde la posición angular de 5° a 55° .

Los bebés tienden a aumentar su peso rápidamente en los primeros meses de vida (aproximadamente 20 gramos al día), llegando a ser fundamental que el esquema de control implementado tenga la capacidad de soportar estos cambios. Con la finalidad de simular estos cambios significativos, se contempló sujetar discos metálicos (0 a 100 gramos) a la altura del tobillo del maniquí. Se planea comparar el esquema de control difuso PI frente a un control PI clásico empleando las mismas condiciones de operación.

Tabla 3.1. Requerimientos de la rehabilitación.

Posición del paciente	Decúbito/Prono
Edades	6 meses
Posición angular de operación	5° - 55° (50°)
Restricción angular de la rehabilitación	No debe superar 60°
Comportamiento de la rehabilitación	Suave
Tiempo en ejecutar la flexión de la pierna	3 segundos
Tiempo en ejecutar la extensión de la pierna	5 segundos
Tiempo que se mantiene en flexión	5 segundos
Tiempo entre ciclo y ciclo	5 segundos

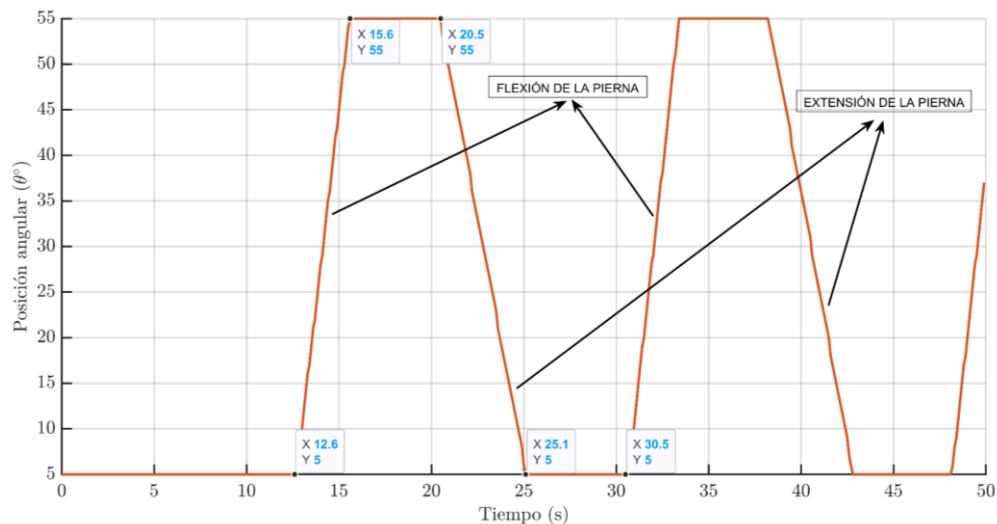


Figura. 3.6 Trayectoria de rehabilitación recomendada por el especialista.

3.3.1.2. Metodología de diseño de un controlador difuso PI

La metodología de diseño e implementación de un controlador difuso PI es iterativa y consiste en 6 pasos básicos (véase Figura 3.8). Para el diseño del controlador difuso PI tomaremos como base los datos de entrada/salida de la planta adquiridos bajo un esquema controlador PI clásico. Se hizo un modelo de la planta basado en red neuronal LSTM y se efectúan simulaciones bajo los esquemas de control difuso elegidos. Los parámetros del controlador difuso se ajustan de acuerdo con los resultados obtenidos durante la simulación, además se verifica que las acciones de control y el error en el seguimiento de la trayectoria se encuentren en rangos válidos. El controlador difuso se valida a través de prueba experimentales, y en caso no se llegue a los resultados esperados, se repite el ciclo.

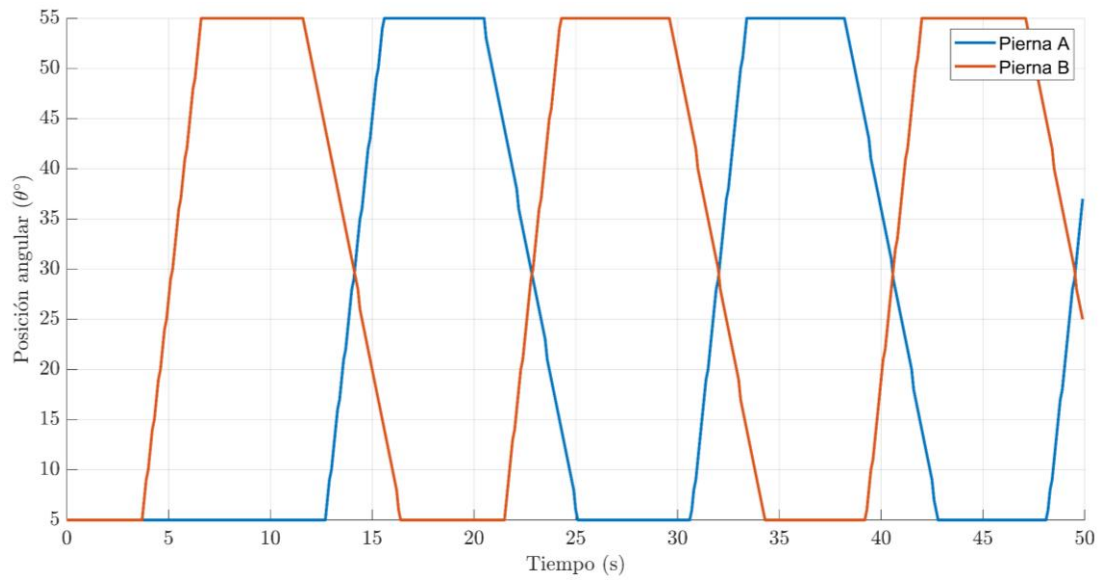


Figura. 3.7 Trayectorias para la rehabilitación intensiva.

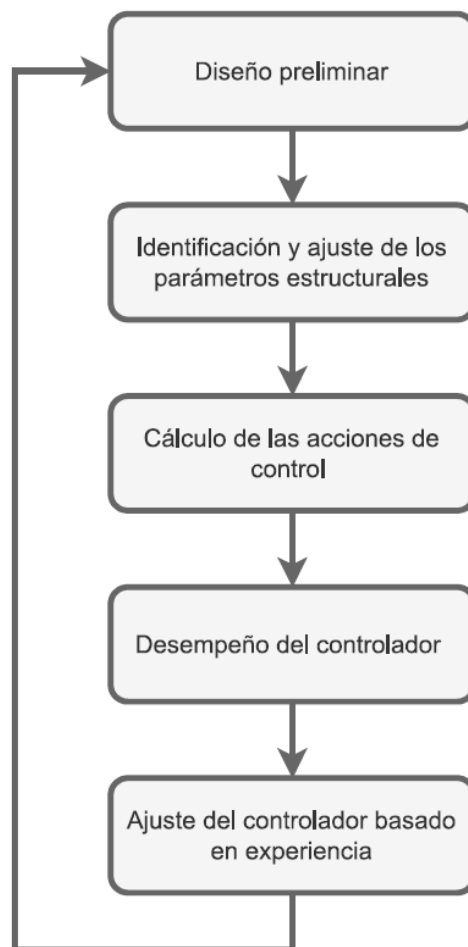


Figura. 3.8 Diseño de los parámetros óptimos de un controlador difuso PI.

3.3.1.3. Diseño preliminar del controlador

Cada actuador neumático requiere de una señal de referencia y una señal de control, por ende, el sistema de control de la posición angular de ambas piernas se modela como dos sistemas SISO independientes, cuya única diferencia reside en la trayectoria de referencia que siguen.

El esquema de control consta de dos sistemas SISO, conformados cada uno por un controlador difuso de 25 reglas, con el error y la derivada del error como entradas. A la salida de cada controlador difuso PI se agrega un integrador discreto y un factor de escala en serie (Figura 3.9) para generar la señal de control que opera un actuador. El primer actuador cuenta con una señal de referencia θ_d^1 , un error e_1 y una derivada del error Δe_1 como señales de entrada al controlador difuso, una derivada de la señal de control Δu_1 como salida del controlador difuso, un factor de escala K_c^1 y una señal de control u_c^1 . El segundo actuador cuenta con una señal de referencia θ_d^2 , un error e_2 y una derivada del error Δe_2 como señales de entrada al controlador difuso, una derivada de la señal de control Δu_2 como salida del controlador difuso, un factor de escala K_c^2 y una señal de control u_c^2 . Tomando en cuenta la estructura del controlador difuso, se desarrolla el pseudocódigo para su posterior implementación (Anexo A1).

Elegimos funciones membresía triangulares para todos los conjuntos difusos, con la finalidad de reducir la complejidad del diseño y computo. Dividimos las entradas y las salidas en 5 conjuntos difusos.

$$e = \{NB, NS, ZO, PS, PB\}; \Delta e = \{NB, NS, ZO, PS, PB\}; \Delta u = \{NB, NS, ZO, PS, PB\}$$

El controlador utiliza 25 reglas difusas generales (Tabla 3.2), donde NB representa negativo grande, PB es positivo grande, NS es negativo pequeño, PS es positivo pequeño, ZO es aproximadamente cero.

Tabla 3.2. Tabla de reglas difusas.

$\begin{matrix} \text{de} \\ e \end{matrix}$	NB	NS	ZO	PS	PB
NB	NB	NB	NB	NS	ZO
NS	NB	NS	NS	ZO	PS
ZO	NB	NS	ZO	PS	PB
PS	NS	ZO	PS	PS	PB
PB	ZO	PS	PB	PB	PB

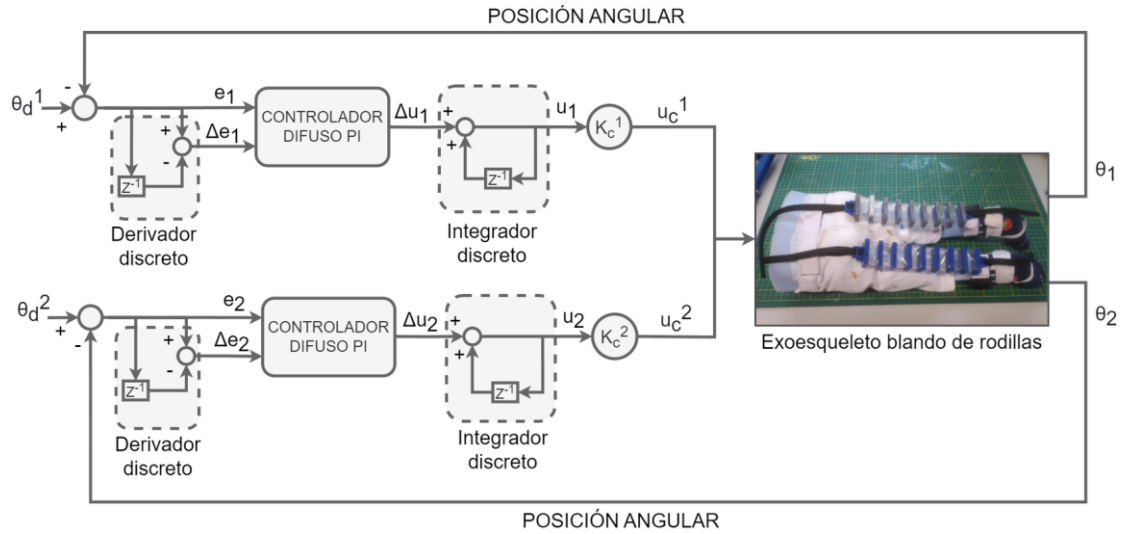


Figura. 3.9 Configuración A del controlador difuso PI para ambas piernas.

3.3.1.4. Identificación y ajuste de los parámetros estructurales

Se realizaron pruebas experimentales para ajustar un controlador PI clásico (véase Figura 3.11) y controlar la trayectoria de rehabilitación de 5° a 55° , llegando a obtenerse una respuesta óptima con los parámetros integral y proporcional de $K_i = 1.5$, $K_p = 0.18$ respectivamente, usando una tasa de muestreo de 0.05 Hz durante la adquisición.

Se emplearon las librerías de Deep Learning de MATLAB & SIMULINK para identificar un modelo dinámico basado en redes neuronales recurrentes LSTM. En identificación de sistemas dinámicos, es recomendable realizar la adquisición de datos en un control lazo cerrado que demuestre estabilidad, ya que el modelo será usado para validar un controlador. Se identifica un modelo local que trabaja de manera óptima para la trayectoria de referencia trapezoidal requerida para la rehabilitación. En caso se quiera trabajar con una trayectoria de referencia de distinta frecuencia, deberá identificarse un nuevo modelo.

La arquitectura de red neuronal LSTM requiere la configuración del número de capas intermedias, número de entradas, número de salidas, número máximo de épocas, tasa de aprendizaje, algoritmo de optimización principalmente. Posteriormente, los datos de entrada/salida, las características de las capas y las opciones de entrenamiento se almacenan cada una en una variable y se agregan a la configuración de entrenamiento usando la función de MATLAB "*trainNetwork*", momento en el cual ya podemos iniciar el aprendizaje.

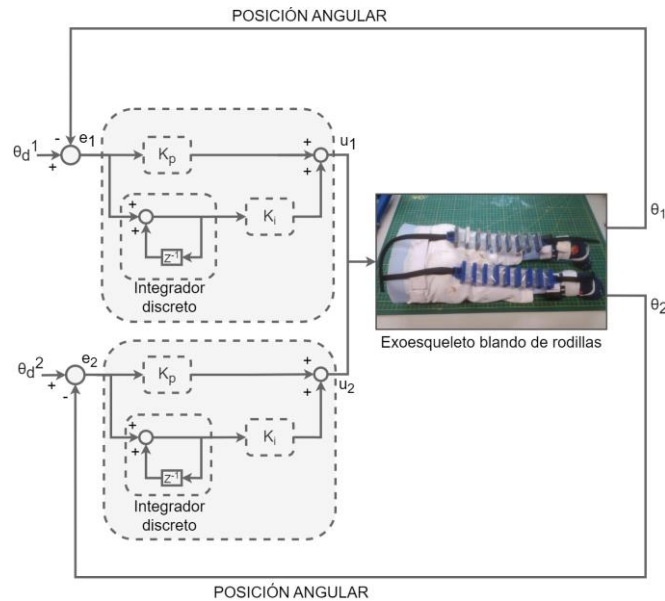


Figura. 3.10 Controlador PI para ambas piernas.

Es posible verificar el comportamiento de la salida del modelo identificado empleando la función "*predict*" de MATLAB y una data de entrada referencial de similares atributos a la disponible durante el entrenamiento. Esta herramienta será útil para la etapa de validación donde contrastaremos la diferencia entre la salida del modelo y la salida experimental, usando parte de los datos entrada/salida adquiridos. Los datos de entrenamiento y validación se obtuvieron durante las pruebas experimentales con un controlador PI usando la trayectoria de rehabilitación recomendada. Fueron necesarios 1000 datos para el entrenamiento, y 900 para la validación a una tasa de 0.05 Hz (Figura 3.12).

Finalmente, se identifican los parámetros estructurales del controlador difuso PI, tomando como base una distribución uniforme de las funciones de membresía sobre el universo de discurso de las diferentes variables de entrada y salida. Los parámetros de diseño del controlador difuso están asociados a los rangos de operación de los universos de discurso de las variables.

3.3.1.5. Cálculos de las acciones de control

Aplicamos como método de razonamiento difuso un sistema de inferencia de MANDANI, que considera la regla de composición mínimo – máximo, y un método de defusificación por centroide. Se procederá a realizar un ajuste del controlador difuso a través de prueba y error utilizando simulación. Todas las operaciones mencionadas se configuran utilizando el toolbox de lógica difusa de SIMULINK.

3.3.1.6. Desempeño del controlador difuso

El desempeño del controlador difuso se evalúa a partir del comportamiento de la señal de error del seguimiento de la trayectoria de rehabilitación durante el periodo de la simulación. Además, se verifica que las señales de control se encuentren en rangos de operación válidos y sin oscilaciones importantes.

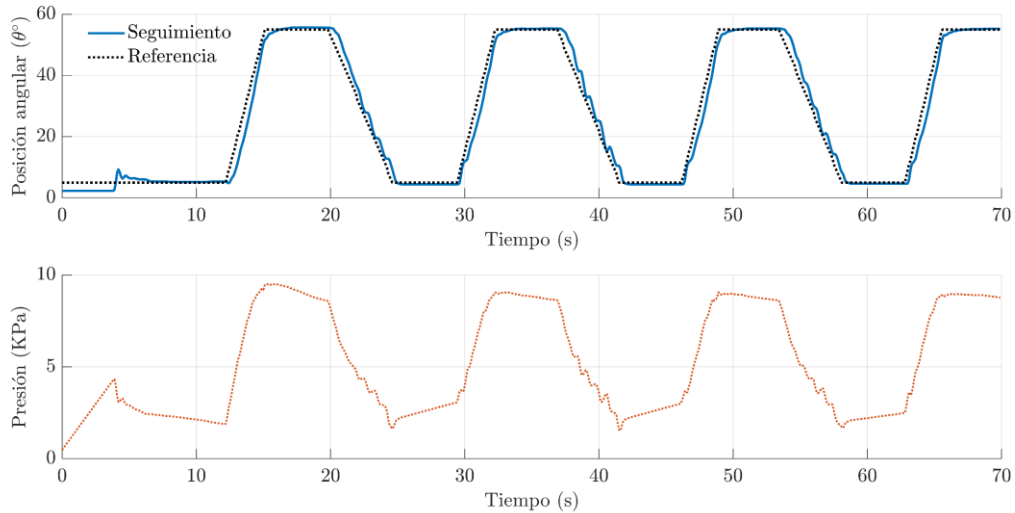


Figura. 3.11 Datos de posición angular y presión (entrada/salida) del sistema.

3.3.1.7. Ajuste del controlador basado en experiencia

A partir de los resultados obtenidos durante la simulación, tomamos la decisión de modificar las funciones de membrecía, el factor de escala o las reglas difusas, de manera que se llegue a obtener el mejor comportamiento del seguimiento de la trayectoria durante las pruebas experimentales. Si no se obtienen los resultados esperados, se volverá realizar una iteración más del proceso de diseño. Las librerías de "*scikit – learn*" de Python facilitaron la implementación del controlador difuso PI bajo diferentes configuraciones de control.

3.4. Estabilidad y robustez del sistema frente a diferentes perturbaciones

Se evalúa el controlador difuso frente a cargas que exceden las condiciones de diseño, con la finalidad de analizar la estabilidad y robustez del sistema de control. Se probará el comportamiento del controlador difuso PI en caso exista un incremento de la masa en las piernas del bebé, por ende, se procederá a agregar pesos metálicos de manera incremental (40, 60, 80, 100 gramos) en el maniquí de prueba (véase Figura 3.13).

3.5. Comparación del controlador difuso PI con respecto a los requisitos médicos y un controlador PI.

Para el diseño del controlador difuso PI se han considerado los requisitos de la rehabilitación mencionados en la Tabla 3.1. Por otro lado, se compara el comportamiento del controlador difuso PI con respecto a un controlador PI clásico. Para llevar a cabo esta comparación se utiliza el parámetro NRMSE (error cuadrático medio normalizado) del seguimiento de la trayectoria de rehabilitación con respecto a la referencia. El valor del parámetro NRMSE se determinará para cada uno de los experimentos realizados. Para poder realizar la evaluación, los datos adquiridos en formato CSV a través de la Raspberry Pi son transferidos al entorno de MATLAB. El valor de NRMSE se determina a través de la ecuación (3.1) que está integrada en el conjunto de funciones de MATLAB con el nombre de *"goodnessOfFit"*.

$$fit(i) = \frac{\|xref(:,i) - x(:,i)\|}{\|xref(:,i) - \text{mean}(xref(:,i))\|} \quad (3.1)$$

Donde $\|$ indica la norma-2 de un vector. *fit* es un vector fila de longitud N e $i = 1, \dots, N$, donde N es el número de canales.

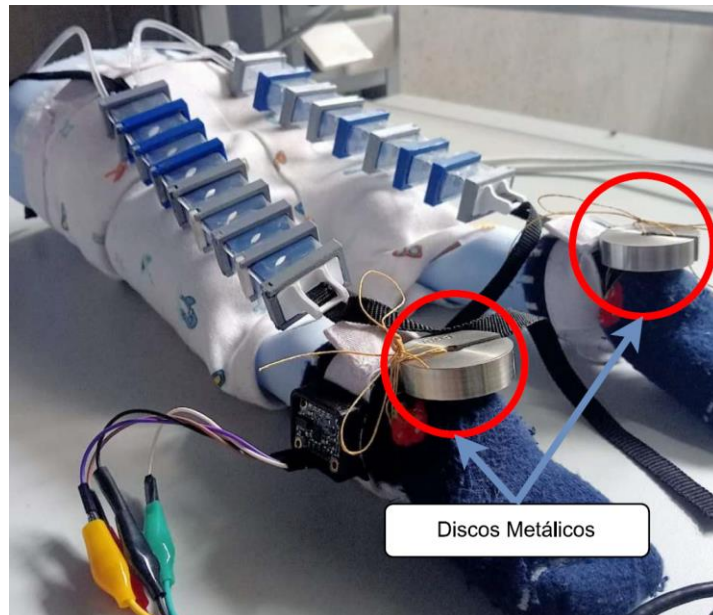


Figura. 3.12 Maniquí de bebé con discos metálicos.

CAPITULO IV

ANÁLISIS Y RESULTADOS

4.1. Resultados experimentales usando un controlador PI clásico

Se observa un comportamiento ligeramente oscilatorio durante la fase de extensión de la pierna A, por otro lado, el comportamiento de la pierna B es estable y sin oscilaciones (véase Figura 4.1). Las señales de presión en ambas piernas siguen un patrón regular, pero es observable una ligera oscilación durante la fase de extensión para el patrón de presión de la pierna A, que es constante durante todos los ciclos ejecutados (véase Figura 4.2). Las señales de error durante el seguimiento de la trayectoria de ambas piernas siguen un patrón regular durante todo el experimento (véase Figura 4.3). La señal de error oscila de $[-17.37^\circ, 11.56^\circ]$ en la pierna A, y de $[-14.75^\circ, 7.12^\circ]$ en la pierna B, cuando se realizan pruebas sin masas adicionales. Posteriormente, pese a que se agregaron 100 gramos de masa adicional por pierna, el rango de oscilación del error permaneció en $[-14.37^\circ, 9.06^\circ]$ en la pierna A, y $[-15.75^\circ, 9.18^\circ]$ en la pierna B (véase Tabla 4.1).

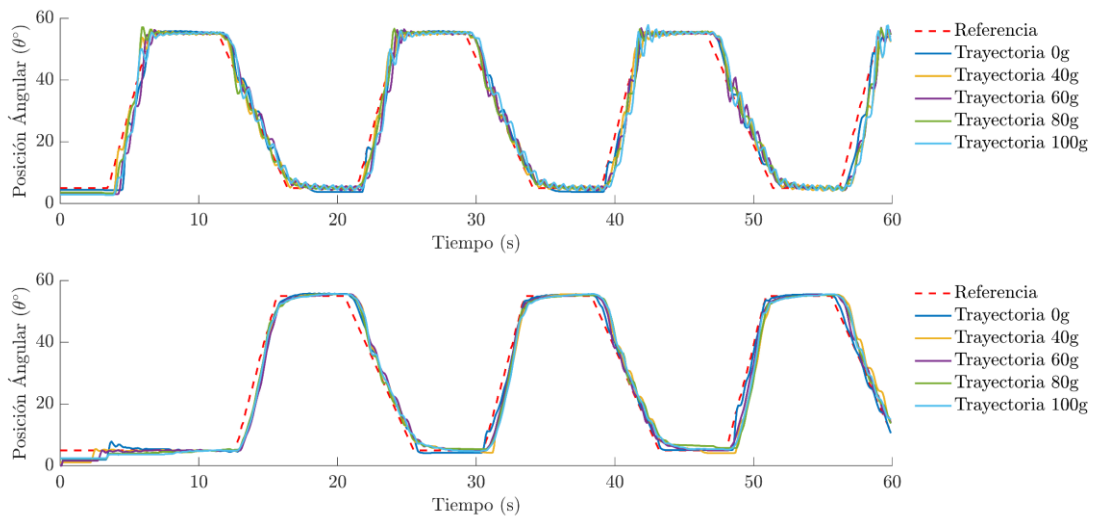


Figura. 4.1 Seguimiento de la posición angular del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.

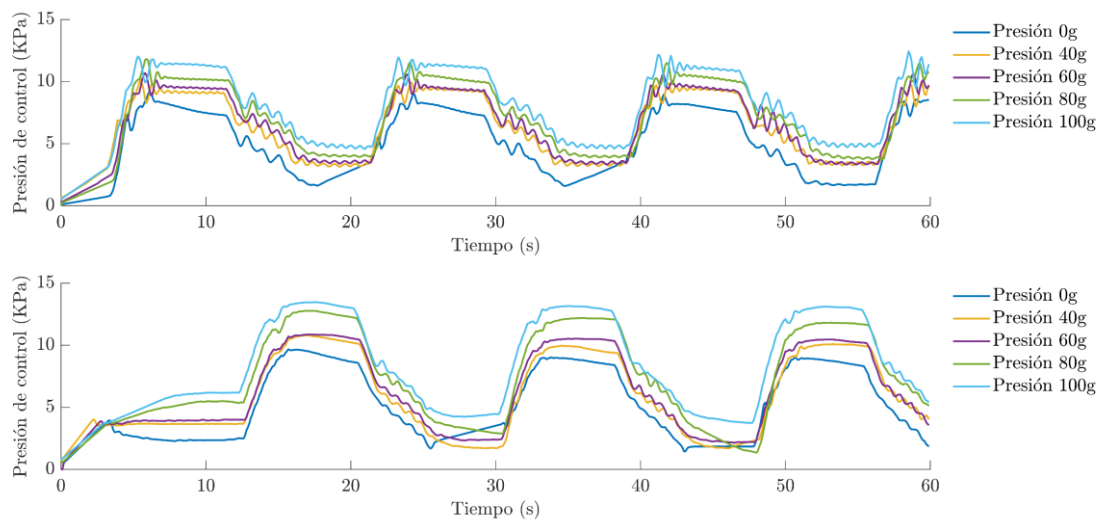


Figura. 4.2 Presión de control del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.

Tabla 4.1. Rangos de la señal de error para el controlador PI clásico.

Masas (g)	Controlador PI	
	Pierna A	Pierna B
0	$[-17.37^\circ, 11.56^\circ]$	$[-14.75^\circ, 7.12^\circ]$
40	$[-13.31^\circ, 7.50^\circ]$	$[-11.68^\circ, 8.81^\circ]$
60	$[-13.62^\circ, 7.68^\circ]$	$[-11.18^\circ, 8.43^\circ]$
80	$[-15.31^\circ, 8.31^\circ]$	$[-14.25^\circ, 8.56^\circ]$
100	$[-14.37^\circ, 9.06^\circ]$	$[-15.75^\circ, 9.18^\circ]$

4.2. Resultados del diseño del controlador difuso

4.2.1. Resultados de la identificación y ajuste de los parámetros estructurales

Se tomaron los datos experimentales (entrada/salida) de la Pierna B bajo un esquema de control de lazo cerrado (Controlador PI), ejecutado en la etapa anterior. El entrenamiento se llevó a cabo con una red neuronal que cuenta con 12 capas ocultas, 1200 épocas, una tasa de aprendizaje de 0.012 y un algoritmo de optimización de Adam (Figura 4.4) en un tiempo de 1 min. 4 segundos. Los resultados usando los datos de validación se muestran en las Figuras 4.5, 4.6 y 4.7. Se obtuvo un porcentaje de precisión de NRMSE = 92.399% durante la validación del modelo de red neuronal LSTM.

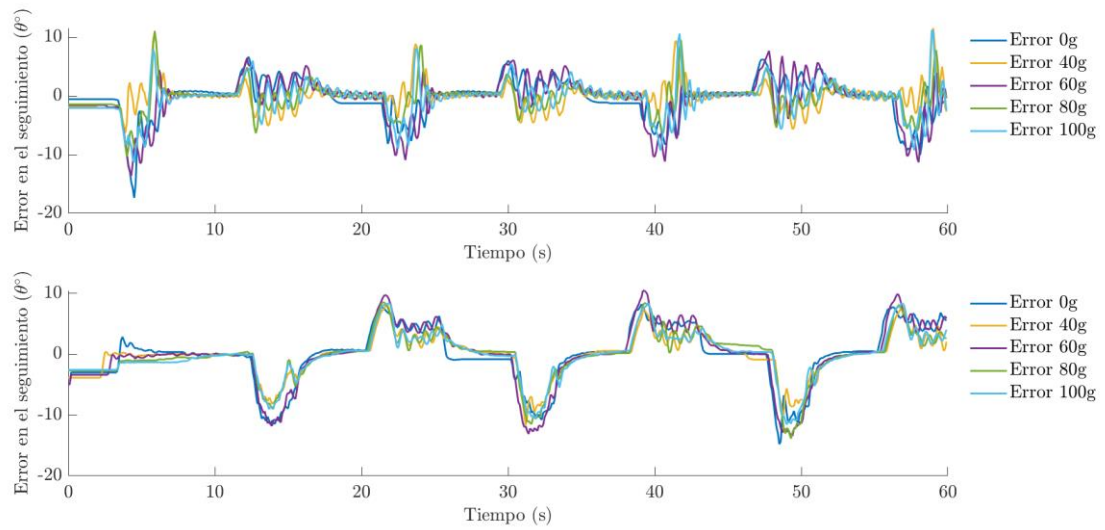


Figura. 4.3 Error de la posición angular del controlador PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando diferentes masas de prueba.

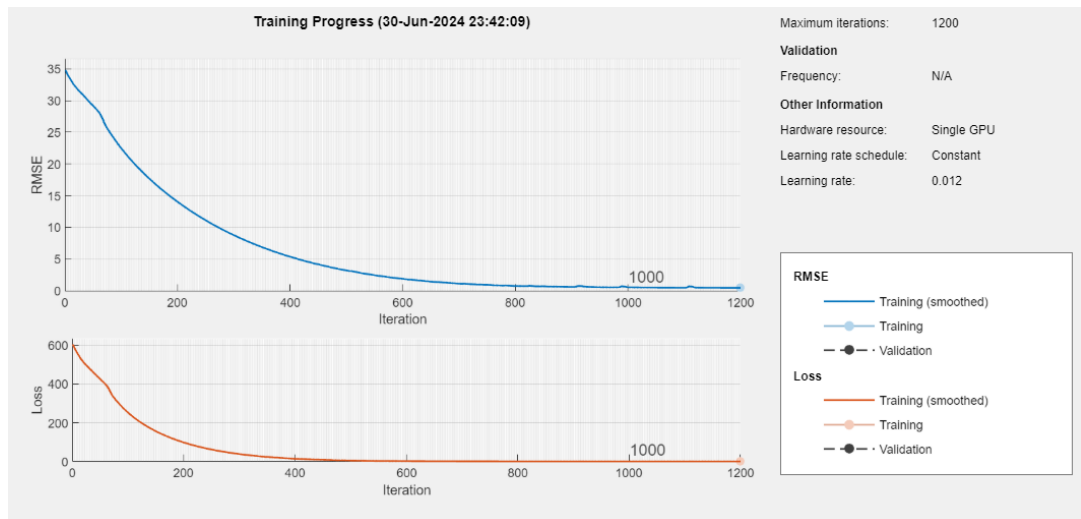


Figura. 4.4 Proceso de entrenamiento de la red neuronal LSTM.

En la Figura. 4.5 se observa que el modelo simulado sigue perfectamente los datos de validación experimentales. La relación entre la presión y la posición angular para el modelo simulado y los datos de validación experimentales se observan en la Figura. 4.6. Se observa una diferencia entre la fase de extensión y la fase de flexión debido a las no linealidades presentes en el sistema. El comportamiento del error es estable a excepción de algunos picos que llegaron hasta los 10° (Figura. 4.7).

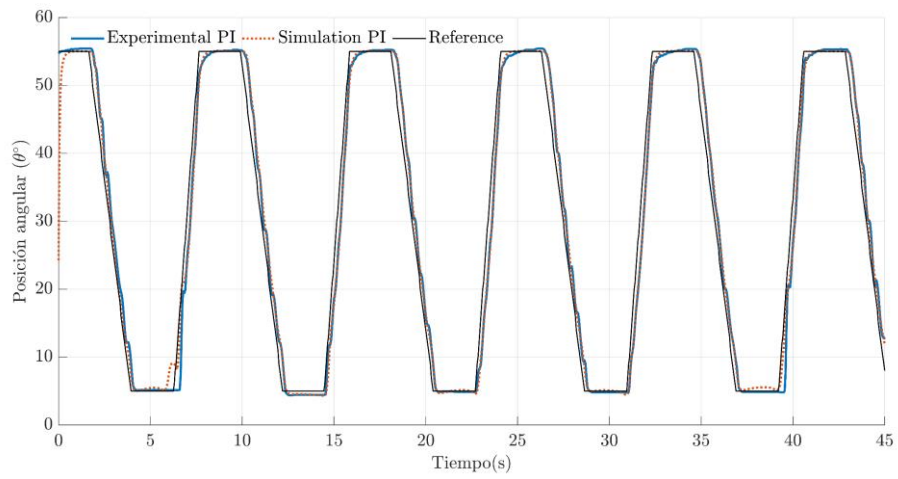


Figura. 4.5 Comportamiento de la salida real y simulada del sistema.

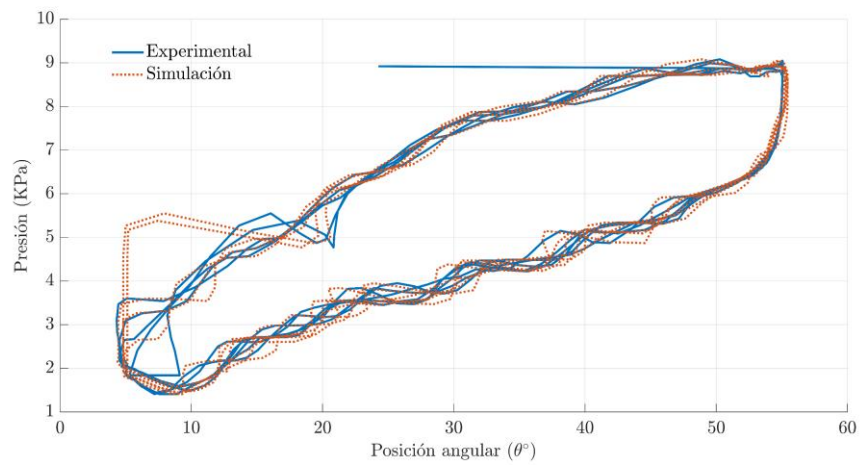


Figura. 4.6 Comportamiento de la Presión versus Posición Angular.

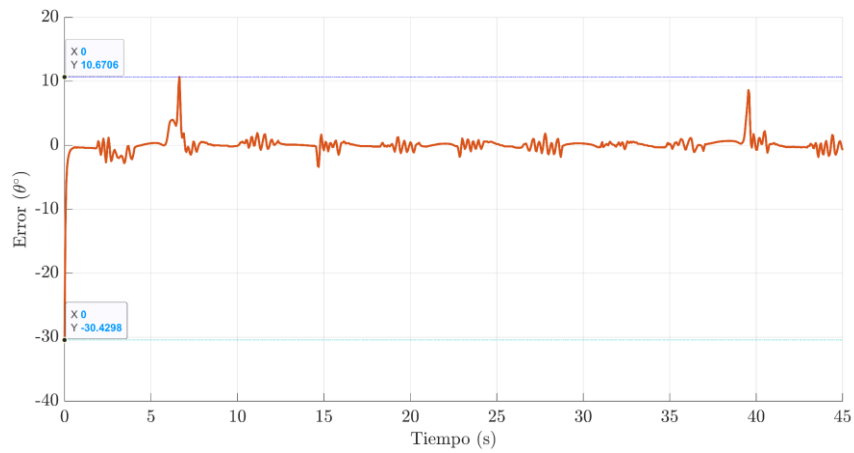


Figura. 4.7 Comportamiento del error del modelo identificado.

El modelo identificado se integró en un programa desarrollado en SIMULINK con la finalidad de ajustar los parámetros del controlador difuso PI (véase Figura. 4.8). Los parámetros estructurales del controlador difuso PI son los universos de discurso de las entradas (e y Δe), las salidas (Δu), y el factor de escala K_c , ya que la distribución de las funciones de membresía ya ha sido definida de manera uniforme para todas las variables.

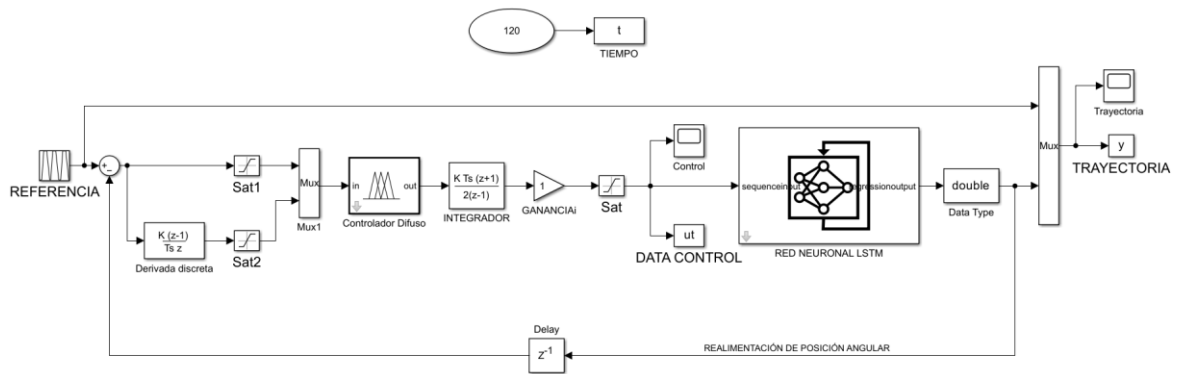


Figura. 4.8 Programa en SIMULINK del controlador difuso PI.

(Fuente: Elaboración propia)

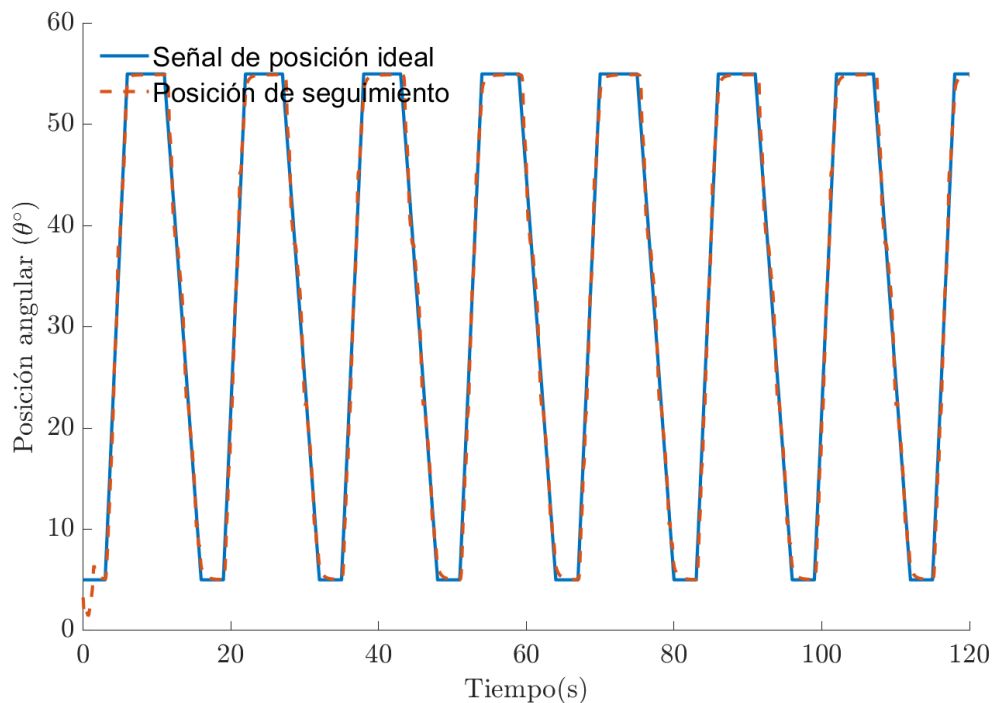


Figura. 4.9 Simulación de la trayectoria del controlador difuso PI.

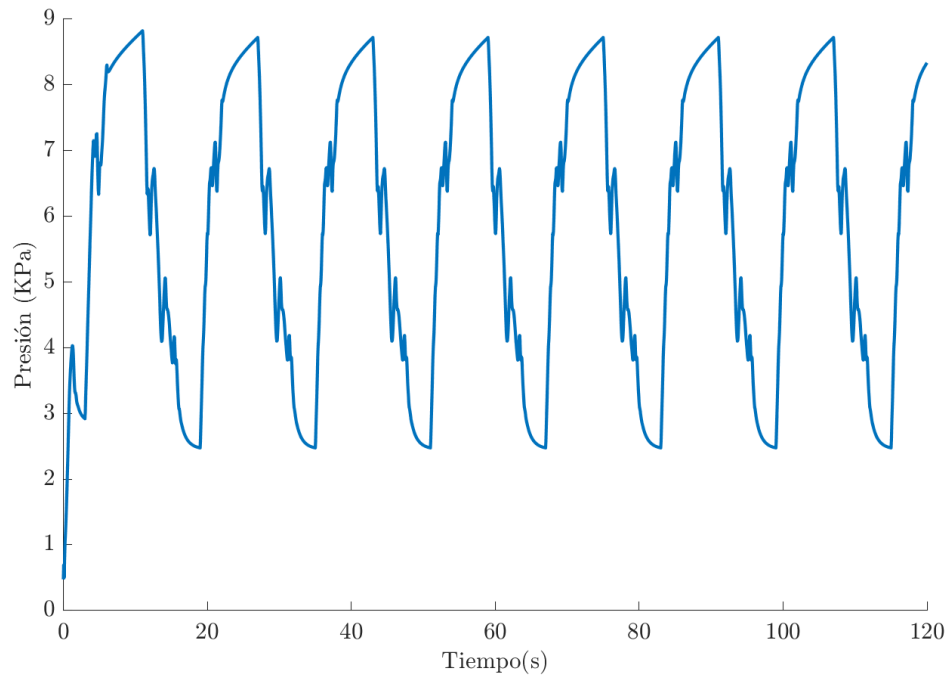


Figura. 4.10 Simulación de la señal de control usando un control difuso PI.

4.2.2. Resultados del cálculo de las acciones de control del controlador difuso PI

Después de varias iteraciones, se llegó a obtener los parámetros que generaban el menor error en el seguimiento de la trayectoria (véase Figura. 4.9). El factor de escala K_c toma un valor de 1 durante las simulaciones. Los universos de discurso para e y Δe son elegidos como $[-50, 50]$ grados y $[-250, 250]$ grados/seg respectivamente. El universo de discurso de la salida Δu es elegido como $[-50, 50]$ KPa/seg. La señal de control tiene un comportamiento estable como se muestra en la Figura. 4.10. Las acciones de control del controlador difuso PI se visualiza en la superficie de inferencia difusa (véase Figura. 4.11).

4.2.3. Resultados del desempeño del controlador difuso PI

El controlador difuso presenta una señal de error que oscila entre $[-6^\circ, 9^\circ]$, y se mantiene estable durante todo el tiempo de simulación (véase Figura. 4.12).

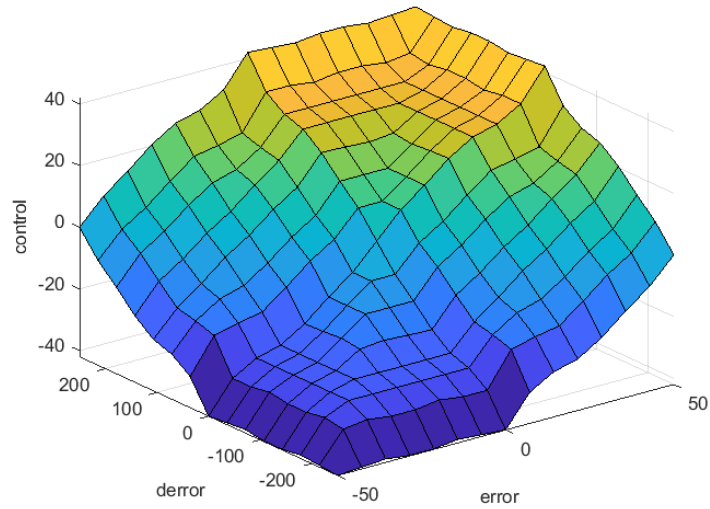


Figura. 4.11 Superficie de control durante la simulación.

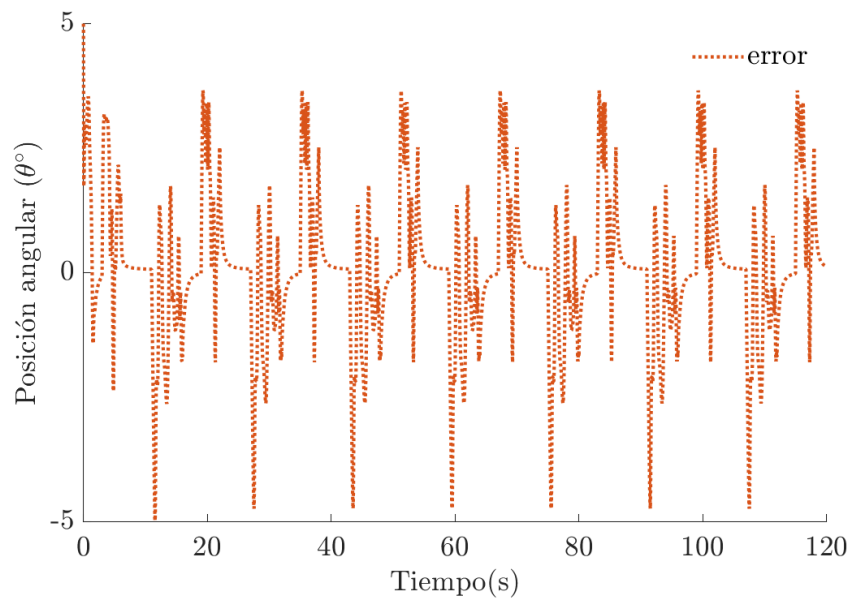


Figura. 4.12 Señal de error del seguimiento de la trayectoria durante la simulación.

4.2.4. Resultados del ajuste basado en experiencia del controlador difuso PI

Después de múltiples experimentos, los universos de discurso que mostraron mejores resultados para e y Δe son $[-44, 44]$ grados y $[-260, 260]$ grados/seg respectivamente. El universo de discurso de la salida Δu se ajustó hasta $[-45, 45]$ KPa/seg. Los conjuntos difusos se distribuyen proporcionalmente a lo largo del universo de discurso (Figuras. 4.13, 4.14, 4.15). El factor de escala K_c seleccionado fue 1.

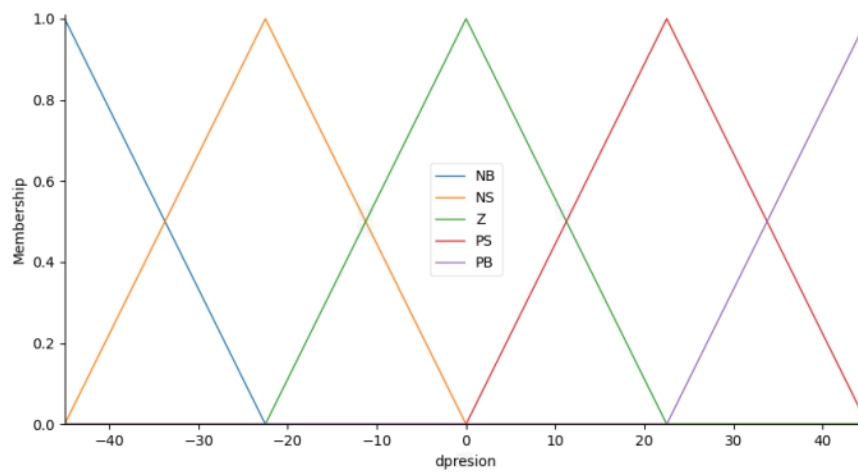


Figura. 4.13 Conjuntos difusos de la derivada de la presión.

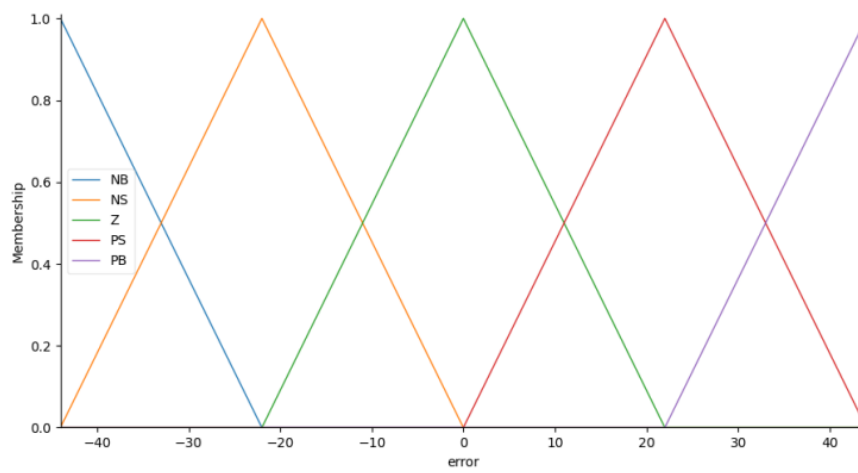


Figura. 4.14 Conjuntos difusos del error.

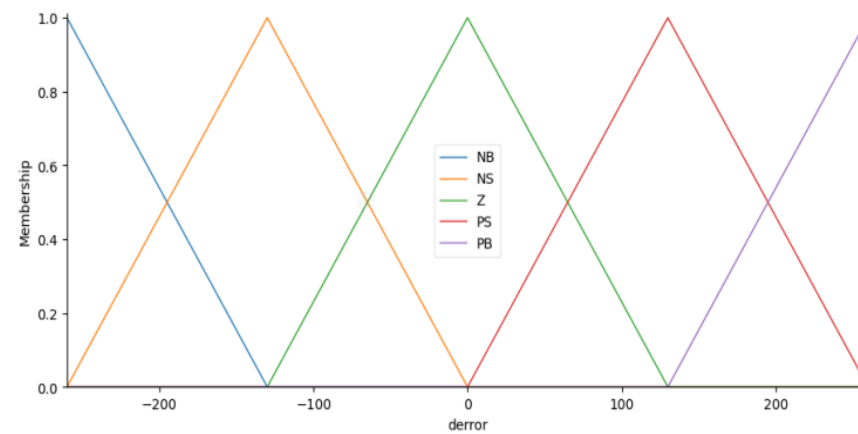


Figura. 4.15 Conjuntos difusos de la derivada del error.

4.3. Resultados de la estabilidad y robustez frente a diferentes perturbaciones del controlador difuso PI.

Se ejecutaron pruebas con el prototipo por un intervalo de 5 minutos. El seguimiento de la trayectoria de rehabilitación cuando el sistema no tiene carga adicional para ambas piernas se muestra en la Figura. 4.16. Se observa un comportamiento oscilante durante la fase de extensión de ambas piernas, siendo más pronunciado en la Pierna A. Después de agregar 100 gramos adicionales por pierna, el controlador permanece estable y aún puede dar seguimiento efectivo a la señal de referencia. Las señales de presión que controlan a los reguladores electrónicos se muestran en las Figura. 4.17. La presión de control oscila entre [0, 9.51] Kpa en la Pierna A, y entre [0, 9.89] Kpa en la Pierna B, cuando el sistema está libre de carga adicional. Cuando adicionamos 100 gramos en cada pierna, la presión de control oscila entre [0, 13.46] Kpa en la Pierna A, y entre [0, 14.82] Kpa en la Pierna B. La señal de error del seguimiento de la trayectoria (Figura. 4.18) oscila entre [-14.50°, 10.00°] para la Pierna A, y entre [-16.43°, 8.50°] para la Pierna B, cuando se ejecuta el ejercicio sin masas adicionales. Después de adicionar 100 gramos en cada pierna el error oscila entre [-11.12°, 8.25°] para la Pierna A, y entre [-9.68°, 7.18°] para la Pierna B.

Tabla 4.2. Señal de error para el controlador difuso PI.

Masas (g)	Controlador difuso PI	
	Pierna A	Pierna B
0	[-14.50°, 10.00°]	[-16.43°, 8.50°]
40	[-21.06°, 9.18°]	[-14.12°, 12.62°]
60	[-19.31°, 13.43°]	[-17.50°, 11.75°]
80	[-21.18°, 11.75°]	[-22.50°, 14.62°]
100	[-11.12°, 8.25°]	[-9.68°, 7.18°]

Tabla 4.3. Presión de control para el controlador difuso PI.

Masas (g)	Controlador difuso PI	
	Pierna A	Pierna B
0	[0, 9.51] Kpa	[0, 9.89] Kpa
40	[0, 11.09] Kpa	[0, 12.06] Kpa
60	[0, 12.38] Kpa	[0, 11.62] Kpa
80	[0, 12.42] Kpa	[0, 12.51] Kpa
100	[0, 13.46] Kpa	[0, 14.82] Kpa

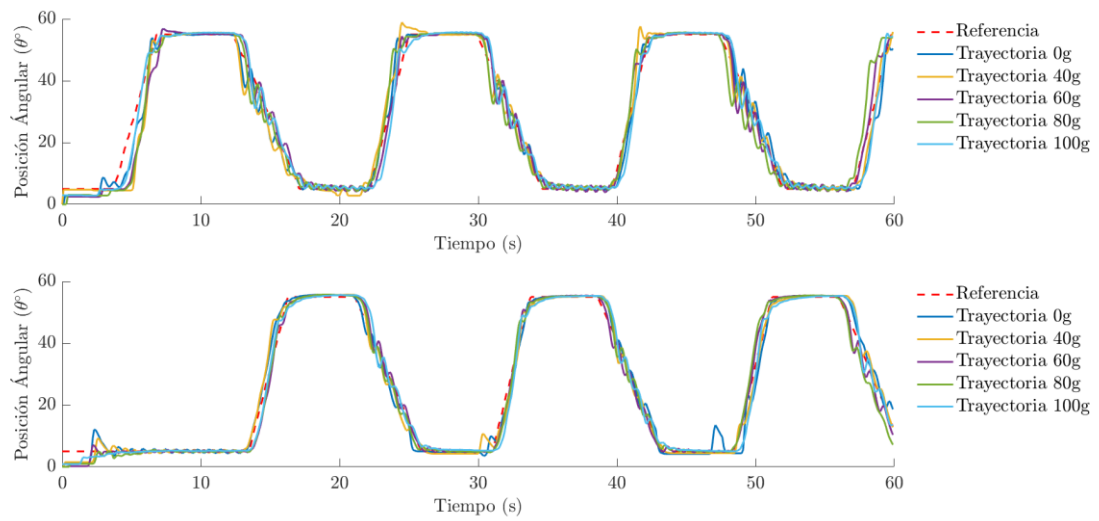


Figura. 4.16 Seguimiento de la posición angular del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.

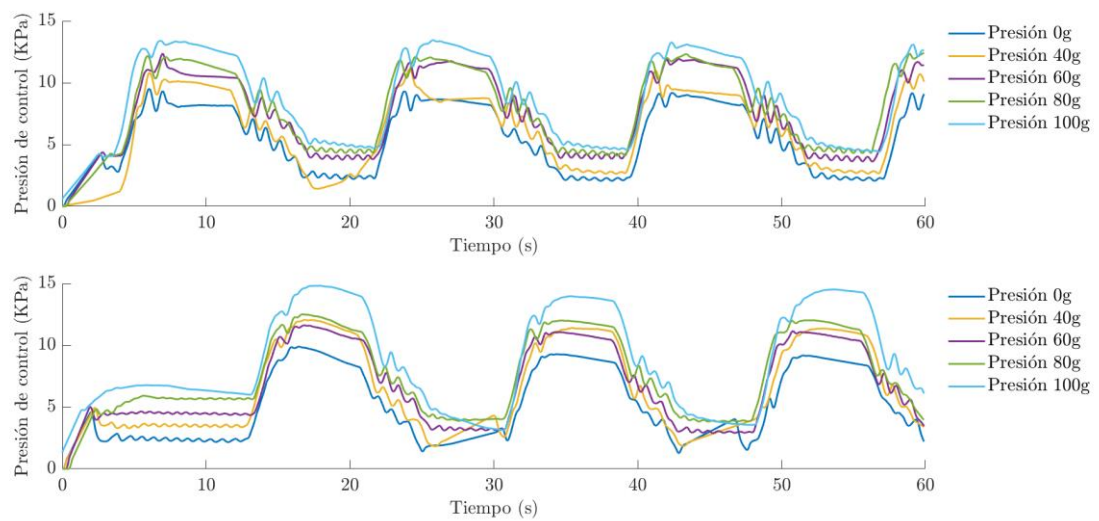


Figura. 4.17 Presión de control del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando distintas masas de prueba.

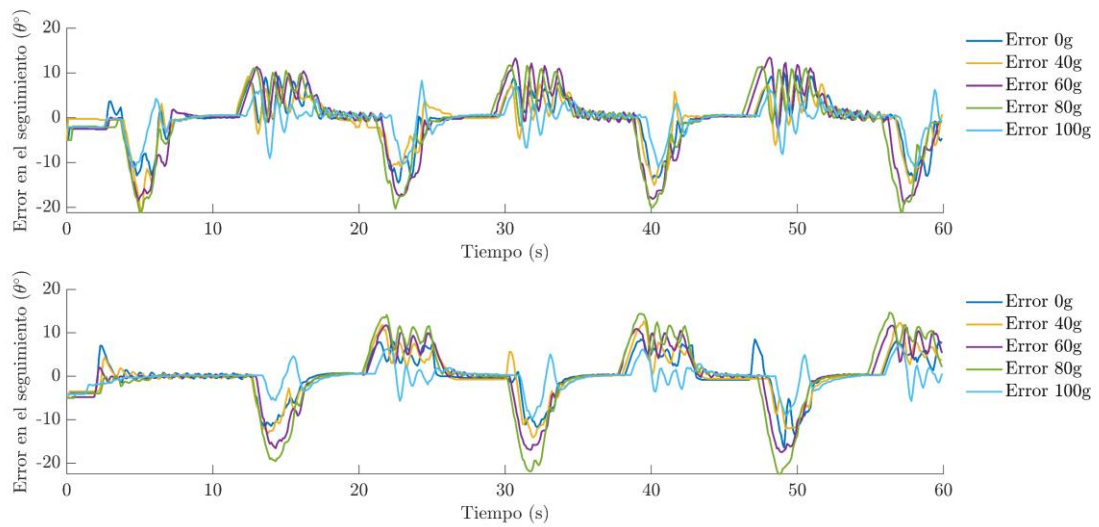


Figura. 4.18 Error de la posición angular del controlador difuso PI en ambas piernas (superior: pierna A, inferior: pierna B) considerando diferentes masas de prueba.

4.4. Análisis y discusión de los resultados

- En la Tabla 4.3 se observan los valores de NRMSE para el control PI y control difuso PI cuando realizamos experimentos usando diferentes masas. La diferencia entre el NRMSE de ambos controladores. Los valores de NRMSE de 0 y 100 gramos son respectivamente 84.734% y 84.583% para la pierna A, 88.415% y 88.172% para la pierna B, utilizando el controlador difuso PI. Por otro lado, para el controlador PI, los valores de NRMSE de 0 y 100 gramos son 84.210% y 77.059% respectivamente para la pierna A, 89.092% y 82.555% para la pierna B. En conclusión, el controlador difuso PI mantiene su robustez aún con 100 gramos adicionales, en cambio, el controlador PI reduce su rendimiento bajo las mismas condiciones de operación.
- Las señales de presión y las señales de error se mantienen estables en el tiempo durante todos los experimentos realizados con el controlador difuso PI.
- El comportamiento del controlador difuso PI presenta ligeras oscilaciones durante la fase de extensión de ambas piernas. Estas oscilaciones son más significativas en la pierna A, incluso es observable en el patrón de la señal de presión. Por otro lado, el controlador PI presenta oscilaciones de menor magnitud en la pierna A, que se ven incrementadas cuando se agregan cuando se agregan masas mayores a 40 gramos. Este comportamiento ligeramente oscilatorio en una de las piernas se debe a las incertidumbres en la fabricación del músculo artificial.

- El rendimiento del controlador difuso PI está afectado debido a que la tasa de muestreo es mucho más oscilatoria durante la ejecución del algoritmo en comparación del controlador PI clásico. En el caso del algoritmo de control PI, se observa un mejor control de la tasa de muestreo, permaneciendo casi constante y cercano al valor asignado de 0.05.

Tabla 4.4. Valores de NRMSE para el controlador difuso PI y PI clásico.

Masas (g)	NRMSE			
	Controlador PI		Controlador Difuso PI	
	Pierna A	Pierna B	Pierna A	Pierna B
0	84.210%	89.092%	84.734%	88.415%
40	82.506%	81.300%	80.152%	88.767%
60	79.326%	84.849%	82.602%	90.479%
80	81.914%	82.766%	78.222%	88.708%
100	77.059%	82.555%	84.583%	88.172%

4.5. Contratación de la hipótesis

El controlador difuso PI diseñado presenta un error en el seguimiento de la trayectoria y una robustez superior a un controlador PI, cuando se trabajan con masas de 0 a 100 gramos como se muestra en la Tabla 4.3. Por ende, la hipótesis principal planteada inicialmente resulta válida considerando las condiciones propuestas en el experimento.

CONCLUSIONES

1. Se selecciono una Raspberry Pi modelo 4B, 2GB RAM, 1.8GHz como tarjeta de adquisición, procesamiento y control del sistema. La posición angular de ambas piernas se midió empleando un sensor IMU BNO055 que se ajusta fácilmente en la pierna del bebé debido a su tamaño reducido. El control de la flexión/extensión de los músculos artificiales de impulso por vacío requiere de dos reguladores electroneumáticos SMC ITV2091-21N2BS5, una bomba de vacío RS-2, un tanque de reserva de vacío, conexiones neumáticas con ajuste a presión, y tubos neumáticos. Cada regulador electroneumático depende la excitación de un convertidor digital/análogo (DAC) MCP4725 de 12 bits (soporta comunicación I2C). El seguimiento del control se visualiza en una pantalla de LCD de 14", que se comunica a la tarjeta de control Raspberry Pi usando un puerto mini HMI. La programación con Python facilita el desarrollo debido a sus librerías disponibles, que incluyen funciones de adquisición de datos a través de I2C, y control difuso.
2. El error durante el seguimiento de la trayectoria en ambas piernas del maniquí se evaluó a través del parámetro NRMSE, calculado para cada experimento. El control difuso PI mantiene un rendimiento superior comparado con el control PI clásico si comparamos el error del seguimiento de la trayectoria de referencia. El control difuso PI presenta ligeras oscilaciones durante la fase de extensión de ambas piernas, por otro lado, el control PI, solo presenta oscilaciones significativas en una sola pierna. A pesar de ello, ambos controladores cumplen los límites de error requeridos por la aplicación. Este rendimiento se mantiene si ambos controladores trabajan sobre los límites de diseño locales establecidos para la aplicación.
3. La robustez del sistema se evaluó considerando la modificación de las masas de ambas piernas del maniquí. Se evaluó el parámetro NRMSE por pierna para cada uno de los experimentos realizados con diferentes masas. Después de adicionar 100 gramos por pierna, el controlador difuso PI (84.583% y 88.172%) presentó una robustez superior al controlador PI clásico (77.059% y 82.555%), manteniendo el error de seguimiento de la trayectoria de referencia estable.

RECOMENDACIONES

1. El músculo artificial neumático utilizado presenta incertidumbres elevadas debido a la fabricación, esto afecta gravemente el rendimiento del controlador, por lo tanto, se requiere una estandarización del método de fabricación y la utilización material más robustos.
2. Como indicamos, la gravedad tiene el efecto de incrementar las no linealidades del sistema, sobre todo si se requieren velocidades angulares considerables. Este efecto se mitiga agregando una fuga artificial en el actuador. Energéticamente hablando la fuga es un fenómeno no deseado dentro de un sistema mecánico ya que representa una pérdida de presión innecesaria en el sistema. Esto lleva a efectos como el incremento de la presión y energía que utiliza el sistema. Por lo tanto, se recomienda eliminar la fuga artificial del actuador, y aplicar un método de compensación de la gravedad en caso se usen estrategias de control libres de modelos, o un control no lineal en caso se usen estrategias de control basadas en modelo. En casos se utilicen métodos de compensación de la gravedad, las incertidumbres se compensan utilizando estrategias de control robusto.
3. El potencial del controlador difuso PI debe ser explorado usando una programación de más bajo nivel como ANSI C o C++, ya que de esa manera se tiene un mejor control de la tasa de muestreo del controlador.
4. Llevar a cabo las tareas de adquisición, procesamiento, control y visualización en una Raspberry Pi afecta la tasa de muestreo del controlador, en consecuencia, su rendimiento. Por lo tanto, es recomendable separar las etapas y realizar las actividades con diferentes sistemas embebidos.

BIBLIOGRAFÍA

- [1] **Agnanto, D., Arifin, A., & Babgei, A. F.** (2021). Multi-input Multi-output Fuzzy Logic Controller for Hybrid Exoskeleton and Functional Electrical Stimulation for Hand Movements Rehabilitation of Hemiparesis Patients. Proceedings - 2021 International Seminar on Intelligent Technology and Its Application: Intelligent Systems for the New Normal Era, ISITIA 2021. PP 272–277. doi: 10.1109/ISITIA52817.2021.9502200.
- [2] **Aliman, N., Ramli, R., Mohamed Haris, S., et al.** (2022). A robust adaptive-fuzzy-proportional-derivative controller for a rehabilitation lower limb exoskeleton. Engineering Science and Technology, an International Journal. vol. 35. PP 101097. doi: 10.1016/J.JESTCH.2022.101097.
- [3] **Aliman, N., Ramli, R., Mohamed Haris, S., et al.** (2022). A robust adaptive-fuzzy-proportional-derivative controller for a rehabilitation lower limb exoskeleton. Engineering Science and Technology, an International Journal. vol. 35. PP 101097. doi: 10.1016/J.JESTCH.2022.101097.
- [4] **Asín-Prieto, G., José Urendes Jiménez, E., Rodríguez Cianca, D., et al.** (2022). Development of a Series Elastic Tendon Actuator (SETA) Based on Gait Analysis for a Knee Assistive Exosuit. Actuators 2022, Vol. 11, Page 166. vol. 11.no. 6. PP 166. doi: 10.3390/ACT11060166.
- [5] **Baud, R., Manzoori, A. R., Ijspeert, A., et al.** (2021). Review of control strategies for lower - limb exoskeletons to assist gait. Journal of NeuroEngineering and Rehabilitation. PP 1–34. doi: 10.1186/s12984-021-00906-3.
- [6] **Cappello, L., Binh, D. K., Yen, S., et al.** (2016). Design and Preliminary Characterization of a Soft Wearable Exoskeleton for Upper Limb.

- [7] **Castelli, E.** (2011). Robotic movement therapy in cerebral palsy. *Developmental medicine and child neurology*. vol. 53.no. 6. PP 481. doi: 10.1111/J.1469-8749.2011.03987.X.
- [8] **Centeno, E., Mendoza, M. J., Pinedo, C. F., et al.** (2023). Incremental PI-like Fuzzy Logic Control of a Vacuum-Powered Artificial Muscle for Soft Exoskeletons. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*. vol. 2023-June. PP 53–58. doi: 10.1109/ARSO56563.2023.10187413.
- [9] **Chen, C. F., Du, Z. J., He, L., et al.** (2019). Active Disturbance Rejection with Fast Terminal Sliding Mode Control for a Lower Limb Exoskeleton in Swing Phase. *IEEE Access*. vol. 7. PP 72343–72357. doi: 10.1109/ACCESS.2019.2918721.
- [10] **Chen, C., Zhang, Y., Li, Y., et al.** (2020). Iterative Learning Control for a Soft Exoskeleton with Hip and Knee Joint Assistance. *Sensors* 2020, Vol. 20, Page 4333. vol. 20.no. 15. PP 4333. doi: 10.3390/S20154333.
- [11] **Chen, L., Wang, C., Wang, J., et al.** (2019). Design of hybrid phase sliding mode control scheme for lower extremity exoskeleton. *Applied Sciences (Switzerland)*. vol. 9.no. 18. doi: 10.3390/app9183754.
- [12] **Chen, Z., Li, Z., & Chen, C. L. P.** (2017). Disturbance Observer-Based Fuzzy Control of Uncertain MIMO Mechanical Systems with Input Nonlinearities and its Application to Robotic Exoskeleton. *IEEE Transactions on Cybernetics*. vol. 47.no. 4. PP 984–994. doi: 10.1109/TCYB.2016.2536149.
- [13] **Chhabra, H., Mohan, V., Rani, A., et al.** (2020). Robust nonlinear fractional order fuzzy PD plus fuzzy I controller applied to robotic manipulator. *Neural Computing and Applications*. vol. 32.no. 7. PP 2055–2079. doi: 10.1007/S00521-019-04074-3/METRICS.
- [14] **Chiaradia, D., Xiloyannis, M., Solazzi, M., et al.** (2020). Rigid Versus Soft Exoskeletons: Interaction Strategies for Upper Limb Assistive Technology. *Wearable Robotics: Systems and Applications*. PP 67–90. doi: 10.1016/B978-0-12-814659-0.00004-7.

- [15] **Chiaradia, D., Xiloyannis, M., Solazzi, M., et al.** (2019). Comparison of a soft exosuit and a rigid exoskeleton in an assistive task. vol. 22. Springer International Publishing. PP 415–419.
- [16] **Cimolin, V., Germiniasi, C., Galli, M., et al.** (2018). Robot-Assisted Upper Limb Training for Hemiplegic Children with Cerebral Palsy. *Journal of Developmental and Physical Disabilities* 2018 31:1. vol. 31.no. 1. PP 89–101. doi: 10.1007/S10882-018-9632-Y.
- [17] **Copp, A. J., Adzick, N. S., Chitty, L. S., et al.** (2015). Spina bifida. *Nature Reviews Disease Primers* 2015 1:1. vol. 1.no. 1. PP 1–18. doi: 10.1038/nrdp.2015.7.
- [18] **Daachi, M. E., Madani, T., Daachi, B., et al.** (2015). A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis. *Applied Soft Computing*. vol. 34. PP 324–336. doi: 10.1016/J.ASOC.2015.04.034.
- [19] **Dinh, B. K., Xiloyannis, M., Antuvan, C. W., et al.** (2017). Hierarchical Cascade Controller for Assistance Modulation in a Soft Wearable Arm Exoskeleton. vol. 3766.no. c. PP 1–8. doi: 10.1109/LRA.2017.2668473.
- [20] **Driankov, D., Hellendoorn, H., & Reinfrank, M.** (1993). *FKBC Design Parameters. An Introduction to Fuzzy Control*. PP 103–144. doi: 10.1007/978-3-662-11131-4_3.
- [21] **Falzarano, V., Marini, F., Morasso, P., et al.** (2019). Devices and Protocols for Upper Limb Robot-Assisted Rehabilitation of Children with Neuromotor Disorders. *Applied Sciences* 2019, Vol. 9, Page 2689. vol. 9.no. 13. PP 2689. doi: 10.3390/APP9132689.
- [22] **Gervasoni, S., Lussi, J., Viviani, S., et al.** (2022). Magnetically Assisted Robotic Fetal Surgery for the Treatment of Spina Bifida. *IEEE Transactions on Medical Robotics and Bionics*. vol. 4.no. 1. PP 85–93. doi: 10.1109/TMRB.2022.3146351.
- [23] **Gevers, M.** (2005). Identification for Control: From the Early Achievements to the Revival of Experiment Design*. *European Journal of Control*. vol. 11.no. 4–5. PP 335–352. doi: 10.3166/EJC.11.335-352.

- [24] **Gmelig Meyling, C., Verschuren, O., Rentinck, I. R., et al.** (2022). Physical rehabilitation interventions in children with acquired brain injury: a scoping review. *Developmental medicine and child neurology*. vol. 64.no. 1. PP 40–48. doi: 10.1111/DMCN.14997.
- [25] **Gold, J. T. & Salsberg, D. H.** (2021). *Pediatric Disorders: Cerebral Palsy and Spina Bifida. Medical Aspects of Disability for the Rehabilitation Professional*. doi: 10.1891/9780826132284.0015.
- [26] **Gollob, S. D., Mendoza, M. J., Koo, B. H. B., et al.** (2023). A length-adjustable vacuum-powered artificial muscle for wearable physiotherapy assistance in infants. *Frontiers in Robotics and AI*. vol. 10. PP 1190387. doi: 10.3389/FROBT.2023.1190387/BIBTEX.
- [27] **Gollob, S. D., Park, C., Koo, B. H. B., et al.** (2021). A Modular Geometrical Framework for Modelling the Force-Contraction Profile of Vacuum-Powered Soft Actuators. *Frontiers in Robotics and AI*. vol. 8. PP 606938. doi: 10.3389/FROBT.2021.606938/BIBTEX.
- [28] **Gonzalez, J. & Yu, W.** (2018). Non-linear system modeling using LSTM neural networks. *IFAC-PapersOnLine*. vol. 51.no. 13. PP 485–489. doi: 10.1016/J.IFACOL.2018.07.326.
- [29] **Goo, A., Laubscher, C. A., Farris, R. J., et al.** (2020). Design and Evaluation of a Pediatric Lower-Limb Exoskeleton Joint Actuator. PP 1–16.
- [30] **Gulati, S. & Sondhi, V.** (2018). Cerebral Palsy: An Overview. *Indian journal of pediatrics*. vol. 85.no. 11. PP 1006–1016. doi: 10.1007/S12098-017-2475-1.
- [31] **Gull, M. A., Bai, S., & Bak, T.** (2020). A Review on Design of Upper Limb Exoskeletons. *Robotics 2020*, Vol. 9, Page 16. vol. 9.no. 1. PP 16. doi: 10.3390/ROBOTICS9010016.
- [32] **Gull, M. A., Bai, S., & Bak, T.** (2020). A Review on Design of Upper Limb Exoskeletons. *Robotics 2020*, Vol. 9, Page 16. vol. 9.no. 1. PP 16. doi: 10.3390/ROBOTICS9010016.

- [33] **Hadders-Algra, M.** (2018). Early human motor development: From variation to the ability to vary and adapt. *Neuroscience and biobehavioral reviews*. vol. 90. PP 411–427. doi: 10.1016/J.NEUBIOREV.2018.05.009.
- [34] **Han, S., Wang, H., & Tian, Y.** (2018). Adaptive computed torque control based on RBF network for a lower limb exoskeleton. *Proceedings - 2018 IEEE 15th International Workshop on Advanced Motion Control, AMC 2018*. PP 35–40. doi: 10.1109/AMC.2019.8371059.
- [35] **Hasan, S. K.** (2022). Radial basis function-based exoskeleton robot controller development. *IET Cyber-Systems and Robotics*. vol. 4.no. 3. PP 228–250. doi: 10.1049/CSY2.12057.
- [36] **Ismail, R., Ariyanto, M., Perkasa, I. A., et al.** (2019). Soft Elbow Exoskeleton for Upper Limb Assistance Incorporating Dual Motor-Tendon Actuator. *Electronics* 2019, Vol. 8, Page 1184. vol. 8.no. 10. PP 1184. doi: 10.3390/ELECTRONICS8101184.
- [37] **Jandial, R.** (2020). Primary Myelomeningocele Closure. *Core Techniques in Operative Neurosurgery*. PP 489–493. doi: 10.1016/B978-0-323-52381-3.00092-7.
- [38] **Jeong, J., Yasir, I. Bin, Han, J., et al.** (2019). Design of Shape Memory Alloy-Based Soft Wearable Robot for Assisting Wrist Motion. *Applied Sciences* 2019, Vol. 9, Page 4025. vol. 9.no. 19. PP 4025. doi: 10.3390/APP9194025.
- [39] **Jonsson, U., Eek, M. N., Sunnerhagen, K. S., et al.** (2019). Cerebral palsy prevalence, subtypes, and associated impairments: a population-based comparison study of adults and children. *Developmental medicine and child neurology*. vol. 61.no. 10. PP 1162–1167. doi: 10.1111/DMCN.14229.
- [40] **Ka, M., Hong, D., Huu, C., et al.** (2015). Minimizing Human-Exoskeleton Interaction Force using Compensation for Dynamic Uncertainty Error with Adaptive RBF Network. *Journal of Intelligent & Robotic Systems*. doi: 10.1007/s10846-015-0251-x.

- [41] **Kim, S. K., Park, D., Yoo, B., et al.** (2021). Overground Robot-Assisted Gait Training for Pediatric Cerebral Palsy. *Sensors* (Basel, Switzerland). vol. 21.no. 6. PP 1–12. doi: 10.3390/S21062087.
- [42] **Koch, M. A. & Font-Llagunes, J. M.** (2021). Lower-limb exosuits for rehabilitation or assistance of human movement: A systematic review. *Applied Sciences* (Switzerland). vol. 11.no. 18. doi: 10.3390/app11188743.
- [43] **Koh, T. H., Cheng, N., Yap, H. K., et al.** (2017). Design of a soft robotic elbow sleeve with passive and intent-controlled actuation. *Frontiers in Neuroscience*. vol. 11.no. OCT. PP 597. doi: 10.3389/FNINS.2017.00597/BIBTEX.
- [44] **Kovacic, Z. & Bogdan, S.** (2021). Fuzzy Controller Design. *Fuzzy Controller Design*. PP 25–90. doi: 10.1201/9781420026504-5/FUZZY-CONTROLLER-DESIGN-ZDENKO-KOVACIC-STJEPAN-BOGDAN.
- [45] **Kwon, J., Park, J. H., Ku, S., et al.** (2019). A Soft Wearable Robotic Ankle-Foot-Orthosis for Post-Stroke Patients. *IEEE Robotics and Automation Letters*. vol. 4.no. 3. PP 2547–2552. doi: 10.1109/LRA.2019.2908491.
- [46] **Lee, D. K. & Sansom, J. K.** (2019). Early Treadmill Practice in Infants Born With Myelomeningocele: A Pilot Study. *Pediatric physical therapy : the official publication of the Section on Pediatrics of the American Physical Therapy Association*. vol. 31.no. 1. PP 68–75. doi: 10.1097/PEP.0000000000000554.
- [47] **Li, Z., Ren, Z., Zhao, K., et al.** (2020). Human-Cooperative Control Design of a Walking Exoskeleton for Body Weight Support. *IEEE Transactions on Industrial Informatics*. vol. 16.no. 5. PP 2985–2996. doi: 10.1109/TII.2019.2900121.
- [48] **Ljung, L., Andersson, C., Tiels, K., et al.** (2020). Deep Learning and System Identification. *IFAC-PapersOnLine*. vol. 53.no. 2. PP 1175–1181. doi: 10.1016/J.IFACOL.2020.12.1329.
- [49] **Long, Y. & Peng, Y.** (2022). Extended State Observer-Based Nonlinear Terminal Sliding Mode Control With Feedforward Compensation for Lower Extremity

Exoskeleton. IEEE Access. vol. 10. PP 8643–8652. doi: 10.1109/ACCESS.2021.3049879.

- [50] **Ma, Y.** (2020). Motion stability enhanced controller design. Dynamics and Advanced Motion Control of Off-Road UGVs. PP 191–218. doi: 10.1016/B978-0-12-818799-9.00008-6.
- [51] **Mefoued, S. & Belkhiat, D. E. C.** (2019). A Robust Control Scheme Based on Sliding Mode Observer to Drive a Knee-Exoskeleton. Asian Journal of Control. vol. 21.no. 1. PP 439–455. doi: 10.1002/ASJC.1950.
- [52] **Mendoza, M. J., Gollob, S. D., Lavado, D., et al.** (2021). A vacuum-powered artificial muscle designed for infant rehabilitation. Micromachines. vol. 12.no. 8. PP 971. doi: 10.3390/M12080971/S1.
- [53] **ben Miled, S., Loeuillet, L., Duong Van Huyen, J. P., et al.** (2020). Severe and progressive neuronal loss in myelomeningocele begins before 16 weeks of pregnancy. American journal of obstetrics and gynecology. vol. 223.no. 2. PP 256.e1-256.e9. doi: 10.1016/J.AJOG.2020.02.052.
- [54] **Mucchiani, C., Liu, Z., Sahin, I., et al.** (2022). Closed-loop Position Control of a Pediatric Soft Robotic Wearable Device for Upper Extremity Assistance. PP 1514–1519. doi: 10.1109/RO-MAN53752.2022.9900650.
- [55] **Narayan, J., Abbas, M., & Dwivedy, S. K.** (2022). Robust adaptive backstepping control for a lower-limb exoskeleton system with model uncertainties and external disturbances. <https://doi.org/10.1080/00051144.2022.2119498>. vol. 64.no. 1. PP 145–161. doi: 10.1080/00051144.2022.2119498.
- [56] **Narendra, K. S. & Parthasarathy, K.** (1990). Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks. vol. 1.no. 1. PP 4–27. doi: 10.1109/72.80202.
- [57] **Nassour, J., Zhao, G., & Grimmer, M.** (2021). Soft pneumatic elbow exoskeleton reduces the muscle activity, metabolic cost and fatigue during holding and carrying

of loads. Scientific Reports 2021 11:1. vol. 11.no. 1. PP 1–14. doi: 10.1038/s41598-021-91702-5.

- [58] **Nassour, J., Zhao, G., & Grimmer, M.** (2021). Soft pneumatic elbow exoskeleton reduces the muscle activity, metabolic cost and fatigue during holding and carrying of loads. Scientific Reports 2021 11:1. vol. 11.no. 1. PP 1–14. doi: 10.1038/s41598-021-91702-5.
- [59] **Nelles, O.** (2001). Linear Dynamic System Identification. Nonlinear System Identification. PP 457–546. doi: 10.1007/978-3-662-04323-3_14.
- [60] **Nguyen, H. T.** (2003). A first course in fuzzy and neural control. PP 301.
- [61] **Niu, J., Song, Q., & Wang, X.** (2013). Fuzzy PID control for passive lower extremity exoskeleton in swing phase. ICEIEC 2013 - Proceedings of 2013 IEEE 4th International Conference on Electronics Information and Emergency Communication. PP 185–188. doi: 10.1109/ICEIEC.2013.6835483.
- [62] **Oliveira, A. C. D., Rose, C. G., Warburton, K., et al.** (2019). Exploring the Capabilities of Harmony for Upper-Limb Stroke Therapy. IEEE ... International Conference on Rehabilitation Robotics : [proceedings]. vol. 2019. PP 637–643. doi: 10.1109/ICORR.2019.8779558.
- [63] **Palanivel, V. & Burrough, M.** (2021). Acquired brain injury in children, and their rehabilitation: where we are now? Paediatrics and Child Health. vol. 31.no. 5. PP 176–180. doi: 10.1016/J.PAED.2021.02.001.
- [64] **Pamungkas, D. S., Caesarendra, W., Susanto, S., et al.** Overview: Types of lower limb exoskeletons, *Electronics (Switzerland)*, vol. 8.no. 11. MDPI AG, Nov. 01, 2019. doi: 10.3390/electronics8111283.
- [65] **Pamungkas, D. S., Caesarendra, W., Susanto, S., et al.** (2019). Overview: Types of lower limb exoskeletons. *Electronics (Switzerland)*. vol. 8.no. 11. doi: 10.3390/electronics8111283.

- [66] **Pan, D., Gao, F., Miao, Y., et al.** (2015). Co-simulation research of a novel exoskeleton-human robot system on humanoid gaits with fuzzy-PID/PID algorithms. *Advances in Engineering Software*. vol. 79. PP 36–46. doi: 10.1016/J.ADVENGSOFT.2014.09.005.
- [67] **Pérez Vidal, A. F., Rumbo Morales, J. Y., Ortiz Torres, G., et al.** (2021). Soft exoskeletons: Development, requirements, and challenges of the last decade. *Actuators*. vol. 10.no. 7. PP 1–26. doi: 10.3390/act10070166.
- [68] **Putu Riantini, N., Budi Rahayu, U., Waspada, E., et al.** (2023). Physiotherapy Management in Spina Bifida Cases. *FISIO MU: Physiotherapy Evidences*. vol. 4.no. 3. PP 193–200. doi: 10.23917/FISIOMU.V4I3.22023.
- [69] **Qassim, H. M. & Wan Hasan, W. Z.** (2020). A Review on Upper Limb Rehabilitation Robots. *Applied Sciences* 2020, Vol. 10, Page 6976. vol. 10.no. 19. PP 6976. doi: 10.3390/APP10196976.
- [70] **Rahmani, M. & Rahman, M. H.** (2019). Adaptive Neural Network Fast Fractional Sliding Mode Control of a 7-DOF Exoskeleton Robot. *International Journal of Control, Automation and Systems* 2019 18:1. vol. 18.no. 1. PP 124–133. doi: 10.1007/S12555-019-0155-1.
- [71] **Rehmat, N., Zuo, J., Meng, W., et al.** (2018). Upper limb rehabilitation using robotic exoskeleton systems: a systematic review. *International Journal of Intelligent Robotics and Applications*. vol. 2.no. 3. PP 283–295. doi: 10.1007/s41315-018-0064-8.
- [72] **Rezage, G. al & Tokhi, M. O.** (2016). Fuzzy PID control of lower limb exoskeleton for elderly mobility. 2016 20th IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2016 Proceedings. doi: 10.1109/AQTR.2016.7501310.
- [73] **Romero-Ayuso, D.** (2021). Future Challenges in Research in Children with Neurodevelopmental Disorders. *Children*. vol. 8.no. 5. doi: 10.3390/CHILDREN8050328.

- [74] **Rozensztrauch, A., Iwańska, M., & Baglaj, M.** (2021). The Quality of Life of Children with Myelomeningocele: A Cross-Sectional Preliminary Study. *International journal of environmental research and public health*. vol. 18.no. 20. doi: 10.3390/IJERPH182010756.
- [75] **Schneider, J., Mohr, N., Aliatakis, N., et al.** (2021). Brain malformations and cognitive performance in spina bifida. *Developmental Medicine and Child Neurology*. vol. 63.no. 3. PP 295–302. doi: 10.1111/DMCN.14717/ABSTRACT.
- [76] **Schwedersky, B. B., Flesch, R. C. C., & Dangui, H. A. S.** (2019). Practical Nonlinear Model Predictive Control Algorithm for Long Short-Term Memory Networks. *IFAC-PapersOnLine*. vol. 52.no. 1. PP 468–473. doi: 10.1016/J.IFACOL.2019.06.106.
- [77] **Sharma, R., Gaur, P., Bhatt, S., et al.** (2021). Optimal fuzzy logic-based control strategy for lower limb rehabilitation exoskeleton. *Applied Soft Computing*. vol. 105. PP 107226. doi: 10.1016/J.ASOC.2021.107226.
- [78] **Siddique, N.** (2014). Mathematics of fuzzy control. *Studies in Computational Intelligence*. vol. 517. PP 57–93. doi: 10.1007/978-3-319-02135-5_4/COVER.
- [79] **Siddique, N.** (2014). Introduction. *Studies in Computational Intelligence*. vol. 517. PP 1–9. doi: 10.1007/978-3-319-02135-5_1/COVER.
- [80] **Siddique, N.** (2014). Fuzzy control. *Studies in Computational Intelligence*. vol. 517. PP 95–135. doi: 10.1007/978-3-319-02135-5_5/COVER.
- [81] **Soares, L. M. dos S., Rozane, J. M. S. G., & Carvalho, R. de P.** (2019). Motor performance of children with cerebral palsy in anterior reach. *Clinical biomechanics (Bristol, Avon)*. vol. 68. PP 158–162. doi: 10.1016/J.CLINBIOMECH.2019.06.007.
- [82] **Sun, J., Wang, J., Yang, P., et al.** (2020). Adaptive Finite Time Control for Wearable Exoskeletons Based on Ultra-local Model and Radial Basis Function Neural Network. *International Journal of Control, Automation and Systems* 2020 19:2. vol. 19.no. 2. PP 889–899. doi: 10.1007/S12555-019-0975-7.

- [83] **Teng, L. & Bai, S.** (2019). Fuzzy Sliding Mode Control of An Upper-Limb Exoskeleton Robot. Proceedings of the IEEE 2019 9th International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics, CIS and RAM 2019. PP 12–17. doi: 10.1109/CIS-RAM47153.2019.9095811.
- [84] **Thapar, A. & Rutter, M.** (2015). Neurodevelopmental disorders. Rutter's Child and Adolescent Psychiatry: Sixth Edition. PP 31–40. doi: 10.1002/9781118381953.CH3.
- [85] **Tran, H. T., Cheng, H., Rui, H., et al.** (2016). Evaluation of a Fuzzy-Based Impedance Control Strategy on a Powered Lower Exoskeleton. International Journal of Social Robotics. vol. 8.no. 1. PP 103–123. doi: 10.1007/S12369-015-0324-9/METRICS.
- [86] **Ullah, K., Basit, A., Ullah, Z., et al.** (2021). Automatic Generation Control Strategies in Conventional and Modern Power Systems : A Comprehensive Overview.
- [87] **Wang, A., Li, W., & yu, J.** (2021). PD Controller of a Lower Limb Exoskeleton Robot Based on Sliding Mode RBF Neural Network. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. vol. 387. PP 427–436. doi: 10.1007/978-3-030-82562-1_40/COVER.
- [88] **Wang, T., Zhang, B., Liu, C., et al.** (2022). A Review on the Rehabilitation Exoskeletons for the Lower Limbs of the Elderly and the Disabled. Electronics (Switzerland). vol. 11.no. 3. PP 1–16. doi: 10.3390/electronics11030388.
- [89] **Wu, Q., Chen, B., & Wu, H.** (2019). Neural-network-enhanced torque estimation control of a soft wearable exoskeleton for elbow assistance. Mechatronics. vol. 63. PP 102279. doi: 10.1016/J.MECHATRONICS.2019.102279.
- [90] **Wu, Q., Wang, X., Chen, B., et al.** (2018). Design and Fuzzy Sliding Mode Admittance Control of a Soft Wearable Exoskeleton for Elbow Rehabilitation. IEEE Access. vol. 6. PP 60249–60263. doi: 10.1109/ACCESS.2018.2875550.
- [91] **Wu, Q., Zhang, Y., & Chen, Y.** (2022). Design, control, and experimental verification of a soft knee exoskeleton for rehabilitation during walking. Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control

Engineering. vol. 236.no. 1. PP 138–152. doi: 10.1177/09596518211014325/ASSET/IMAGES/LARGE/10.1177_09596518211014325-FIG18.JPEG.

- [92] **Wu, Q., Zhang, Y., & Chen, Y.** (2021). Design, control, and experimental verification of a soft knee exoskeleton for rehabilitation during walking. <https://doi.org/10.1177/09596518211014325>. vol. 236.no. 1. PP 138–152. doi: 10.1177/09596518211014325.
- [93] **Xiloyannis, M., Annese, E., Canesi, M., et al.** (2019). Design and validation of a modular one-to-many actuator for a soft wearable exosuit. *Frontiers in Neurorobotics*. vol. 13. PP 39. doi: 10.3389/FNBOT.2019.00039/BIBTEX.
- [94] **Xiloyannis, M., Chiaradia, D., Frisoli, A., et al.** (2019). Physiological and kinematic effects of a soft exosuit on arm movements. *Journal of neuroengineering and rehabilitation*. vol. 16.no. 1. doi: 10.1186/S12984-019-0495-Y.
- [95] **Yang, Y., Huang, D., & Dong, X.** (2019). Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning. *Neurocomputing*. vol. 323. PP 256–264. doi: 10.1016/J.NEUCOM.2018.09.085.
- [96] **Ye, X., Chen, C., Shi, Y., et al.** (2021). A Time Division Multiplexing Inspired Lightweight Soft Exoskeleton for Hip and Ankle Joint Assistance. PP 1–18.
- [97] **Yi, Y., Lindemann, M., Colligs, A., et al.** (2011). Economic burden of neural tube defects and impact of prevention with folic acid: a literature review. *European journal of pediatrics*. vol. 170.no. 11. PP 1391–1400. doi: 10.1007/S00431-011-1492-8.
- [98] **Yu, H., Choi, I. S., Han, K. L., et al.** (2018). Development of a upper-limb exoskeleton robot for refractory construction. *Control Engineering Practice*. vol. 72. PP 104–113. doi: 10.1016/J.CONENGPRAC.2017.09.003.
- [99] **Zhang, B., Liu, S., Li, Y., et al.** (2021). Sliding Mode Control Algorithm of Upper Limb Exoskeleton Rehabilitation Robot Based on RBF Neural Network. 11th International Conference on Intelligent Control and Information Processing, ICICIP 2021. PP 35–40. doi: 10.1109/ICICIP53388.2021.9642219.

- [100] **Zhang, L., Huang, Q., Cai, K., et al.** (2020). A Wearable Soft Knee Exoskeleton Using Vacuum-Actuated Rotary Actuator. *IEEE Access*. vol. 8. PP 61311–61326. doi: 10.1109/ACCESS.2020.2983790.
- [101] **Zhang, P., Zhang, J., & Zhang, Z.** (2020). Design of RBFNN-Based Adaptive Sliding Mode Control Strategy for Active Rehabilitation Robot. *IEEE Access*. vol. 8. PP 155538–155547. doi: 10.1109/ACCESS.2020.3018737.
- [102] **Zhang, X., Wang, H., Tian, Y., et al.** (2018). Model-free based neural network control with time-delay estimation for lower extremity exoskeleton. *Neurocomputing*. vol. 272. PP 178–188. doi: 10.1016/J.NEUCOM.2017.06.055.
- [103] **Zhao, W. & Song, A.** (2020). Active Motion Control of a Knee Exoskeleton Driven by Antagonistic Pneumatic Muscle Actuators. *Actuators 2020*, Vol. 9, Page 134. vol. 9.no. 4. PP 134. doi: 10.3390/ACT9040134.
- [104] **Zhou, J., Yang, S., & Xue, Q.** (2021). Lower limb rehabilitation exoskeleton robot: A review. *Advances in Mechanical Engineering*. vol. 13.no. 4. PP 1–17. doi: 10.1177/16878140211011862.

ANEXOS

A.1 Pseudocódigo controlador difuso PI

1:	Inicializar variables;
2:	Definir funciones de membresía (triangulares) y universos de discurso de e, de, y du; <ul style="list-style-type: none"> • Universo de discurso de “e” = [-a, a]; • Universo de discurso de “de” = [-b, b]; • Universo de discurso de “de” = [-c, c];
3:	Definir las 25 reglas difusas;
4:	Generar modelo del controlador difuso C_1 ;
5:	Mientras $N \leq \text{Número}_{\text{ciclos}}$:
6:	$\text{tiempo} = \text{obtener}(\text{tiempo}_{\text{transcurrido}})$;
7:	$\text{ángulo} = \text{obtener}(\text{posición angular}_{\text{actual}})$;
8:	$\text{Target} = \text{generar}(\text{trayectoria})$;
9:	$\text{tiempo}_{\text{actual}} = \text{obtener}(\text{tiempo}_{\text{transcurrido}}) - \text{tiempo}_{\text{pasado}}$;
10:	$\text{error} = \text{Target} - \text{ángulo}$;
11:	Derivar: $d_{\text{error}} = \frac{\text{error} - \text{error}_{\text{previo}}}{\text{tiempo}_{\text{actual}}}$
12:	$\text{error}_{\text{previo}} = \text{error}$;
13:	Actualizar entrada y saturar controlador: $C_1(\text{error}) = \text{mínimo}(\text{máximo}(\text{error}, -a), a)$;
14:	Actualizar entrada y saturar controlador: $C_1(d_{\text{error}}) = \text{mínimo}(\text{máximo}(d_{\text{error}}, -b), b)$;
15:	Computar simulación del controlador C_1 ;
16:	Actualizar salida: $du_{\text{control}} = \text{salida}(C_1)$;
17:	Integrar du_{control} : $u_{\text{control}} = u_{\text{control}} + \text{tiempo}_{\text{actual}} * \left(\frac{du_{\text{control}} + \text{prev} du_{\text{control}}}{2} \right)$;
18:	Generar señal de control total: $u_{\text{control}_t} = u_{\text{control}} * Ki$
19:	$\text{prev} du_{\text{control}} = du_{\text{control}}$;
20:	$\text{tiempo}_{\text{pasado}} = \text{tiempo}_{\text{actual}}$;
21:	Acondicionar señal: $\text{Voltaje}_{\text{dac}} = \frac{5 * 65535}{\text{Vol}_{\text{rasp}} * \text{limite}_{\text{regulador}}}$
22:	Actualizar valor de Voltaje: $\text{Valor}_{\text{dac}} = \text{Voltaje}_{\text{dac}}$

A.2 Código fuente para la identificación en MATLAB

```
% -----  
% Use LSTM Network for nonlinear System Identification  
% -----  
close all; clc; clear;  
%  
type PID2_1_1_0_intensivo.csv  
PID2_1_1_0_intensivo = readmatrix('PID2_1_1_0_intensivo.csv');  
save('PID2_1_1_0_intensivo.mat')  
load PID2_1_1_0_intensivo.mat  
  
% data  
y = PID2_1_1_0_intensivo(:,2);  
u = PID2_1_1_0_intensivo(:,4);  
  
M = 1;  
C = 1000;  
y = y(M:C);  
u = u(M:C);  
  
Ts = 0.05;  
z = iddata(y,u,Ts);  
zv = z(M:C);  
  
figure(1);  
  
subplot(211);  
plot(y,'k','linewidth',2);  
title('Salida');  
xlabel('Muestras');ylabel('Posición angular (°)');  
  
subplot(212);  
plot(u,'r','linewidth',2);  
title('Entrada');  
xlabel('Muestras');ylabel('Presión (KPa)')
```

```
% Preprocessing data -> deep learning toolbox
% -----
XTrain=cell(1,1);
YTrain=cell(1,1);
XTrain{1}=zv.u';
YTrain{1}=zv.y';
```

```
% Edit Options Neural Net
% -----
numHiddenUnits=13;
featureDimension=1; % u(t)
```

```
layers=[ ...
sequenceInputLayer(featureDimension), ...
lstmLayer(numHiddenUnits, ...
'OutputMode','sequence'), ...
fullyConnectedLayer(1), ...
regressionLayer];
```

```
options=trainingOptions('adam', ...
'MaxEpochs',1000, ...
'InitialLearnRate',0.012, ...
'Plots','training-progress');
```

```
% Train neural network command
% -----
[net,tr]=trainNetwork(XTrain,YTrain,layers,options);
```

```
%% Plot testing results
% -----
load PID2_1_1_0_intensivo.mat
```

```
y = PID2_1_1_0_intensivo(:,2);
u = PID2_1_1_0_intensivo(:,4);
```

```

M = 1001;
P = 1905;
uv = u(M:P);
uv = uv';
yv = predict(net,uv);
yv = yv';
% y = y(:,2);
ys = y(M:P);
e = yv - ys;
cost_func = 'NRMSE';
fit=goodnessOfFit(yv,ys,cost_func);
fitpercent_train = 100*(1-fit);

figure
plot(e,'r','linewidth',2);
legend('error');
xlabel('Muestras');ylabel('ÁNGULO(°)');
figure
plot(yv,uv,'r:',ys,uv,'b','linewidth',2)
legend('Simulación','Real');
xlabel('Posición angular (°)');ylabel('Presión de control (KPa)');
figure
plot(ys,'b','linewidth',2)
xlabel('Muestras');ylabel('Posición angular (°)');

hold on
plot(yv,'r:',linewidth,2)
legend('Real','Simulada');

```

A.3 Código fuente para la adquisición, control y visualización en Python

```
## Controlador difuso PI, 25 rules, rehabilitación intensiva, ambas piernas
#####

import time
import board
import busio
import adafruit_bno055
import adafruit_mcp4725
import sys
import datetime
import csv
import os

import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.gridspec import GridSpec
import threading
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.widgets import Button
from matplotlib.widgets import TextBox

import time
from random import randrange

import skfuzzy as fuzz
from skfuzzy import control as ctrl

from functions_fuzzy_controller import fuzzy_controller25
from tracking_signal import trapezoidal

# Configuramos la gráfica
fig = plt.figure(figsize=(10,8))
gs = GridSpec(2, 2, figure=fig)
```

```

axth1 = plt.subplot(gs[0, 0])
axth1.grid(True)
axth2 = plt.subplot(gs[0, 1])
axth2.grid(True)
axu1 = plt.subplot(gs[1, 0])
axu1.grid(True)
axu2 = plt.subplot(gs[1, 1])
axu2.grid(True)

axth1.set_xlabel('Muestras')
axth1.set_ylabel('Ángulo pierna A (grados)')
axth1.set_ylim(-5, 90)
axth1.set_xlim(0, 400)
axth2.set_xlabel('Muestras')
axth2.set_ylabel('Ángulo pierna B (grados)')
axth2.set_ylim(-5, 90)
axth2.set_xlim(0, 400)

axu1.set_xlabel('Muestras')
axu1.set_ylabel('Presion pierna A(kPa)')
axu1.set_ylim(-5, 20)
axu1.set_xlim(0, 400)
axu2.set_xlabel('Muestras')
axu2.set_ylabel('Presion pierna B (kPa)')
axu2.set_ylim(-5, 20)
axu2.set_xlim(0, 400)

line_th1, = axth1.plot([], [], lw=2)
line_th2, = axth2.plot([], [], lw=2)
line_t1, = axth1.plot([], [], lw=2)
line_t2, = axth2.plot([], [], lw=2)
line_u1, = axu1.plot([], [], lw=2)
line_u2, = axu2.plot([], [], lw=2)

def init():

```

```

# Creamos un plot/frame
line_th1.set_data([], [])
line_th2.set_data([], [])
line_t1.set_data([], [])
line_t2.set_data([], [])
line_u1.set_data([], [])
line_u2.set_data([], [])
return line_th1,line_t1,line_u1,line_th2,line_t2,line_u2

gtheta1 = []
gtheta2 = []
gframe = []
t1_m = []
t2_m = []
u1_m =[]
u2_m =[]

# Iniciamos el bus I2C
i2c = busio.I2C(board.SCL, board.SDA)

# Iniciamos los sensores BNO055 de ambas piernas
bno2 = adafruit_bno055.BNO055_I2C(i2c, address=0x28)
bno2.mode = adafruit_bno055.NDOF_MODE
bno1 = adafruit_bno055.BNO055_I2C(i2c, address=0x29)
bno1.mode = adafruit_bno055.NDOF_MODE

#####
# Iniciamos los MCP4725 de ambas piernas

dac1 = adafruit_mcp4725.MCP4725(i2c, address=0x60)
dac2 = adafruit_mcp4725.MCP4725(i2c, address=0x63)
# Máximo voltaje de entrada (medido en Raspberry)
Vol_rasp = 5.20 #Voltios.

```

```
#####
# Valores iniciales
limite = 30.0 # Limite de la señal de control [kPa]
u1 = 0.0     # Señal de control Pierna A [kPa]
u2 = 0.0     # Señal de control Pierna B [kPa]
Ki_1 = 0.0   # Factor Ki del controlador en Pierna A
Kd_1 = 0.0   # Factor Kd del controlador en Pierna A
Ki_2 = 0.0   # Factor Ki del controlador en Pierna B
Kd_2 = 0.0   # Factor Kd del controlador en Pierna B
prev_error_1 = 0.0 # Error previo en Pierna A
prev_error_2 = 0.0 # Error previo en Pierna B
prev_control1 = 0 # Señal de control previa del controlador en Pierna A
prev_control2 = 0 # Señal de control previa del controlador en Pierna B
u_control1i = 0 # Parte integral de la señal de control en Pierna A
u_control2i = 0 # Parte integral de la señal de control en Pierna B
tm = 0.050
tv = 0.035

# Conjuntos del controlador en Pierna A
Sa1 = np.array([0.50, 0.50, 0.50]);
Sb1 = np.array([0.50, 0.50, 0.50]);
Sc1 = np.array([0.50, 0.50, 0.50]);

# Limites de las funciones de membresía de controlador de Pierna A
Ta1 = np.array([44, 260, 45])
simulacion1 = fuzzy_controller25(Ta1,Sa1,Sb1,Sc1,tm)

# Conjuntos del controlador en Pierna B
Sa2 = np.array([0.50, 0.50, 0.50]);
Sb2 = np.array([0.50, 0.50, 0.50]);
Sc2 = np.array([0.50, 0.50, 0.50]);

# Limites de las funciones de membresía de controlador de Pierna B
Ta2 = np.array([44, 260, 45])
simulacion2 = fuzzy_controller25(Ta2,Sa2,Sb2,Sc2,tm)
```

```

# Valores máximos y mínimos de ángulo
Amin = np.array([5,5]) # [grados]
Amax = np.array([55,55]) # [grados]

# Settings needed for trapezoidal reference
tsubida = 3 # t2-t1
tbajada = 5 # t4-t3
testatico = 5 # t3-t2
tinicial = 5 # t1

total = tsubida + tbajada + testatico + tinicial

tn_1 = np.array([0,0,0,0,0,0])
tn_1[0] = tinicial
tn_1[1] = tsubida + tn_1[0]
tn_1[2] = testatico + tn_1[1]
tn_1[3] = tbajada + tn_1[2]

tn_2 = np.array([0,0,0,0,0,0])
tn_2[0] = tinicial
tn_2[1] = tsubida + tn_2[0]
tn_2[2] = testatico + tn_2[1]
tn_2[3] = tbajada + tn_2[2]
# Variable de referencia:
target = np.array([0,0])

# Angulo
angulo1 = 0
angulo2 = 0

#Control signal
u1 = 0
u2 = 0

# Función para actualizar las ganancias del controlador
Ki_1 = 1

```

Kd_1 = 0

Ki_2 = 1

Kd_2 = 0

with open('fuzzy2_25_1_5_0_intensivo.csv', 'a', newline='') as csvfile:

field_names1 = ['Sample1', 'Theta1', 'Reference1', 'Control signal1','Muestreo1']

writer = csv.DictWriter(csvfile, fieldnames=field_names1)

writer.writerow({'Sample1': 'Sample1','Theta1': 'Theta1',

'Reference1': 'Reference1', 'Control signal1': 'Control

signal1','Muestreo1':'Muestreo1'})

with open('fuzzy2_25_2_5_0_intensivo.csv', 'a', newline='') as csvfile:

field_names2 = ['Sample2', 'Theta2', 'Reference2', 'Control signal2','Muestreo2']

writer = csv.DictWriter(csvfile, fieldnames=field_names2)

writer.writerow({'Sample2': 'Sample2','Theta2': 'Theta2',

'Reference2': 'Reference2', 'Control signal2': 'Control

signal2','Muestreo2':'Muestreo2'})

This function is called periodically from FuncAnimation

def animate(i, lineth1,linet1,lineu1,lineth2,linet2,lineu2):

global error_1, error_2, prev_error_1, prev_error_2

global derivada_error_2, derivada_error_1

global prev_control1, prev_control2, u_control1, u_control2, u_control1i,u_control2i

global Amin,Amax,tn_1,tn_2

global angulo1, angulo2, target, u1, u2

global gtheta1, gtheta2, gframe, t1_m, u1_m, t2_m, u2_m

t = time.time() - tx #[s]

tin = time.time()

[tn_1, target[0]] = trapezoidal(t,tn_1,target[0],Amin[0],Amax[0],tsubida,tbajada,total)

[tn_2, target[1]] = trapezoidal(t,tn_2,target[1],Amin[1],Amax[1],tsubida,tbajada,total)

Read the orientation of BNO055

#angulo1 = bno1.euler

#angulo2 = bno2.euler

#angulo2 = angulo2[1]

```

#angulo2 = angulo2*(-1) - 7
#angulo1 = angulo1[1] - 15

angulo1 = round(randrange(15, 65))
angulo2 = round(randrange(15, 65))

# Converts orientation to an error value for the PID controller
error_1 = round(target[0] - angulo1,2)
error_2 = round(target[1] - angulo2,2)

# Calculates the integral and derivative terms
d_error_1 = round((error_1 - prev_error_1) / tm, 2)
prev_error_1 = round(error_1, 2)

d_error_2 = round((error_2 - prev_error_2) / tm, 2)
prev_error_2 = round(error_2, 2)

# Actualiza los valores de entrada del controlador difuso y saturamos:
simulacion1.input['error'] = min(max(round(error_1, 2), -Ta1[0]), Ta1[0])
simulacion1.input['derror'] = min(max(round(d_error_1, 2), -Ta1[1]), Ta1[1])
simulacion2.input['error'] = min(max(round(error_2, 2), -Ta2[0]), Ta2[0])
simulacion2.input['derror'] = min(max(round(d_error_2, 2), -Ta2[1]), Ta2[1])

simulacion1.compute()
u_control1 = simulacion1.output['voltaje']
u_control1i = round(u_control1i + (tm * ((u_control1 + prev_control1)/2)), 2)
u_control1t = u_control1i * Ki_1 + u_control1 * Kd_1
prev_control1 = round(u_control1, 2)

simulacion2.compute()
u_control2 = simulacion2.output['voltaje']
u_control2i = round(u_control2i + (tm * ((u_control2 + prev_control2)/2)), 2)
u_control2t = u_control2i * Ki_2 + u_control2 * Kd_2
prev_control2 = round(u_control2, 2)

# Limits the output to the limits of the regulator (0-[limit])

```

```

u1 = min(max(u_control1t, 0), limite)
u2 = min(max(u_control2t, 0), limite)

# Converts voltage to a 12-bit value (MCP4725)
valor1 = int(5 * 65535 / Vol_rasp * u1 / limite)
valor2 = int(5 * 65535 / Vol_rasp * u2 / limite)

# Write the value -> MCP4725
#dac1.value = valor1
#dac2.value = valor2

# Almacenamos los datos
gtheta1.append(angulo1)
gtheta2.append(angulo2)
gframe.append(i)
t1_m.append(target[0])
t2_m.append(target[1])
u1_m.append(u1)
u2_m.append(u2)

lineth1.set_data(range(len(gtheta1)), gtheta1)
linet1.set_data(range(len(gtheta1)), t1_m)
lineu1.set_data(range(len(gtheta1)), u1_m)
lineth2.set_data(range(len(gtheta2)), gtheta2)
linet2.set_data(range(len(gtheta2)), t2_m)
lineu2.set_data(range(len(gtheta2)), u2_m)

if len(gtheta1) & len(gtheta2) > 400:
    gtheta1.pop(0)
    t1_m.pop(0)
    u1_m.pop(0)
    gtheta2.pop(0)
    t2_m.pop(0)
    u2_m.pop(0)

time.sleep(tv)

```

```

tmd = time.time() - tin
#print(tm1)

with open('fuzzy2_25_1_5_0_intensivo.csv', 'a', newline='') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames=field_names1)
    writer.writerow({'Sample1': len(gframe), 'Theta1': angulo1, 'Reference1': target[0],
'Control signal1': u1, 'Muestreo1': tmd})
    with open('fuzzy2_25_2_5_0_intensivo.csv', 'a', newline='') as csvfile:
        writer = csv.DictWriter(csvfile, fieldnames=field_names2)
        writer.writerow({'Sample2': len(gframe), 'Theta2': angulo2, 'Reference2': target[1],
'Control signal2': u2, 'Muestreo2': tmd})

    return lineth1, linet1, lineu1, lineth2, linet2, lineu2

# Tiempo inicial:
tx = time.time()

# Set up plot to call animate() function periodically
try:
    anim = animation.FuncAnimation(fig, animate, init_func=init,
                                  fargs=(line_th1, line_t1, line_u1, line_th2, line_t2, line_u2),
                                  frames=20, interval=tm*100, blit=True)

    plt.show()

except KeyboardInterrupt:
    # quit
    dac1.value = 0
    dac2.value = 0
sys.exit()

```