

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de Estrategias de Control Inteligente para un Sistema de
Soldadura por Fricción con Agitación

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Automatización y Control

Presentado por:

Richard Genaro Cobos Franco

GUAYAQUIL - ECUADOR

Año: 2021

DEDICATORIA

El presente proyecto lo dedico primeramente a mis padres, quienes han sido mi apoyo a lo largo de mi vida y a mi novia quien ha sido un apoyo emocional en toda mi trayectoria universitaria.

AGRADECIMIENTOS

Mi más sincero agradecimiento a los profesores de la ESPOL por haberme impartido sus conocimientos, a mi tutor por haber dedicado su tiempo a mi tesis y por sus consejos que me serán útiles en mi vida académica, a mi revisor de tesis quien me ayudó a mejorarla con sus comentarios y al profesor Wilfredo Angulo quien ayudó considerablemente al proyecto con sus conocimientos.

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; *Richard Genaro Cobos Franco* y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Autor

COMITÉ EVALUADOR



Firmado digitalmente por
**SANTIAGO
DAVID SALAS**

Santiago D. Salas, Ph.D.

PROFESOR TUTOR

**GABRIELA
PAOLA
SALCAN
REYES**

Firmado
digitalmente por
**GABRIELA PAOLA
SALCAN REYES**
Fecha: 2021.06.27
17:51:12 -05'00'

Gabriela P. Salcán, Ph.D.

PROFESOR EVALUADOR

RESUMEN

En este trabajo se propuso una estrategia de control inteligente como alternativa para el proceso de soldadura por fricción con agitación para una aleación de aluminio. Un modelo empírico validado y representado por funciones de transferencia fue reformulado a un conjunto de ecuaciones diferenciales ordinarias, para observar la dinámica de las fuerzas en el dominio tiempo. Se verificó que el proceso en estudio es estable mediante criterios de estabilidad de Lyapunov. Adicionalmente, se definieron fronteras operacionales del proceso, explotando datos experimentales etiquetados disponibles, esto permitió establecer un set-point del sistema en el espacio de fuerzas, localizado en una región con máxima probabilidad de lograr materiales libres de defecto y con buenas propiedades mecánicas. El controlador inteligente fue comparado con un conjunto de controladores PI desacoplados y un MPC lineal para verificar su competitividad. Los experimentos computacionales permitieron verificar la capacidad del sistema de control propuesto para rechazar perturbaciones inducidas y lograr mantener el set-point deseado. Los resultados comprueban que el controlador inteligente puede competir con otras estrategias de control tradicionales, obteniendo el menor error global acumulado durante las simulaciones.

Palabras Clave: Red neuronal recurrente, Control Inteligente, Soldadura por fricción con agitación. Set-point de soldadura libre de defectos.

ABSTRACT

In this work, an intelligent control strategy was proposed as an alternative to the friction stir welding process in the case of an aluminum alloy. A validated empirical model was translated to a set of ordinary differential equations, allowing the observation of the system dynamics in function of the variables of interest. The process was verified to be stable by employing Lyapunov stability theorems. Additionally, process operational boundaries were defined, by exploiting available labeled experimental data, which by themselves, allow the definition of the defect-free set-point in the force space, located in a maximum likelihood region for obtaining defect-free welds and joints with good mechanical properties. The intelligent controller was compared to a set of PI decoupled controllers and a linear MPC to verify its competitiveness. Computational experiments verified the proposed control system capacity to reject induced disturbances, and achieve and maintain the desired set-point. Results prove the intelligent controller can compete with other traditional control frameworks, attaining the least accumulated global error during the simulations.

Keywords: Recurrent neural network, Intelligent control, Friction stir welding, Defect-free set-point.

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS	V
SIMBOLOGÍA	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	X
CAPÍTULO 1	
1	
1. Introducción	1
1.1 Descripción del problema.....	1
1.2 Justificación del problema.....	1
1.3 Objetivos.....	
2	
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos.....	2
1.4 Marco teórico.....	2
1.4.1 Soldadura por fricción con agitación.....	2
1.4.2 Defectos de soldadura.....	4
1.4.3 Control de fuerzas en FSW.....	5
1.4.4 Control de otras variables en FSW.....	5
1.4.5 Redes neuronales artificiales.....	6
1.4.6 Redes neuronales recurrentes.....	7
1.4.7 Gated recurrent unit.....	8
CAPÍTULO 2	
10	
2. Metodología	
10	
2.1 Modelamiento del proceso y selección de punto de referencia.....	10
2.1.1 Modelo dinámico.....	11
2.1.2 Análisis de estabilidad.....	12
2.1.3 Selección de referencia o set-point.....	14

2.2	Arquitectura de control inteligente.....	16
2.2.1	Módulos GRU.....	16
2.2.2	Sistema en lazo cerrado y entrenamiento.....	17
2.3	Experimentos y comparaciones.....	19
2.3.1	Controlador PI.....	20
2.3.2	MPC.....	23
2.3.3	Controlador inteligente.....	25
CAPÍTULO 3		
28		
3.	Resultados Y ANÁLISIS	28
3.1	Comparación de respuestas del sistema en lazo cerrado.....	28
3.2	Comparación del esfuerzo de los controladores.....	31
3.3	Comparación del error acumulado.....	34
3.4	Comparación empleando perturbaciones aleatorias.....	36
3.5	Ventajas y desventajas de cada sistema de control según los resultados obtenidos.....	39
CAPÍTULO 4		
41		
4.	Conclusiones Y Recomendaciones	41
4.1	Conclusiones.....	41
4.2	Recomendaciones.....	43
BIBLIOGRAFÍA		
APÉNDICE		
APÉNDICE A		
APÉNDICE B		
APÉNDICE C		
APÉNDICE D		
APÉNDICE E		

ABREVIATURAS

FSW	Friction Stir Welding (Soldadura por Fricción con Agitación)
ANN	Artificial Neural Network (Red Neuronal Artificial)
ISO	International Organization for Standardization (Organización Internacional de Normalización)
PID	Proporcional – Integrativo – Derivativo
MPC	Model Predictive Control (Control Predictivo basado en Modelo)
CNN	Convolutional Neural Network (Red Neuronal Convolutiva)
SIANN	Shift-invariant Artificial Neural Network (Red Neuronal Artificial Invariante en el Espacio)
RNN	Recurrent Neural Network (Red Neuronal Recurrente)
LSTM	Long-Short Term Memory
GRU	Gated Recurrent Unit
PI	Proporcional – Integrativo
RGA	Relative Gain Array (Matriz de Ganancias Relativas)

SIMBOLOGÍA

F_x	Fuerza de corte
F_y	Fuerza normal
F_z	Fuerza axial
v	Velocidad traslacional
ω	Velocidad rotacional
d	Profundidad
t	Instante de tiempo
x, X	Vector de estados
A	Matriz de relación de estados y sus variaciones en el tiempo
U	Vector de entradas transformado
x_0	Vector de estados inicial
p_1, p_2	Polos de la fuerza axial
τ_x	Constante de tiempo de la fuerza de corte
τ_y	Constante de tiempo de la fuerza normal
$\alpha_z, \beta_z, \gamma_z$	Exponentes asociados a la fuerza axial
$\alpha_x, \beta_x, \gamma_x$	Exponentes asociados a la fuerza de corte
$\alpha_y, \beta_y, \gamma_y$	Exponentes asociados a la fuerza normal
z	Cero del sistema
K_z	Ganancia estática de la fuerza axial
K_x	Ganancia estática de la fuerza de corte
K_y	Ganancia estática de la fuerza normal
Φ	Matriz fundamental del sistema
x^*	Núcleo de los estados
λ	Valores propios de la matriz A
kN	Kilo-newton
n	Número de módulos GRU
h_i	Vector de salida del módulo i
F_i	Vector de olvido del módulo i
σ	Función sigmoide
A_f, B_f, C_f, D_f	Matrices con pesos asociados a los vectores de olvido

\mathbf{h}_i	Vetor de activación del módulo i
ϕ	Función tangente hiperbólica
A_h, B_h, C_h, D_h	Matrices con pesos asociados a los vectores de activación
T	Tiempo límite de simulación
N	Número de iteraciones o epochs de entrenamiento
Y_{sp}	Set-point o referencia
μ	Media de la distribución de las perturbaciones
GHz	Giga-hertz
GB	Giga-byte
mm	Milímetro
s	Segundo
krev	Kilo-revolución
min	Minuto
p	Probabilidad de que una distribución Bernoulli sea 1
F_{max}	Magnitud de las perturbaciones en el primer escenario
D	Matriz de desacoplo
$C(s)$	Matriz diagonal con las funciones de transferencia del controlador PI
Y	Vector de salidas del proceso
k_{Px}, k_{Py}, k_{Pz}	Constantes proporcionales del controlador PI
k_{Ix}, k_{Iy}, k_{Iz}	Constantes integrativas del controlador PI
$G(s)$	Matriz de funciones de transferencia del proceso
$G^*(s)$	Matriz de funciones de transferencia desacopladas del proceso
$Q(s)$	Matriz con las ganancias estáticas de la diagonal de $G(s)$
J	Función objetivo por minimizar
e_{Fz}, e_{Fx}, e_{Fy}	Error de las variables de salida del proceso
$\Delta_d, \Delta_v, \Delta_\omega$	Variación de las variables de entrada
n_p	Horizonte de predicción
n_c	Horizonte de control
Net	Controlador inteligente representado como función
θ^*	Parámetros óptimos del controlador inteligente
θ	Parámetros entrenables del controlador inteligente

ÍNDICE DE FIGURAS

Figura 1.1 Esquema del proceso FSW	3
Figura 1.2 Defectos de soldadura típicos. a) Sin defecto. b) Falta de fusión. c) Porosidad. d) Cavidades [23].....	4
Figura 1.3 Esquema de una red neuronal prealimentada. Adaptado de [41]	6
Figura 1.4 Esquema general de una RNN [48]	8
Figura 1.5 Esquema de los módulos de una RNN original, una LSTM y una GRU [51] ..	9
Figura 2.1 Entradas y respuesta del proceso FSW en lazo abierto	13
Figura 2.2 Diagrama de densidad de soldaduras libres de defecto	14
Figura 2.3 Vista tridimensional de los datos en el espacio de fuerzas.....	15
Figura 2.4 Representación en diagrama de bloques de una GRU con arquitectura mínima	17
Figura 2.5 Diagrama del proceso en lazo cerrado empleando el sistema de control propuesto	19
Figura 2.6 Matriz de ganancias relativas y matriz de desacoplamiento	22
Figura 2.7 Función de transferencia desacoplada.	22
Figura 2.8 Modelo discretizado de la planta	23
Figura 2.9 Valores nominales de las entradas y salidas en el diseño del MPC	24
Figura 2.10 Modelo implementado en Python empleando tensores de Pytorch	25
Figura 2.11 Controlador inteligente implementado en Pytorch	26
Figura 2.12 Resumen de la red del controlador inteligente	27
Figura 3.1 Comparación de la respuesta de la fuerza de corte F_x entre diferentes sistemas de control al rechazar perturbaciones	29
Figura 3.2 Comparación de la respuesta de la fuerza normal F_y entre diferentes sistemas de control al rechazar perturbaciones	29
Figura 3.3 Comparación de la respuesta de la fuerza axial F_z entre diferentes sistemas de control al rechazar perturbaciones	30
Figura 3.4 Comparación de la profundidad d entre diferentes sistemas de control al rechazar perturbaciones	32
Figura 3.5 Comparación de la velocidad traslacional v entre diferentes sistemas de control al rechazar perturbaciones.....	32
Figura 3.6 Comparación de la velocidad rotacional ω entre diferentes sistemas de control	

al rechazar perturbaciones	33
Figura 3.7 Serie de tiempo del error de los tres controladores en el sistema perturbado	34
Figura 3.8 Serie de tiempo del error del MPC y del controlador inteligente en el sistema perturbado	34
Figura 3.9 Respuesta de la fuerza de corte F_x entre diferentes sistemas de control al rechazar perturbaciones aleatorias.....	36
Figura 3.10 Respuesta de la fuerza normal F_y entre diferentes sistemas de control al rechazar perturbaciones aleatorias.....	37
Figura 3.11 Respuesta de la fuerza axial F_z entre diferentes sistemas de control al rechazar perturbaciones aleatorias.....	37
Figura 3.12 Serie de tiempo del error del MPC y del controlador inteligente en el sistema perturbado aleatoriamente	38
Figura A.1 Esquema del modelo del proceso implementado en Simulink	
Figura A.2 Función de transferencia de la planta	
Figura D.1 Afinamiento del controlador PI de la fuerza axial en seguimiento de referencia	
Figura D.2 Afinamiento del controlador PI de la fuerza axial en rechazo de perturbaciones	
Figura D.3 Afinamiento del controlador PI de la fuerza normal en rechazo de perturbaciones	
Figura D.4 Esquema del controlador PI con desacoplo implementado en Simulink	
Figura D.5 Salidas del sistema en afinamiento del MPC para rechazar perturbaciones	
Figura D.6 Entradas del sistema en afinamiento del MPC para rechazar perturbaciones	
Figura D.7 Esquema del MPC implementado en Simulink	
Figura E.1 Entrenamiento del controlador inteligente	
Figura E.2 Evolución del error del controlador en el proceso de entrenamiento	
Figura E.3 Muestra aleatoria de pesos del controlador inteligente entrenado	

ÍNDICE DE TABLAS

Tabla 2.1 Error obtenido al final del entrenamiento del controlador inteligente empleando diferente número de módulos.....	¡Error! Marcador no definido.
Tabla 3.1 Error cuadrático medio empleando diferentes controladores y porcentaje de mejora del controlador inteligente	36
Tabla 3.2 Error cuadrático medio del sistema ante perturbaciones aleatorias y porcentaje de mejora del controlador inteligente.....	40
Tabla A.1 Constantes del proceso FSW	
Tabla C.1 Valores experimentales de los puntos utilizados para determinar la referencia	

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

Los sistemas de soldadura convencionales, como la soldadura por arco, tienen numerosas aplicaciones en la industria. Sin embargo, la fusión y posterior cristalización de los materiales de proceso (aporte y base) puede presentar diversas complicaciones para la obtención de la calidad deseada. Estas complicaciones pueden incrementar la porosidad del material o causar otros defectos microestructurales [1]. La soldadura por fricción con agitación –o FSW por sus siglas en inglés– es una técnica de soldadura en estado sólido. Esta técnica no requiere temperaturas muy elevadas para operación [2]. Sin embargo, la complejidad del proceso termo-mecánico dificulta la realización de un modelo analítico para este sistema. En este sentido, se han investigado diversos enfoques con modelos empíricos para la representación de sistemas de FSW [3], [4]. Un reto importante en este tipo de sistemas es la selección del conjunto de parámetros de soldadura capaces de obtener materiales libres de defectos y con buenas propiedades mecánicas. Los procedimientos generalmente se basan en prueba y error para encontrar el conjunto de parámetros adecuado [5].

1.2 Justificación del problema

Con el fin de reducir los defectos de las uniones generadas durante el proceso FSW, en contraste con otras técnicas de soldadura, la necesidad de un modelo que permita establecer una estrategia de control apropiada es fundamental para evaluar su correcto funcionamiento, respuesta dinámica y operación [3]. Un sistema de control inteligente puede aprender la dinámica del sistema FSW para así poder aplicar acciones correctivas y obtener la calidad deseada [4]. El modelo obtenido puede emplearse para otros sistemas de control tradicionales, e incluso puede utilizarse para la estimación de parámetros de operación en función de los requerimientos de calidad especificados, el material a soldar y las aplicaciones industriales específicas [6].

1.3 Objetivos

1.3.1 Objetivo General

Diseñar un sistema de control inteligente para un proceso de soldadura por fricción con agitación empleando redes neuronales.

1.3.2 Objetivos Específicos

1. Definir un modelo matemático del proceso de soldadura por fricción con agitación para las respectivas simulaciones.
2. Diseñar sistemas de control mediante métodos convencionales con la finalidad de establecer las comparaciones pertinentes.
3. Diseñar un sistema de control empleando redes neuronales y entrenarlo por el método supervisado empleando gradientes.
4. Realizar un análisis comparativo de los diferentes sistemas de control mediante simulaciones.

1.4 Marco teórico

1.4.1 Soldadura por fricción con agitación

El proceso de soldadura por fricción con agitación (FSW por sus siglas en inglés) es un método de soldadura en estado sólido, que ha demostrado efectividad en lograr juntas con alta resistencia para metales ligeros y otros metales de diferente naturaleza [7] [8] [9]. De hecho, FSW puede obtener propiedades mecánicas excepcionales, lo cual lo hace atractivo para varias aplicaciones industriales [10] [11].

El proceso FSW utiliza una herramienta rotacional, la cual es inmersa y se mueve a lo largo de la junta entre las dos piezas a soldar, esta herramienta no se consume en el proceso. La rotación de la herramienta dentro del material genera calor, lo cual suaviza el material metálico, permitiendo que la herramienta mezcle las piezas para así crear la junta. La Figura 1.1 muestra un esquema del proceso FSW.

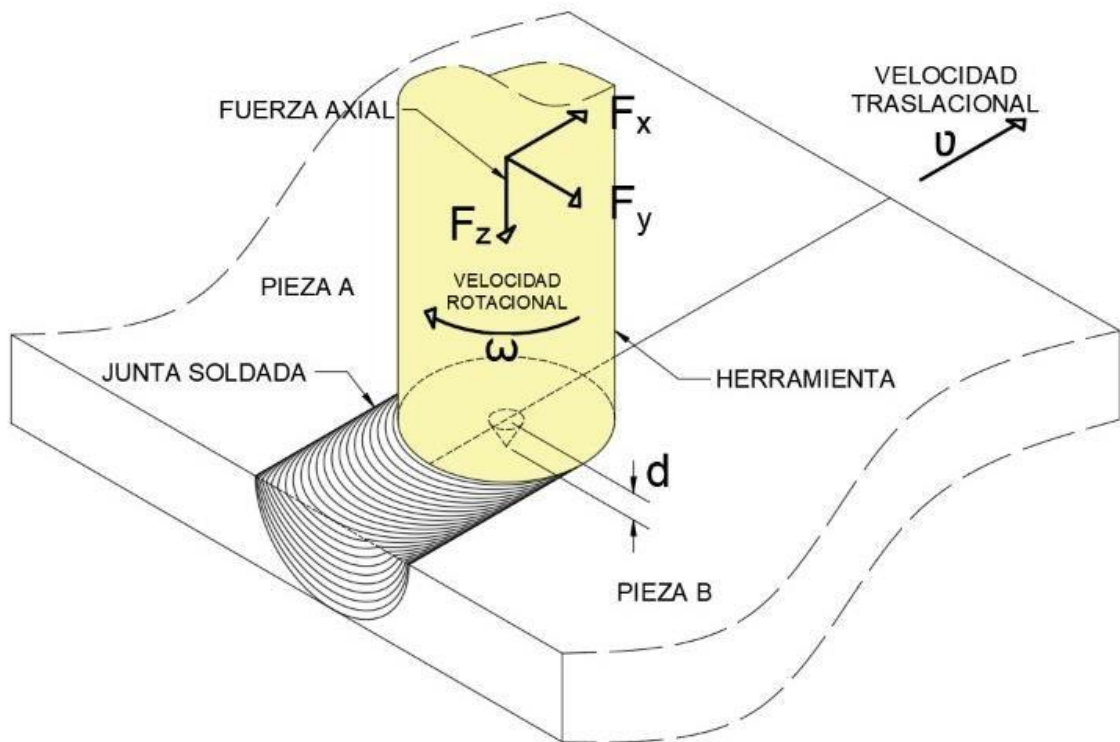


Figura 1.1 Esquema del proceso FSW

El proceso FSW involucra fenómenos físicos complejos que incluyen cambios metalúrgicos, flujo de material, transferencia de calor y acoplamiento termomecánico. Se han realizado estudios sobre las principales dinámicas analizadas en este proceso que incluyen el flujo de material, la distribución de temperatura y las fuerzas involucradas en el mecanismo [12] [13] [14] [15] [16]. Otras contribuciones se han enfocado en predecir la calidad de las piezas resultantes para la posterior toma de decisiones y para el monitoreo del sistema [17] [18].

Para obtener modelos simplificados de FSW, otros trabajos han realizado la identificación del sistema para construir representaciones empíricas en el dominio de Laplace en base a observaciones experimentales [19]. Estos modelos tienen la capacidad de reproducir el sistema sin requerir conocimiento del fenómeno físico que toma lugar. Otro método basado en datos para modelar sistemas FSW es el uso de redes neuronales artificiales (ANN por sus siglas en inglés) para reproducir la dinámica y monitorear el proceso [20] [21]. Sin embargo, estos modelos

empíricos son limitados porque son válidos solamente para el rango de datos empleados y para un material en específico.

1.4.2 Defectos de soldadura

Un defecto de soldadura es una falla que compromete la utilidad de una pieza soldada. Hay una gran variedad de defectos de soldadura y estas imperfecciones son clasificadas de acuerdo con la norma ISO 6520 [22]. Estas imperfecciones se deben en gran parte a un mal proceso de soldadura, influenciado por los parámetros utilizados, y todas ellas se traducen en malas propiedades mecánicas del material final. La Figura 1.2 muestra ejemplos de defectos de soldadura y de una soldadura sin defecto para comparación. La Figura 1.2.b muestra una falta de fusión en el centro, lo cual suele ser causado si se presenta óxido de aluminio al momento de soldar. Las Figuras 1.2.c y 1.2.d ilustran presencia de porosidades y cavidades en la junta respectivamente, las cuales pueden ser causadas por impurezas de materiales extraños presentes en la junta [22].

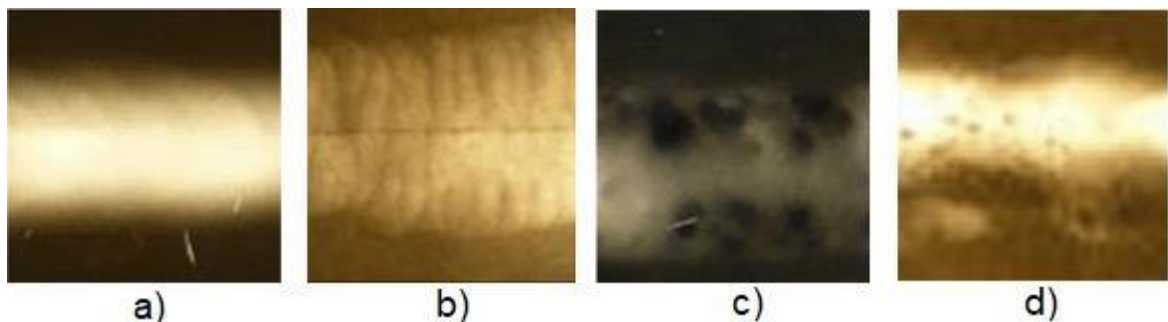


Figura 1.2 Defectos de soldadura típicos. a) Sin defecto. b) Falta de fusión. c) Porosidad. d) Cavidades [23]

Existen varias contribuciones que se enfocan a determinar los parámetros adecuados del proceso FSW a fin de obtener soldaduras sin defectos. Entre estos cabe destacar los trabajos de Dewan et al. [17] y Huggett et al. [18], quienes introdujeron una representación visual de regiones de soldadura. Esta representación es construida con datos recolectados de procesos y define zonas de soldadura caliente, soldadura fría y soldadura ideal. Una soldadura caliente se da cuando se introduce calor en exceso debido a una alta velocidad rotacional de la herramienta, baja velocidad traslacional y una alta fuerza axial. Una soldadura fría está asociada a una baja tasa de calor debido a una baja velocidad rotacional,

alta velocidad traslacional y baja fuerza axial. Una soldadura ideal es aquella que se logra con la tasa de calor necesaria para no presentar defectos en el material.

1.4.3 Control de fuerzas en FSW

Varias alternativas y arquitecturas de control han explorado el paradigma de controlar el proceso de FSW mediante las fuerzas que intervienen en el sistema [24] [25]. Entre estas, Zhao et al. [26] diseñaron e implementaron controladores no lineales para las fuerzas generadas en FSW, manteniendo las diferentes fuerzas constantes para lograr sus objetivos de control. Davis et al. [27] implementaron un controlador adaptativo empleando observadores. En este estudio se encontró una correlación entre la potencia del eje y la fuerza axial, permitiendo el uso de mediciones energéticas como retroalimentación. Davis et al. [28] diseñaron un controlador difuso adaptativo para mantener una potencia constante en el proceso FSW. Zhao et al. [29] propusieron un controlador sobre la fuerza axial, incluyendo una compensación desfasada para favorecer a la estabilidad y al desempeño del sistema. El uso de ANNs u otras estrategias de control equivalentes aplicado a sistemas FSW es limitado [30]. Adicionalmente, contribuciones previas no consideran la obtención de piezas libres de defecto y con buenas propiedades mecánicas para el diseño de los controladores.

1.4.4 Control de otras variables en FSW

Existen varias contribuciones que han evaluado principalmente los efectos de la temperatura como una medida indirecta para el control de calidad de la soldadura. En este contexto, Fehrenbacher et al. [31] [32] implementaron un sistema de control, con medición de la temperatura interfacial entre la pieza y la herramienta de trabajo, para ser utilizada como señal de retroalimentación; esta medición se realizó en tiempo real empleando sensores sin contacto. Uno de los hallazgos de este trabajo fue que un insuficiente contacto entre la herramienta y el material puede producir soldaduras de baja calidad. Adicionalmente, la calidad de la soldadura se puede mantener, a pesar de las perturbaciones externas en el sistema, controlando la temperatura y otros parámetros del proceso [33].

Otras alternativas, como controladores proporcional-integrador-derivativo (PID) y controladores predictivos basados en modelo (MPC por sus siglas en inglés) han demostrado ser arquitecturas de control adecuadas para los sistemas de FSW. Taysom et al. [34] encontraron que el desempeño de cada controlador (PID o MPC) varía con las características del problema en el que se los emplea. Debido a las capacidades particulares de los controladores, estos cumplen con determinados objetivos de control dependiendo del caso. Finalmente, Bachman et al. [35] propusieron un MPC adaptativo para control de temperatura en un sistema FSW, obteniendo una reducción en el tiempo de asentamiento en lazo cerrado, evitando problemas relacionados a inestabilidad.

1.4.5 Redes neuronales artificiales

Una red neuronal artificial (ANN por sus siglas en inglés) es un conjunto de nodos inter-conectados con cierta estructura, la cual es dependiente de la arquitectura de la red. Cada nodo y conexión tienen pesos asociados, los cuales son ajustados durante el proceso de aprendizaje de la red [36]. Las redes neuronales son ampliamente utilizadas en los campos de visión por computadora [37] [38], sistemas de recomendación [39] y recientemente en sistemas de control [40]. La Figura 1.3 muestra una representación gráfica de una red neuronal artificial.

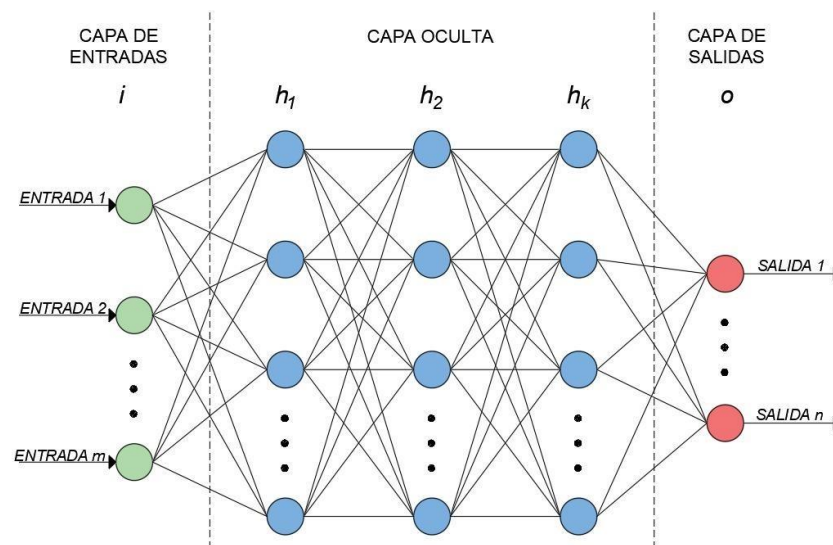


Figura 1.3 Esquema de una red neuronal prealimentada. Adaptado de [41]

En el caso de aprendizaje supervisado, el proceso de aprendizaje toma lugar a través de diferenciación automática o “backpropagation” [42]. Esto consiste en realizar la inferencia normal, luego calcular el valor de la función de pérdida seleccionada y regresar los gradientes a través de la red para luego actualizar sus parámetros, utilizando el optimizador escogido.

En el caso de aprendizaje por refuerzo [43], el proceso de aprendizaje se da mediante un sistema de políticas y recompensas. La red, en este caso también llamada agente, infiere una respuesta dada una entrada, y según la calidad de la respuesta y de acuerdo con la política empleada, se da una recompensa correspondiente. Posteriormente, se ajustan los parámetros de la red.

Existen muchas arquitecturas de ANNs y varias jerarquías de clasificación, algunos tipos de arquitecturas se desempeñan mejor en ciertas tareas. Las redes neuronales prealimentadas tienen conexiones entre todos los nodos de capas consecutivas y estas conexiones no forman ciclos [44], esta es la red neuronal más simple. Las redes neuronales convolucionales (CNN por sus siglas en inglés), también conocidas como redes neuronales artificiales invariantes en el espacio (SIANN por sus siglas en inglés), son redes que emplean kernels (matrices convolucionales). Estas matrices se mueven a través del espacio de la entrada en forma de una ventana móvil, para así encontrar los parámetros que reflejen características de simetría traslacional [45]. Las redes CNN son ampliamente utilizadas en el campo de visión por computadora. Finalmente, se tiene a las redes neuronales recurrentes, las cuales serán explicadas en la siguiente sección.

1.4.6 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN por sus siglas en inglés), son ANNs donde las conexiones entre nodos forman un grafo dirigido a través de una secuencia temporal. Esto les permite exhibir un comportamiento temporal dinámico [46]. Las RNNs pueden utilizar sus estados internos para procesar secuencias de entradas con tamaño variable. Esto las hace aplicable a tareas como reconocimiento de escritura y de habla [47].

En las RNNs, las salidas de todas las neuronas normalmente se conectan a las entradas de todas las neuronas de la siguiente capa. Esta es la topología más general. Otras topologías pueden representarse asignándole cero al peso de algunas conexiones para simular la falta de conexiones entre dichas neuronas. La Figura 1.4 muestra una representación de una RNN, donde en cada instante t , ingresa el estado z_t y se entrega la salida x_t .

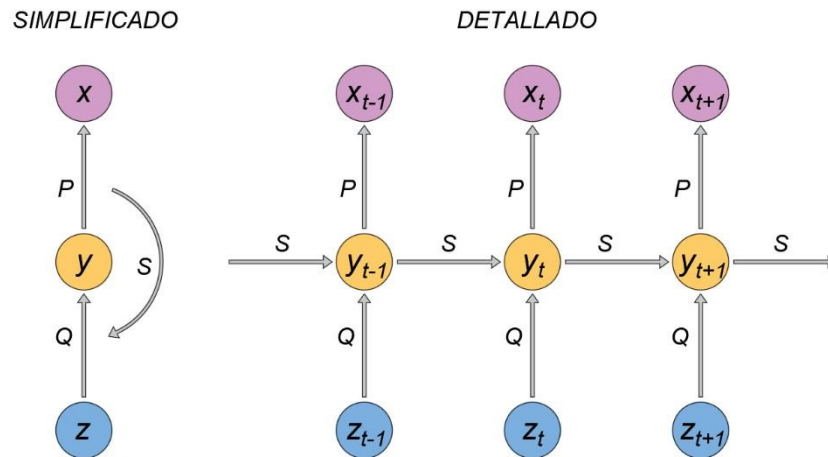


Figura 1.4 Esquema general de una RNN [48]

La Figura 1.4 muestra solamente la representación global de una RNN en su forma más general. Sin embargo, existen algunas variaciones en las conexiones entre los diferentes módulos de la red. La variación de RNN más notable es la “*Long shortterm memory*” (LSTM) [49], la cual fue diseñada para evitar el problema del desvanecimiento del gradiente al entrenar RNNs. Las LSTM son mejoradas con puertas para prevenir errores generados en el proceso de entrenamiento cuando se tiene desvanecimiento o explosión de gradientes [50].

1.4.7 Gated recurrent unit

Gated recurrent units (GRU) son mecanismos en RNNs introducidos por Bengio et al. [51]. La GRU es similar a la LSTM en que ambas consisten en olvidar paulatinamente el gradiente mediante puertas. Sin embargo, la GRU tiene menos parámetros que la LSTM ya que no contiene una puerta a la salida de la unidad. La Figura 1.5 muestra un esquema de las diferencias entre los distintos módulos de RNNs. Existen algunas variaciones en la arquitectura normal de una GRU, una de

ellas es una forma simplificada denominada “*minimal gated unit*” [52], esta combina la puerta para olvidar el gradiente y la puerta para actualización del módulo.

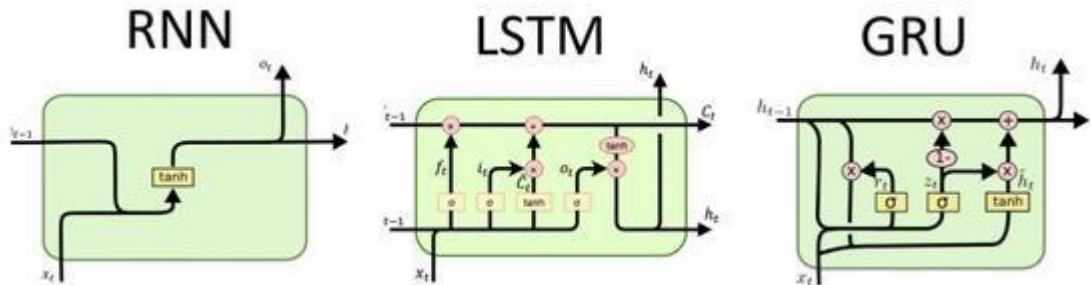


Figura 1.5 Esquema de los módulos de una RNN original, una LSTM y una GRU [51]

El desempeño de las redes GRU en ciertas tareas de modelamiento de señales, como de habla y música, y procesamiento de lenguaje natural, ha sido similar al de las LSTM, a pesar de que la complejidad de las LSTM es mayor que la de las GRU. Sin embargo, las redes GRU han mostrado un mejor desempeño en datasets más pequeños [52], debido a que su menor número de conexiones se ajusta mejor a datasets con limitada cantidad de datos o dimensionalidad, en comparación a arquitecturas con mayor capacidad o complejidad.

CAPÍTULO 2

2. METODOLOGÍA

En el presente trabajo se estudia la respuesta dinámica de un sistema de control inteligente como alternativa para el proceso FSW. El material y sistema bajo estudio corresponde a una aleación de aluminio. Los datos experimentales utilizados corresponden a la aleación Al 6061-T651 [53], mientras que el modelo empírico fue desarrollado considerando la aleación Al 6061-T6 [19]. La consistencia del diseño experimental es garantizada porque ambos materiales son esencialmente iguales. A continuación, el modelo empírico validado fue inicialmente convertido de funciones de transferencia a un conjunto de ecuaciones diferenciales ordinarias en el dominio del tiempo. El modelo permite visualizar y estudiar la dinámica de las variables de interés, las cuales corresponden a las fuerzas involucradas en el proceso, en función de las variables que se pueden manipular.

Para que el proceso produzca soldaduras libres de defectos, los datos experimentales etiquetados permitieron construir las fronteras operacionales que definen una región con alta probabilidad de obtener condiciones para que la suelta no presente defectos. Por consiguiente, la referencia fue definida en un diagrama de fuerzas tridimensional para finalmente implementar el sistema de control inteligente. Se realizaron experimentos computacionales para comprobar el desempeño del controlador donde se priorizó la evaluación y capacidad en rechazar perturbaciones. Además, se visualizó su capacidad para alcanzar el punto de referencia y consecuentemente un producto sin defectos. La respuesta del controlador inteligente es comparada con un conjunto de controladores PI desacoplados y una estrategia MPC lineal para verificar su desempeño, competitividad y aplicabilidad al problema planteado.

2.1 Modelamiento del proceso y selección de punto de referencia

Para diseñar un sistema de control de fuerzas adecuado, el modelamiento de las fuerzas involucradas y sus relaciones con las entradas relevantes es una tarea crítica. De esta manera, un modelo relacionando las variables de entrada

(profundidad, velocidad traslacional y velocidad rotacional) y las variables de salida (fuerza axial, fuerza de corte y fuerza normal) es definido. Adicionalmente, se define la referencia del sistema de control en términos del espacio de fuerzas. Su definición es basada en obtener características de soldadura sin defectos.

2.1.1 Modelo dinámico

El modelo utilizado en este trabajo se basa en el modelo empírico propuesto por Zhao et al. [19]. En términos generales, el modelo dinámico es un sistema lineal no autónomo y no homogéneo de ecuaciones diferenciales con condiciones iniciales, que representa el comportamiento dinámico de las variables mecánicas de interés y las variables manipuladas. En este contexto, el modelo consiste en encontrar el vector de variables de estado $\mathbf{x}(t) = (F_z(t), \dot{F}_z(t), F_x(t), F_y(t))$, de tal manera que:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{U}(t); \forall t \in (0, +\infty); \\ \mathbf{x}(0) &= \mathbf{x}_0, \end{aligned} \quad (2.1)$$

donde:

$$\mathbf{A} = \begin{bmatrix} -p_0 & 1 & 0 & 0 \\ p_1 & -p_2 & 0 & 0 \\ 0 & 0 & -1/\tau_x & 0 \\ 0 & 0 & 0 & -1/\tau_y \end{bmatrix},$$

$$\mathbf{U}(t) = [u_1, u_2, u_3, u_4]^T, u_4 = 0,$$

$$u_2 = \left(\frac{d}{dt} + \alpha \right) (\beta \dot{v}(t) + v(t)) (\gamma \dot{\omega}(t) + \omega(t)) K_z d^{\alpha z - 1} v^{\beta z - 1} \omega^{\gamma z - 1}(t),$$

$$u_3 = K_x d^{\alpha x} v^{\beta x} \omega^{\gamma x}(t), \tau_x$$

$$u_4 = K_y d^{\alpha y} v^{\beta y} \omega^{\gamma y}(t).$$

τ_y

Donde, $F_z(t)$, $F_x(t)$ y $F_y(t)$ representan la fuerza axial, de corte y normal respectivamente, mientras que $d(t)$ es la profundidad de inmersión de la herramienta en la pieza, $v(t)$ es la velocidad traslacional de la pieza y $\omega(t)$ es la velocidad rotacional de la misma. Una descripción gráfica de estas variables se puede apreciar en la Figura 1.1. Los parámetros son: p_1, p_2, τ_x, τ_y y las constantes dadas son: $K_x, K_y, K_z, z, \alpha_x, \alpha_y, \alpha_z, \beta_x, \beta_y, \beta_z, \gamma_x, \gamma_y, \gamma_z$. Los valores de los parámetros mencionados y la matriz de transferencia del proceso se muestran en el Apéndice A.

2.1.2 Análisis de estabilidad

Un factor clave que garantiza el comportamiento de un sistema dinámico a largo plazo es la estabilidad de sus soluciones. En nuestro caso, la existencia de soluciones del modelo dinámico propuesto en la ecuación 2.1 es garantizada debido a que A es una matriz con coeficientes constantes. Además, las funciones $d(t)$, $v(t)$ y $\omega(t)$, y sus primeras derivadas, son continuas y acotadas de acuerdo con el comportamiento reportado por Zhao et al. [19]. De hecho, de acuerdo con [54], existe una matriz fundamental $\Phi(t)$ asociada con el sistema homogéneo $\dot{x}(t) = Ax(t)$ tal que una matriz exponencial para el sistema no homogéneo es $e^{At} = \Phi(t)\Phi^{-1}(0)$ y su solución está dada por:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\xi)}U(\xi)d\xi. \quad (2.2)$$

En base a esta solución, se obtiene el siguiente criterio de estabilidad, en el marco de referencia de Lyapunov [55]:

- Para el sistema lineal regular $\dot{x}(t) = Ax(t)$, el núcleo $x^*(t) = x(t); Ax(t) = 0$ es Lyapunov estable en $t \geq 0$ si y solo si la solución es acotada en $t \rightarrow \infty$.

- Todas las soluciones del modelo dinámico en la ecuación 2.1 dadas por la ecuación 2.2, tienen la misma propiedad de estabilidad Lyapunov del núcleo del sistema del sistema lineal $\dot{x}(t) = Ax(t)$.

La prueba de los dos axiomas del criterio de estabilidad de Lyapunov se detalla en el Apéndice B. Por otro lado, analizando la estabilidad del sistema mediante inspección de la matriz A, se tiene que los valores propios de esta matriz son:

$$\begin{matrix} p_1 & p_2 \\ \lambda = -1/\tau_x & \\ & [-1/\tau_y] \end{matrix} \quad (2.3)$$

Los valores propios mostrados en la ecuación 2.3 corresponden a los polos del sistema en lazo abierto y debido a que todos ellos son negativos (referirse al Apéndice A), se concluye que el sistema es estable en lazo abierto. La Figura 2.1 muestra la respuesta del sistema en lazo abierto.

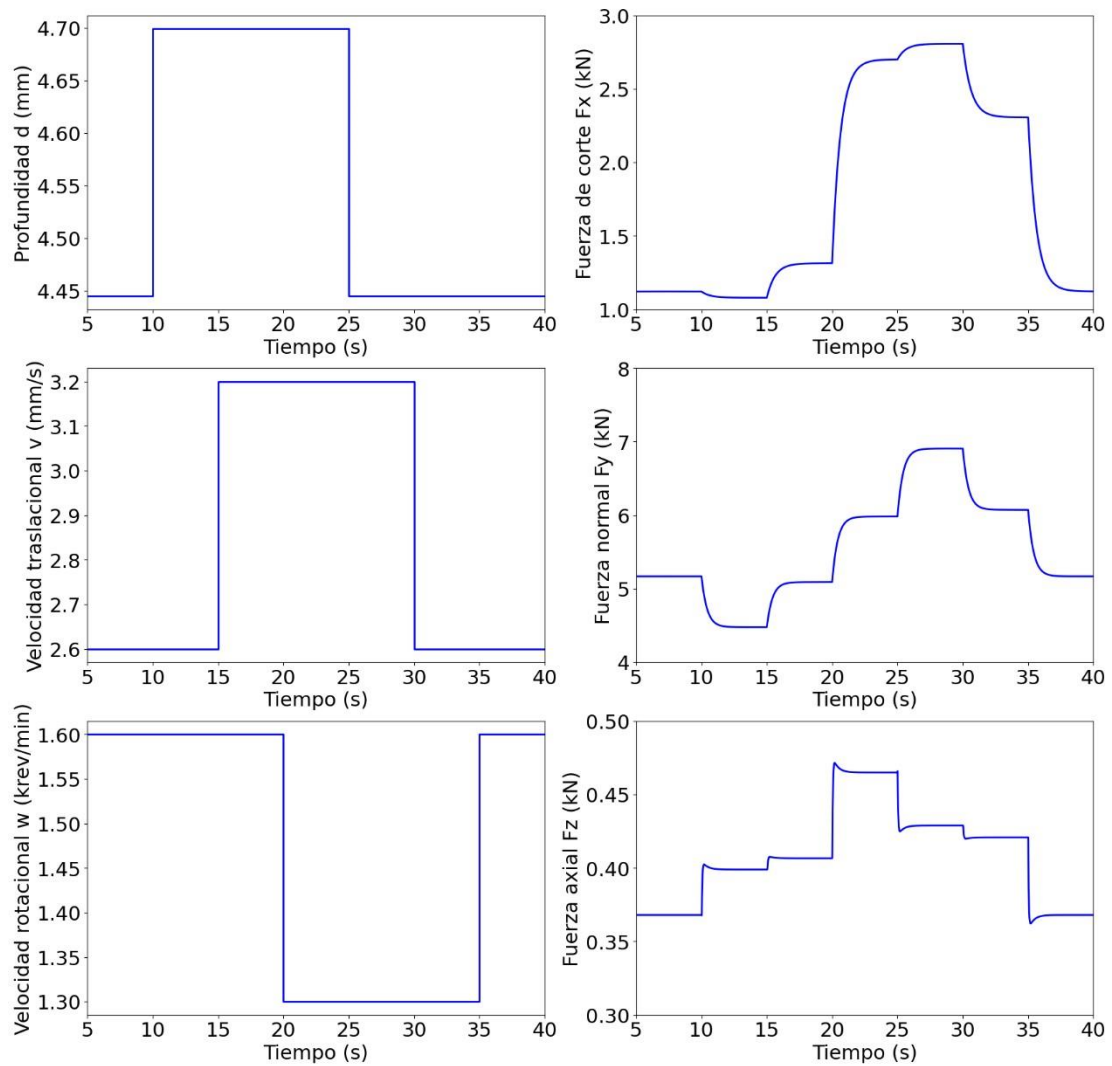


Figura 2.1 Entradas y respuesta del proceso FSW en lazo abierto

Se puede apreciar en la Figura 2.1 que un incremento en la profundidad d causa una ligera disminución en la fuerza de corte F_x , una disminución moderada en la fuerza normal F_y y un incremento en la fuerza axial F_z . Un aumento en la velocidad traslacional v provoca un aumento en F_x y en F_y , mientras que en F_z solo causa un aumento leve. Finalmente, una disminución en la velocidad rotacional ω causa un gran aumento en F_x , mientras que en F_z y en F_y el aumento es moderado. Lo contrario se observa si se revierten los valores de las entradas.

2.1.3 Selección de referencia o set-point

Para determinar el set-point óptimo a seguir por el controlador en un dominio de fuerzas dado (F_x, F_y, F_z) , se hizo uso de los datos experimentales de [53]. Estos

datos fueron procesados para definir regiones que representan una alta probabilidad de obtener soldaduras libres de defecto y mejores propiedades mecánicas en el material final. La Figura 2.2 muestra la densidad de las soldaduras libres de defecto, en la cual las zonas internas corresponden a regiones con mayor probabilidad de obtención de estos resultados, en las zonas externas las propiedades de soldadura se deterioran y pueden aparecer ciertos defectos. El número de regiones es un parámetro completamente visual, por lo cual se escoge arbitrariamente, lo más importante es el centro de estas curvas de nivel.

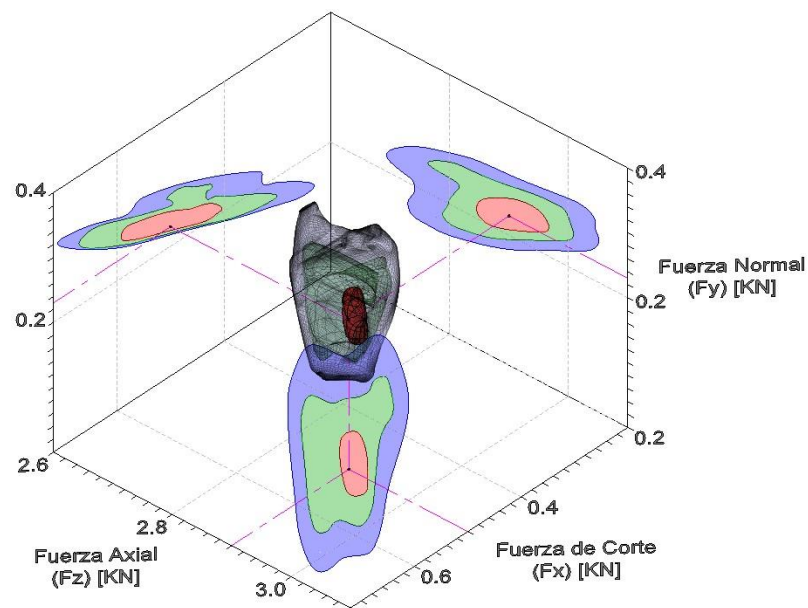


Figura 2.2 Diagrama de densidad de soldaduras libres de defecto

La Figura 2.3 muestra una vista tridimensional de los datos experimentales en el espacio de fuerzas. Un buen producto es aquel que se encuentra libre de defectos y tiene buenas propiedades mecánicas; un producto mediano es aquel que no tiene defectos, pero sus propiedades mecánicas se encuentran debajo del umbral deseado; mientras que un mal producto es aquel que tiene defectos en sus granos o microestructura. La cruz representa el punto de mayor magnitud en el espacio de fuerzas, siguiendo la estimación por kernels de densidad en los datos experimentales etiquetados. Este punto se localiza en la región con mayor probabilidad de lograr las características deseadas en el proceso de soldadura.

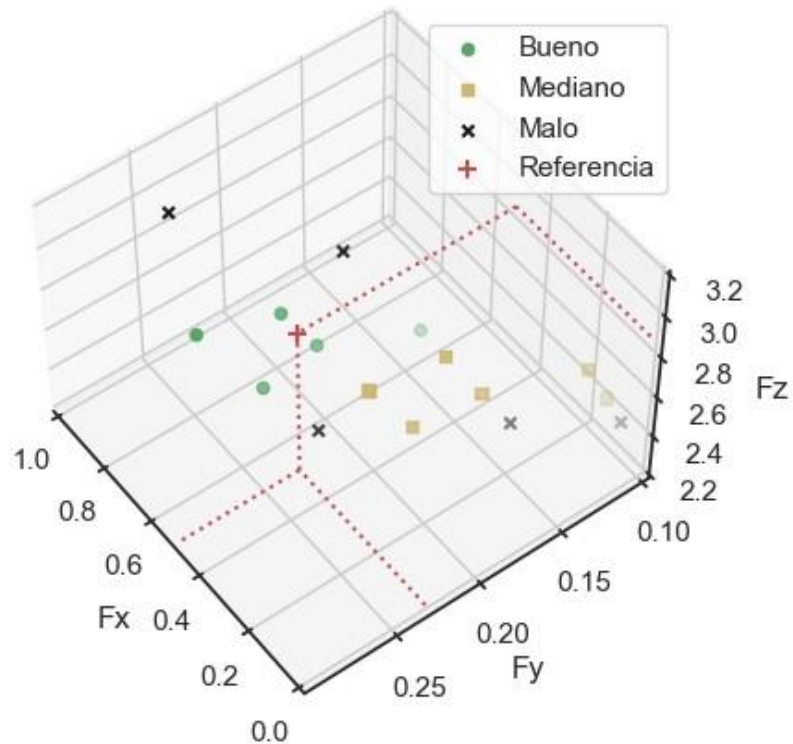


Figura 2.3 Vista tridimensional de los datos en el espacio de fuerzas

En este trabajo, la cruz fue utilizada como una referencia tridimensional localizada en el dominio de fuerzas para comparar la capacidad de alcanzar esta condición por las diferentes arquitecturas de control. Esta referencia se localiza en el punto $F_x = 0.5 \text{ kN}$, $F_y = 0.23 \text{ kN}$, $F_z = 2.9 \text{ kN}$. La señal de error es calculada como la distancia euclidiana entre la posición actual y la referencia deseada. Más detalles y el procesamiento de los datos experimentales de [53] se encuentran en el Apéndice C.

2.2 Arquitectura de control inteligente

En este trabajo se propone un sistema de control inteligente basado en redes neuronales. La selección de la arquitectura de la red generalmente se hace de acuerdo con el problema particular. Para utilizar apropiadamente la información disponible del sistema dinámico bajo estudio, la arquitectura del controlador escogida fue la de una RNN. El controlador RNN tiene n módulos temporales, los cuales corresponden a las capas de la red y sus conexiones con las entradas del sistema, con los módulos anteriores y con los siguientes módulos son determinadas dependiendo del tipo de RNN a emplear. El controlador RNN, para $1 \leq i \leq n$, tiene la siguiente formulación:

$$\begin{aligned}
\mathbf{h}_0 &= \mathbf{0}, \\
\mathbf{h}_i &= f_i(\mathbf{X}_{t-n+i-1}, \mathbf{U}_{t-n+i-1}, \mathbf{h}_{i-1}), \\
\mathbf{U}_t &= \mathbf{h}_n.
\end{aligned} \tag{2.4}$$

Para que la ecuación 2.4 sea válida, las dimensiones de la salida de la última capa deben ser iguales a las dimensiones del vector de entradas del proceso (3 en el proceso bajo estudio). Las dimensiones de los demás módulos son arbitrarias y pueden variar entre sí, lo único que se debe considerar es que la entrada a la capa inicial sea un vector de ceros.

2.2.1 Módulos GRU

La arquitectura de cada módulo temporal del controlador inteligente corresponde al de una GRU [51]. Se seleccionó GRU como la arquitectura de cada módulo debido a que su mecanismo requiere menor cantidad de parámetros si se la compara con una LSTM [49]. En este caso, el módulo GRU es más adecuado para el problema bajo estudio ya que puede llegar a desarrollar sus funciones requeridas con una menor cantidad de datos y dimensionalidad.

Se escogió la clase de GRU que contiene la mínima cantidad de puertas, la cual combina la puerta de actualización con la puerta de olvido. Cada módulo tiene la siguiente formulación:

$$\begin{aligned}
\mathbf{F}_i &= \sigma(A_{f,i}\mathbf{X}_{t-n+i-1} + B_{f,i}\mathbf{U}_{t-n+i-1} + C_{f,i}\mathbf{h}_{i-1} + D_{f,i}), \\
\hat{\mathbf{h}}_i &= \phi(A_{h,i}\mathbf{X}_{t-n+i-1} + B_{h,i}\mathbf{U}_{t-n+i-1} + C_{h,i}(\mathbf{F}_i \odot \mathbf{h}_{i-1}) + D_{h,i}), \\
\mathbf{h}_i &= (1 - \mathbf{F}_i) \odot \mathbf{h}_{i-1} + \mathbf{F}_i \odot \hat{\mathbf{h}}_i.
\end{aligned} \tag{2.5}$$

En la ecuación 2.5, \mathbf{F}_i es el vector de olvido, $\hat{\mathbf{h}}_i$ es el vector de activación, \mathbf{h}_i es el vector de salida del módulo, σ y ϕ denotan a las funciones de activación, usualmente corresponden a las funciones sigmoide y tangente hiperbólica respectivamente. $A_{f,i}$, $B_{f,i}$, $C_{f,i}$, $D_{f,i}$, $A_{h,i}$, $B_{h,i}$, $C_{h,i}$, $D_{h,i}$ son matrices y vectores con valores que se ajustan en el paso de optimización del proceso de aprendizaje o

entrenamiento. i denota el número del módulo actual ($1 \leq i \leq n$) y t indica el tiempo en dominio discreto. Una representación visual del esquema de un módulo GRU se observa en Figura 2.4.

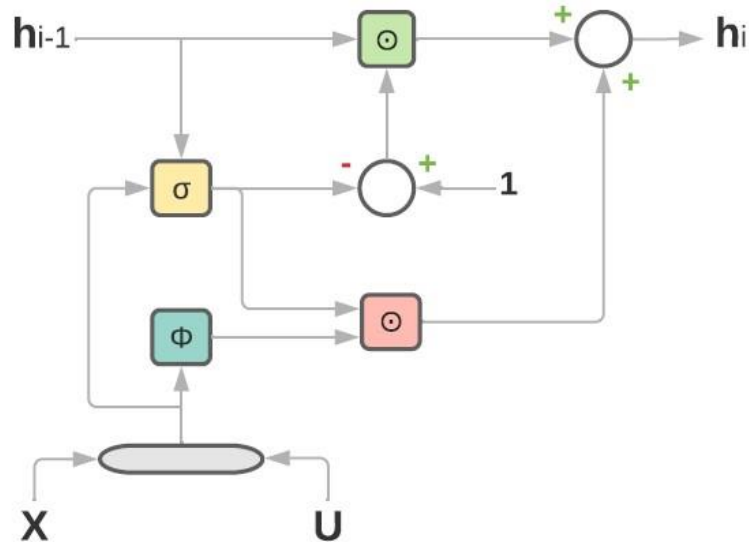


Figura 2.4 Representación en diagrama de bloques de una GRU con arquitectura mínima

2.2.2 Sistema en lazo cerrado y entrenamiento

El controlador propuesto, previo a ser comparado con otras alternativas, debe entrenarse para la tarea a desempeñar. Dado un conjunto de estados X y entradas del proceso anteriores U , la nueva entrada del proceso U_t es determinada por el controlador y alimentada al modelo diferenciable definido en la sección 2.1.1. Se calcula el valor de la función de pérdida, en este caso corresponde al error cuadrático medio, y se regresan los gradientes a través de la red para actualizar todos los pesos variables de la red en el proceso de optimización.

En ambas tareas, entrenamiento e inferencia, los estados del proceso y las entradas previas son alimentadas al controlador en una ventana móvil de tamaño fijo n . Aunque el proceso de entrenamiento no se realiza en línea, la forma de generación y alimentación de datos al modelo simula un entrenamiento en línea.

Se obtiene un sistema lazo cerrado con los pesos del controlador definidos aleatoriamente, se alimenta al controlador los estados iniciales y el controlador genera una entrada al proceso y este se alimenta al proceso, este paso se repite hasta llegar a un tiempo límite T . Al final se calcula la función de pérdida y se realiza el paso de optimización y se actualizan los pesos, todo esto corresponde a un “*epoch*” (paso de entrenamiento). En cada epoch el ruido alimentado al proceso es distinto, ya que sigue una distribución normal con media cero. Se sigue entrenando al controlador hasta llegar al número de epochs definido.

La Figura 2.5 ilustra el flujo de las señales del sistema en lazo cerrado. La figura también muestra el flujo del entrenamiento y la inferencia. El entrenamiento corresponde a la inferencia más el cómputo de la función de pérdida y el paso del optimizador para la posterior actualización de los pesos del controlador.

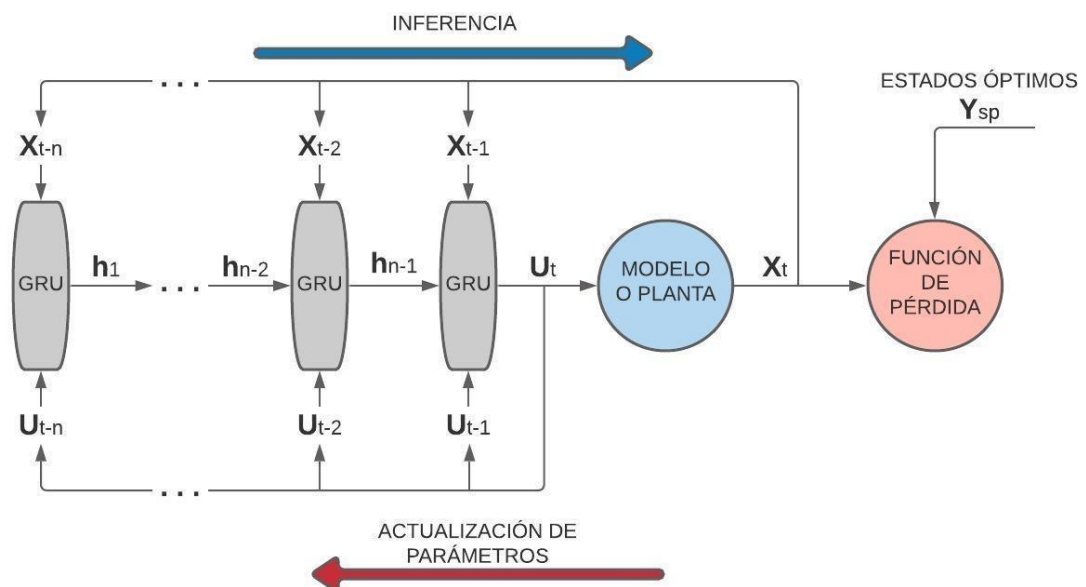


Figura 2.5 Diagrama del proceso en lazo cerrado empleando el sistema de control propuesto

En el proceso de inferencia, los últimos pasos son saltados, lo cual hace que el método sea significativamente más rápido que otros métodos basados en optimización, debido a que las operaciones necesarias son transformaciones

lineales con simples funciones de activación no lineales y el paso de optimización, computacionalmente costoso, ya no es necesario.

2.3 Experimentos y comparaciones

El sistema de control inteligente propuesto fue comparado con otras dos estrategias de control tradicionales, un conjunto de controladores PI desacoplados y un MPC lineal [56]. Las respuestas fueron simuladas y evaluadas aplicando las mismas perturbaciones en los tres diferentes sistemas de lazo cerrado para observar el desempeño de los controladores y sus capacidades para rechazar perturbaciones.

Se escogieron perturbaciones paso de subida y bajada para analizar la respuesta del sistema en ambas situaciones. Las perturbaciones suceden en las salidas para analizar el desempeño del controlador ante influencias externas. Se planteó dos escenarios, en el primero se definió los tiempos de inicio y final de estas perturbaciones y la magnitud de estas. Entre $t = 20s$ y $t = 50s$ se perturbó a la fuerza de corte F_x en $0.05kN$, entre $t = 30s$ y $t = 60s$ se perturbó a la fuerza normal F_y en $0.05kN$, finalmente, entre $t = 40s$ y $t = 70s$ se perturbó a la fuerza axial F_z en $0.5kN$.

En el segundo escenario, también se realizaron perturbaciones paso, pero los tiempos de inicio y final de estas perturbaciones, así como las magnitudes de las mismas son generadas aleatoriamente. Los tiempos de inicio y final de las perturbaciones de cada una de las fuerzas siguen una distribución Bernoulli con $p = 0.3$, mientras que la magnitud es definida por una distribución uniforme en $[0, F_{max}]$ donde F_{max} corresponde a los valores de las perturbaciones en el primer escenario.

El esquema de evaluación de respuesta se plantea en condiciones justas para evitar sesgos. El modelo dinámico fue implementado en Python y en un ambiente de Matlab-Simulink, el esquema del modelo se encuentra en el Apéndice A.

2.3.1 Controlador PI

El controlador PI desacoplado fue diseñado con un desacoplamiento estático, y las variables fueron emparejadas siguiendo las siguientes combinaciones: profundidad (d) y fuerza axial (F_z), velocidad traslacional (v) y fuerza normal (F_y), y velocidad rotacional (ω) y fuerza de corte (F_x). El emparejamiento se realizó empleando la matriz de ganancias relativas estáticas (RGA por sus siglas en inglés). Esta arquitectura de control fue afinada para minimizar el error transitorio de la respuesta del sistema al inducir perturbaciones.

La ley de control en el caso del control PI desacoplado es de la siguiente forma:

$$\mathbf{U}(s) = DC(s)[\mathbf{Y}_{sp}(s) - \mathbf{Y}(s)] \quad (2.6)$$

Donde $\mathbf{U}(s)$ es el vector de entradas, D es la matriz cuadrada de desacople, $C(s)$ es una matriz diagonal con los parámetros del controlador, $\mathbf{Y}_{sp}(s)$ es el vector de set-points y $\mathbf{Y}(s)$ es el vector de salidas del sistema. La matriz $C(s)$ tiene la siguiente forma:

$$C(s) = \begin{bmatrix} k & & & \\ k_{pz} + I_z/s & & & \\ & k_{px} + k_{Ix}/s & & \\ & & k_{Iy}/s & \\ & & & k_{py} + \end{bmatrix} \quad (2.7)$$

La matriz de desacoplo estático D se la obtiene empleando la ecuación:

$$D = G^{-1}(0)Q(0) \quad (2.8)$$

Donde G es la matriz de funciones de transferencia del proceso y $Q(0)$ corresponde a la matriz diagonal de las ganancias estáticas de la matriz G :

$$Q(0) = \begin{bmatrix} G(0)_{11} & 0 & 0 \\ 0 & G(0)_{22} & 0 \\ & & & \end{bmatrix} \quad (2.9)$$

$$0 \quad 0 \quad G(0)_{33}$$

El objetivo de la matriz de desacoplo es hacer 0 las ganancias estáticas cruzadas, manteniendo la misma ganancia en los componentes diagonales de la función de transferencia del proceso. Para establecer los pares de entradas y salidas, se obtiene la matriz de ganancias relativas empleando la siguiente ecuación:

$$RGA = G(0) \cdot G(0)^{-1} \quad (2.10)$$

Donde la operación \cdot representa el producto Hadamard o elemento a elemento. Figura 2.6 muestra la matriz de ganancias relativas y la matriz de desacoplamiento, en dichas matrices las filas corresponden en orden a la fuerza axial (F_z), fuerza de corte (F_x) y fuerza normal (F_y); mientras que las columnas corresponden en orden a la profundidad (d), velocidad rotacional (ω) y velocidad traslacional (v).

RGA0 =	D =																		
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">1.3460</td> <td style="padding: 2px 10px;">-0.8540</td> <td style="padding: 2px 10px;">0.5080</td> </tr> <tr> <td style="padding: 2px 10px;">0.1494</td> <td style="padding: 2px 10px;">2.5871</td> <td style="padding: 2px 10px;">-1.7365</td> </tr> <tr> <td style="padding: 2px 10px;">-0.4954</td> <td style="padding: 2px 10px;">-0.7331</td> <td style="padding: 2px 10px;">2.2285</td> </tr> </table>	1.3460	-0.8540	0.5080	0.1494	2.5871	-1.7365	-0.4954	-0.7331	2.2285	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">0.8370</td> <td style="padding: 2px 10px;">-1.1282</td> <td style="padding: 2px 10px;">2.1812</td> </tr> <tr> <td style="padding: 2px 10px;">1.8459</td> <td style="padding: 2px 10px;">-6.0353</td> <td style="padding: 2px 10px;">16.5893</td> </tr> <tr> <td style="padding: 2px 10px;">8.6583</td> <td style="padding: 2px 10px;">-16.6668</td> <td style="padding: 2px 10px;">71.0248</td> </tr> </table>	0.8370	-1.1282	2.1812	1.8459	-6.0353	16.5893	8.6583	-16.6668	71.0248
1.3460	-0.8540	0.5080																	
0.1494	2.5871	-1.7365																	
-0.4954	-0.7331	2.2285																	
0.8370	-1.1282	2.1812																	
1.8459	-6.0353	16.5893																	
8.6583	-16.6668	71.0248																	

Figura 2.6 Matriz de ganancias relativas y matriz de desacoplamiento

Figura 2.7 muestra el modelo de la planta desacoplado $G^* = GD$. Se puede apreciar que, aunque no se desacople por completo el modelo, las ganancias cruzadas son muy pequeñas en comparación a las ganancias diagonales de la función de transferencia.

$\frac{48.298 (s+2.25) (s+2.65)^2 (s+25.5)^2}{(s+25.5)^3 (s+2.65)^3}$	$\frac{-2.8016e-12 (s+2.65)^2 (s+25.5)}{(s+25.5)^3 (s+2.65)^3}$	$\frac{4.4408e-13 (s+25.5) (s+2.65)^2}{(s+25.5)^3 (s+2.65)^3}$
0	$\frac{-0.67507 (s+1.575)^2}{(s+1.575)^3}$	$\frac{-7.5533e-34}{(s+1.575)^3}$
$\frac{3.4627e-16 (s+2.375)}{(s+2.375)^3}$	$\frac{1.2578e-16 (s+2.375)}{(s+2.375)^3}$	$\frac{0.074528 (s+2.375)^2}{(s+2.375)^3}$

Figura 2.7 Función de transferencia desacoplada.

Con el modelo desacoplado se encuentran los parámetros de los controladores PI y la matriz $C(s)$ queda definida como:

$$C(s) = \begin{bmatrix} 0.752 + 60.1/s & 0 & 0 \\ 0 & -2.43 - 11.5/s & 0 \\ 0 & 0 & 33.2 + 237/s \end{bmatrix} \quad (2.11)$$

Resultados del afinamiento de los controladores PI y el esquema del controlador implementado en Simulink se encuentran en el Apéndice D. Este sistema de control fue implementado en el ambiente Matlab-Simulink y en Python obteniendo los mismos resultados.

2.3.2 MPC

El MPC lineal fue diseñado para minimizar el error en las salidas del sistema, y con menor importancia, minimizar la variabilidad de las señales manipuladas. El horizonte de predicción seleccionado fue de 10, el horizonte de control seleccionado fue de 3 y el tiempo de muestreo fue de 0.1 segundos. Este controlador también fue afinado para responder favorablemente a las perturbaciones del proceso.

Para implementar el MPC, fue necesario llevar el modelo linealizado del proceso a tiempo discreto. Figura 2.8 muestra el modelo de la planta discretizado. Las filas corresponden a las variables F_z , F_x y F_y en ese orden; mientras que las columnas corresponden a las variables d , ω y v en ese orden.

1.7024 (z-0.7973)	-0.48979 (z-0.7973)	0.062117 (z-0.7973)
(z-0.7672) (z-0.07808)	(z-0.7672) (z-0.07808)	(z-0.7672) (z-0.07808)
-0.019294	-0.06246	0.015181
(z-0.8543)	(z-0.8543)	(z-0.8543)
-0.048018	-0.0093433	0.0066338
(z-0.7886)	(z-0.7886)	(z-0.7886)

Figura 2.8 Modelo discretizado de la planta

Los valores nominales de las variables de salida corresponden al valor del set-point determinado en la sección 2.1.3, mientras que los valores nominales de las variables de entrada corresponden a los valores que cuando ingresan al modelo linealizado en estado estacionario, resultan en el set-point seleccionado. Figura 2.9 muestra estos valores nominales.

Plant Inputs					
Channel	Type	Name	Unit	Nominal Value	Scale Factor
u(1)	MV	d		3.97992923622094	1
u(2)	MV	w		1.44166678069738	1
u(3)	MV	v		4.79406230433704	1

Plant Outputs					
Channel	Type	Name	Unit	Nominal Value	Scale Factor
y(1)	MO	Fz		2.9	1
y(2)	MO	Fx		0.5	1
y(3)	MO	Fy		0.23	1

Figura 2.9 Valores nominales de las entradas y salidas en el diseño del MPC

Debido a que no hay restricciones duras para las entradas y salidas del sistema, la única restricción que se añadió es que ninguna de las variables del sistema puede ser negativa.

Debido a que se desea dar mayor importancia a las salidas del sistema, los pesos asignados a las variaciones de las entradas del sistema son menores a los de las

salidas. Respecto a las salidas, para equilibrar el hecho que la variable F_z es de un orden de magnitud mayor que las otras dos salidas, se emplea un peso menor para esta variable. Estas condiciones se reflejan en la siguiente función objetivo del proceso de optimización del MPC:

$$J = 0.3 \sum_{i=1}^{n_p} e_{i,F_z}^2 + \sum_{i=1}^{n_p} [e_{i,F_x}^2 + e_{i,F_y}^2] + 0.05 \sum_{i=1}^{n_c} [\Delta_{i,d}^2 + \Delta_{i,\omega}^2 + \Delta_{i,v}^2] \quad (2.12)$$

Donde J es la función objetivo a minimizar por parte del MPC, e corresponde a la señal de error de las variables de salida, Δ corresponde a la variación de las variables de entrada, n_p es el horizonte de predicción y n_c es el horizonte de control.

El MPC fue completamente diseñado en el MPC Designer de Matlab, luego fue implementado y probado en un entorno Matlab-Simulink. Imágenes del afinamiento del controlador y su implementación en Simulink se encuentran en el Apéndice D.

2.3.3 Controlador inteligente

El controlador que emplea la RNN fue completamente diseñado y probado en Python bajo el ambiente PyTorch. El modelo fue transformado a tensores de Pytorch para agilizar los procesos de entrenamiento e inferencia. El proceso de entrenamiento consistió en 500 epochs o iteraciones (N), cada iteración tuvo una duración (T) de 100 s, con perturbaciones generadas aleatoriamente siguiendo distribuciones normales con $\mu = Y_{sp}$.

La Figura 2.10 muestra el modelo de la planta implementado con tensores de Pytorch para mayor facilidad de entrenamiento del controlador inteligente. La Figura 2.11 muestra el controlador implementado en Pytorch.


```

def model(X, t, u, dudt, K, pwt, Ia, rel, tau, mat, per):
    X = torch.from_numpy(X).float().to(device)
    dF = (K * torch.prod(u**(pwt-1)*(Ia*dudt+u), axis=1) - torch.matmul(
        rel, X) + per) / tau + torch.matmul(mat, X)
    X_temp = X + step_size * dF
    return X_temp

```

Figura 2.10 Modelo implementado en Python empleando tensores de Pytorch

```

class Net(torch.nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.rnn = torch.nn.GRU(7, 32, n)
        self.body = torch.nn.Linear(32, 3)

    def forward(self, x):
        x, _ = self.rnn(x)
        x = self.body(x)
        return x

```

Figura 2.11 Controlador inteligente implementado en Pytorch

Para el entrenamiento del controlador, la función de pérdida J escogida es el estándar error cuadrático medio, similar al caso del MPC, con la diferencia de que los parámetros entrenables de la red θ se codifican en el proceso de entrenamiento minimizando dicha función de pérdida:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} J(\operatorname{Net}(\theta, X, U), Y_{sp}) \quad (2.13)$$

$$\theta^* = \underset{\theta_i}{\operatorname{argmin}} \sum_j \sum [\operatorname{Net}(\theta_{i,j}, X_{i,j}, U_{i,j}) - Y_{sp,i,j}]^2$$

En la Ecuación 2.13, θ^* corresponde a los parámetros óptimos del controlador, X es el vector de estados, U es el vector de entradas del paso anterior y Y_{sp} corresponde al set-point.

El tamaño de cada capa en los módulos (número de filas en las matrices

$A_f, B_f, C_f, D_f, A_h, B_h, C_h, D_h$) fue de 32, a excepción del último módulo, el cual por consistencia debía tener un tamaño de 3, igual al número de entradas del proceso. Mientras que el número de módulos GRU n , fue determinado mediante experimentos variando dicho hiper-parámetro entre 1 y 5. La Tabla 2.1 muestra el mínimo error transitorio obtenido al finalizar el entrenamiento con cada número de módulos.

Tabla 2.1 Error obtenido al final del entrenamiento del controlador inteligente empleando diferente número de módulos

Número de módulos (n)	Error mínimo obtenido
1	2.225
2	1.264
3	1.081
4	0.959
5	0.957

De acuerdo con la Tabla 2.1, el error transitorio es cada vez menor a medida que se incrementa el valor de n . Sin embargo, esta mejora cada vez es más pequeña. El cambio entre $n = 4$ y $n = 5$ es mínimo, por lo que se selecciona $n = 4$ para el proceso de comparación con los demás controladores. Figura 2.12 muestra el resumen de la red del controlador inteligente.

```

=====
Layer (type:depth-idx)          Output Shape          Param #
=====
├GRU: 1-1                       [-1, 1, 32]          22,944
├Linear: 1-2                     [-1, 1, 3]           99
=====
Total params: 23,043
Trainable params: 23,043
Non-trainable params: 0

```

Figura 2.12 Resumen de la red del controlador inteligente

El número total de parámetros de la red es de 23043, un número grande en comparación a los sistemas de control tradicionales y pequeño en comparación a otras aplicaciones de redes neuronales. En el Apéndice E se encuentra una

muestra de los valores del controlador entrenado y el progreso de la minimización de la función de pérdida durante el entrenamiento.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

Los experimentos computacionales se realizaron empleando un computador con un procesador de octava generación Intel i7 de 2.2 GHz. Con 8 GB de RAM y una tarjeta gráfica Nvidia GTX 1060 con 6 GB de vRAM.

Primero se muestran los resultados de la comparación de las salidas del proceso en lazo cerrado con perturbaciones en un paso de tiempo dado y proveyendo tiempo al sistema, entre perturbaciones, para que éste se estabilice. Seguido, se presenta la comparación de las salidas de los controladores, después se muestra la comparación de los errores obtenidos y finalmente, se presenta una comparación con perturbaciones de paso en tiempos definidos aleatoriamente.

3.1 Comparación de respuestas del sistema en lazo cerrado

Las Figuras 3.1-3.3 muestran el comportamiento dinámico de los controladores para cada fuerza evaluada, considerando sus capacidades para rechazar perturbaciones. Perturbaciones artificiales fueron inducidas al sistema para observar las correspondientes respuestas dinámicas. Las mismas perturbaciones fueron empleadas para cada sistema de control.

Se escogen perturbaciones paso en las salidas para simular efectos de variables externas desconocidas que influyan directamente en las salidas del proceso. Se realizan perturbaciones paso de subida y bajada para observar si el sistema regresa a las condiciones de equilibrio después de haber sido perturbado. En cada simulación se perturba las tres variables de salida con una separación de 10 segundos entre cada perturbación, empezando a los 10 segundos con F_z , luego F_x y finalmente F_y . Luego se da una perturbación en sentido contrario con la misma magnitud para regresar cada variable a su estado sin perturbación. Estas perturbaciones también se realizan con 10 segundos de separación y se empieza a los 40 segundos de la simulación, las variables son perturbadas en el mismo orden que en el caso de las perturbaciones paso de subida.

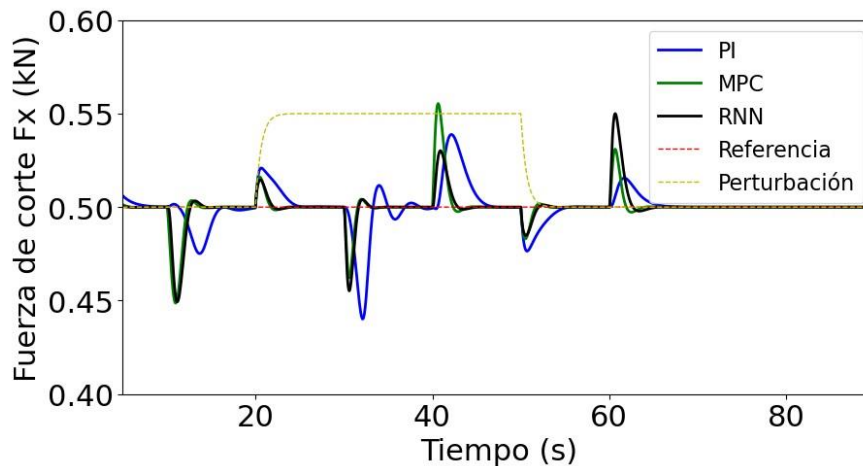


Figura 3.1 Comparación de la respuesta de la fuerza de corte F_x entre diferentes sistemas de control al rechazar perturbaciones

En la Figura 3.1 se observa la comparación de la fuerza de corte F_x empleando los tres controladores. Se observa que la respuesta del controlador PI tiene tiempos de asentamiento mayores en todas las perturbaciones. El tiempo de asentamiento del MPC y del controlador inteligente son iguales a diferencia del segundo 60 en el cual hay una perturbación de bajada en F_y , aquí el tiempo de asentamiento del MPC es ligeramente menor que el del controlador inteligente. La variable se desvía del setpoint significativamente en todos los casos, sin embargo, en la mayoría de los casos el PI se desvía mayormente, a excepción de los segundos 10 y 60. El MPC y el controlador RNN tienen un desempeño similar, a excepción del segundo 40 donde el RNN es mejor y el segundo 60 donde el MPC mejor.

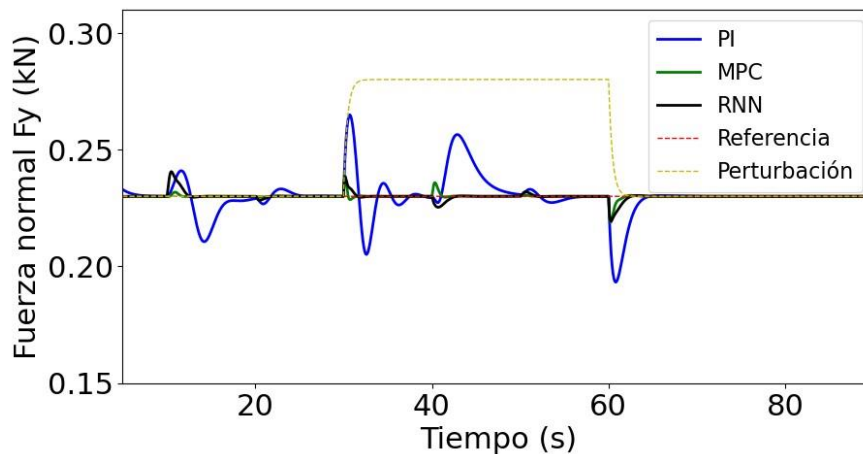


Figura 3.2 Comparación de la respuesta de la fuerza normal F_y entre diferentes sistemas de control al rechazar perturbaciones

En la Figura 3.2 se observa la comparación de la fuerza de corte F_y empleando los tres controladores. Se observa que el controlador PI logra cumplir el objetivo de seguir la referencia en el sistema perturbado pero su desempeño es considerablemente inferior al de los otros dos controladores, esto puede deberse a que el afinamiento se realizó en el sistema linealizado y fue hecho antes de realizar el desacoplo. En el caso del MPC y del controlador inteligente, ambos se desempeñan similarmente, las diferencias radican en el segundo 10 donde el MPC es mejor y en el segundo 40 donde el controlador RNN es mejor.

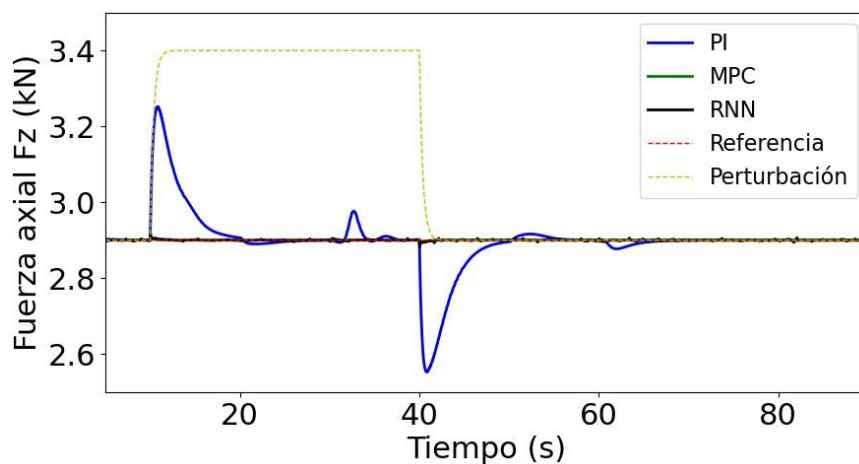


Figura 3.3 Comparación de la respuesta de la fuerza axial F_z entre diferentes sistemas de control al rechazar perturbaciones

En la Figura 3.3 se observa la comparación de la fuerza de corte F_y empleando los tres controladores. El controlador PI tiene un desempeño considerablemente menor que los otros dos controladores en los segundos 10 y 40, los cuales corresponden a las perturbaciones de subida y bajada de la salida F_z . Es importante notar que el desempeño del controlador PI respecto a la magnitud de la perturbación no ha cambiado significativamente en comparación a las Figuras 3.1 y 3.2, sino que los otros dos controladores se ajustan rápidamente a esta señal dando como resultado un error transitorio despreciable. A pesar de que se trató de compensar con un menor peso a F_z en la función de optimización del MPC, se observa que este controlador se desempeña mejor en esta variable en comparación con las otras dos salidas. En el caso del controlador inteligente es de esperarse que F_z contribuya más a la dirección del gradiente en el proceso de entrenamiento, debido a que F_z

tiene una mayor magnitud que las otras dos variables, causando que el sistema de control sea mucho mejor en controlar esta variable.

En los casos del MPC y del controlador inteligente propuesto, el controlador logra mantener al sistema en su referencia deseada, mientras que el controlador PI se desempeña significativamente peor, presentando acoplamientos entre las variables, a pesar de que los controladores han sido desacoplados, esto puede darse debido a que el desacoplo fue estático y este no garantiza el desacoplo en la etapa transitoria. En el caso de la fuerza normal (Figura 3.2), hay un pequeño retraso en la respuesta del controlador inteligente cuando se compara con la respuesta del MPC. Esto se puede haber inducido por la presencia de un mínimo local durante el proceso de entrenamiento, en el cual la fuerza normal presenta menor importancia en comparación a las otras dos fuerzas.

Se debe notar que los estados iniciales para la simulación ilustradas en las Figuras 3.1-3.3, corresponden a las referencias propuestas para lograr soldaduras sin defectos. Estas condiciones iniciales fueron seleccionadas, debido a que, aunque el MPC lineal y el controlador RNN logran alcanzar la referencia propuesta en todos los casos de las tres fuerzas, los controladores PI desacoplados presentan inestabilidad al alcanzarla desde el reposo. Esta condición es particularmente notoria cuando el sistema comienza con las condiciones $F_z(0) = 0$, $F_x(0) = 0$ y $F_y(0) = 0$.

3.2 Comparación del esfuerzo de los controladores

Las señales de las variables manipuladas, la profundidad (d), la velocidad traslacional (v) y la velocidad rotacional (ω), son ilustradas en Figuras 3.4-3.6.

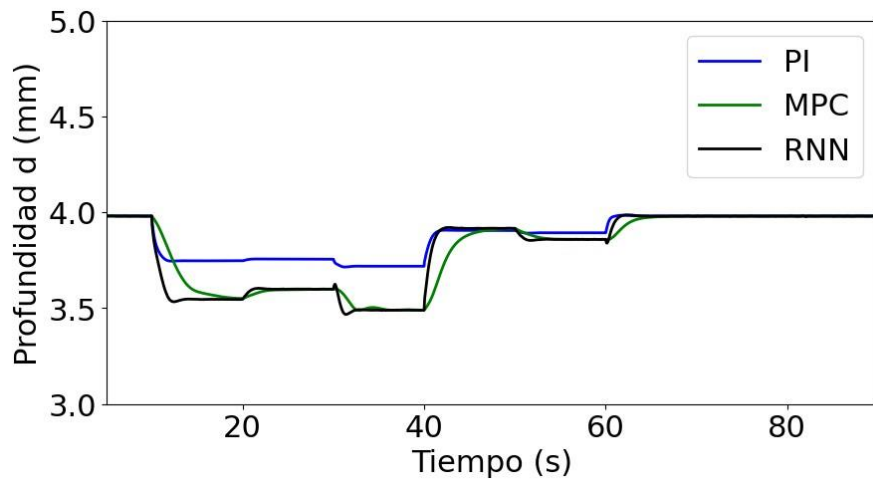


Figura 3.4 Comparación de la profundidad d entre diferentes sistemas de control al rechazar perturbaciones

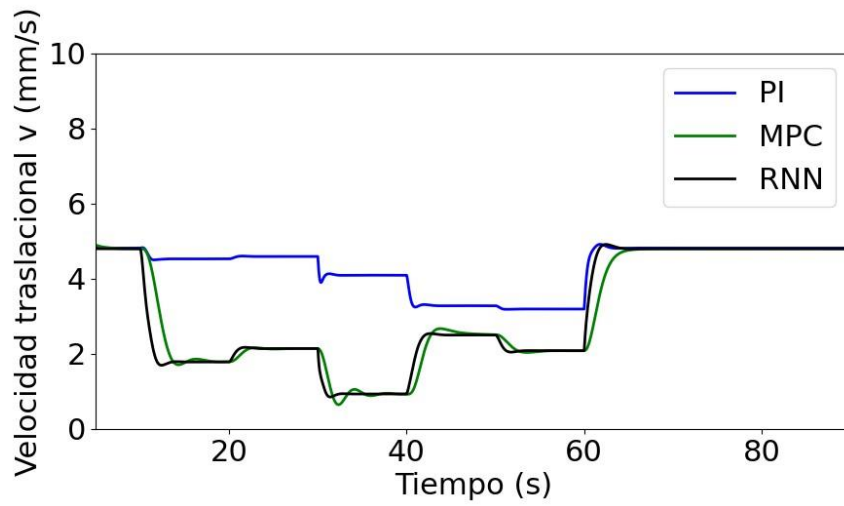


Figura 3.5 Comparación de la velocidad traslacional v entre diferentes sistemas de control al rechazar perturbaciones

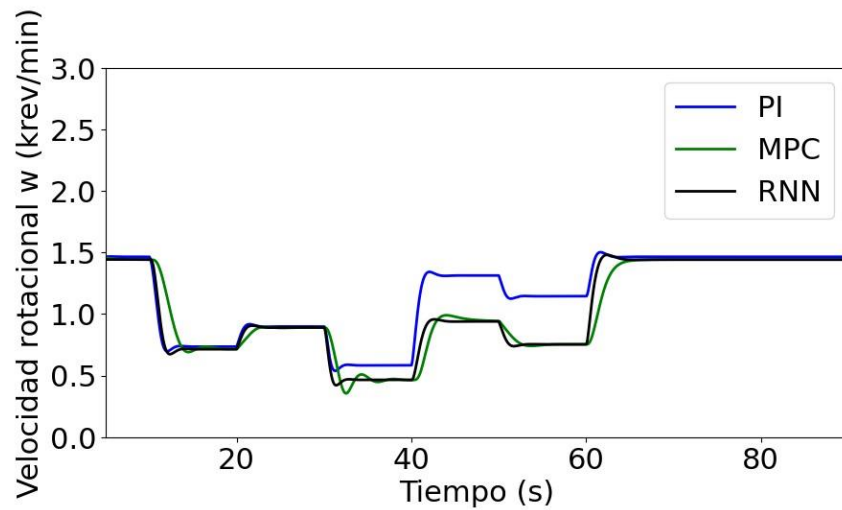


Figura 3.6 Comparación de la velocidad rotacional ω entre diferentes sistemas de control al rechazar perturbaciones

Se puede observar que las acciones de control de los controladores PI desacoplados son considerablemente diferentes cuando se comparan a las otras arquitecturas bajo estudio. Esto es particularmente notorio en el caso de la velocidad traslacional, la cual es significativamente diferente hasta que el tiempo alcanza los 60 segundos. Las mismas condiciones de estado estacionario se consiguen debido a la no linealidad del proceso con respecto a las entradas, y al hecho que diferentes combinaciones posibles de entradas pueden resultar en salidas del proceso similares, es decir, la inyectividad del proceso no es garantizada si el proceso es no lineal. Este comportamiento es consistente con el hecho de que los controladores PI desacoplados fallan en alcanzar la referencia en un tiempo prudente.

En el caso del MPC, las variables manipuladas muestran pequeños retrasos en cada paso, y también exhiben *overshoots* en algunas de las perturbaciones, mientras que las variables manipuladas en el caso del controlador RNN presentan dinámicas más suaves y atenuadas. Además, aunque el MPC lineal y el controlador inteligente demuestran tener un comportamiento similar, y ambos convergen a las mismas condiciones estacionarias en cada paso. El controlador RNN muestra menos oscilaciones en su acción de control en comparación al MPC, pero ambos controladores tienen un desempeño similar en el rechazo de las perturbaciones.

3.3 Comparación del error acumulado

Las Figuras 3.7 y 3.8 ilustran la efectividad de los tres controladores en lograr la referencia deseada, la cual fue definida en la sección 2.1.3, bajo diferentes perturbaciones en términos del error porcentual. El error es calculado como la distancia euclidiana entre la localización actual en el espacio de fuerzas y las coordenadas de la referencia propuesta.

En este contexto, lo que se evaluó es la rapidez y efectividad para retornar al punto deseado, el cual tiene la máxima probabilidad en obtener piezas soldadas libres de defectos y con buenas propiedades mecánicas.

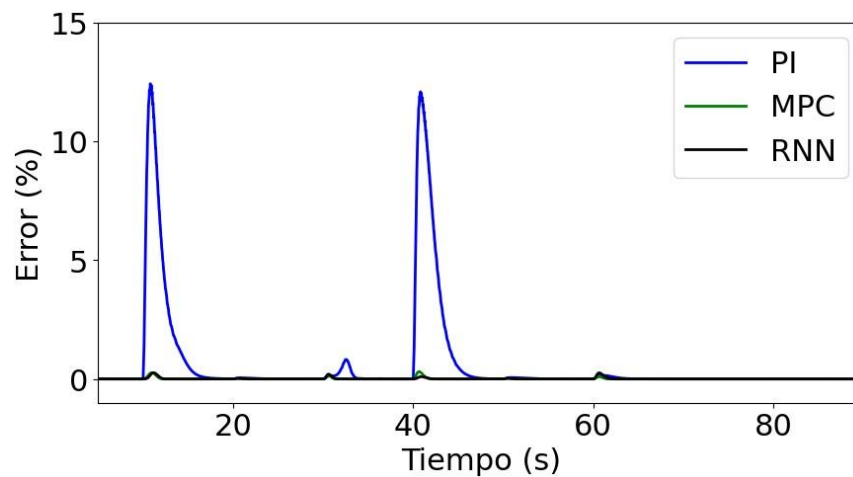


Figura 3.7 Serie de tiempo del error de los tres controladores en el sistema perturbado

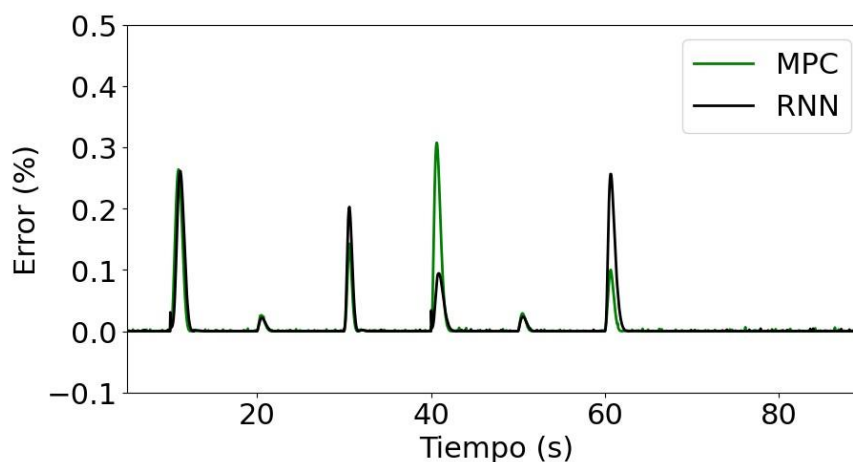


Figura 3.8 Serie de tiempo del error del MPC y del controlador inteligente en el sistema perturbado

La Figura 3.7 muestra la serie de tiempo del error porcentual de la simulación introducida previamente en las Figuras 3.1-3.6. Se puede observar que el error de los controladores PI desacoplados es considerablemente mayor que el error de los otros controladores, especialmente a los 10 y 40 segundos. En estos instantes se incorporó las perturbaciones en la fuerza axial (F_z) (Figura 3.3).

Para una mejor apreciación de las diferencias en el error entre el MPC lineal y el controlador inteligente, Figura 3.8 ilustra la evolución del error de estos dos controladores bajo las diferentes perturbaciones. Se puede observar que el desempeño del MPC es mejor cuando la fuerza normal (F_y) es la variable perturbada; mientras que el controlador inteligente tiene un mejor desempeño cuando la perturbación es introducida en la fuerza axial (F_z). Finalmente, cuando la fuerza de corte (F_x) es la variable perturbada, aparentemente ambos controladores responden favorablemente con un desempeño similar.

La Tabla 3.1 enlista el error cuadrático medio porcentual para todos los controladores y una comparación en contraste con el controlador inteligente.

Tabla 3.1 Error cuadrático medio empleando diferentes controladores y porcentaje de mejora del controlador inteligente

Controlador	Error cuadrático medio (%)	Porcentaje de mejora (%)
PI desacoplado	2.315	38.10
MPC	1.644	12.83
RNN	1.433	---

El controlador inteligente es el que tiene el menor error acumulado, es decir, es el que menos se desvía de la referencia propuesta. La estrategia de control empleando los controladores PI desacoplados es la que obtiene considerablemente mayor error que las otras dos, además, también exhibe problemas significativos al arrancar el proceso. El MPC lineal demuestra un mayor error que el controlador inteligente, aunque fue minuciosamente diseñado y afinado para eliminar perturbaciones, sin embargo y a pesar de mostrar una respuesta aceptable, su desempeño es inferior al del controlador inteligente.

El hecho de que ambos controladores, MPC y RNN, se desempeñan de manera ejemplar, puede atribuirse al paso de optimización. Este paso no fue realizado en línea en el caso del controlador inteligente, mientras que el MPC siempre optimiza en línea, considerando todas las entradas y salidas simultáneamente. Además, el proceso de alimentación de datos a la RNN y su entrenamiento fue de manera similar al esquema de optimización de un MPC. Ambos controladores reciben un vector de estados y entradas en una ventana de tiempo dada. La diferencia radica en que el MPC optimiza en línea de acuerdo con las predicciones futuras del modelo asociado, mientras que el controlador inteligente optimiza fuera de línea y codifica la dinámica del sistema en los pesos de la red.

3.4 Comparación empleando perturbaciones aleatorias

Finalmente, se realizó una comparación entre los sistemas de control con respecto a su respuesta ante perturbaciones aleatorias cada 5 segundos. El objetivo es analizar sus respuestas dinámicas para ajustar el proceso continuamente sin llegar al estado estacionario. Figuras 3.9-3.11 muestran las respuestas del sistema en lazo cerrado con estas perturbaciones.

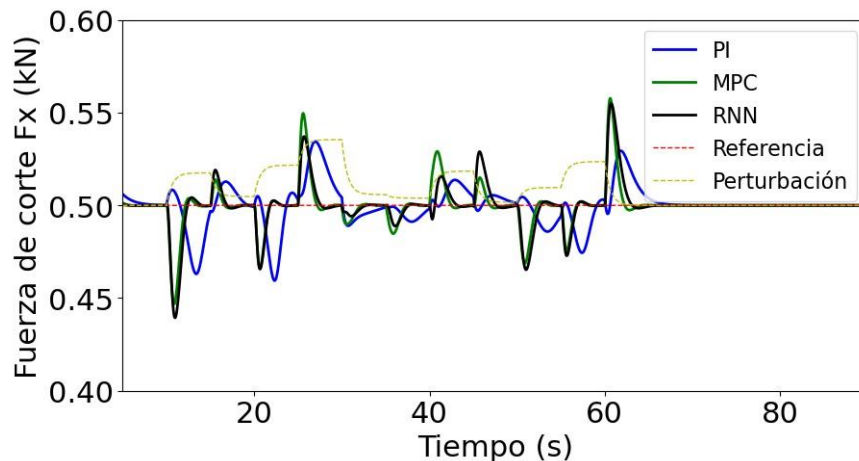


Figura 3.9 Respuesta de la fuerza de corte F_x entre diferentes sistemas de control al rechazar perturbaciones aleatorias

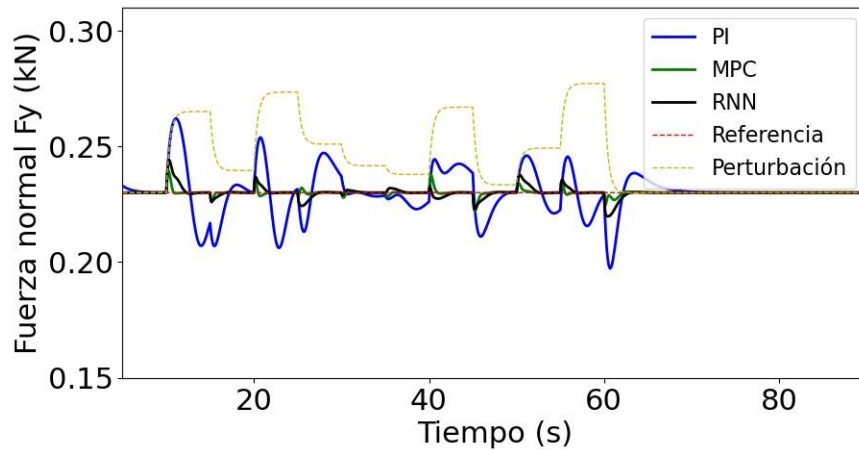


Figura 3.10 Respuesta de la fuerza normal F_y entre diferentes sistemas de control al rechazar perturbaciones aleatorias

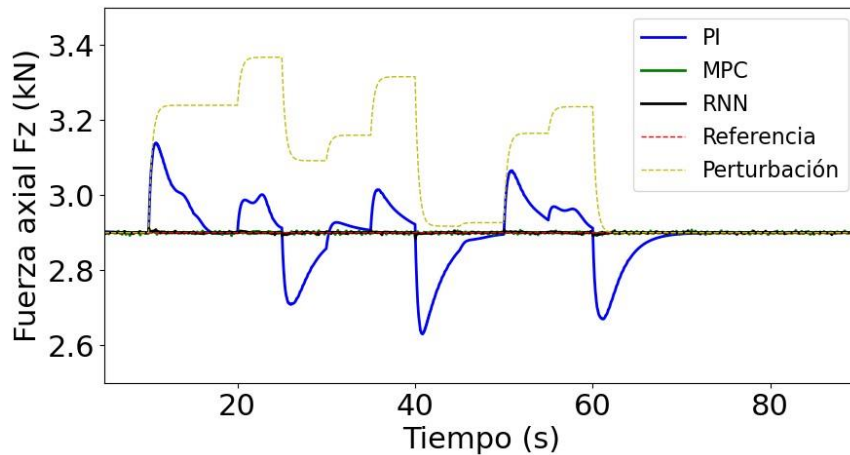


Figura 3.11 Respuesta de la fuerza axial F_z entre diferentes sistemas de control al rechazar perturbaciones aleatorias

Se puede apreciar que los resultados mostrados en las Figuras 3.9-3.11 son consistentes con los resultados mostrados en las Figuras 3.1-3.3. El conjunto de controladores PI desacoplados no puede mantener el sistema en la referencia, desviándose significativamente por largos periodos de tiempo.

A pesar de ser un escenario más desafiante, el MPC lineal y el controlador inteligente logran reestablecer el sistema al set-point propuesto en tiempos cortos. Las mayores desviaciones de las salidas del sistema ante las perturbaciones, empleando estos dos controladores, se observan en la fuerza de corte (F_x). Donde la magnitud del error llega a ser igual en ambos casos a la del controlador PI; sin

embargo, el MPC y el controlador RNN son más rápidos que el conjunto de controladores PI al momento de regresar a la referencia seleccionada.

Para observar mejor la comparación entre el controlador inteligente y el MPC, se muestra la serie de tiempo del error cuadrático medio de estos dos controladores en la Figura 3.12. Se omite a los controladores PI desacoplados, debido a que se aprecia en las Figuras 3.9-3.11 que su error asociado es considerablemente mayor al de los otros dos controladores.

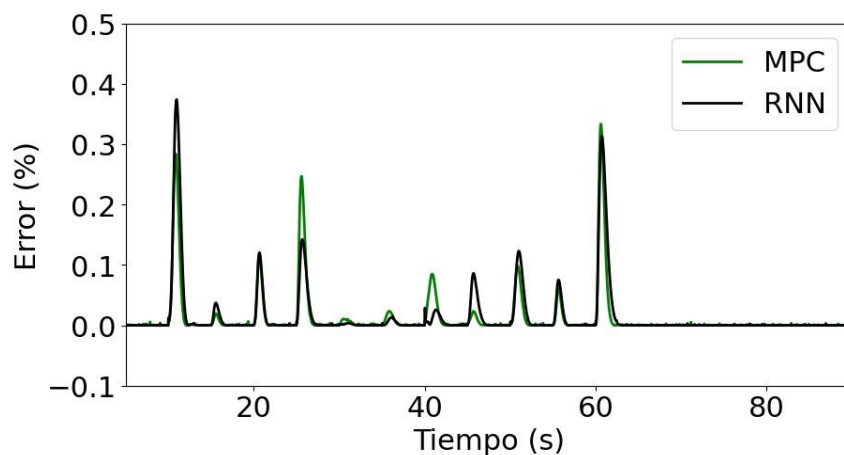


Figura 3.12 Serie de tiempo del error del MPC y del controlador inteligente en el sistema perturbado aleatoriamente

Se puede apreciar que ambos controladores son bastante competitivos en el problema dado. Los picos se dan cada 5 segundos, el cual es el tiempo en el que se da cada perturbación con magnitud aleatoria. En algunas ocasiones el controlador inteligente logra menor error, mientras que en otras ocasiones el MPC se desempeña mejor. La Tabla 3.2 resume los resultados mostrados en la Figura 3.12, y también muestra el porcentaje de mejora del controlador inteligente en comparación a sus dos contrapartes.

Tabla 3.2 Error cuadrático medio del sistema ante perturbaciones aleatorias y porcentaje de mejora del controlador inteligente

Controlador	Error cuadrático medio (%)	Porcentaje de mejora (%)
PI desacoplado	1.993	36.78
MPC	1.504	16.22
RNN	1.260	---

Se observa en la Tabla 3.2 que nuevamente, el controlador inteligente logra obtener el menor error cuadrático acumulado en toda la serie de tiempo, mientras que el controlador PI desacoplado sigue obteniendo el mayor error. Sin embargo, a diferencia de los resultados mostrados en la Tabla 3.1, el conjunto de controladores PI logró desempeñarse mejor en este experimento, a diferencia del MPC, cuyo desempeño disminuyó en esta simulación.

Debido a que el último experimento fue más desafiante que el primero, la métrica del error cuadrático medio puede ser engañosa. Sin embargo, el porcentaje de mejora es una medida relativa entre los tres controladores y se puede notar que el controlador PI desacoplado subió su nivel de desempeño mientras que el MPC bajó; de acuerdo con el hecho de que el porcentaje de mejora del controlador RNN bajó en el caso del controlador PI y subió en el caso del MPC.

3.5 Ventajas y desventajas de cada sistema de control según los resultados obtenidos

El sistema de control PI desacoplado es el más sencillo de implementar, es el más intuitivo y es el controlador que requiere de menor velocidad de procesamiento y de memoria. Solo posee dos parámetros por cada par de entrada y salida, y estos parámetros se los fija con anterioridad, por lo cual tiende a ser menos robusto a nuevas situaciones. Los resultados muestran que su desempeño es inferior al de los otros controladores y si se incrementa la magnitud de las perturbaciones o se disminuye el tiempo entre perturbaciones, este es el primer controlador que falla en estabilizar al sistema.

El MPC es menos sencillo e intuitivo de implementar en comparación al controlador PI y requiere mayor velocidad de procesamiento y de memoria. El MPC tiene que encontrar las entradas del sistema óptimas para minimizar su función objetivo en cada paso, lo cual es un proceso costoso que debe ejecutarse en cada paso de tiempo. En los resultados obtenidos se observa que el MPC se desempeña de manera favorable y tiene resultados similares al controlador inteligente.

Finalmente, el controlador inteligente es el menos intuitivo de implementar debido a su inferior trayectoria en comparación al MPC, requiere mayor cantidad de memoria debido a su alto número de parámetros, pero requiere menor velocidad de cómputo debido a que todo el proceso de entrenamiento se realiza con anterioridad y en inferencia solo se realizan operaciones básicas. En los resultados obtenidos se observa que el controlador inteligente se desempeña de manera favorable y tiene resultados similares al MPC.

En las Tablas 3.1 y 3.2 los errores absolutos carecen de mayor significado debido a que la magnitud y tiempo entre perturbaciones son arbitrarios. Sin embargo, el valor relativo de los errores si es una medida válida para comparar los controladores y aunque la diferencia es mínima, si se observa un mejor desempeño por parte del controlador inteligente en comparación al MPC, aunque el MPC también puede lograr el objetivo de rechazar perturbaciones al proceso con un error transitorio aceptable.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En este trabajo, un sistema de control inteligente empleando una red neuronal con arquitectura RNN y módulos GRU fue propuesto. El controlador fue entrenado con la finalidad de obtener piezas soldadas libres de defectos y con buenas propiedades mecánicas.

Se empleó un modelo empírico del proceso validado para la aleación estudiada. Dicho modelo presentó no linealidades con respecto a las entradas y fue transformado a un conjunto de ecuaciones diferenciales para poder ser empleado junto al sistema de control propuesto.

Aplicando los teoremas de estabilidad de Lyapunov se probó que el modelo es estable en lazo abierto. Estos teoremas son aplicables al sistema FSW, ya que el modelo del proceso es lineal en relación con los estados del mismo, sin importar el hecho de que el modelo sea no lineal con respecto a las entradas del proceso.

El punto de referencia óptimo en el espacio de fuerzas para el proceso FSW fue determinado empleando kernels con los datos etiquetados, para obtener regiones de alta probabilidad, particularizándolo para el caso del material Al 6061.

Al momento de comparar el controlador inteligente con otras estrategias de control comunes, el controlador RNN demostró tener un desempeño notablemente superior al de los controladores PI desacoplados en todos los casos estudiados bajo perturbaciones similares.

Los experimentos computacionales permitieron verificar la capacidad del sistema de control de propuesto para rechazar perturbaciones inducidas y lograr mantener el set-point deseado.

El controlador inteligente logró tener un mejor desempeño que el MPC lineal, con la excepción del caso en el cual la fuerza normal era la variable perturbada.

El porcentaje de mejora del sistema de control propuesto sobre los otros controladores es significativo, obteniendo el menor error acumulativo en las variables de salida del proceso. Este porcentaje de mejora también demuestra que el controlador inteligente propuesto puede ser considerado como una alternativa competitiva para procesos FSW.

El sistema de control inteligente logra, bajo perturbaciones previamente definidas, un 38.10% mejor desempeño que los controladores PI desacoplados y un 12.83% mejor desempeño que el MPC lineal; y logra, bajo perturbaciones semi aleatorias, un 36.78% mejor desempeño que los controladores PI desacoplados y un 16.22% mejor desempeño que el MPC lineal; siguiendo la métrica del error cuadrático acumulado.

El control RNN puede verse como una estrategia de control óptima, donde los resultados de la optimización son codificados en los parámetros de la red neuronal.

El sistema de control propuesto es altamente eficiente en lograr la respuesta deseada, debido a que, al momento de inferir, el controlador inteligente solamente debe calcular una serie de operaciones lineales con funciones de activación simples.

4.2 Recomendaciones

Se recomienda explorar el paradigma de aprendizaje en línea para el sistema de control. Esto permitiría adaptarse a las variaciones de los parámetros en la planta, por ejemplo: diferentes materiales, diferentes parámetros de calidad, entre otros.

Emplear un proceso de entrenamiento basado en políticas, como es el caso del aprendizaje por refuerzo, en lugar de utilizar “*backpropagation*”, para poder utilizar modelos de procesos no diferenciables.

Implementar el sistema de control inteligente en un proceso FSW real, para validar el modelo, la estrategia propuesta y las simulaciones realizadas.

Al momento de implementar el sistema de control propuesto se debe verificar el modelo del proceso, debido a que los pesos de la red codifican un sistema óptimo en función del modelo del proceso empleado.

Debido a que las fuerzas involucradas en el proceso sirven como aproximación de la calidad final del producto. Se puede incorporar la variable de temperatura como otra variable que serviría como una aproximación a una medida de calidad del proceso de soldado.

Los sistemas que emplean redes neuronales son considerados buenos escalando con la dimensionalidad del problema. Se puede explotar este hecho si se emplea en procesos con alta dimensionalidad, como por ejemplo una refinería de petróleo.

BIBLIOGRAFÍA

- [1] R. W. Messler Jr, Principles of welding: processes, physics, chemistry, and metallurgy, John Wiley & Sons, 2008.
- [2] D. Lohwasser, "Friction Stir Welding in A350," in *EADS Research Workshop, Al RBUS*, 2005.
- [3] A. P. Reynolds and W. Tang, "Alloy, tool geometry, and process parameter effects on friction stir weld energies and resultant FSW joint properties," *Friction stir welding and processing*, pp. 15-23, 2001.
- [4] K. J. Colligan, J. Xu and J. R. Pickens, "Welding tool and process parameter effects in friction stir welding of aluminum alloys," *Friction stir welding and processing II*, vol. 181, 2003.
- [5] R. S. Mishra and Z. Y. Ma, "Friction stir welding and processing," *Materials science and engineering: R: reports*, vol. 50, pp. 1-78, 2005.
- [6] K. J. Colligan, "Relationships between process variables related to heat generation in friction stir welding of aluminum," in *Proc. 136th TMS Annual Meeting Friction Stir Welding & Processing IV Symposia*, 2007.
- [7] G. Cam, "Friction stir welded structural materials: beyond Al-alloys," *International Materials Reviews*, vol. 56, pp. 1-48, 2011.
- [8] G. Cam and G. Ipekoglu, "Recent developments in joining of aluminum alloys," *The International Journal of Advanced Manufacturing Technology*, vol. 91, pp. 18511866, 2017.
- [9] G. Cam, G. Ipekoglu, T. Kucukomeroglu and S. Aktarer, "Applicability of friction stir welding to steels," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 80, 2017.
- [10] A. Lakshminarayanan, V. Balasubramanian and K. Elangovan, "Effect of welding processes on tensile properties of AA6061 aluminium alloy joints," *The International Journal of Advanced Manufacturing Technology*, vol. 40, pp. 286-296, 2009.
- [11] A. Tribe and T. W. Nelson, "Study on the fracture toughness of friction stir welded API X80," *Engineering Fracture Mechanics*, vol. 150, pp. 58-69, 2015.
- [12] S. B. Aziz, M. W. Dewan, D. J. Huggett, M. A. Wahab, A. M. Okeil and T. W. Liao, "A fully coupled thermomechanical model of friction stir welding (FSW) and

- numerical studies on process parameters of lightweight aluminum alloy joints," *Acta Metallurgica Sinica (English letters)*, vol. 31, pp. 1-18, 2018.
- [13] P. A. Colegrove and H. R. Shercliff, "3-Dimensional CFD modelling of flow round a threaded friction stir welding tool profile," *Journal of materials processing technology*, vol. 169, pp. 320-327, 2005.
- [14] M. P. Iqbal, A. Tripathi, R. Jain, R. P. Mahto, S. Pal and P. Mandal, "Numerical modelling of microstructure in friction stir welding of aluminium alloys," *International Journal of Mechanical Sciences*, vol. 185, 2020.
- [15] R. V. Vignesh, R. Padmanaban, M. Arivarasu, S. Thirumalini, J. Gokulachandran and M. S. S. S. Ram, "Numerical modelling of thermal phenomenon in friction stir welding of aluminum plates," in *IOP Conference Series: Materials Science and Engineering*, 2016.
- [16] H. Zhang, Z. Zhang and J. Chen, "The finite element simulation of the friction stir welding process," *Materials Science and Engineering*, vol. 403, pp. 340-348, 2005.
- [17] M. W. Dewan, D. J. Huggett, T. W. Liao, M. A. Wahab and A. M. Okeil, "Prediction of tensile strength of friction stir weld joints with adaptive neuro-fuzzy inference system (ANFIS) and neural network," *Materials & Design*, vol. 92, pp. 288-299, 2016.
- [18] D. Huggett, T. Liao, M. Wahab and A. Okeil, "Prediction of friction stir weld quality without and with signal features," *The International Journal of Advanced Manufacturing Technology*, vol. 95, pp. 1989-2003, 2018.
- [19] X. Zhao, P. Kalya, R. G. Landers and K. Krishnamurthy, "Empirical dynamic modeling of friction stir welding processes," *Journal of manufacturing science and engineering*, vol. 131, 2009.
- [20] T. Kohler, M. Schiele, M. Glaser, K. Schrickler, J. Bergmann and K. Augsburg, "Insitu monitoring of hybrid friction diffusion bonded EN AW 1050/EN CW 004A lap joints using artificial neural nets," *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, vol. 234, pp. 766785, 2020.
- [21] H. Okuyucu, A. Kurt and E. Arcaklioglu, "Artificial neural network application to the friction stir welding of aluminum plates," *Materials & design*, vol. 28, pp. 78-84, 2007.
- [22] ISO, "Welding and allied processes-Classification of geometric imperfections in metallic materials-Part 1: Fusion welding," 2007.

- [23] O. Waelkhalifa, Abouelatta, I. Elamirgadelmawla and I. Elewa, "Classification of Welding Defects Using Gray Level Histogram Techniques via Neural Network," 2014.
- [24] W. R. Longhurst, A. M. Strauss and G. E. Cook, "Enabling automation of friction stir welding: the modulation of weld seam input energy by traverse speed force control," *Journal of dynamic systems, measurement, and control*, vol. 132, 2010.
- [25] W. R. Longhurst, A. M. Strauss, G. E. Cook and P. A. Fleming, "Torque control of friction stir welding for manufacturing and automation," *The International Journal of Advanced Manufacturing Technology*, vol. 51, pp. 905-913, 2010.
- [26] X. Zhao, P. Kalya, R. G. Landers and K. Krishnamurthy, "Design and implementation of nonlinear force controllers for friction stir welding processes," *Journal of Manufacturing Science and Engineering*, vol. 130, 2008.
- [27] T. A. Davis, Y. C. Shin and B. Yao, "Observer-based adaptive robust control of friction stir welding axial force," *IEEE/ASME transactions on mechatronics*, vol. 16, pp. 1032-1039, 2010.
- [28] T. A. Davis, P. D. Ngo and Y. C. Shin, "Multi-level fuzzy control of friction stir welding power," *The International Journal of Advanced Manufacturing Technology*, vol. 59, pp. 559-567, 2012.
- [29] S. Zhao, Q. Bi and Y. Wang, "An axial force controller with delay compensation for the friction stir welding process," *The International Journal of Advanced Manufacturing Technology*, vol. 85, pp. 2623-2638, 2016.
- [30] B. Eren, M. A. Guvenc and S. Mistikoglu, "Artificial intelligence applications for friction stir welding: a review," *Metals and Materials International*, pp. 1-27, 2020.
- [31] A. Fehrenbacher, N. A. Duffie, N. J. Ferrier, F. E. Pfefferkorn and M. R. Zinn, "Toward automation of friction stir welding through temperature measurement and closed-loop control," *Journal of manufacturing science and engineering*, vol. 133, 2011.
- [32] A. Fehrenbacher, N. A. Duffie, N. J. Ferrier, F. E. Pfefferkorn and M. R. Zinn, "Effects of tool-workpiece interface temperature on weld quality and quality improvements through temperature control in friction stir welding," *The International Journal of Advanced Manufacturing Technology*, vol. 71, pp. 165-179, 2014.
- [33] A. Fehrenbacher, C. B. Smith, N. A. Duffie, N. J. Ferrier, F. E. Pfefferkorn and M. R. Zinn, "Combined temperature and force control for robotic friction stir welding," *Journal of Manufacturing Science and Engineering*, vol. 136, 2014.

- [34] B. S. Taysom, C. D. Sorensen and J. D. Hedengren, "A comparison of model predictive control and PID temperature control in friction stir welding," *Journal of Manufacturing Processes*, vol. 29, pp. 232-241, 2017.
- [35] A. Bachmann, J. Gamper, M. Krutzlinger, A. Zens and M. Zaeh, "Adaptive modelbased temperature control in friction stir welding," *The International Journal of Advanced Manufacturing Technology*, vol. 93, pp. 1157-1171, 2017.
- [36] J. J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, pp. 3-10, 1988.
- [37] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.
- [38] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural computation*, vol. 28, pp. 815-825, 2016.
- [39] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, pp. 138, 2019.
- [40] Y. Wang, K. Velswamy and B. Huang, "A novel approach to feedback control with deep reinforcement learning," *IFAC*, vol. 51, pp. 31-36, 2018.
- [41] F. Bre, J. Gimenez and V. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks," *Energy and Buildings*, vol. 158, 2017.
- [42] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," 1985.
- [43] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.
- [44] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85-117, 2015.
- [45] W. Zhang, "Shift-invariant pattern recognition neural network and its optical architecture," in *Proceedings of annual conference of the Japan Society of Applied Physics*, 1988.
- [46] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, 2018.

- [47] H. Sak, A. W. Senior and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.
- [48] W. Bao, J. Yue and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," vol. 12, 2017.
- [49] S. a. S. J. Hochreiter, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735-1780, 1997.
- [50] F. A. Gers, N. N. Schraudolph and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of machine learning research*, vol. 3, pp. 115143, 2002.
- [51] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [52] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [53] S. Lim, S. Kim, C.-G. Lee and S. Kim, "Tensile behavior of friction-stri-welded Al 6061-T651," *Metallurgical and Materials Transactions A*, vol. 35, pp. 2829-2835, 2004.
- [54] G. Teschl, *Ordinary differential equations and dynamical systems*, vol. 140, American Mathematical Soc., 2012.
- [55] L. Barreira, C. Valls and C. Valls, *Stability of nonautonomous differential equations*, vol. 1926, Springer, 2008.
- [56] J. A. Romagnoli and A. Palazoglu, *Introduction to process control*, CRC Press, 2020.

APÉNDICE

APÉNDICE A

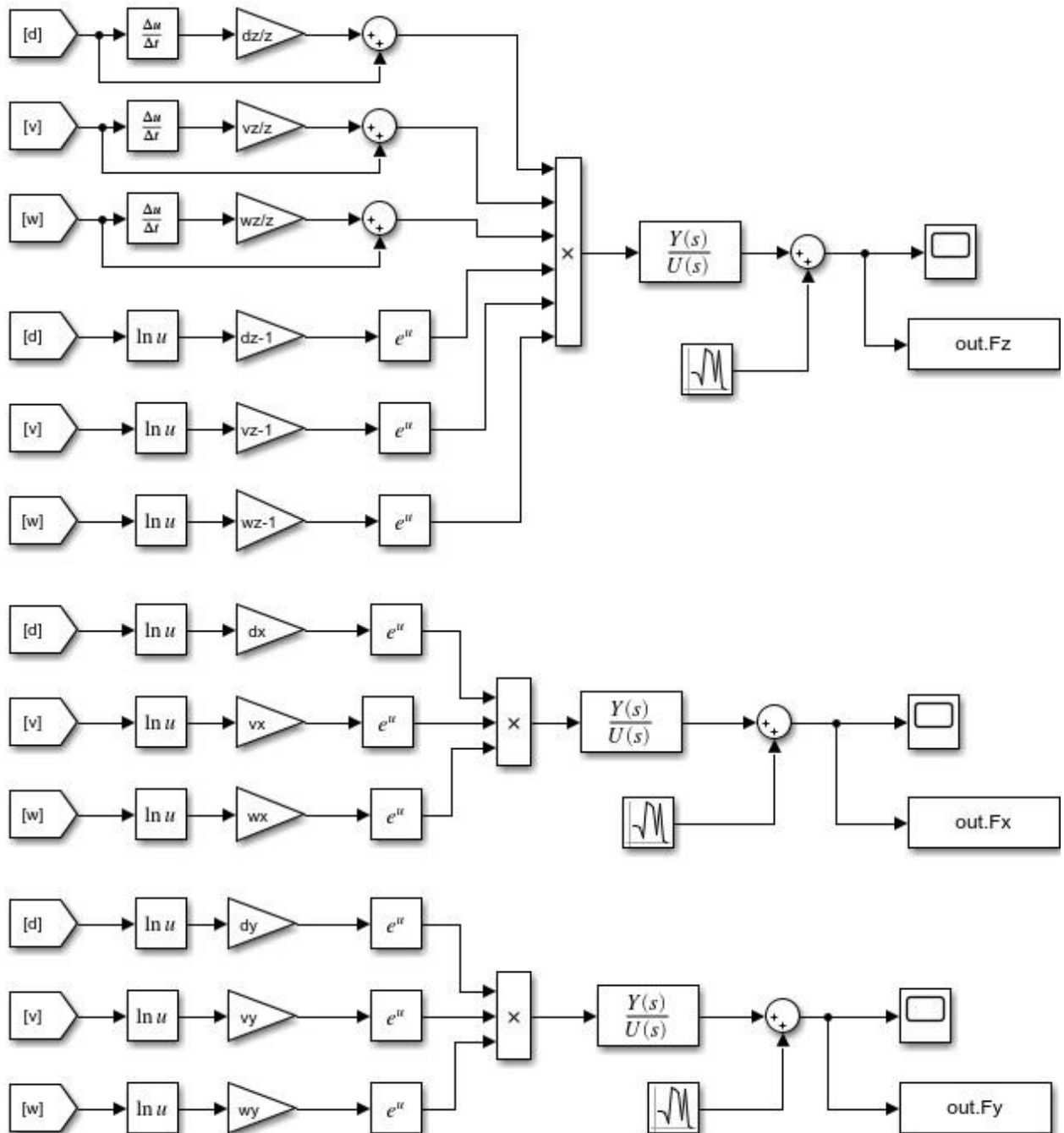


Figura A.1 Esquema del modelo del proceso implementado en Simulink

Tabla A.1 Constantes del proceso FSW

Constante	Valor
p_1	-25.5
p_2	-2.65

z	2.25
K_z	3.86
K_x	0.704
K_y	20.8
α_z	2.207
β_z	0.097
γ_z	-0.23
α_x	-1.054
β_x	0.999
γ_x	-1.236
α_y	-3.93
β_y	0.654
γ_y	-0.277
τ_x	0.635
τ_y	0.421

$$\begin{array}{ccc}
 \frac{48.298 (s+2.25)}{(s+25.5) (s+2.65)} & \frac{-13.895 (s+2.25)}{(s+25.5) (s+2.65)} & \frac{1.7623 (s+2.25)}{(s+25.5) (s+2.65)} \\
 \frac{-0.20853}{(s+1.575)} & \frac{-0.67507}{(s+1.575)} & \frac{0.16408}{(s+1.575)} \\
 \frac{-0.53946}{(s+2.375)} & \frac{-0.10497}{(s+2.375)} & \frac{0.074528}{(s+2.375)}
 \end{array}$$

Figura A.2 Función de transferencia de la planta

APÉNDICE B

Las pruebas de los dos axiomas de Lyapunov se detallan a continuación:

La prueba del primer punto se da suponiendo primero que el núcleo es Lyapunov estable. Por definición, el núcleo es Lyapunov estable si existe un $\delta > 0$ tal que $\|x_0\| < \delta \Rightarrow \|x(t)\| < \varepsilon$ para todo $t \geq 0$. Para las cuatro soluciones linealmente independientes $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$ de $\dot{x}(t) = Ax(t)$, se considera la matriz fundamental $\Phi(t) = [\Phi_1, \Phi_2, \Phi_3, \Phi_4]$ que satisface la condición inicial $\Phi(0) = \delta_{-2} I$. Por lo consiguiente, cualquier solución para $\dot{x}(t) = Ax(t)$, con condiciones iniciales arbitrarias, puede ser escrita como $x(t) = \Phi(t)c$ donde c es el vector columna de componentes c_j , ($1 \leq j \leq 4$), tal que $c = \Phi^{-1}(0)x_0$. Para cualquier $j = 1, 2, 3, 4$, debido a que $\|\Phi_j(0)\| = \delta_{-2} < \delta$, la estabilidad Lyapunov implica que $\|\Phi_j(t)\| < \varepsilon$; por consiguiente:

$$\|x(t)\| = \|\Phi(t)c\| = \left\| \sum_{j=1}^4 c_j \Phi_j(t) \right\| \leq \sum_{j=1}^4 \|c_j\| \|\Phi_j(t)\| \leq \varepsilon \sum_{j=1}^4 \|c_j\| < \infty. \quad (B.1)$$

Luego, suponemos que toda solución de $\dot{x}(t) = Ax(t)$ es acotada y consideramos que $\Phi(t)$ es una matriz fundamental. Dado que x es acotada, entonces existe una constante $M > 0$ tal que $\|\Phi(t)\| < M$ para todo $t \geq 0$ y cualquier norma matricial inducida $\|\cdot\|$. Dado cualquier $\varepsilon > 0$, se escoge $\delta = \frac{\varepsilon}{M\|\Phi^{-1}(0)\|} > 0$. Como cualquier

solución de $\dot{x}(t) = Ax(t)$ tiene la forma $x(t) = \Phi(t)\Phi^{-1}(0)x_0$, entonces para $\|x_0\| < \delta$ se tiene:

$$\|x(t)\| \leq \|\Phi(t)\| \|\Phi^{-1}(0)\| \|x_0\| \leq M \|\Phi^{-1}(0)\| \delta = \varepsilon. \quad (B.2)$$

Para probar el segundo punto se define a $x^*(t)$ como una solución del modelo dinámico en la ecuación 2.1, cuya estabilidad se debe determinar. Si se define a $x(t)$ como otra solución, y $y(t) = x(t) - x^*(t)$. Se tiene que:

$$\begin{aligned}\dot{\mathbf{y}}(t) &= A\mathbf{y}(t) \\ \mathbf{y}(0) &= \mathbf{x}_0 - \mathbf{x}_0^*\end{aligned}\tag{B.3}$$

La estabilidad Lyapunov de $\mathbf{x}^*(t)$ implica que para todo $\varepsilon > 0$, existe un $\delta > 0$, tal que $\|\mathbf{x}_0 - \mathbf{x}_0^*\| < \delta \Rightarrow \|\mathbf{x}(t) - \mathbf{x}^*(t)\| < \varepsilon$ para todo $t \geq 0$. En términos de \mathbf{y} , esto es equivalente a $\|\mathbf{y}_0\| < \delta \Rightarrow \|\mathbf{y}(t)\| < \varepsilon$, lo cual es la condición para la estabilidad Lyapunov del núcleo de acuerdo con el primer punto.

APÉNDICE C

Tabla C.1 Valores experimentales de los puntos utilizados para determinar la referencia

<i>rev/min</i>	<i>m/min</i>	Calidad	<i>krev/min</i>	<i>mm/min</i>	<i>mm/s</i>	F_z (kN)	F_x (kN)	F_y (kN)
1000	0.2	Malo	1	200	3.333333	3.07871	0.543697	0.196757
1000	0.3	Malo	1	300	5	3.202209	0.815216	0.256504
1400	0.2	Bueno	1.4	200	3.333333	2.849439	0.35871	0.179247
1400	0.3	Bueno	1.4	300	5	2.96374	0.537846	0.233677
1600	0.1	Medio	1.6	100	1.666667	2.583576	0.152172	0.109778
1600	0.2	Medio	1.6	200	3.333333	2.763256	0.304134	0.172738
1600	0.3	Bueno	1.6	300	5	2.874101	0.456016	0.225192
1600	0.4	Bueno	1.6	400	6.666667	2.955433	0.607847	0.271808
2000	0.1	Medio	2	100	1.666667	2.454324	0.115493	0.103198
2000	0.2	Medio	2	200	3.333333	2.625015	0.230826	0.162385
2000	0.3	Medio	2	300	5	2.730315	0.346098	0.211694
2000	0.4	Bueno	2	400	6.666667	2.807578	0.461332	0.255516
2500	0.1	Malo	2.5	100	1.666667	2.331539	0.087655	0.097012
2500	0.2	Malo	2.5	200	3.333333	2.493691	0.175188	0.152651
2500	0.3	Medio	2.5	300	5	2.593722	0.262675	0.199005
2500	0.4	Malo	2.5	400	6.666667	2.667119	0.350132	0.240201

APÉNDICE D

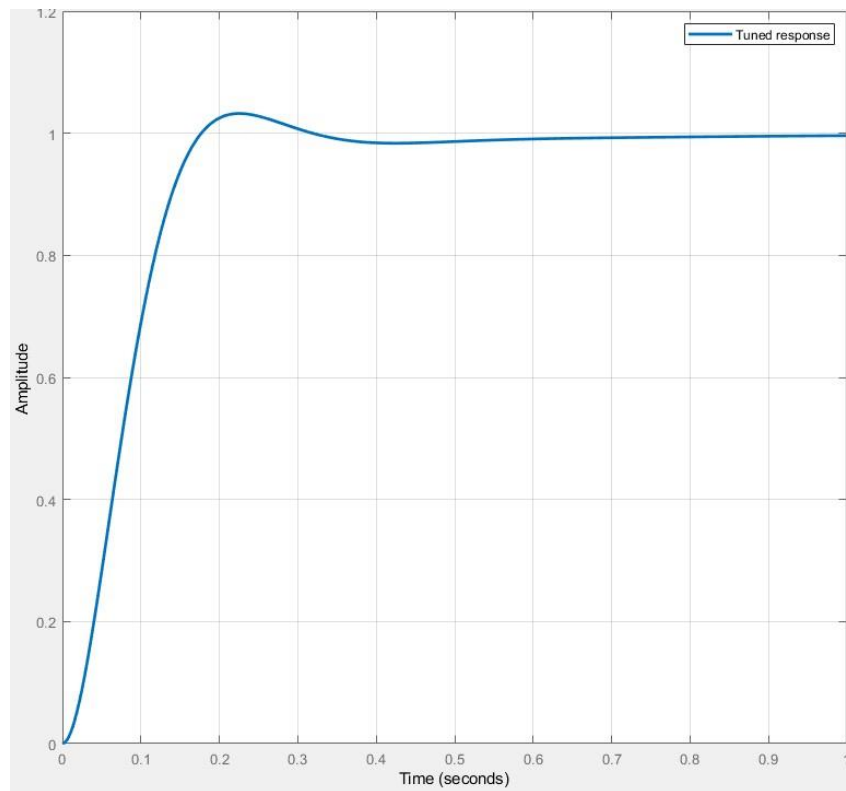


Figura D.1 Afinamiento del controlador PI de la fuerza axial en seguimiento de referencia

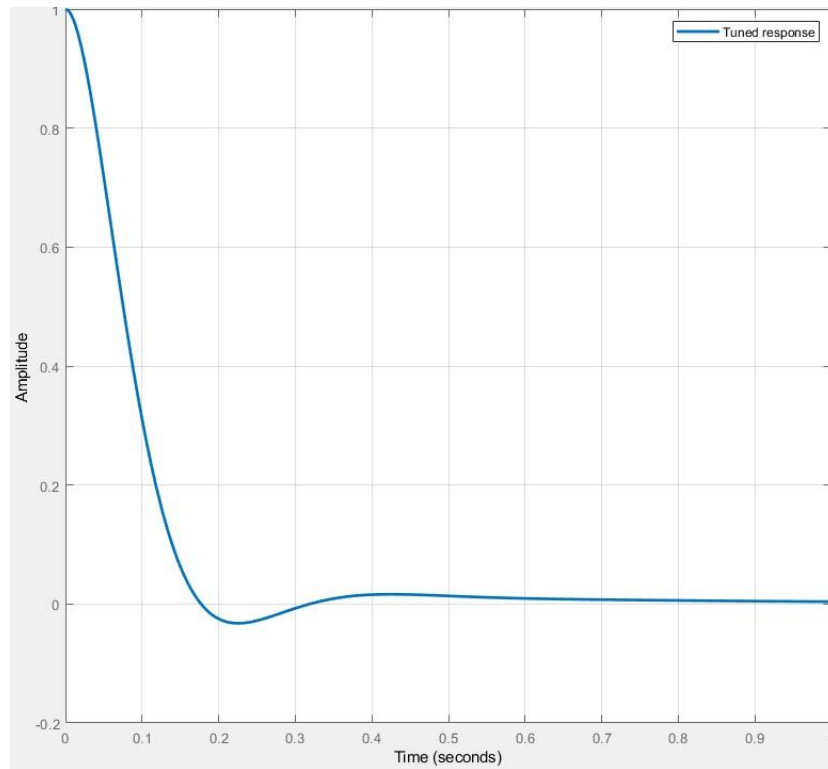


Figura D.2 Afinamiento del controlador PI de la fuerza axial en rechazo de perturbaciones

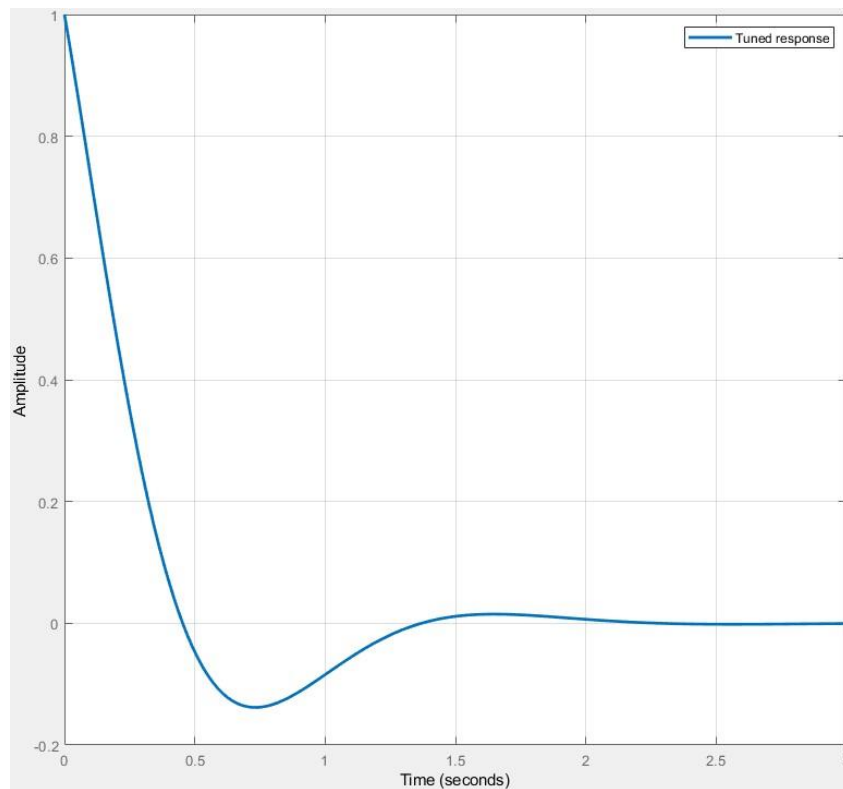


Figura D.4 Esquema del controlador PI con desacoplo implementado en Simulink

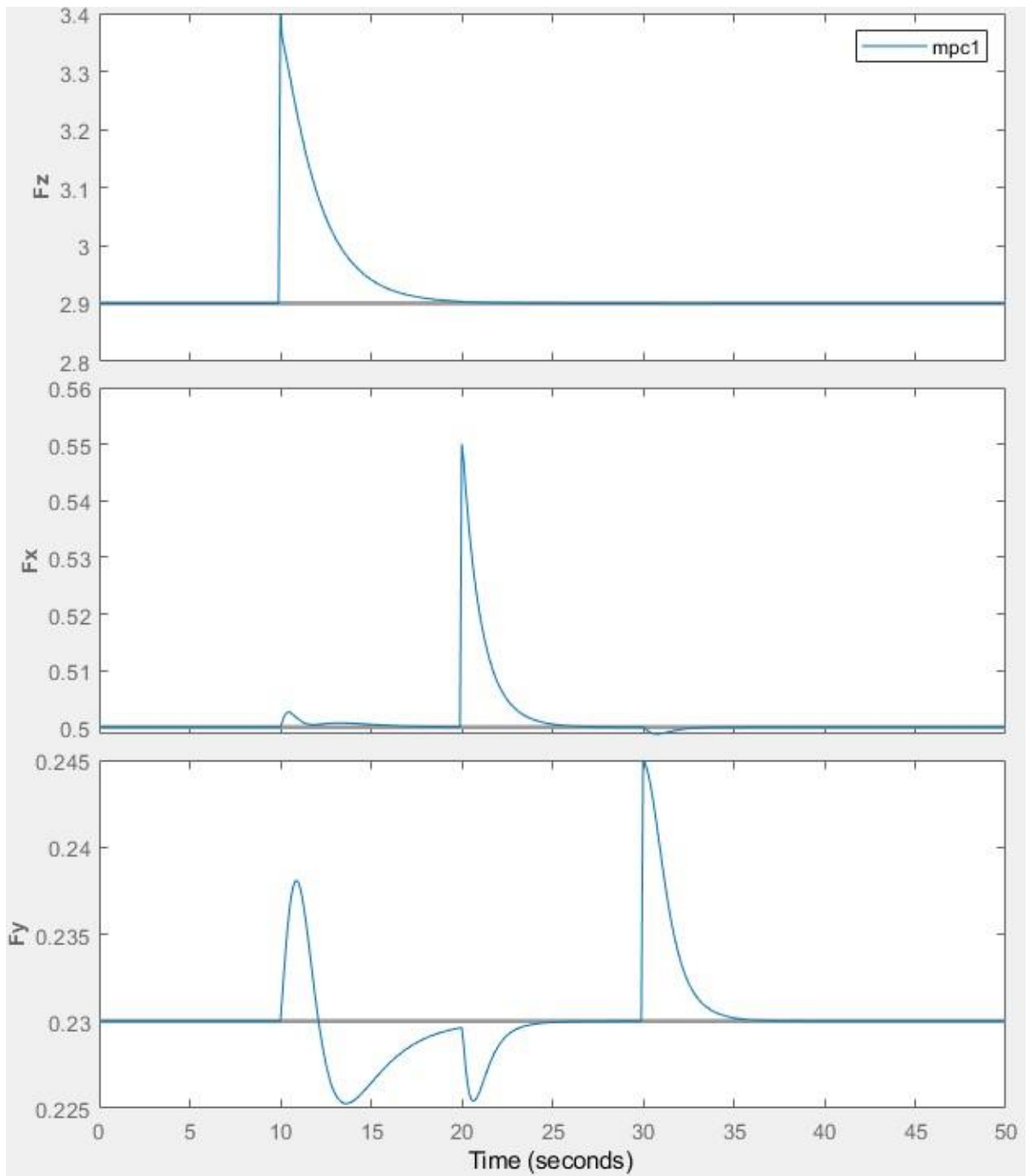


Figura D.5 Salidas del sistema en afinamiento del MPC para rechazar perturbaciones

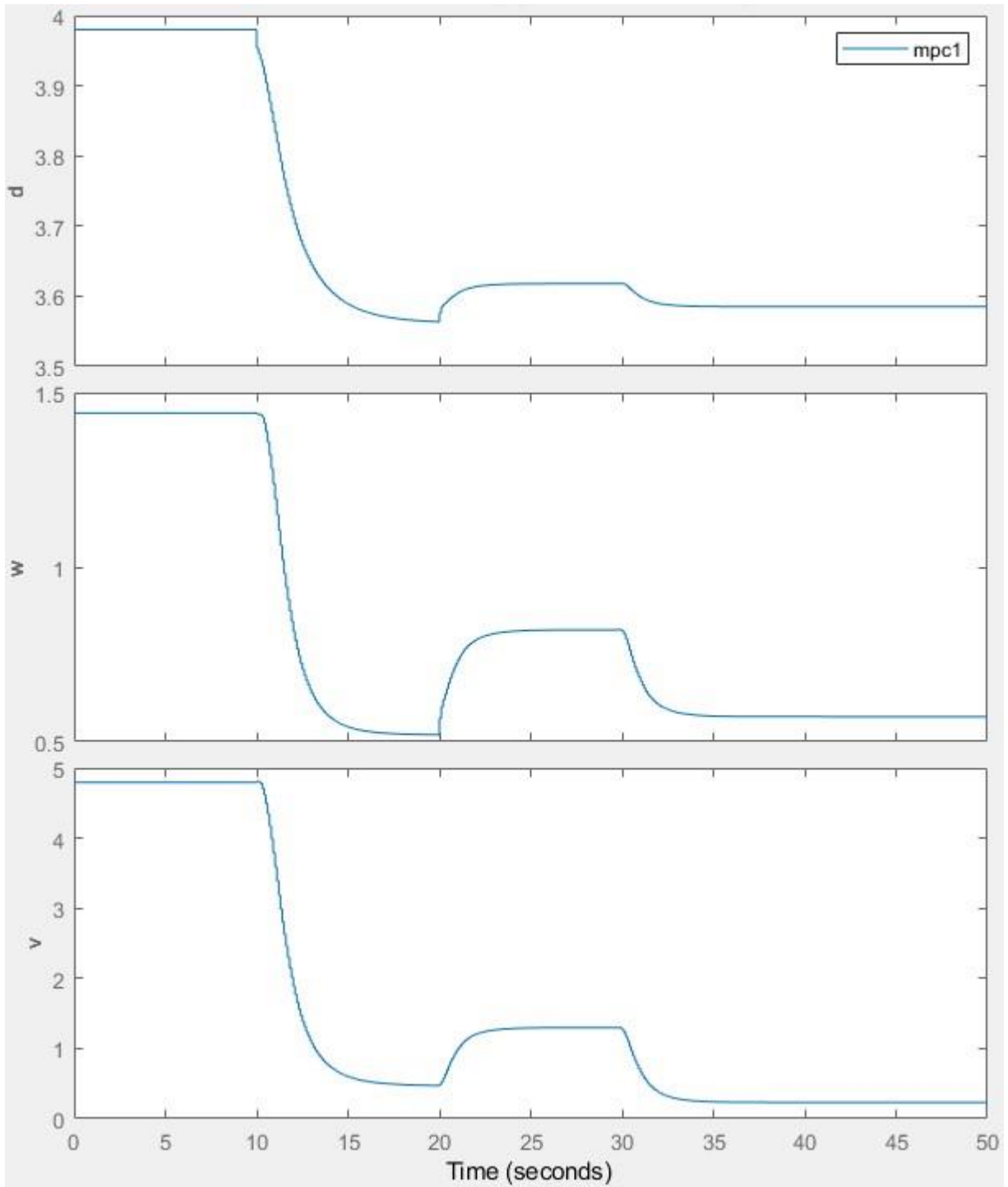


Figura D.6 Entradas del sistema en afinamiento del MPC para rechazar perturbaciones

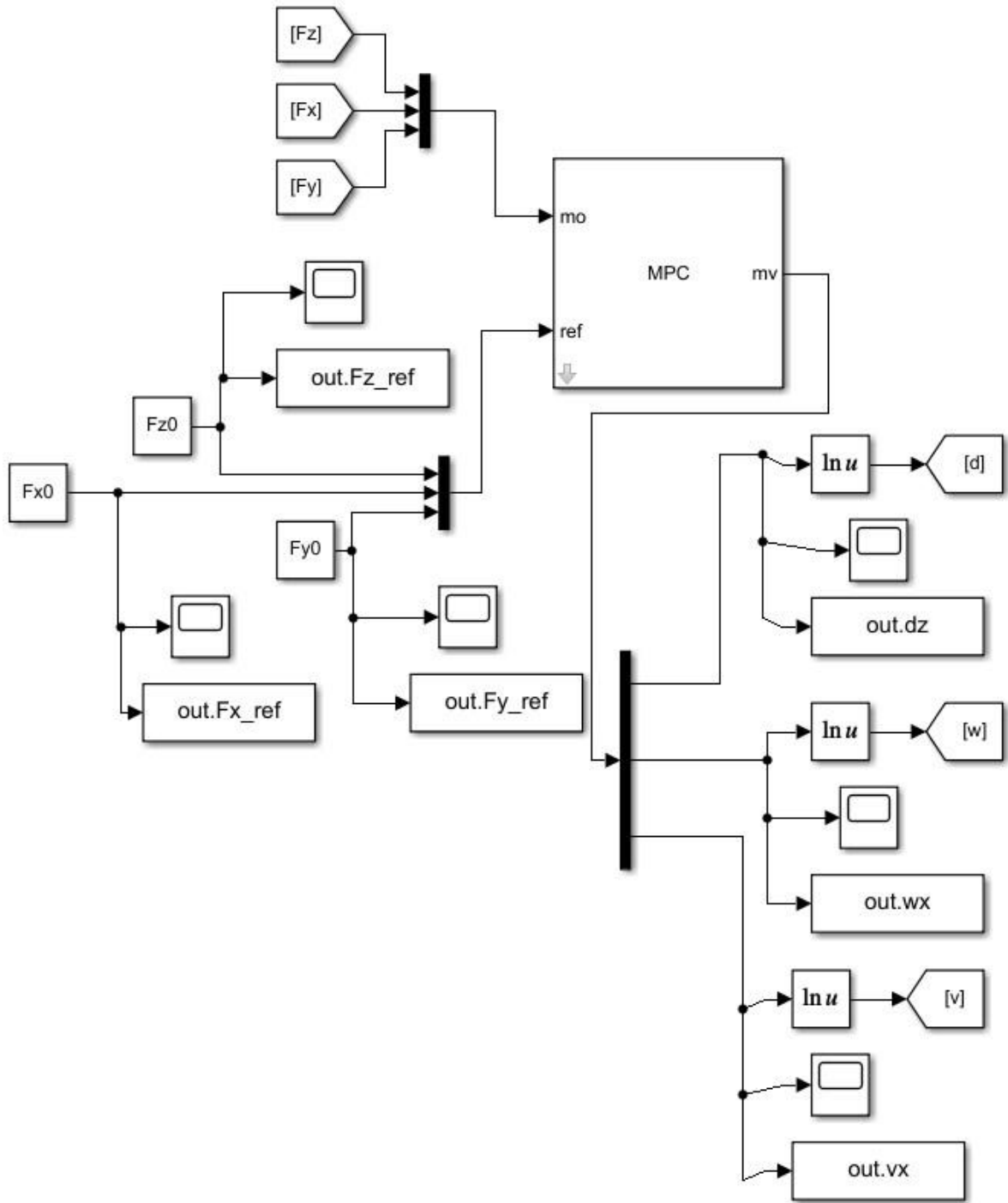


Figura D.7 Esquema del MPC implementado en Simulink

APÉNDICE E

```
Administrator: Anaconda Prompt
Epoch loss: 0.012691709340661763, Best loss: 0.011455595907699317, - 1
Epoch loss: 0.011696043142080307, Best loss: 0.011455595907699317, - 2
Epoch loss: 0.017344007152616976, Best loss: 0.011455595907699317, - 3
Epoch loss: 0.023853922843933107, Best loss: 0.011455595907699317, - 4
Epoch loss: 0.012906651558876037, Best loss: 0.011455595907699317, - 5
Epoch loss: 0.019960999639630318, Best loss: 0.011455595907699317, - 6
Epoch loss: 0.01284544633988291, Best loss: 0.011455595907699317, - 7
Epoch loss: 0.013298615684807301, Best loss: 0.011455595907699317, - 8
Epoch loss: 0.014421758825406432, Best loss: 0.011455595907699317, - 9
Epoch loss: 0.023207905131578447, Best loss: 0.011455595907699317, - 10
Epoch loss: 0.01486799272298813, Best loss: 0.011455595907699317, - 11
Epoch loss: 0.03380843288004398, Best loss: 0.011455595907699317, - 12
Epoch loss: 0.018699473497085273, Best loss: 0.011455595907699317, - 13
Epoch loss: 0.01136030944943428, Best loss: 0.01136030944943428, - 0
Epoch loss: 0.011437736542820931, Best loss: 0.01136030944943428, - 1
Epoch loss: 0.014992759013175965, Best loss: 0.01136030944943428, - 2
Epoch loss: 0.017378779895305632, Best loss: 0.01136030944943428, - 3
Epoch loss: 0.017994427455961703, Best loss: 0.01136030944943428, - 4
Epoch loss: 0.013373179644346237, Best loss: 0.01136030944943428, - 5
Epoch loss: 0.010860048635601997, Best loss: 0.010860048635601997, - 0
Epoch loss: 0.010288239004313946, Best loss: 0.010288239004313946, - 0
Epoch loss: 0.009588456066399813, Best loss: 0.009588456066399813, - 0
```

Figura E.1 Entrenamiento del controlador inteligente

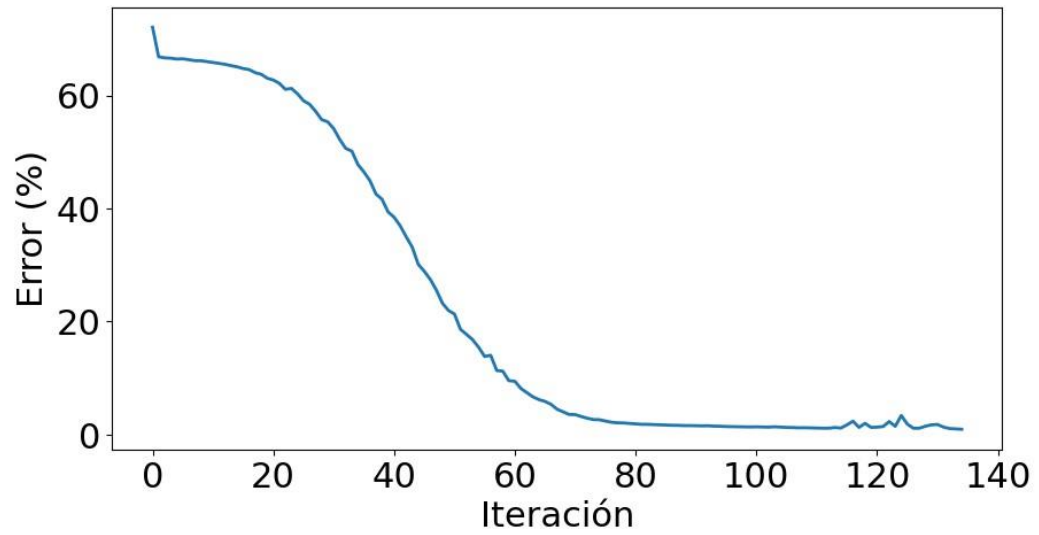


Figura E.2 Evolución del error del controlador en el proceso de entrenamiento

```
device='cuda:0')
tensor([-3.3496e-03, -1.0747e-02, -4.8915e-02,  2.8787e-02, -4.5512e-02,
        5.9325e-02, -9.8079e-03, -2.9461e-02,  2.9554e-02, -6.7003e-02,
        -3.6476e-02,  1.0504e-02,  4.7910e-02, -1.3975e-02, -3.3074e-02,
        5.5695e-02,  8.6569e-03, -4.5963e-02,  6.2512e-02,  6.3385e-02,
        -3.2591e-02, -6.0888e-02,  2.0533e-02, -1.0245e-02,  2.2099e-02,
        6.1354e-02,  6.7261e-02, -5.3302e-02, -3.3798e-02, -3.9221e-02,
        3.5490e-02, -3.4687e-02, -1.3765e-02,  3.3714e-02, -4.7199e-02,
        4.1396e-02,  1.6743e-02,  3.0085e-02,  2.5228e-02, -1.6158e-02,
        7.2705e-03,  1.7871e-02, -7.0055e-02,  1.2069e-02,  2.4724e-02,
        3.0884e-02,  3.2090e-02, -6.6462e-02, -4.7768e-02,  3.6561e-02,
        3.1661e-02,  6.2341e-02, -5.3188e-02, -6.9470e-02,  1.1537e-02,
        4.8083e-02,  4.8377e-02,  4.8453e-03, -2.9490e-02, -2.8955e-02,
        -3.3541e-02, -5.8802e-02, -6.1044e-02, -3.1220e-02, -1.9374e-03,
        -1.7888e-02,  3.9366e-02,  2.9562e-02, -6.9809e-02, -2.4572e-03,
        1.0469e-02,  1.9569e-02,  2.1311e-03, -3.0753e-03,  5.4505e-02,
        3.7976e-02,  4.1536e-02,  3.9118e-02, -1.2032e-02, -2.6302e-02,
        3.7785e-02,  2.6385e-02,  6.4612e-02, -1.9964e-03,  4.1252e-02,
        1.2459e-02,  2.2107e-02, -4.3645e-02, -2.0194e-02,  3.0872e-02,
        -4.8484e-02, -5.5478e-02,  5.5085e-02,  1.7852e-02, -5.3204e-02,
        4.3849e-02, -1.1553e-02, -5.6308e-02,  1.2042e-02,  5.9821e-02,
        -7.0277e-02, -3.7558e-02, -1.7738e-02, -4.7011e-02, -4.9158e-02,
        4.7192e-02,  2.9443e-02, -6.5240e-02,  2.4879e-02,  4.6082e-02,
        4.4847e-03,  6.6322e-02,  1.4677e-03,  3.2623e-02, -4.7044e-02,
        1.5942e-02, -5.6593e-02, -4.6366e-02, -5.6244e-02, -4.9730e-02,
        6.8906e-02,  1.9728e-02, -6.0683e-02,  1.8987e-02,  5.7174e-02,
        -1.6353e-02,  1.9826e-02, -3.1364e-02, -1.4495e-03, -5.1772e-02,
        -3.9030e-02,  2.7475e-02, -2.2150e-02, -5.9798e-02,  5.9356e-02,
        -3.8487e-02, -3.7051e-02, -4.2644e-02, -5.9550e-02,  6.5868e-02,
        -1.7148e-02, -1.8319e-03, -7.6090e-03, -1.5599e-02, -6.5357e-02,
        -9.9098e-03, -5.6109e-02,  1.9837e-02, -7.0357e-02,  3.9782e-02,
        -1.2236e-02, -5.0108e-02,  4.6938e-02,  3.7250e-03, -2.3349e-02,
        6.9683e-02,  3.0142e-02, -4.2880e-02, -3.5964e-02, -4.0919e-02,
        -2.3427e-02,  6.3483e-02, -2.9868e-02,  2.7697e-02, -2.6510e-02,
        1.7647e-02, -2.6897e-02,  4.9819e-02,  6.3605e-03,  1.8759e-02,
        -4.0316e-02,  3.6465e-02, -1.3077e-02,  4.4440e-03, -2.8276e-02,
        5.2884e-02, -4.3228e-02,  4.8692e-03, -5.3537e-02, -3.7556e-02,
        -5.4366e-03, -6.3546e-02,  3.4515e-02, -5.6824e-02,  3.3695e-02,
        -2.9815e-02,  3.3483e-02,  4.6904e-02, -6.6246e-02, -1.7223e-02,
        -6.8740e-03,  1.5745e-02,  5.6350e-02, -9.1805e-03,  1.6210e-02,
        3.8562e-02,  6.4834e-02, -5.6545e-02,  7.0141e-02,  4.9483e-02,
        -4.8027e-02, -1.7715e-02, -2.1273e-02, -1.0931e-02, -4.1466e-02,
        2.2482e-02, -7.5162e-03,  4.6604e-02, -6.6787e-02,  3.9604e-02,
```

Figura E.3 Muestra aleatoria de pesos del controlador inteligente entrenado