

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ingeniería Electrónica

TESIS DE MAESTRÍA EN CIENCIAS

**Neuro-Identificación de Sistemas Dinámicos No Lineales Afines con la
Entrada con Incertidumbre en la Retroalimentación**

presentada por
Ing. César Octavio Márquez Aguirre

como requisito para la obtención del grado de
Maestro en Ciencias en Ingeniería Electrónica

Director de tesis
Dr. Juan Reyes Reyes

Cuernavaca, Mor., 1° de junio de 2016.

OFICIO No. DIE/118/2016


Asunto: Aceptación de documento de tesis

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. César Octavio Márquez Aguirre** con número de control **M14CE077** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **“NEURO-IDENTIFICACIÓN DE SISTEMAS DINÁMICOS NO LINEALES AFINES CON LA ENTRADA CON INCERTIDUMBRE EN LA RETROALIMENTACIÓN”** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

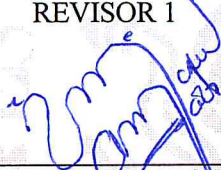
DIRECTOR DE TESIS

CODIRECTOR DE TESIS


Dr. Juan Reyes Reyes
Doctor en Ciencias en la Especialidad de
Control Automático
Cédula profesional 4214833

REVISOR 1

REVISOR 2


Dr. Víctor Manuel Alvarado Martínez
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 8031070


Dr. José Francisco Gómez Aguilar
Doctor en Física
Cédula profesional 9124781

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente.

CMAZ/lrr

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Mor., 1 de junio de 2016
OFICIO No. SAC/184/2016

Asunto: Autorización de impresión de tesis

ING. CÉSAR OCTAVIO MÁRQUEZ AGUIRRE
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“Neuro-Identificación de Sistemas Dinámicos no Lineales Afines con la Entrada con Incertidumbre en la Retroalimentación”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

“CONOCIMIENTO Y TECNOLOGIA AL SERVICIO DE MEXICO”

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

Dedicatoria

A Dios:

Por permitirme cumplir este logro en mi vida académica, por enseñarme a encarar las adversidades y nunca darme por vencido.

A Mis Padres:

A quienes debo todo en la vida, por brindarme su apoyo incondicional en este sueño que hoy se materializa. Por ser la motivación para ser mejor ser humano día a día y para superarme en cada aspecto de mi vida. Los amo.

A Mis Hermanas:

Por su apoyo incondicional, por nunca dejarme a mi suerte y estar ahí cuando las necesito. Las amo.

A Esa Persona Especial N.M.:

Por ser una de las personas más importantes en mi vida, Por cada momento que he pasado a tu lado, por ser una de mis principales fuentes de inspiración, por estar ahí siempre que necesite tu apoyo.

A Mis Amigos y Familiares:

Por su apoyo en esta meta que el día de hoy se cumple, por sus consejos y ayuda siempre que lo necesite.

Mil Gracias por todo. Esta tesis es dedicada a ustedes.

Agradecimientos

A Mis Padres y Mis Hermanas: Por darme todo en la vida, porque gracias a ellos y su apoyo incondicional hoy puedo materializar esta meta de mi vida, por ser la principal motivación para superarme día a día en todos los aspectos, por ser mi sustento siempre. Los amo.

A Esa Persona Especial N.M.: Por ser una de las principales motivaciones para ser mejor ser humano, por tu apoyo durante este tiempo, por escucharme cuando lo necesite, por todos y cada uno de los momentos a tu lado, muchas gracias.

Al Dr. Juan Reyes Reyes: Por su tiempo, consejos, enseñanzas, dedicación, paciencia y motivación para el desarrollo de la presente tesis, por su amistad muchas gracias.

Al Dr. Manuel Alejandro Jiménez Lizárraga: Por el apoyo para la realización de la estancia en la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León, de la cual se desprende una parte importante de la presente tesis.

Al Dr. Víctor Manuel Alvarado Martínez y al Dr. José Francisco Gómez Aguilar: Por sus acertados comentarios, observaciones, disponibilidad y ayuda para la culminación de la presente tesis.

A Todo el personal del Área de Control Automático del CENIDET: A los Profesores que me impartieron clases, por sus enseñanzas y compartir sus conocimientos: Dr. Juan Reyes Reyes, Dr. Carlos Manuel Astorga Zaragoza, Dr. Manuel Adam Medina, Dr. Ricardo Fabricio Escobar Jiménez, Dr. Gerardo Vicente Guerrero Ramírez, Dr. José Francisco Gómez Aguilar, Dr. Víctor Manuel Alvarado Martínez, Dr. Luis Gerardo Vela Valdés, Dr. Enrique Quintero-Mármol Márquez . A Lorena Ruiz por siempre estar al pendiente, su ayuda y apoyo durante toda mi maestría.

A Mis Compañeros y Amigos: Hugo Arellano, Jesús Reyes, Adrián Velázquez, Oscar Gutiérrez, Saúl Beltrán, Ivonne Guerrero, Patricia Vázquez, Arturo Robles, Diego Carbot, José Manuel Martínez, Moisés Ramos, Juan Joachin, Jashiel Pérez, Didhier Montiel, Susana Sainz, Montserrat Alegría, Alberto Flores, Gerardo Ortiz y todos los faltantes pero no menos importantes, por todos y cada uno de los momentos vividos a lo largo de esta etapa en CENIDET.

Al Tecnológico Nacional de México y al Centro Nacional de Investigación y Desarrollo Tecnológico: Por Proporcionarme los medios necesarios para desarrollarme de manera correcta y ampliar mis conocimientos para mi crecimiento profesional.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT): Por el apoyo económico brindado durante mis estudios de maestría.

Resumen

Esta tesis de maestría presenta en primer lugar el diseño de una metodología de identificación en línea mediante redes neuronales dinámicas cuyo error de identificación sea estable para un sistema dinámico no lineal afín con la entrada cuya incertidumbre es parcial en el término no lineal de retroalimentación, dicha metodología es formalizada por medio del planteamiento de un teorema.

El teorema planteado tiene su base en el análisis estilo Lyapunov desarrollado. Dicho análisis también permite conocer la estructura del umbral neuronal y del algoritmo de aprendizaje de la red neuronal.

La efectividad de esta metodología se muestra en el desarrollo de 4 simulaciones de sistemas dinámicos no lineales, los cuales son: un eslabón simple, un sistema de dos tanques de área transversal no uniforme conectados en cascada, un sistema biológico depredador-presa y un péndulo invertido.

Además gracias a la estancia realizada en la Facultad de Ciencias Físico-Matemáticas de la Universidad Autónoma de Nuevo León, se presentan los resultados de la aplicación de la metodología propuesta en un sistema real el cual es un servomotor modular de la empresa INTECO.

Finalmente se propone una metodología para aproximar al sistema no lineal afín con la entrada dentro de una región de operación, mediante la utilización de los datos arrojados por 3 de las simulaciones elaboradas y la aplicación de una red neuronal estática para proponer un modelo neuronal que aproxime a dichos sistemas dinámicos dentro de una región de operación.

Abstract

This thesis firstly presents the design of online identification methodology through dynamic neural networks whose identification error is stable for a nonlinear dynamic system affine to the input whose uncertainty is partial in the nonlinear feedback term, this methodology is formalized through the approach of a theorem.

The raised theorem is based on the Lyapunov analysis developed. This analysis also reveals the structure of neuronal threshold and learning algorithm of the neural network.

The effectiveness of this methodology is shown in the development of four simulations of nonlinear dynamics systems, which are: A simple link, a two tanks of nonuniform transversal area connected in cascade, a biologic system predator-prey and inverted pendulum.

Besides, thanks to the research stay at the Facultad de Ciencias Físico-Matemáticas at the Universidad Autonoma de Nuevo Leon, the results of the implementation of the proposed methodology in a real system, which is a modular servomotor company INTECO are presented.

Finally a methodology is proposed to approximate the nonlinear system affine to the input into a region of operation, by utilizing the data obtained from 3 of the simulations and the application of a static neural network to propose a neural model that approximates such dynamic systems within a region of operation.

Nomenclatura

x	Vector de variables de estado.
$\dot{x} = \frac{dx}{dt}$	Derivada del vector de variables de estado con respecto al tiempo.
$f(x)$	Función no lineal de retroalimentación.
$g(x)$	Función no lineal asociada a la entrada.
u	Entrada del sistema no lineal.
$h(x)$	Función no lineal de salida.
$f_0(x)$	Parte conocida de la función $f(x)$.
$\Delta_f(x)$	Incertidumbre de la función $f(x)$.
φ, σ	Funciones de activación de una red neuronal.
Ω	Conjunto de datos en un sistema de coordenadas bidimensionales, tridimensionales o p-dimensionales.
A^{-1}	Inversa de una matriz.
A^+	Pseudoinversa de una matriz.
A^T	Transpuesta de una matriz.
$tr\{A\}$	Traza de una matriz.
$\ (\cdot)\ _p$	Norma p de (\cdot) .
$\ (\cdot)\ _\infty$	Norma ∞ de (\cdot) .
$\ (\cdot)\ _2 = \ (\cdot)\ $	Norma 2 o norma euclidiana de (\cdot) .
$sign(\bullet)$	Función signo de \bullet .
$V(x)$	Función candidata de Lyapunov.
$\delta_f(x)$	Cota superior de la norma euclidiana de la incertidumbre.
W	Matriz de pesos sinápticos de la red neuronal.
\hat{x}	Vector de variables de estado estimadas por la red neuronal.
ψ	Umbral Neuronal.
L	Matriz de ganancias del factor de corrección.
\dot{W}	Algoritmo de aprendizaje de la red neuronal.
e	Error de estimación.
$W\varphi(\hat{x}) + \psi$	Función de aproximación de la incertidumbre.

Ω_k	Conjunto de datos formado en el instante k.
\bar{x}_k	Muestra de interés en el instante k.
\bar{W}_k	Matriz de pesos sinápticos de la aproximación neuronal en el instante k.
\hat{x}_k	Valor estimado de interés en el instante k.
Ω_0	Valor inicial del conjunto de datos Ω_k
χ	Elemento del k-ésimo Conjunto Ω
${}^N w_i$	i-ésimo peso sinápticos de la capa N de la red neuronal.
σ_i	i-ésima función de activación de la red neuronal estática.
${}^N \theta_i$	i-ésimo umbral de la capa N de la red neuronal estática.
Φ	Matriz generada por las funciones de activación y los parámetros de diseño de la red neuronal estática.
${}^H W^*$	Matriz de pesos sinápticos de la segunda capa de la red neuronal estática.

Índice General

	Pág.
Nomenclatura	XV
I. Introducción	1
1.1. Estado del Arte	3
1.1.1. Los Sistemas No Lineales Con Incertidumbre Abordados con un Enfoque de Redes Neuronales	3
1.1.2. Identificación de Sistemas no Lineales	5
1.1.2.1. Identificación de sistemas no lineales en general	5
1.1.2.2. Identificación en Línea	6
1.1.2.3. Identificación Fuera de Línea	8
1.1.2.4. Neuro-Identificación	9
1.1.3. Ejemplos de Aplicación de Redes Neuronales Dinámicas	10
1.2. Planteamiento del Problema	12
1.3. Motivación	12
1.4. Objetivo y Metas	13
1.4.1. Objetivo	13
1.4.2. Metas	13
II. Marco Teórico	15
2.1. Las Redes Neuronales Artificiales	15
2.1.1. Neurona Artificial	15
2.1.2. Arquitectura de las Redes Neuronales Artificiales	15
2.1.2.1. Modelado Estático de Redes Neuronales	16
2.1.2.2. Modelado Dinámico de Redes Neuronales	17
2.2. Aproximación de Funciones	17
2.3. Conjunto Finito de datos o Nube de Datos	18
2.4. Aproximación de Funciones mediante Redes Neuronales	19
2.5. Teorema de Cybenko	19
2.6. Modelo de red neuronal de dos capas para aproximación de funciones	20

	Pág.
2.7. Pseudoinversa de una Matriz	20
2.8. Algunas Propiedades de las Matrices	21
2.9. Normas y Cotas Superiores de Productos Vectoriales y Matrices	21
2.10. Estabilidad por Criterio de Lyapunov	23
2.10.1. Principio del Teorema de Estabilidad de Lyapunov	23
2.10.1.1. Función Candidata de Lyapunov	23
2.10.1.2. Teorema de Estabilidad de Lyapunov	23
III. Resultados Principales	25
3.1. Aproximación Neuronal Dinámica de un sistema no lineal afín con la entrada con incertidumbre en la retroalimentación	25
3.2. Modelo Neuronal Dinámico para aproximar un Sistema No Lineal con Incertidumbre dentro de una Región de Operación	29
3.2.1. Propuesta para construir el conjunto de datos para la aproximación de la incertidumbre	30
IV. Simulaciones	33
4.1. Simulación 1: Identificación e Interpretación de la Fricción en un Manipulador de Eslabón Simple	33
4.1.1. Planteamiento del Problema	33
4.1.2. Objetivos de la Simulación	34
4.1.3. Desarrollo de la Simulación	34
4.2. Simulación 2: Estimación de Una Fuga en un par de Tanques de Área Transversal no uniforme conectados en Cascada	46
4.2.1. Planteamiento del Problema	46
4.2.2. Objetivos de la Simulación	47
4.2.3. Desarrollo de la Simulación	47
4.3. Simulación 3: Estimación de la competencia intraespecífica en la especie de la presa dentro de un sistema depredador-presa	62
4.3.1. Planteamiento del Problema	62
4.3.2. Objetivos de la Simulación	63
4.3.3. Desarrollo de la Simulación	63
4.4. Simulación 4: Identificación de Incertidumbre en un Péndulo Invertido	71
4.4.1. Planteamiento del Problema	71
4.4.2. Objetivos de la Simulación	72
4.4.3. Desarrollo de la Simulación	72
4.5. Implementación del Algoritmo de identificación en el Sistema Servo Modular	90

	Pág.
4.6. Obtención del Modelo Neuronal Dinámico para aproximar al sistema no lineal con incertidumbre en una región de operación.	125
4.6.1. Obtención del Modelo con los datos de la Simulación 1, correspondiente al sistema no lineal del Eslabón Simple con cambio de fricción	125
4.6.2. Obtención del Modelo con los datos de la Simulación 2, correspondiente al sistema no lineal de tanques con una fuga	131
4.6.3. Obtención del Modelo con los datos de la Simulación 3, correspondiente al sistema no lineal de depredador-presa	134
V. Conclusiones y trabajos futuros	139
5.1. Conclusiones	139
5.2. Trabajos Futuros	140
Bibliografía	141
Anexos	150
A. Manipulación Algebraica para mostrar que el Error es Acotado	151
B. Convergencia de los Pesos Sinápticos de la Simulación 3	155
C. Desarrollo del Sistema de Péndulo Invertido	157

Índice de Figuras

	Pág.
2.1. Neurona Artificial Estática Modelo McCullock Pitts.	16
2.2. Estructura de una Red Neuronal Dinámica.	17
2.3. Nube de datos bidimensional Ω y función de aproximación \hat{y}	18
2.4. Red Neuronal de 2 Capas.	20
3.1. Estructura propuesta de la red neuronal propuesta.	26
3.2. Red Neuronal Estática Propuesta.	31
4.1. Sistema de un eslabón con cambio de fricción.	33
4.2. Posición Angular del Sistema.	36
4.3. Velocidad Angular.	37
4.4. Posición Angular Estimada.	37
4.5. Velocidad Angular Estimada.	38
4.6. Comparativa entre la Posición Angular del Sistema y la Posición Angular Estimada.	38
4.7. Error de Estimación de la Posición Angular.	39
4.8. Comparativa entre la Velocidad Angular del Sistema y Velocidad Angular Estimada.	39
4.9. Diferencia entre la velocidad angular y la velocidad angular estimada.	40
4.10. Incertidumbre del Eslabón Simple.	41
4.11. Componente Neuronal $W\varphi(\hat{x})$	41
4.12. Comparación de Δ_f y $W\varphi(\hat{x})$	42
4.13. Error de Estimación de la Incertidumbre del Sistema.	42
4.14. Error Cuadrático Medio a lo largo del tiempo de simulación.	43
4.15. Pesos Sinápticos de la Red Neuronal Dinámica.	43
4.16. Perfil de $l \cos(\theta)$	44
4.17. Fricción Estimada.	44
4.18. Localización de \hat{x}_0	45
4.19. Distancia Estimada.	45
4.20. Esquema general del Sistema de tanques de área transversal no uniforme.	46
4.21. Perfil de Flujo de Entrada.	49
4.22. Niveles de los dos tanques.	49

	Pág.
4.23. Niveles Estimados de los dos tanques.	50
4.24. Comparativa entre Nivel y Nivel Estimado del tanque 1.	50
4.25. Error de Estimación del Nivel del tanque 1.	51
4.26. Comparativa entre Nivel y Nivel Estimado del tanque 2.	51
4.27. Error de Estimación del Nivel del tanque 2.	52
4.28. Flujo de Salida del tanque 2.	52
4.29. Flujo de Salida Estimado del tanque 2.	53
4.30. Comparativa de Flujos de Salida Real y Estimado del tanque 2.	53
4.31. Error de Estimación del Flujo de Salida del tanque 2.	54
4.32. Incertidumbre del Sistema de Tanques.	55
4.33. Componente Neuronal $W\varphi(\hat{x})$	55
4.34. Comparación de Δ_f y $W\varphi(\hat{x})$	56
4.35. Error de Estimación la Incertidumbre del Sistema.	56
4.36. Error Cuadrático Medio a lo largo del tiempo de simulación.	57
4.37. Pesos Sinápticos de la Red Neuronal Dinámica.	57
4.38. Flujo de la fuga del tanque 2.	58
4.39. Flujo de la fuga Estimado del tanque 2.	58
4.40. Comparativa de Flujos de Fuga Real y Estimado del tanque 2.	59
4.41. Error de Estimación del Flujo de Fuga del tanque 2.	59
4.42. Flujo de Salida Total del tanque 2.	60
4.43. Flujo de Salida Total Estimado del tanque 2.	60
4.44. Comparativa de Flujos de Salida Total Real y Estimado del tanque 2.	61
4.45. Error de Estimación del Flujo de Salida Total del tanque 2.	61
4.46. Población de las dos especies.	64
4.47. Población Estimada de las dos especies.	65
4.48. Comparativa entre Población y Población Estimada de la especie de la presa. . .	65
4.49. Error de Estimación de la población de la especie de la presa.	66
4.50. Comparativa entre la población y la población estimada en la especie del depredador. .	66
4.51. Error de Estimación de la población en la especie del depredador.	67
4.52. Incertidumbre del Sistema.	68
4.53. Componente Neuronal $W\varphi(\hat{N})$	68
4.54. Comparación de Δ_f y $W\varphi(\hat{N})$	69
4.55. Error de Estimación la Incertidumbre del Sistema.	69
4.56. Error Cuadrático Medio a lo largo del tiempo de simulación.	70
4.57. Pesos Sinápticos de la Red Neuronal Dinámica.	70
4.58. Sistema de péndulo invertido.	71

4.59.	Sistema de péndulo invertido de Matlab/Simulink.	72
4.60.	Señal de Control.	74
4.61.	Posición Angular del Péndulo.	75
4.62.	Posición Angular Estimada del Péndulo.	75
4.63.	Comparativa de la Posición del Péndulo real y estimada.	76
4.64.	Error de Estimación de Posición Angular del Péndulo.	76
4.65.	Velocidad Angular del Péndulo.	77
4.66.	Velocidad Angular Estimada del Péndulo.	77
4.67.	Comparativa de la Velocidad del Péndulo real y estimada.	78
4.68.	Error de Estimación de Velocidad Angular del Péndulo.	78
4.69.	Posición del Carro.	79
4.70.	Posición Estimada del Carro.	79
4.71.	Comparativa de la Posición del Carro real y estimada.	80
4.72.	Error de Estimación de Posición del Carro.	80
4.73.	Velocidad del Carro.	81
4.74.	Velocidad Estimada del Carro.	81
4.75.	Comparativa de la Velocidad del Carro real y estimada.	82
4.76.	Error de Estimación de Velocidad del Carro.	82
4.77.	Incertidumbre del Péndulo Invertido.	83
4.78.	Componente Neuronal $W\varphi(\hat{x})$	83
4.79.	Comparación de Δ_f y $W\varphi(\hat{x})$	84
4.80.	Comparación de Δ_{f_1} y $W\varphi(\hat{x})_1$	84
4.81.	Comparación de Δ_{f_2} y $W\varphi(\hat{x})_2$	85
4.82.	Comparación de Δ_{f_3} y $W\varphi(\hat{x})_3$	85
4.83.	Comparación de Δ_{f_4} y $W\varphi(\hat{x})_4$	86
4.84.	Error de estimación de la Incertidumbre en el Péndulo Invertido.	86
4.85.	Error de estimación de la Incertidumbre $e_{\Delta_{f_1}}$ en el Péndulo Invertido.	87
4.86.	Error de estimación de la Incertidumbre $e_{\Delta_{f_2}}$ en el Péndulo Invertido.	87
4.87.	Error de estimación de la Incertidumbre $e_{\Delta_{f_3}}$ en el Péndulo Invertido.	88
4.88.	Error de estimación de la Incertidumbre $e_{\Delta_{f_4}}$ en el Péndulo Invertido.	88
4.89.	Error Cuadrático Medio a lo largo del tiempo de simulación.	89
4.90.	Pesos Sinápticos de la Red Neuronal Dinámica.	89
4.91.	Fotografía del Sistema Servo Modular.	90
4.92.	Posición Angular en Experimento 1.	92
4.93.	Posición Angular en Experimento 2.	92
4.94.	Posición Angular en Experimento 3.	93
4.95.	Posición Angular en Experimento 4.	93
4.96.	Posición Angular en Experimento 5.	94

	Pág.
4.97. Velocidad Angular en Experimento 1.	94
4.98. Velocidad Angular en Experimento 2.	95
4.99. Velocidad Angular en Experimento 3.	95
4.100. Velocidad Angular en Experimento 4.	96
4.101. Velocidad Angular en Experimento 5.	96
4.102. Posición Angular Estimada en Experimento 1.	97
4.103. Posición Angular Estimada en Experimento 2.	97
4.104. Posición Angular Estimada en Experimento 3.	98
4.105. Posición Angular Estimada en Experimento 4.	98
4.106. Posición Angular Estimada en Experimento 5.	99
4.107. Velocidad Angular Estimada en Experimento 1.	99
4.108. Velocidad Angular Estimada en Experimento 2.	100
4.109. Velocidad Angular Estimada en Experimento 3.	100
4.110. Velocidad Angular Estimada en Experimento 4.	101
4.111. Velocidad Angular Estimada en Experimento 5.	101
4.112. Comparación de Posición en el Experimento 1.	102
4.113. Comparación de Posición en el Experimento 2.	102
4.114. Comparación de Posición en el Experimento 3.	103
4.115. Comparación de Posición en el Experimento 4.	103
4.116. Comparación de Posición en el Experimento 5.	104
4.117. Error de Estimación de posición angular en el Experimento 1	104
4.118. Error de Estimación de posición angular en el Experimento 2	105
4.119. Error de Estimación de posición angular en el Experimento 3	105
4.120. Error de Estimación de posición angular en el Experimento 4	106
4.121. Error de Estimación de posición angular en el Experimento 5	106
4.122. Comparación de Velocidad en el Experimento 1.	107
4.123. Comparación de Velocidad en el Experimento 2.	107
4.124. Comparación de Velocidad en el Experimento 3.	108
4.125. Comparación de Velocidad en el Experimento 4.	108
4.126. Comparación de Velocidad en el Experimento 5.	109
4.127. Error de Estimación de velocidad angular en el Experimento 1	109
4.128. Error de Estimación de velocidad angular en el Experimento 2	110
4.129. Error de Estimación de velocidad angular en el Experimento 3	110
4.130. Error de Estimación de velocidad angular en el Experimento 4	111
4.131. Error de Estimación de velocidad angular en el Experimento 5	111
4.132. Incertidumbre del Sistema Servo Modular en el Experimento 1.	112

	Pág.
4.133. Incertidumbre del Sistema Servo Modular en el Experimento 2.	112
4.134. Incertidumbre del Sistema Servo Modular en el Experimento 3.	113
4.135. Incertidumbre del Sistema Servo Modular en el Experimento 4.	113
4.136. Incertidumbre del Sistema Servo Modular en el Experimento 5.	114
4.137. Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 1.	114
4.138. Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 2.	115
4.139. Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 3.	115
4.140. Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 4.	116
4.141. Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 5.	116
4.142. Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 1.	117
4.143. Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 2.	117
4.144. Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 3.	118
4.145. Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 4.	118
4.146. Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 5.	119
4.147. Error de Estimación de la Incertidumbre del Sistema en el Experimento 1	119
4.148. Error de Estimación de la Incertidumbre del Sistema en el Experimento 2. . . .	120
4.149. Error de Estimación de la Incertidumbre del Sistema en el Experimento 3. . . .	120
4.150. Error de Estimación de la Incertidumbre del Sistema en el Experimento 4. . . .	121
4.151. Error de Estimación de la Incertidumbre del Sistema en el Experimento 5. . . .	121
4.152. Error Cuadrático Medio en el experimento 1.	122
4.153. Error Cuadrático Medio en el experimento 2.	122
4.154. Error Cuadrático Medio en el experimento 3.	123
4.155. Error Cuadrático Medio en el experimento 4.	123
4.156. Error Cuadrático Medio en el experimento 5.	124
4.157. Trayectoria de la incertidumbre a lo largo de $0 \leq t \leq 4$	125
4.158. Red Neuronal Estática para la Simulación 1.	126
4.159. Nube de Datos del Eslabón fuera del líquido.	127
4.160. Nube de Datos del Eslabón fuera del líquido.	127
4.161. Aproximación de la Nube de Datos del Eslabón fuera del líquido.	128
4.162. Nube de Datos del Eslabón dentro del líquido.	129
4.163. Nube de Datos del Eslabón dentro del líquido.	129
4.164. Aproximación de la Nube de Datos del Eslabón dentro del líquido.	130
4.165. Nube de Datos del Sistema de Tanques.	131
4.166. Nube de Datos de la Simulación 2.	131
4.167. Red Neuronal Estática para la Simulación 2.	132
4.168. Aproximación de la Nube de Datos de la Simulación 2.	133

4.169. Nube de Datos del Sistema de Depredador-Presa.	134
4.170. Nube de Datos de la Simulación 3.	134
4.171. Red Neuronal Estática para la Simulación 3.	135
4.172. Aproximación de la Nube de Datos de la Simulación 3.	136
B.1. Pesos Sinápticos de la Red Neuronal Dinámica de la Simulación 3.	155
C.1. Sistema de Péndulo Invertido.	157

Índice de Tablas

	Pág.
4.1. Datos usados en la Simulación 1.	36
4.2. Datos usados en la Simulación 2.	48
4.3. Datos usados en la Simulación 3.	64
4.4. Datos usados en la Simulación 4.	74
4.5. Datos usados en la aproximación de la simulación 1.	126
4.6. Datos usados en la aproximación de la simulación 2.	132
4.7. Datos usados en la aproximación de la simulación 3.	135

I. Introducción

Un sistema no lineal expresado en su manera más general posee la siguiente estructura:

$$\begin{aligned}\dot{x} &= F(x, u, t), \\ y &= h(x),\end{aligned}\tag{1.1}$$

donde F es una función no lineal que depende del estado x , la entrada u y el tiempo t , y la salida y es una función no lineal que tiene como argumento al estado x . Sin embargo, una amplia gama de sistemas no lineales pueden expresar la parte $F(x, u, t)$ de la ecuación dinámica, como la suma de un término $f(x)$ que expresa la función no lineal de retroalimentación, más otro término no lineal $g(x)$ que multiplica a la entrada u [1]. Esta clase de sistema no lineal se le conoce como *sistema no lineal afín con la entrada*. La estructura de esta clase de sistemas es:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u, \\ y &= h(x).\end{aligned}\tag{1.2}$$

Las imprecisiones en los modelos, los parámetros o variables, inevitables en cualquier aplicación práctica, pueden tener efectos adversos en los sistemas de control. De hecho, no sólo pueden degradar su comportamiento y eficacia, sino que también pueden conducir a la inestabilidad del sistema.

El conocimiento total del modelo de una planta en general es muy difícil tenerlo, por lo que siempre existe algún grado de incertidumbre.

La *incertidumbre* surge por errores en el modelado, tanto en los parámetros como en las funciones no lineales que aparecen en las ecuaciones diferenciales.

El modelo de la incertidumbre se puede representar como la suma de una parte conocida $f_0(x)$ más una parte desconocida representada por $\Delta_f(x)$

$$f(x) = \underbrace{f_0(x)}_{\text{Parte Conocida}} + \underbrace{\Delta_f(x)}_{\text{Parte Desconocida}}.\tag{1.3}$$

Si se construye un modelo de la planta antes de que suceda algún deterioro a lo largo del tiempo, se tendría un problema en el que no se conocen completamente los parámetros del modelo, a esta falta de conocimiento sobre el modelo de la planta se le llama *incertidumbre paramétrica* [2].

Un ejemplo: Cuando diseñamos un circuito eléctrico o electrónico lo hacemos según un valor nominal de resistencia. Ahora bien, éstas presentan un valor real que oscila dentro de unas tolerancias que marcan la incertidumbre en el componente lineal.

A su vez, es posible que en el modelo que se está construyendo no se hayan tomado en cuenta por alguna razón, ciertas dinámicas de la planta, en este caso la estructura real de la planta tendrá diferencia con respecto a la estructura del modelo matemático, es decir, se refiere a los suposiciones realizadas por los investigadores en la elección y diseño del modelo, a esta falta de información del modelo de la planta se le conoce como *incertidumbre estructural* [2].

Existen ciertos efectos de la incertidumbre que deben tenerse en cuenta, los cuales son:

- Sobre el Modelo: No se reciben los datos esperados o se reciben con un error mayor al esperado.
- Sobre el Controlador: Se produce al retroalimentar, es decir, el controlador del sistema recibe datos con incertidumbre, por lo que el principal efecto es mayor esfuerzo de control, es decir, mayor energía; esto es porque el controlador está diseñado a partir de la respuesta de la señal controlada y no por manipulación del modelo. Lo anterior puede ejemplificarse de manera general como un profesor de biología, al cual invitan a realizar una plática de un tema en particular de biología a dos distintos grupos de personas, el primer grupo son estudiantes con conocimientos en biología, mientras que el segundo grupo solo sabe que no tienen ningún conocimiento de biología, no conoce nada más, en el primer caso para el orador supone un menor esfuerzo dar la plática al grupo de estudiantes, mientras que para el segundo grupo le supone un esfuerzo mayor porque tiene que sondear primero, para poder saber cómo dar la plática. De igual manera para el controlador supone un menor esfuerzo al conocer el modelo o al menos una parte, es decir, sin tener incertidumbre, por el contrario al tener incertidumbre supone un esfuerzo mayor para poder controlar el sistema.

La evolución del control se ha visto motivada por tres aspectos principales: la necesidad de tratar con sistemas complejos; la necesidad de cumplir con demanda de nuevos modelos de control y la necesidad de obtener esos modelos con el menor conocimiento de la planta y el medio ambiente, es decir la necesidad de controlar bajo incertidumbre.

El uso de las *Redes Neuronales Dinámicas (RNDs)* o también llamadas recurrentes han mostrado que poseen mejor desempeño en la identificación y en aproximación de funciones. A su vez, una variedad de arquitecturas de redes neuronales han sido propuestas para el control de sistemas no lineales con incertidumbre [3–6].

Una de las principales ventajas de las redes neuronales en el diseño de control no lineal es precisamente la capacidad de construir mapeos no lineales complejos.

1.1. Estado del Arte

Luego de la revisión de la literatura, se realizó la clasificación de los artículos revisados, colocándolos en 3 grupos, el primero “Sistemas No Lineales Con Incertidumbre que Utilizan un Enfoque de Redes Neuronales”; el segundo llamado “Identificación”, que a su vez se dividió en “Identificación de sistemas no lineales en general”, “Identificación en Línea”, “Identificación Fuera de Línea” y “Neuro-Identificación”; Por último el tercer grupo nombrado “Ejemplos de Aplicación de Redes Neuronales Dinámicas”.

1.1.1. Los Sistemas No Lineales Con Incertidumbre Abordados con un Enfoque de Redes Neuronales

En la referencia [7] es utilizada una estimación de Estados desconocidos por medio de redes neuronales multicapa utilizando el aprendizaje y capacidades de mapeo no lineales de redes neuronales, los sistemas de control propuestos pueden ser mostrados para dar cabida a una clase más amplia de las incertidumbres de modelado que un LQR (linear-quadratic regulator por sus siglas en inglés) convencional.

En la referencia [8] es utilizado un esquema de control adaptable basado en redes neuronales para los sistemas no lineales en forma estricta de retroalimentación, se supone que las incertidumbres son desconocida, aunque todavía satisface ciertas condiciones de crecimiento que se caracterizan por “funciones delimitantes” compuestas por funciones conocidas multiplicado por constantes desconocidas.

En la referencia [9] el Neuro-identificador incluye una modificación robusta con el fin de garantizar la estabilidad de Lyapunov. Se aplica el enfoque de estabilidad de entrada a estado para acceder a los algoritmos de entrenamiento robustos de tiempo discreto de las redes neuronales recurrentes.

Dentro de la referencia [10] se realiza una comparación de distintos algoritmos de aprendizaje, luego de dicha comparación se llega a la conclusión de que el algoritmo de aprendizaje basado en un filtro de Kalman, tiene mejores propiedades que otros algoritmos de aprendizaje, aunque este algoritmo es más complejo y sensible a la naturaleza de los ruidos; una de las características en las que resalta dicho algoritmo de aprendizaje es una convergencia más rápida, Además el filtro

de Kalman utilizado como la base del algoritmo de aprendizaje es el filtro de Kalman Extendido. También se utiliza el método de Lyapunov para demostrar que el entrenamiento basado en el filtro de Kalman es estable.

En la Referencia [11] centran su investigación en un diseño de control robusto de retroalimentación de la salida por H_∞ basado en una red neuronal, cuya red utiliza el modelo SNNM (standard neural network model, por sus siglas a inglés), cuyo fin es eliminar el efecto de los errores de aproximación, incertidumbres paramétricas, y las perturbaciones externas.

En la referencia [12] se propone una nueva estructura dinámica de red neuronal difusa (DSNFN por sus siglas a inglés) para abordar los problemas de seguimiento adaptable de sistemas no lineales inciertos con múltiples entradas múltiples salidas (MIMO por sus siglas a inglés). Utiliza una red neuronal difusa de cuatro capas (NFN por sus siglas a inglés) para estimar las incertidumbres del sistema en línea. La característica principal de este DSNFN es que puede aumentar o disminuir el número de reglas difusas en el tiempo basado en los errores de seguimiento. Mediante un esquema de control híbrido, que combina el control de modos deslizantes y el control adaptable de estimación limitada con diferentes pesos, mejora el rendimiento del sistema mediante la supresión de la influencia de las perturbaciones externas y los errores de aproximación.

En la referencia [13] se presenta un control robusto adaptable con enfoque en una red neuronal para sistemas no lineales con incertidumbre y de retroalimentación estricta. Los términos desconocidos en pasos intermedios se transmiten y aproximan por una sola red neuronal en el último paso. De esta manera, la estructura del controlador diseñado es mucho más simple, y la ley de control y la ley de adaptación se pueden dar directamente.

En la referencia [14] se propone un control robusto adaptable para una clase de sistemas no lineales con retroalimentación pura con dinámicas no modeladas y signos de ganancia desconocidos utilizando redes neuronales con la función de base radial (RBFNNs). Además las incertidumbres dinámicas se tratan mediante una señal dinámica. Usando el teorema de valor medio y la desigualdad de Young, sólo un parámetro de aprendizaje necesita ser afinado en línea en cada paso de la recursividad.

La referencia [15] se centra en la capacidad de aprendizaje de redes neuronales con función de base radial, utilizada en controladores neuronales adaptables para un grupo de sistemas no lineales de tiempo discreto con incertidumbre, donde las estructuras de los sistemas son idénticas, pero las señales de referencia son diferentes.

En la referencia [16] se investiga el diseño de un control óptimo robusto basado en redes neuronales, para una clase de sistemas no lineales inciertos través enfoque de programación dinámica adaptable.

En [17] Se considera el problema de seguimiento para una clase de sistemas dinámicos no holonómicos (sistema que no es capaz de modificar su dirección instantáneamente sin necesidad de rotar previamente) con incertidumbres. Para hacer frente a las incertidumbres, un sistema de control de la red neuronal recurrente se desarrolla sin requerir conocimiento explícito de la dinámica del sistema. Comparando con los resultados existentes, el sistema de control resultante tiene una estructura más simple, y se puede tratar con incertidumbres paramétricas, así como las incertidumbres no paramétricas, sin embargo, garantiza la estabilidad asintótica de la dinámica de error de seguimiento.

En [18] se propone una red neuronal adaptable como control de superficie dinámica para una clase de sistemas no lineales con retardo de tiempo con incertidumbres dinámicas e histéresis desconocida. La red neuronal se utiliza para describir las no linealidades y las dinámicas desconocidas de los sistemas no lineales con retardo de tiempo.

Los artículos analizados en esta sección utilizan a las redes neuronales dinámicas como controlador, donde la parte de la incertidumbre es asumida mas no estimada por la red neuronal, solamente en las referencias [8] y [14] se propone una estructura de la incertidumbre. Con lo anterior se puede observar que existe un nicho de oportunidad al estimar la parte de la incertidumbre para poder extraer una información mayor de dichos sistemas. De los demás artículos se destacan los distintos tipos de arquitectura de las redes neuronales dinámicas y los algoritmos de aprendizaje utilizados.

1.1.2. Identificación de Sistemas no Lineales

1.1.2.1. Identificación de sistemas no lineales en general

En la referencia [19] se propone una nueva red neuronal difusa recurrente, que tiene la forma de espacio de estado estándar, la estabilidad entrada-estado se aplica a acceder a algoritmos de entrenamiento robustos para la identificación del sistema.

En la referencia [20] se aborda la identificación del sistema dinámico no lineal usando redes neuronales difusas. Se centra tanto la incertidumbre estructurada y la incertidumbre de parámetros. La contribución principal es que se propone un marco analítico integrado para la selección automatizada de la estructura de la red neuronal difusa, además de identificación de parámetros y de la red de histéresis de conmutación. Para hacer frente a la estructura de la incertidumbre, se propone una estrategia de histéresis de conmutación para permitir que el identificador neuronal difuso tenga un rendimiento garantizado a lo largo del proceso de conmutación.

Dentro de la referencia [21] se propone un nuevo modelo de red neuronal recurrente con retardo (TDRNN). El modelo TDRNN propuesto tiene ventajas especiales como es una estructura simple, mayor rango de resolución en la memoria. A partir de entonces, se desarrolla el algoritmo de retropropagación para el TDRNN. Para garantizar la convergencia rápida, las tasas óptimas de aprendizaje adaptable también se derivan en el sentido de tipo discreto de criterio de estabilidad de Lyapunov. También un identificador de TDRNN y un controlador TDRNN se construyen para llevar a cabo la identificación y control de los sistemas no lineales.

En la referencia [22] se utiliza un procedimiento recursivo para la identificación de los modelos lineales conmutados a partir de datos de entrada-salida. A partir de algunos valores iniciales de los vectores de parámetros que representan los diferentes submodelos. Dado el estado discreto estimado, el vector de parámetros asociado se actualiza sobre la base de mínimos cuadrados recursivos o cualquier identificador lineal adaptable que sea rápido. Al elegir adecuadamente el criterio de asignación de los datos, el método en línea propuesto puede extenderse a lidiar también con la identificación de modelos afines a trozos.

En la referencia [23] utiliza una red neuronal artificial como un método de identificación para sistemas no lineales con histéresis. El enfoque establece una red neuronal basada en modelo de Bouc-Wen. Los pesos de la red diseñada corresponden a los parámetros del modelo Bouc-Wen.

En la referencia [24] Se propone el diseño de una red neuronal recurrente y un enfoque de reducción de modelo con el fin de mejorar el equilibrio entre la complejidad y la calidad de los modelos de identificación del sistema no lineal de una caja negra. El diseño de la red neuronal propuesto, basado en una arquitectura de tres capas, ayuda para reducir el número de parámetros del modelo después de la fase de entrenamiento sin ninguna pérdida significativa de precisión de la estimación. La reducción es conseguida por una elección conveniente de las funciones de activación y las condiciones iniciales de los pesos sinápticos.

En los trabajos analizados en esta sección utilizan la identificación para estimar el modelo del sistema, pero no para estimar la parte de incertidumbre, por lo cual existe esa brecha para ser abordada. De nueva cuenta se utilizan distintas arquitecturas y algoritmos de aprendizaje que pueden ser analizados para su aplicación en el presente tema de tesis.

1.1.2.2. Identificación en Línea

En la referencia [25] se utilizan las redes neuronales en conjunto con los mínimos cuadrados recursivos para la identificación del modelo de procesos no lineales variantes en el tiempo. El cumplimiento de los criterios de convergencia y el exceso de parametrización son un problema menor. El enfoque de actualización se aplica a un CSTR no isotérmico con parámetros variables en el tiempo y se demuestra su rendimiento.

En la referencia [26] se desarrolló una red neuronal basada en un algoritmo de retropropagación mejorado. Los resultados indican que las redes neuronales proporcionan un método muy eficaz de la aplicación de la integración de sensores para el monitoreo en línea de las anomalías.

En la referencia [27] se propone un modelo de red neuronal artificial (ANN) para la identificación de sistemas no lineales en tiempo discreto con dinámicas desconocidas. La necesidad de la capa oculta se elimina mediante la expansión del patrón de entrada por polinomios de Chebyshev. El método de mínimos cuadrados recursivos con factor de olvido se utiliza como algoritmo de aprendizaje en línea para el parámetro actualizado.

En la referencia [28] se desarrolla un procedimiento generalizado de identificación y control de una clase de sistemas dinámicos no lineales variables en el tiempo con retardo. La red neuronal recurrente se desarrolla para dar cabida a la identificación en línea, que los pesos de la red neuronal se actualizan de forma iterativa y adaptable a través de los errores de modelo.

La referencia [29] presenta una metodología novedosa de identificación sistemática para el modelado afín en línea de procesos multivariantes utilizando redes neurodifusas adaptables. El enfoque propuesto introduce un procedimiento integrado para estimar simultáneamente una serie de redes neurodifusas adaptables con estructuras dinámicas sencillas y compactas para realizar una identificación del modelo afín multivariable. Un algoritmo de filtro de Kalman extendido (EKF) se desarrolló para ajustar los parámetros neurodifusos libres correspondientes a las reglas difusas creadas.

En la referencia [30] se proponen dos Redes Neuronales (NN) para la identificación de sistemas no lineales a través de redes neuronales dinámicas con diferentes escalas de tiempo, incluyendo los fenómenos tanto rápidos y lentos. El primer identificador utiliza las señales de salida del sistema real para la identificación del sistema. Las leyes de actualización en línea para redes neuronales dinámicas se han desarrollado utilizando la función de Lyapunov. En el segundo identificador de NN, todas las señales de salida de sistema no lineal se reemplazan con las variables de estado de las redes de neuronas. Se propone el algoritmo de identificación en línea con la función de zona muerta para mejorar el rendimiento de identificación del sistema no lineal.

En la referencia [31] se presenta un nuevo algoritmo de identificación en línea para conducir un modelo dinámico no lineal afín y para los procesos de variables en el tiempo. El nuevo algoritmo está ideado sobre la base de un enfoque adaptable de modelado neuro-difuso. Dos modelos neuro-difusos adaptables se identifican secuencialmente sobre la base de los datos más recientes del proceso de entrada-salida para realizar un modelo de tipo afín en línea. Se explora el uso de diferentes medidas de evaluación, incluyendo la comparación con un sistema adaptable de inferencia neuro-difusa (ANFIS) como el pionero y el sistema neuro-fuzzy adaptable más popular con potentes funciones de modelado.

La referencia [32] presenta un método de identificación novedoso para sistemas no lineales, que incluye los aspectos de fenómeno rápido y lento a través de redes dinámicas de múltiples capas neuronales (NN) con escalas en dos ocasiones. El algoritmo de aprendizaje propuesto es similar a la regla de propagación conocida de los perceptrones multicapa pero con los términos de corrección que garantizan errores de seguimiento acotados y pesos limitados. El enfoque de pasividad se utiliza para demostrar que el identificador neuronal propuesto es robusto y evita la necesidad de condiciones persistentemente excitantes (PE).

En esta sección se puede resaltar los algoritmos de identificación en línea que son utilizados, destacando [29], [30] y [32], el primero por ser capaz de identificar en línea un proceso multivariable y ser aplicado en un modelo afín con la entrada, mientras el segundo es utilizado para identificar fenómenos tanto rápidos y lentos, también porque las leyes de actualización en línea para redes neuronales dinámicas se han desarrollado utilizando la función de Lyapunov, el último resalta por utilizar el enfoque de pasividad para demostrar que el identificador es robusto.

1.1.2.3. Identificación Fuera de Línea

En la referencia [33] se desarrolla una idea novedosa de identificación de sistemas dinámicos no lineales a través de una red neuronal inspirada en constructivismo. La red propuesta se conoce como red de crecimiento múltiple-expertos (GMN). En GMN, el espacio del problema se descompone en la superposición de las regiones de dominio, la experiencia y modelos expertos locales se clasifican de acuerdo a su nivel de conocimiento. La salida de la red se calcula por la suave combinación de modelos lineales locales.

En la referencia [34] se aborda la necesidad de tener conocimiento parcial de la dinámica del sistema no lineal y que además sea relajado para el desarrollo de un nuevo enfoque para programación dinámica aproximada (ADP) mediante un proceso de dos partes: la identificación del sistema en línea y la formación de control sin conexión óptima. En primer lugar, en el proceso de identificación del sistema, una red neuronal (NN) se sintoniza para aprender la dinámica completa de la planta, de modo que puede mostrar la estabilidad asintótica local del error de identificación. Luego, utilizando sólo el modelo de sistema de NN aprendido, fuera de línea se intenta que resulte en una nueva ley de control óptimo.

En esta sección, los dos artículos analizados pueden ser de utilidad, el primero por la división que se hace en la región de dominio y el enfoque de constructivismo, mientras que la segunda por el método utilizado para mostrar la estabilidad asintótica del error de identificación, además en el segundo trabajo se propone una nueva ley de control para el sistema utilizando el aprendizaje de la red neuronal.

1.1.2.4. Neuro-Identificación

En la referencia [35] se muestra la técnica de modelado de redes de avance de los sistemas dinámicos, así como el control Modelo Basado en Redes de Adelanto (FNMB). También se demuestra que el controlador FNMB es capaz de realizar el control de desacoplamiento casi perfecto sobre un sistema dinámico MIMO acoplado altamente no lineal.

En la referencia [36] se propone un nuevo método de control adaptable mediante redes neuronales que puede realizar el control adaptable y la identificación de una clase de plantas controladas con incertidumbres lineales y no lineales. Este método utiliza una única red neuronal tanto para el control y la identificación, y una condición de estabilidad asintótica local.

En la referencia [37] se estudia un sistema no lineal para identificación en línea a través de las redes neuronales dinámicas. La principal contribución de este trabajo es que se aplica el enfoque de la pasividad a acceder a varias nuevas propiedades estables de neuro-identificación. El algoritmo de gradiente descendente para el ajuste de peso es estable en un sentido y robusto para cualquier incertidumbre acotada.

En la referencia [38] se proponen perceptrones multicapa recurrentes de tiempo continuo (RMLP) para identificar los sistemas no lineales. Usando el teorema de la función de aproximación para perceptrones multicapa (MLP), se consigue que RMLP pueden aproximar cualquier sistema dinámico en cualquier grado de precisión. El algoritmo de aprendizaje sugerido es similar a la regla retropropagación bien conocido de la MLP pero con un término adicional que aseguran la estabilidad de error de identificación.

En la referencia [39] las redes neuronales dinámicas de una sola capa se utilizan para el sistema no lineal de identificación en línea. El enfoque de pasividad se aplica para acceder a varias nuevas propiedades de estabilidad del neuroidentificador. El algoritmo de retropropagación con un término modificado que se determina por el aprendizaje fuera de línea puede hacer al neuroidentificador robusto con respecto a cualquier incertidumbre acotada.

En la referencia [40] las redes neuronales dinámicas multicapa se utilizan para la identificación en línea de sistema no lineal. El algoritmo de retropropagación con un término modificado puede hacer al algoritmo de neuroidentificación sea robusto y estable con respecto a cualquier incertidumbre acotada.

En la referencia [41] Se propone un esquema de identificación en línea para mejorar el rendimiento de error residual del Estado frente a las perturbaciones. El esquema propuesto se basa en una ley de adaptación que modifica los pesos para aproximar las no linealidades desconocidas con error acotado. Un modelo de identificación con retroalimentación se introduce para mejorar el rendimiento de error de estado. La retroalimentación se basa en una función de delimitación para estimar un límite superior para los disturbios. A través de una técnica de delimitación y métodos de Lyapunov, se muestran las características del error.

Cada uno de los artículos analizados en esta sección, tratan que la incertidumbre no afecte la estabilidad de los sistemas, pero no estiman dicha incertidumbre, dejando de lado, mucha información que puede ser extraída de dicha estimación; En la referencia [41] se utiliza una función de delimitación para estimar el límite superior de las perturbaciones, además de utilizar el método de Lyapunov para demostrar las características del error de identificación.

1.1.3. Ejemplos de Aplicación de Redes Neuronales Dinámicas

En la referencia [42] se presenta un esquema de identificación y control de la digestión anaerobia, con redes neuronales con entrenamiento en línea. El esquema de control propuesto, tiene como características deseadas el seguimiento, de regulación y de robustez, en varias tareas de control de la digestión anaerobia, incluyendo los puntos de ajuste o variaciones entradas de proceso, incluso en presencia de ruido de medición o en casos de cambios de parámetros de proceso. El rendimiento de los tres algoritmos de entrenamiento, la propagación hacia atrás y dos técnicas de optimización aleatoria diferentes.

En la referencia [43] se abarca el desarrollo de un FES (functional electrical stimulation) por sus siglas en inglés, cuyas técnicas de identificación desempeñan un papel crucial para obtener la información sobre el sistema neuromuscular. Sin embargo, en la práctica los sistemas subyacentes tienen una dinámica no lineal además de incertidumbres asociadas. Las redes neuronales artificiales proporcionan dichos modelos no lineales de adaptación y pueden extenderse a problemas de identificación del sistema o de predicción. La identificación rápida y robusta en línea usando varias capas de redes neuronales recurrentes es abordada en dos pasos: (a) la arquitectura de red de múltiples capas a través de supervisión de señal y el error recurrente de la red neuronal (SERNN) por sus siglas en inglés. (b) la formación sólida a través del método de mínimos cuadrados recursivos modificados, algoritmo que tiene un factor de olvido dinámico.

En la referencia [44] trata de la identificación de un sistema (MIMO) que es un sistema de turbogenerador no lineal, mediante el uso de redes neuronales dinámicas. Se utilizan de Redes Neuronales Adaptables con tiempo de retardo (ATDNN) para llevar a cabo la neuroidentificación. Se muestra analíticamente la capacidad de la estructura propuesta para representar el mapa de entrada-salida no lineal del sistema turbogenerador.

En la referencia [45] se presenta una metodología general de diseño del controlador para el control híbrido neuroinverso con la separación de no linealidades basado en el conocimiento, para los sistemas no lineales industriales. En los sistemas no lineales industriales, los diversos tipos de incertidumbres pueden causar un grave deterioro en el comportamiento del sistema. El neurocontrolador sólo es necesario para controlar una parte de la dinámica no lineal original de los sistemas no lineales industriales que tienen incertidumbre. La estructura de la red neuronal empleada en el neurocontrolador se hace más simple y el consumo de tiempo en el entrenamiento se reduce.

En la referencia [46] el núcleo de un reactor nuclear (VVER) se identifica mediante una estructura autorregresiva lineal múltiple con entradas exógenas (NARX), incluidas las redes neuronales con diferentes intervalos de tiempo y un método de aprendizaje compuesto heurístico, que consiste un aprendizaje off y on-line por lotes. Un controlador del núcleo del reactor nuclear inteligente, posee las capacidades de generación de datos rápidos para la red neuronal NARX y un sistema difuso basado en el conocimiento y la experiencia del operador con el propósito de la toma de decisiones. Esta metodología representa un método innovador de control de núcleo utilizando sistemas neuro-difusos y se puede utilizar para la identificación y control de otras plantas no lineales complejas.

En la referencia [47] se plantea que el factor crucial que afecta a los sistemas de energía modernos hoy en día es el control de flujo de carga. El controlador de flujo de potencia unificado es un medio eficaz para el control del flujo de potencia. El UPFC se controla convencionalmente usando controladores PI. Dos neuroidentificadores son utilizados para identificar la dinámica no lineal de un sistema de energía multimáquina y UPFC, un neuroidentificador para el inversor de derivación y otro para el inversor serie.

En la referencia [48] se propone un sistema neuro-difuso modificado de tipo TS (MTSNFS) para la identificación en línea, que posee seis capas de neuronas para realizar la inferencia difusa. Se propone un algoritmo de aprendizaje híbrido que combina la estimación usando mínimos cuadrados recursivos (RLS) y el aprendizaje derivado ordenado se utiliza para la estimación de parámetros. Tanto la estructura y los parámetros se pueden determinar de forma automática en línea sin un conocimiento a priori.

En la referencia [49] se propone un modelo de espacio de estado con dinámica no lineal, con una red neuronal que utiliza un algoritmo de aprendizaje secuencial capaz de hacer simulación en línea, en la que el modelo predice y se adapta a la llegada de cada nuevo elemento de datos hidrológicos de una manera secuencial, lo que permite la modelización hidrológica en línea. Además utiliza el filtro de Kalman extendido (EKF) como algoritmo de aprendizaje que se desarrolló para entrenar una red neuronal de un perceptrón multicapa (MLP), y se conoce como el método MLP-EKF con actualización de covarianza de ruido (MLP-EKFQ). La propuesta MLP-EKFQ se utilizó para desarrollar un sistema de alerta en tiempo real en línea para predecir las temperaturas de los ríos afectados por la descarga de agua de refrigeración 1 km aguas abajo de una central térmica.

En esta sección se muestran una serie de aplicaciones, cada una de ellas, en distintas áreas y utilizando distintas arquitecturas de redes neuronales, algoritmos de aprendizaje, además se puede ver una panorámica mucho mayor si se analiza qué pasaría si se estimara alguna incertidumbre en estas aplicaciones.

1.2. Planteamiento del Problema

El análisis de estabilidad de sistemas con incertidumbre, siempre ha sido de interés en la comunidad de control; pero por motivo de la incertidumbre, el análisis de estabilidad no resulta sencillo [50]. Esto es debido a que los modelos matemáticos son una aproximación de un sistema físico, *una representación exacta es imposible o difícil de obtener*. Las razones debidas a las cuales se hace imposible o difícil obtener una representación exacta son imprecisiones de cálculo en la determinación de los parámetros, cambio en las constantes de tiempo, desgaste interno, y dinámicas no modeladas.

1.3. Motivación

El análisis de la estabilidad de los sistemas descritos con anterioridad pasa a ser, de simple estabilidad a algo más complejo: Estabilidad Robusta. Esto es, que el sistema a pesar de las incertidumbres sea estable. Cuando un sistema posee incertidumbre de algún tipo, se convierte en una familia de sistemas. Utilizando la metodología de identificación en línea mediante redes neuronales dinámicas, se busca garantizar el buen desempeño de todos los elementos de la familia en algún aspecto particular, en este caso, que haga estable el error de identificación.

Además al identificar la incertidumbre en línea, se puede encontrar un sinnúmero de aplicaciones prácticas para esta aproximación, ejemplos de estas aplicaciones pueden ser realizar un control robusto a incertidumbre, el diagnóstico de desgaste de algún componente de la planta, diagnosticar el comportamiento de la planta bajo distintos factores, detección de desvíos en el comportamiento del sistema e inclusive evaluación del comportamiento del sistema ante perturbaciones.

1.4. Objetivo y Metas

1.4.1. Objetivo

Diseñar una metodología de identificación en línea mediante redes neuronales dinámicas, cuyo error de identificación sea estable, para un sistema no lineal afín con la entrada cuya incertidumbre es parcial en el término no lineal de retroalimentación.

1.4.2. Metas

- Diseño de la arquitectura de la red neuronal dinámica.
- Diseño de la ley de aprendizaje que lleve al error de identificación a estado estable.
- Formalizar el resultado en un teorema que manifieste las condiciones necesarias para acotar el error de identificación.
- Generar un modelo neuronal dinámico dentro de una región de operación para el sistema no lineal con incertidumbre parcial en el término de la retroalimentación.

II. Marco Teórico

2.1. Las Redes Neuronales Artificiales

Idealmente, el objetivo de las Redes de Neuronas Artificiales es llegar a diseñar máquinas con elementos neuronales de procesamiento paralelo, de modo que el comportamiento global de esa red emule, de la forma más fiel posible, los sistemas neuronales de los animales [51].

La gran diferencia entre una máquina neuronal y los programas de computador convencionales es que éstas elaboran en cierta medida, la información de entrada para obtener una salida o respuesta.

No se trata de la aplicación ciega y automática de un algoritmo. De hecho, el proceso de elaboración de la información recibida depende de las distintas características, tanto estructurales como funcionales, de la red.

2.1.1. Neurona Artificial

La neurona artificial es un elemento que posee un estado interno, llamado nivel de activación, y recibe señales que le permiten cambiar de estado [51].

Si se denomina S al conjunto de estados posibles de la neurona, S podrá ser, por ejemplo, $S = \{0, 1\}$, siendo 0 el estado inactivo y 1 el activo.

Las neuronas artificiales poseen una función que les permite cambiar de nivel de activación a partir de las señales que reciben; a dicha función se la denomina función de transición de estado o función de activación.

2.1.2. Arquitectura de las Redes Neuronales Artificiales

Las redes neuronales artificiales, también llamados sistemas neuronales artificiales son un intento de imitar (simular) por lo menos parcialmente la estructura y las funciones del cerebro y el sistema nervioso de los seres vivos [52].

En términos generales, una red neuronal artificial es un sistema de información o señal de procesamiento compuesta de un gran número de elementos de procesamiento simples, llamadas neuronas artificiales o simplemente nodos, que están interconectados por enlaces directos llamado

conexiones y que cooperan para realizar el procesamiento distribuido en paralelo con el fin de resolver una tarea computacional deseada.

Una de las características más atractivas de las redes neuronales artificiales es su capacidad para adaptarse a condiciones especiales, cambiando sus puntos fuertes o estructura de conexión [52].

Una neuronal artificial es diseñada para realizar funciones o cálculos que sean específicos, de manera que la arquitectura de la misma es muy particular para cada problema que se desee resolver con esta técnica.

2.1.2.1. Modelado Estático de Redes Neuronales

Una neurona artificial estática muestra un mapeo estático de las entradas hacia la salida [1]. Una forma básica de este modelo es como el que propuso McCulloch Pitts [53] en la cual se expresa la salida y de la neurona de la siguiente manera

$$y = \varphi(w_1x_1 + w_2x_2 + \dots + w_nx_n + \theta) = \varphi(Wx + \theta),$$

$$y = \varphi\left(\sum_{i=1}^n w_ix_i + \theta\right). \quad (2.1)$$

Otra forma que es común encontrar de los modelos de las neuronas artificiales, es a través de representaciones en esquemas gráficos, los cuales son muy útiles para visualizar la analogía que tienen con una neurona biológica. Para el caso del modelo McCulloch Pitts el esquema correspondiente es el de la Figura 2.1, donde x_i es la i -ésima entrada de la neurona. w_i es el i -ésimo peso de la neurona y modela la acción de la sinapsis de una neurona biológica. El punto suma, el umbral θ y la función de activación $\varphi(u)$ corresponden a el cuerpo de la neurona.

Finalmente la salida y representa el axon de la neurona biológica. La función de activación en general es no lineal y acotada, generalmente se escoge el perfil sigmoidal, pero puede tener entre otros perfiles.

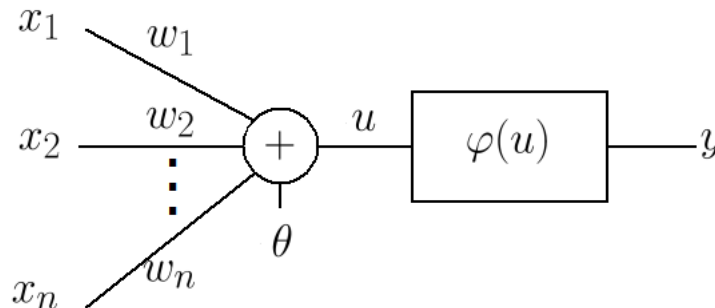


Figura 2.1: Neurona Artificial Estática Modelo McCulloch Pitts.

2.1.2.2. Modelado Dinámico de Redes Neuronales

Las Redes neuronales dinámicas poseen al menos una señal retroalimentada. Esta retroalimentación puede estar hecha en tiempo discreto o en tiempo continuo [1].

Por ejemplo una estructura de red neuronal dinámica en tiempo continuo podría ser como la siguiente

$$\dot{\hat{x}} = \varphi \left(w_1 \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \theta_1 \right), \quad y = \varphi \left(w_2 \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \theta_2 \right). \quad (2.2)$$

La estructura de la red neuronal dinámica mostrada en la ecuación (2.2) se encuentra plasmada en la Figura 2.2.

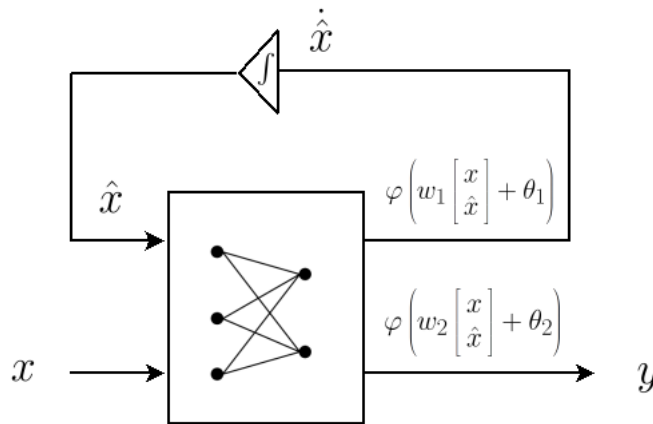


Figura 2.2: Estructura de una Red Neuronal Dinámica.

2.2. Aproximación de Funciones

El problema de aproximación de funciones se divide en dos tipos [54]:

- En el primero, se supone que se conoce la expresión explícita de la función, y lo que se busca es una estructura paramétrica que aproxime la función con algún grado de precisión deseado.
- En el segundo caso, el problema consiste en obtener una función que se ajuste a la relación entrada-salida existente en un conjunto finito de datos.

En el primer caso se busca representar cierta clase de funciones con una arquitectura neuronal, como lo puede ser una suma de potencias a través de manipular series geométricas. Este enfoque permite obtener versiones simplificadas de funciones, con el único fin de ayudar al análisis de los problemas y también para simplificar los cálculos. En el segundo caso está enfocado de manera experimental debido a que no se cuenta con una expresión explícita de la función de la cual se desea obtener una aproximación.

2.3. Conjunto Finito de datos o Nube de Datos

Sea Ω un conjunto que contiene un número finito de vértices en un sistema de coordenadas bidimensionales, tridimensionales, p-dimensionales. Estos vértices se identifican habitualmente como coordenadas X, Y, Z, etc.

Para el caso bidimensional se tiene un conjunto de la forma siguiente

$$\Omega \subset \mathbb{R}^2, \quad (2.3)$$

donde

$$\Omega = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}. \quad (2.4)$$

Si se aborda encontrar un \hat{y} , que es una función que aproxima a Ω puede tener una interpretación gráfica como la de la Figura 2.3

$$\begin{aligned} \hat{y} &= f(x), \\ f(x_i) &\cong y_i \forall (x_i, y_i) \in \Omega. \end{aligned} \quad (2.5)$$

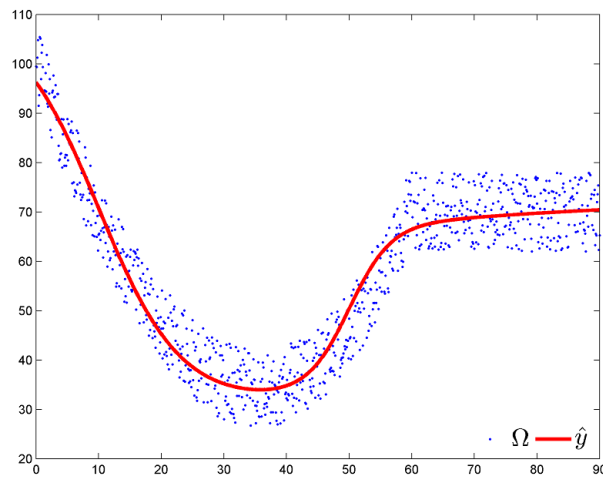


Figura 2.3: Nube de datos bidimensional Ω y función de aproximación \hat{y} .

2.4. Aproximación de Funciones mediante Redes Neuronales

El motivo que ha dado más impulso a la utilización de las redes neuronales en el campo de la identificación y control de procesos, es su demostrada capacidad de aproximar funciones con un grado arbitrario de precisión. Los resultados a este respecto parten de los trabajos de Kolmogorov[55] de carácter general en la disciplina de aproximación de funciones. Además han sido adaptados y refinados por diversos autores como Hecht-Nielsen[56, 57], Kurková[58], Hornik[59], Ito[60], Cybenko[61], etc. al campo de la aproximación de funciones basada en redes neuronales [62]. El resultado general más interesante (formulado independientemente en [59] y [61]) es que las redes neuronales de dos y tres capas con funciones de activación sigmoideas son aproximadores universales.

Otras aportaciones como Hush[63] y Meltser[64], por ejemplo, han abierto las puertas a algoritmos constructivos de redes neuronales que, utilizando como fundamento los teoremas precedentes de existencia de soluciones, crean redes neuronales con errores de aproximación arbitrariamente pequeños.

2.5. Teorema de Cybenko

Cybenko demostró la existencia de una red neuronal de dos capas con N neuronas en la primera capa como en la Figura 2.4 que puede aproximar cualquier función $f \in C(I_n)$ donde I_n representa al cubo unitario n -dimensional $[0, 1]^n$, $C(I_n)$ es el espacio de funciones continuas en I_n [61].

Teorema 2.1. *Sea σ cualquier función sigmoidea. Dada cualquier función continua en $[0, 1]^n$ (o en cualquier conjunto compacto de \mathbb{R}^n) y un $\epsilon > 0$, existen unos vectores ω , α y θ , y una función parametrizada $G(x, \omega, \alpha, \theta)$ de $[0, 1]^n$ en \mathbb{R} tal que*

$$|G(x, \omega, \alpha, \theta) - f(x)| < \epsilon, \quad \forall x \in [0, 1]^n, \quad (2.6)$$

donde

$$G(x, \omega, \alpha, \theta) = \sum_{j=1}^N \alpha_j \sigma(\omega_j^T x + \theta_j), \quad (2.7)$$

y

$$\begin{aligned} \omega_j &\in \mathbb{R}^n, \alpha_j, \theta_j \in \mathbb{R}; \\ \omega &= (\omega_1, \omega_2, \dots, \omega_N), \\ \alpha &= (\alpha_1, \alpha_2, \dots, \alpha_N), \\ \theta &= (\theta_1, \theta_2, \dots, \theta_N). \end{aligned} \quad (2.8)$$

2.6. Modelo de red neuronal de dos capas para aproximación de funciones

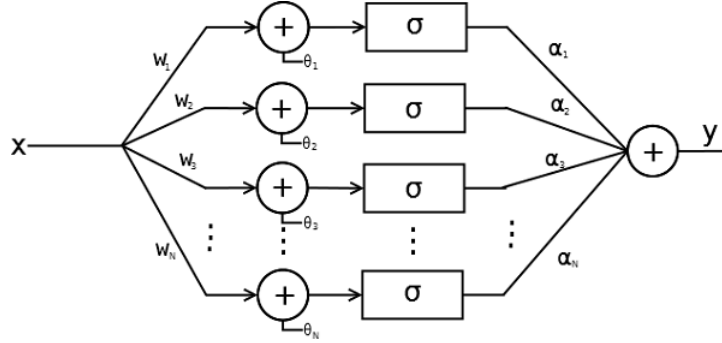


Figura 2.4: Red Neuronal de 2 Capas.

La expresión matemática que representa la arquitectura utilizada en la Figura 2.4 está dada por cualquiera de las ecuaciones (2.9) y (2.10).

$$y = \alpha^T \begin{bmatrix} \sigma(w_1^T x + \theta_1) \\ \sigma(w_2^T x + \theta_2) \\ \sigma(w_3^T x + \theta_3) \\ \vdots \\ \sigma(w_N^T x + \theta_N) \end{bmatrix}, \quad (2.9)$$

o bien

$$y = \sum_{j=1}^N \alpha_j \sigma(w_j^T x + \theta_j). \quad (2.10)$$

2.7. Pseudoinversa de una Matriz

Si A es una matriz de $n \times n$ con columnas linealmente independientes, entonces es invertible, y la única solución de $Ax = b$ es $x = A^{-1}b$. Si $m > n$ y A es de $m \times n$ con columnas linealmente independientes, entonces $Ax = b$ no tiene solución exacta, pero la mejor aproximación está dada por la solución única de mínimos cuadrados $\bar{x} = (A^T A)^{-1} A^T b$. En consecuencia, la matriz $(A^T A)^{-1} A^T$ desempeña el papel de un “inverso de A ” en esta situación [65].

Si A es una matriz con columnas linealmente independientes, entonces la pseudoinversa de A es la matriz A^+ definida por

$$A^+ = (A^T A)^{-1} A^T. \quad (2.11)$$

Observe que si A es de $m \times n$, entonces A^+ es de $n \times m$.

2.8. Algunas Propiedades de las Matrices

En esta sección se muestran algunas propiedades que fueron de utilidad como herramientas para lograr el resultado principal de esta tesis.

La traza de A es la suma de los elementos de la diagonal principal de A , se denota como $tr\{\cdot\}$, donde

$$tr\{A\} = \sum_{i=1}^n \alpha_{ii}. \quad (2.12)$$

Sea $A, B \in \mathbb{R}^{m \times n}$ y $\alpha \in \mathbb{R}$

las propiedades fundamentales de la traza son [66]:

1. $tr\{\alpha A\} = \alpha tr\{A\}$.
2. $tr\{A + B\} = tr\{A\} + tr\{B\}$.
3. $tr\{AB\} = tr\{BA\}$.

Si $A = \alpha_{ij} \in \mathbb{R}^{m \times n}$ entonces la transpuesta de A escrita como A^T , es la matriz $A^T = \gamma_{ij}$ donde $\gamma_{ij} = \alpha_{ji}$ para todas las i y j .

La Transpuesta de A es la matriz que se obtiene intercambiando los renglones de A con las columnas de A , las propiedades formales básicas de la transpuesta son [66]:

1. $(A^T)^T = A$.
2. $(A + B)^T = A^T + B^T$.
3. $(AB)^T = B^T A^T$.

Por lo que si $a \in \mathbb{R}^m, b \in \mathbb{R}^m, M \in \mathbb{R}^{m \times q}, N \in \mathbb{R}^{q \times m}$

$$a^T M N b = tr\{M N b a^T\} = tr\{N b a^T M\} = tr\{b a^T M N\}. \quad (2.13)$$

2.9. Normas y Cotas Superiores de Productos Vectoriales y Matrices

La norma de $\|x\|$ de un vector x es una función de valor real con las propiedades [67]:

1. $\|x\| \geq 0 \forall x \in \mathbb{R}^n$, con $\|x\| = 0 \Leftrightarrow x = 0$.
2. $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$.
3. $\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in \mathbb{R}$ y $x \in \mathbb{R}^n$.

Las tres normas usadas más comúnmente son $\|x\|_1$, $\|x\|_\infty$ y $\|x\|_2$ o norma euclidiana, las cuales tenemos definidas como:

$$\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}, \quad 1 \leq p < \infty, \quad (2.14)$$

$$\|x\|_\infty = \max_i |x_i|, \quad (2.15)$$

$$\|x\|_2 = \|x\| = (x_1^2 + \cdots + x_n^2)^{\frac{1}{2}} = (x^T x)^{\frac{1}{2}}. \quad (2.16)$$

Para todo $x \in \mathbb{R}^n, y \in \mathbb{R}^n$. Las normas matriciales mas comunes son:

$$\|A\|_1 = \max_j \sum_{i=1}^m |a_{ij}|, \quad \|A\|_2 = [\lambda_{\max}(A^T A)]^{\frac{1}{2}}, \quad \text{y} \quad \|A\|_\infty = \max_i \sum_{j=1}^m |a_{ij}|. \quad (2.17)$$

Con las propiedades anteriores se pueden definir los siguientes axiomas, para lo cual considere que $a, b \in \mathbb{R}^n$:

$$a^T b \leq \frac{1}{2} a^T a + \frac{1}{2} b^T b = \frac{1}{2} \|a\|^2 + \frac{1}{2} \|b\|^2. \quad (2.18)$$

$$a^T b \leq \|a\| \cdot \|b\|. \quad (2.19)$$

$$x^T y \leq \|x\|_2 \cdot \|y\|_1, \quad (2.20)$$

$$\|x\|_1 \geq \|x\|_2. \quad (2.21)$$

Sea $sign(a)$ la función signo de a , donde $sign(a) := [sign(a_1), sign(a_2), \dots, sign(a_n)]^T$ de modo que es fácil definir el siguiente axioma

$$\|a\|_1 = \sum_{i=1}^n |a_i| = \sum_{i=1}^n a_i \cdot sign(a_i), \quad (2.22)$$

o bien,

$$\|a\|_1 = a^T \cdot sign(a). \quad (2.23)$$

2.10. Estabilidad por Criterio de Lyapunov

2.10.1. Principio del Teorema de Estabilidad de Lyapunov

De manera general un sistema es estable si su función de energía total V , es una función definida positiva, además es continuamente decreciente, es decir, la derivada con respecto al tiempo de la V total es definida negativa, hasta que alcanza un estado de equilibrio. En un sistema no lineal no tiene por qué existir una manera simple de definir una función de energía, por ello, el método de estabilidad por Lyapunov introduce una función de energía auxiliar denominada *función candidata de Lyapunov* [68].

2.10.1.1. Función Candidata de Lyapunov

Suponga que $V : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función escalar. V es una función candidata de Lyapunov si es una función positiva definida localmente, es decir, la función candidata de Lyapunov es mayor que cero para todo valor en los reales excepto el cero, y el único valor donde tiene un valor de cero es el mismo cero:

$$\begin{aligned} V(0) &= 0, \\ V(x) &> 0 \quad \forall x \in \mathbb{R} \setminus \{0\}. \end{aligned} \tag{2.24}$$

2.10.1.2. Teorema de Estabilidad de Lyapunov

Teorema 2.2. *Considere el sistema $\dot{x} = f(x, t)$, donde $f(0, t) = 0$. Si \exists una función escalar $V(x, t)$ con las primeras derivadas parciales continuas que satisface las condiciones*

$$\bullet V(x, t) \text{ es definida positiva, es decir, } V(x, t) > 0 \quad \forall x, t \neq 0, \tag{2.25}$$

$$\bullet \dot{V}(x, t) \text{ es definida negativa, es decir, } \dot{V}(x, t) < 0 \quad \forall x, t \neq 0. \tag{2.26}$$

entonces el estado de equilibrio en el origen es uniforme y asintóticamente estable.

Si se satisface (2.25) pero \dot{V} es semidefinida negativa, es decir, $\dot{V}(x, t) \leq 0 \quad \forall x$ entonces el estado de equilibrio en el origen es estable.

Con las herramientas descritas en el presente capítulo se plantean las bases para la correcta comprensión de los resultados principales de esta tesis, como por ejemplo las propiedades de las matrices mencionadas o el apartado de normas y cotas superiores de productos vectoriales y matrices, los cuales son absolutamente necesarios para el desarrollo del teorema 3.1.

III. Resultados Principales

La estabilidad en el error de identificación en las redes neuronales dinámicas puede ser determinada mediante la aplicación de un análisis estilo Lyapunov, para el cual en el presente capítulo son descritas cada una de las condiciones necesarias para acotar el error de identificación y hacerlo estable, lo anterior, presentado de manera formal mediante la elaboración de un teorema, el cual es uno de los principales resultados de la presente tesis.

Además se muestra la metodología propuesta para la obtención de un modelo neuronal dinámico que aproxime al sistema no lineal con incertidumbre dentro de una región de operación, elaborado a partir de los datos recopilados del entrenamiento en línea y del elaborado fuera de línea mediante la aplicación de una red neuronal estática, cuya arquitectura se propone en este apartado.

3.1. Aproximación Neuronal Dinámica de un sistema no lineal afín con la entrada con incertidumbre en la retroalimentación

Considere el siguiente sistema no lineal

$$\sum_{SNL} : \begin{cases} \dot{x} = f(x) + g(x)u, \\ y = x. \end{cases} \quad (3.1)$$

donde $f(x) = f_0(x) + \Delta_f(x)$, es decir, la función no lineal $f(x)$ está formada por la suma de una parte conocida $f_0(x)$ y una parte denominada incertidumbre representada por $\Delta_f(x)$, la cual es considerada una incertidumbre aditiva por definición, con lo anterior se deduce la siguiente suposición:

Suposición 1. Suponga que la incertidumbre del sistema $\Delta_f(x)$ está acotada, y solo se conoce que la norma euclidiana de la incertidumbre es menor o igual que un valor dado denominado $\delta_f(x)$, es decir, $\|\Delta_f(x)\|$ tiene como cota superior el valor $\delta_f(x)$, esto es representado por (3.2), el valor de $\delta_f(x)$ es conocido o seleccionado por diseño.

$$\|\Delta_f(x)\| \leq \delta_f(x). \quad (3.2)$$

Entonces considerando la incertidumbre en \sum_{SNL} tenemos:

$$\sum_{SNL} : \begin{cases} \dot{x} = f_0(x) + \Delta_f(x) + g(x)u, \\ y = x. \end{cases} \quad (3.3)$$

La estructura propuesta de la red neuronal dinámica para identificación de la incertidumbre (Figura 3.1) es la siguiente:

$$\sum_{RND} : \begin{cases} \dot{\hat{x}} = f_0(x) + W\varphi(\hat{x}) + \psi + g(x)u + L(x - \hat{x}), \\ y = \hat{x}. \end{cases} \quad (3.4)$$

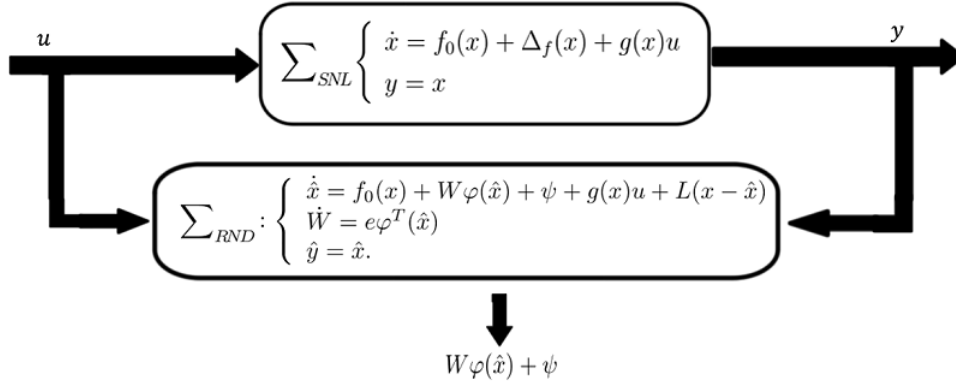


Figura 3.1: Estructura propuesta de la red neuronal propuesta.

Observación 1. Nótese que el sistema (3.3) tiene disponible el estado a la salida, por lo que es aprovechado en la estructura propuesta de la red neuronal dinámica, es decir, la estructura propuesta utiliza la salida del sistema para tener acceso al estado, y con ello aproximar la incertidumbre del sistema.

Observación 2. Nótese que el término neuronal $W\varphi(\hat{x}) + \psi$ de (3.4) se propone para realizar la aproximación de la incertidumbre. Por otro lado el término $L(x - \hat{x})$ del sistema (3.4) se agrega debido a que la dinámica del error puede ser modificada convenientemente mediante la elección de L , garantizando que el error de identificación sea estable.

Con el uso de la red neuronal propuesta, se formula el teorema 3.1, además cabe mencionar que la manera de demostrar el Teorema 3.1, es decir, realizar la estimación de la incertidumbre del sistema, es mediante un análisis estilo Lyapunov aplicado a la arquitectura del de la red neuronal propuesta.

Teorema 3.1. *Considere un sistema \sum_{SNL} de la forma (3.3), el cual es identificado por \sum_{RND} (3.4) y a su vez la Suposición 1 expresada en (3.2) se satisface, sí se selecciona al algoritmo de aprendizaje de la red neuronal \dot{W} como*

$$\dot{W} = e\varphi^T(\hat{x}), \quad (3.5)$$

también si el umbral neuronal ψ se elige como

$$\psi = \text{sign}(e)\delta_f(x), \quad (3.6)$$

donde $\text{sign}(e)$ es la función signo del error de identificación y $\delta_f(x)$ es la cota superior de la incertidumbre; además se asegura que la matriz de ganancias L sea simétrica y definida positiva, es decir

$$L = L^T > 0, \quad (3.7)$$

entonces el error de identificación $e = x - \hat{x}$ es estable y acotado

$$\lim_{t \rightarrow \infty} \|e(t)\| < \infty. \quad (3.8)$$

Demostración: Se selecciona a $V(x)$ como la función candidata de Lyapunov siguiente

$$V(e, W) = \frac{1}{2}e^T e + \frac{1}{2}\text{tr}\{WW^T\}, \quad (3.9)$$

se tiene que la derivada con respecto al tiempo de dicha función candidata es

$$\dot{V}(e, W) = e^T \dot{e} + \text{tr}\{W\dot{W}^T\}. \quad (3.10)$$

Primeramente se define el error de identificación $e = x - \hat{x}$ por lo tanto $\dot{e} = \dot{x} - \dot{\hat{x}}$.

Sustituyendo \sum_{SNL} y \sum_{RND} en la derivada del error de identificación tenemos

$$\begin{aligned} \dot{e} &= f_0(x) + \Delta_f(x) + Bu - f_0(x) - W\varphi(\hat{x}) - \psi - Bu - L(x - \hat{x}), \\ \dot{e} &= \cancel{f_0(\hat{x})} + \Delta_f(x) + \cancel{Bu} - \cancel{f_0(\hat{x})} - W\varphi(\hat{x}) - \psi - \cancel{Bu} - L(x - \hat{x}), \end{aligned} \quad (3.11)$$

en este momento del análisis se tiene que como $e = x - \hat{x}$

$$\dot{e} = \Delta_f(x) - W\varphi(\hat{x}) - \psi - Le. \quad (3.12)$$

Luego manipulando la señal de error en $\dot{V}(x)$ a partir de (3.12) se obtiene que

$$\begin{aligned}\dot{V}(e, W) &= e^T(\Delta_f(x) - W\varphi(\hat{x}) - \psi - Le) + tr\{W\dot{W}^T\}, \\ \dot{V}(e, W) &= e^T\Delta_f(x) - e^T\psi - e^TLe - e^TW\varphi(\hat{x}) + tr\{W\dot{W}^T\},\end{aligned}\quad (3.13)$$

considerando los dos últimos términos de (3.13) se tiene

$$\begin{aligned}-e^TW\varphi(\hat{x}) + tr\{W\dot{W}^T\} &= \\ -tr\{W\varphi(\hat{x})e^T\} + tr\{W\dot{W}^T\} &= \\ tr\{W[-\varphi(\hat{x})e^T + \dot{W}^T]\}.\end{aligned}\quad (3.14)$$

Lo anterior es debido a la propiedades de la traza (2.13).

Luego de las manipulaciones algebraicas anteriores resulta

$$\dot{V}(e, W) = e^T\Delta_f(x) - e^T\psi - e^TLe + tr\{W[-\varphi(\hat{x})e^T + \dot{W}^T]\}.\quad (3.15)$$

Si se selecciona una ley de aprendizaje $\dot{W}^T = \varphi(\hat{x})e^T$ y se sustituye en el término (3.15), una parte de este se cancela y se obtiene

$$\dot{V}(e, W) = e^T\Delta_f(x) - e^T\psi - e^TLe.\quad (3.16)$$

Luego de (3.16), se toma el termino $e^T\Delta_f(x)$ y en el se aplica la propiedad de la norma dada en (2.19), Por lo que

$$e^T\Delta_f(x) \leq \|e\|_1 \|\Delta_f(x)\|.\quad (3.17)$$

Por la propiedad de la norma representada por (2.23) se tiene que $\|e\|_1 = e^T \text{sign}(e)$ por lo tanto

$$e^T\Delta_f(x) \leq e^T \text{sign}(e) \|\Delta_f(x)\|.\quad (3.18)$$

Sustituyendo la cota superior del término (3.18) en (3.16) se obtiene

$$\begin{aligned}\dot{V}(e, W) &\leq e^T \text{sign}(e) \|\Delta_f(x)\| - e^T\psi - e^TLe, \\ \dot{V}(e, W) &\leq e^T[\text{sign}(e) \|\Delta_f(x)\| - \psi] - e^TLe,\end{aligned}\quad (3.19)$$

como la incertidumbre del sistema (3.3) expresada como $\Delta_f(x)$ se supone acotada (3.2) entonces

$$\dot{V}(e, W) \leq e^T[\text{sign}(e)\delta_f(x) - \psi] - e^TLe.\quad (3.20)$$

Por lo que si se selecciona $\psi = \text{sign}(e)\delta_f(x)$ se tiene

$$\dot{V}(e, W) \leq -e^TLe,\quad (3.21)$$

de esta manera sólo se tendría que asegurar que $L = L^T > 0$ con lo que se garantiza que la derivada de la función candidata de Lyapunov propuesta es definida negativa, es decir

$$\dot{V}(e, W) \leq 0, \quad (3.22)$$

por lo tanto se concluye que el error de identificación $e = x - \hat{x}$ es estable y por tanto, acotado, es decir

$$\lim_{t \rightarrow \infty} \|e(t)\| < \infty. \quad (3.23)$$

Las manipulaciones algebraicas necesarias para llegar a la expresión con (3.23), se muestran de manera formal en el anexo A □

3.2. Modelo Neuronal Dinámico para aproximar un Sistema No Lineal con Incertidumbre dentro de una Región de Operación

Un sistema no lineal como el que se muestra en (3.24)

$$\sum_{\star} : \begin{cases} \dot{x} = f_0(x) + \Delta_f(x) + g(x)u, \\ y = x. \end{cases} \quad (3.24)$$

Utilizando la arquitectura neuronal propuesta para la identificación de la incertidumbre, es decir, el modelo neuronal (3.4), se puede obtener un conjunto de datos a partir del cual podemos obtener un modelo neuronal dinámico para aproximar al sistema no lineal con incertidumbre dentro de una región de operación.

De acuerdo al teorema 3.1, la incertidumbre del sistema es aproximada por la componente neuronal $W\varphi(\hat{x})$, entonces, si nos interesa la k-ésima muestra, lo anterior se expresa por la ecuación (3.25)

$$\Delta_f(x_k) \cong W(t_k)\varphi(\hat{x}(t_k)). \quad (3.25)$$

entonces si se selecciona un α pequeño, por la ecuación (3.30) el error de identificación entre el estado estimado y el estado real es pequeño, así que

$$W(t_k)\varphi(\hat{x}(t_k)) \cong W(t_k)\varphi(x(t_k)), \quad (3.26)$$

entonces la incertidumbre del sistema real puede ser aproximada por

$$\Delta_f(x_k) \cong W(t_k)\varphi(x(t_k)), \quad (3.27)$$

3.2.1. Propuesta para construir el conjunto de datos para la aproximación de la incertidumbre

El conjunto de datos se propone construirse de la siguiente manera, sea Ω_k el conjunto construido en la k -ésima muestra de interés, recopilado a partir de la aplicación de la red neuronal propuesta, el criterio utilizado para ello es el siguiente

$$\Omega_k = \Omega_{k-1} \cup (\bar{x}_k, \bar{W}_k \varphi(\hat{x}_k)), k = 1, 2, 3, \dots; \Omega_0 = \phi, \quad (3.28)$$

donde

$$(\bar{x}_k, \bar{W}_k \varphi(\hat{x}_k)) = \begin{cases} (\bar{x}_{k-1}, \bar{W}_{k-1} \varphi(\hat{x}_{k-1})), & \text{Si no cumple con (3.30) y (3.31), } \forall \chi \in \Omega_k. \\ (x(t), W \varphi(\hat{x}(t))), & \text{Si cumple con (3.30) y (3.31), } \forall \chi \in \Omega_k. \end{cases} \quad (3.29)$$

donde la ecuación (3.30) representa que el error de identificación entre el sistema real y el sistema neuronal sea más pequeño o igual que α

$$\|x(t) - \hat{x}(t)\| \leq \alpha, \quad (3.30)$$

y la ecuación (3.31) representa la distancia euclidiana entre los datos recopilados

$$\|x(t) - \chi\| \geq \beta, \forall \chi \in \Omega_k, \quad (3.31)$$

Observación 3. *Nótese que el conjunto Ω_k , solo añade nuevos valores cuando la distancia entre dichos datos es mayor a β y además dichos datos tienen un error de estimación cercano al valor de α*

Observación 4. *Nótese que para propósitos de aproximación los valores de \bar{x}_k corresponden al dominio y $\bar{W}_k \varphi(\hat{x}_k)$ al codominio de la función que se desea aproximar.*

Se desea hacer una aproximación de la incertidumbre mediante una red neuronal estática de la forma que se muestra en la Figura 3.2, donde N es el número de neuronas de la red, lo cual se deja a consideración del diseñador.

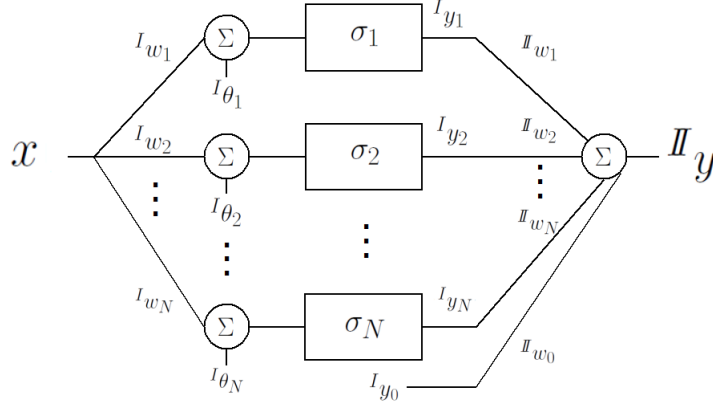


Figura 3.2: Red Neuronal Estática Propuesta.

De aplicación de la red neuronal, obtenemos el vector de pesos sinápticos de la segunda capa, el cual nos ayudara a realizar la aproximación.

Para ello, necesitamos representar de manera matemática la red neuronal utilizada, que está dada por la ecuación (3.32).

$$W_k \varphi(\tilde{x}_k) \cong \mathbb{I} w_1 \sigma_1(\mathbb{I} w_1 \tilde{x}_k + \mathbb{I} \theta_1) + \mathbb{I} w_2 \sigma_2(\mathbb{I} w_2 \tilde{x}_k + \mathbb{I} \theta_2) + \dots + \mathbb{I} w_N \sigma_N(\mathbb{I} w_N \tilde{x}_k + \mathbb{I} \theta_N) + \mathbb{I} w_0 \mathbb{I} y_0, \quad (3.32)$$

donde $k = 1, 2, 3, \dots, \eta$, η es el número total de datos de la nube de datos y $\mathbb{I} y_0 = 1$.

Cabe mencionar que los pesos sinápticos de la primera capa son seleccionados de manera libre por el diseñador, mientras que los pesos sinápticos de la segunda capa serán calculados.

La ecuación (3.32) la transformamos a su forma matricial tenemos que

$$\begin{bmatrix} W_1 \varphi(\tilde{x}_1) \\ W_2 \varphi(\tilde{x}_2) \\ \vdots \\ W_\eta \varphi(\tilde{x}_\eta) \end{bmatrix} \cong \overbrace{\begin{bmatrix} \sigma_1(\mathbb{I} w_1 \tilde{x}_1 + \mathbb{I} \theta_1) & \sigma_2(\mathbb{I} w_2 \tilde{x}_1 + \mathbb{I} \theta_2) & \dots & \sigma_N(\mathbb{I} w_N \tilde{x}_1 + \mathbb{I} \theta_N) & 1 \\ \sigma_1(\mathbb{I} w_1 \tilde{x}_2 + \mathbb{I} \theta_1) & \sigma_2(\mathbb{I} w_2 \tilde{x}_2 + \mathbb{I} \theta_2) & \dots & \sigma_N(\mathbb{I} w_N \tilde{x}_2 + \mathbb{I} \theta_N) & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \sigma_1(\mathbb{I} w_1 \tilde{x}_\eta + \mathbb{I} \theta_1) & \sigma_2(\mathbb{I} w_2 \tilde{x}_\eta + \mathbb{I} \theta_2) & \dots & \sigma_N(\mathbb{I} w_N \tilde{x}_\eta + \mathbb{I} \theta_N) & 1 \end{bmatrix}}^{\Phi} \begin{bmatrix} \mathbb{I} w_1 \\ \mathbb{I} w_2 \\ \vdots \\ \mathbb{I} w_N \\ \mathbb{I} w_0 \end{bmatrix}. \quad (3.33)$$

donde Φ es la matriz generada por las funciones de activación y los parámetros de las simulaciones.

Para obtener el vector de pesos sinápticos de la segunda capa tenemos que

$$\begin{bmatrix} \mathbb{I} w_1^* \\ \mathbb{I} w_2^* \\ \vdots \\ \mathbb{I} w_N^* \\ \mathbb{I} w_0^* \end{bmatrix} \cong \Phi^+ \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\eta \end{bmatrix}, \quad (3.34)$$

$$\mathbb{I} W^* \cong \Phi^+ \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\eta \end{bmatrix}, \quad (3.35)$$

con lo anterior tenemos que la aproximación deseada para la incertidumbre del sistema se expresa en la ecuación (3.36)

$$\Delta_f(x) \cong \mathbb{I} W^* \sigma(I W x + I \theta), \quad (3.36)$$

con ello, del entrenamiento en línea realizado con la red neuronal resultante del teorema 3.1 y del realizado fuera de línea con la red neuronal estática se puede obtener un modelo neuronal que aproxima al sistema (3.24) en la región de operación $x(t)$ cercana al dominio del conjunto Ω_k .

$$\sum_{Idn^*} : \begin{cases} \dot{x} \cong f_0(x) + \mathbb{I} W^* \sigma(I W x + I \theta) + g(x)u, \\ y = x. \end{cases} \quad (3.37)$$

Observación 5. *Nótese que la selección de los valores de α y β tienen una repercusión directa en la elección de los datos que formaran la nube de datos; el valor de α permite incluir datos con un error de identificación menor o igual que α , mientras que el valor de β da la distancia euclidiana mínima entre los datos seleccionados, para evitar repetición de datos o datos muy parecidos.*

Observación 6. *Nótese que entre más grande sea α , mayor sera el error de estimación de la incertidumbre, mientras que β entre más grande sea su valor menor cantidad de datos se tomaran en cuenta; es decir, α indica que tan cercana se desea que sea la estimación, mientras β te dice la cantidad de datos que deseas obtener con ese error de estimación.*

La aplicación de la red neuronal estática a las simulaciones realizadas y la obtención del modelo neuronal para aproximar al sistema no lineal en una región de operación, se muestran en el capítulo de simulaciones.

IV. Simulaciones

El presente capítulo presenta una serie de simulaciones realizadas para demostrar la efectividad de la aplicación del teorema 3.1, además por medio de la obtención de la aproximación de la incertidumbre se trata de extraer la mayor información, es decir, en cada simulación el objetivo principal es la obtención de la aproximación de la parte de incertidumbre, pero luego de la obtención, se busca darle una interpretación para conocer información relevante de la aplicación del sistema.

Además se incluyen las simulaciones de la red neuronal estática para la obtención del modelo que aproxime al modelo no lineal con incertidumbre dentro de una región de operación en cada una de las primeras tres simulaciones realizadas para mostrar la efectividad del teorema 3.1.

4.1. Simulación 1: Identificación e Interpretación de la Fricción en un Manipulador de Eslabón Simple

4.1.1. Planteamiento del Problema

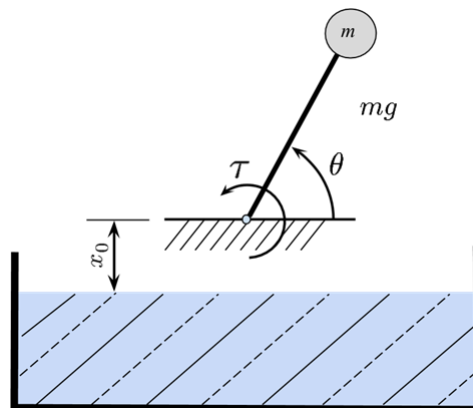


Figura 4.1: Sistema de un eslabón con cambio de fricción.

Considere el sistema no lineal correspondiente a la ecuación diferencial de un eslabón simple:

$$\ddot{\theta}(t) = -\frac{1}{ml^2}\sigma\left(\theta(t), \dot{\theta}(t)\right) - \frac{g}{l} \cdot \cos(\theta(t)) + \frac{1}{ml^2}\tau, \quad (4.1)$$

donde m es la masa del punto terminal del eslabón, l es la longitud del eslabón, g es la constante de gravedad, τ es el par de entrada aplicado a la articulación del eslabón, finalmente $\sigma\left(\theta(t), \dot{\theta}(t)\right)$ es la fricción del sistema, cuya función es la siguiente:

$$\sigma\left(\theta(t), \dot{\theta}(t)\right) = \begin{cases} k_b \dot{\theta}(t), & l \cdot \cos(\theta(t)) \leq -x_0 \\ k_a \dot{\theta}(t), & l \cdot \cos(\theta(t)) > -x_0 \end{cases}, \quad (4.2)$$

La interpretación física del sistema dinámico (4.1) se muestra en la Figura 4.1, la cual muestra al eslabón simple y debajo está a x_0 metros de distancia de la articulación el nivel de líquido de un tanque, así pues, de acuerdo a la variable de articulación $\theta(t)$, el punto terminal del eslabón puede estar o no sumergido en el líquido.

La función $\sigma(\cdot, \cdot)$ representa la fricción conmutada que tiene el sistema debida al cambio del medio del punto terminal, de modo que k_a es el coeficiente de fricción de la articulación y del medio que rodea al punto terminal cuando éste se encuentra fuera del líquido y k_b es el coeficiente de fricción viscosa correspondiente a la articulación y a la correspondiente del medio líquido que rodea al punto terminal.

Se considera que la fricción es la única incertidumbre del sistema, y esta simulación consiste primeramente estimar la fricción del sistema mediante redes neuronales dinámicas y excitando al sistema con un par que haga girar el eslabón y posteriormente a partir del perfil de la fricción estimada, estimar la distancia x_0 .

4.1.2. Objetivos de la Simulación

Partiendo de la condición inicial $\theta(0) = \frac{\pi}{2}$, $\dot{\theta}(0) = 0$ y cuando el eslabón haya dado dos revoluciones se estime la distancia x_0 . Además se sabe que existe un intervalo de θ en el cual el punto terminal del eslabón queda sumergido, es decir, $-l < x_0 < l$.

4.1.3. Desarrollo de la Simulación

De acuerdo con el resultado principal de esta tesis, considerando una Red Neuronal Dinámica de la forma (3.4), y ajustando nuestro sistema no lineal (4.1 y 4.2) considerando la fricción como la incertidumbre del sistema tenemos que:

$$\begin{aligned} \theta(t) &= x_1(t) \ , \\ \dot{\theta}(t) &= \dot{x}_1(t) = x_2(t) \ , \\ \ddot{\theta}(t) &= \ddot{x}_1(t) = \dot{x}_2(t) \ , \\ u &= \tau \ , \end{aligned} \quad (4.3)$$

así que por lo tanto el modelo del Manipulador de Eslabón Simple en espacio de estados que cumple con la forma (3.3), está dado por

$$\dot{x}(t) = \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-1}{ml^2} \sigma(\theta(t), \dot{\theta}(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u , \quad (4.4)$$

donde

$$\begin{aligned} f_0(x) &= \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} , \\ \Delta_f(x) &= \begin{bmatrix} 0 \\ \frac{-1}{ml^2} \sigma(\theta(t), \dot{\theta}(t)) \end{bmatrix} , \\ g(x) &= \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} , \end{aligned} \quad (4.5)$$

la red neuronal dinámica de la forma (3.4) está dada por

$$\dot{\hat{x}}(t) = \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + W\varphi(\hat{x}) + \psi + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u + L(x - \hat{x}) , \quad (4.6)$$

donde por el teorema 3.1 selecciona al umbral neuronal ψ como

$$\psi = \text{sign}(e) \cdot \delta_f(x) , \quad (4.7)$$

donde

$$\delta_f(x) = \frac{k \cdot |\hat{\theta}|}{ml^2}, k > k_b > k_a , \quad (4.8)$$

y finalmente la estructura de la neurona se selecciona

$$\varphi(\hat{x}) = \hat{x} + 0.1 \cdot \tanh(0.5\hat{x}) , \quad (4.9)$$

La siguiente simulación fue realizada con los datos de la Tabla 4.1.

Variable	Valor	Descripción
m	0.2 Kg	Masa del punto terminal del eslabón
l	0.2 m	Longitud del eslabón
T	5	Par de Entrada
k_a	0.5	El coeficiente de fricción de la articulación y del medio que rodea al punto terminal cuando éste se encuentra fuera del líquido.
k_b	2	El coeficiente de fricción viscosa de la articulación y del medio líquido que rodea al punto terminal.
k	2.1	Constante de la Cota Superior.
g	9.81	Constante de Gravedad
x_0	0.05	Distancia a estimar

Tabla 4.1: Datos usados en la Simulación 1.

En la Figura 4.2 se muestra el comportamiento de la posición angular del eslabón, además en ella se indica el momento en que se completan la primera y segunda revolución, además analizando la evolución del crecimiento de la posición se puede observar que la gráfica tiene cambios de pendiente, lo que se puede interpretar como la entrada y salida del eslabón del líquido.

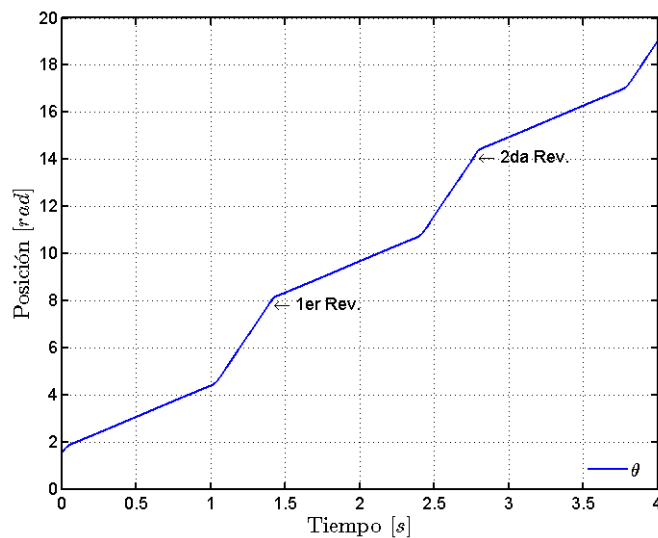


Figura 4.2: Posición Angular del Sistema.

En la Figura 4.3 se puede observar claramente la evolución de la velocidad del eslabón, debido a la disminución de la velocidad del eslabón y luego al aumento de nueva cuenta de la misma, se pueden deducir los cambios del medio que rodea al punto terminal mediante estos cambios de velocidad, es decir, cuando la velocidad disminuye es porque el eslabón está entrando en el líquido, mientras que cuando aumenta sale del líquido y entra en el aire.

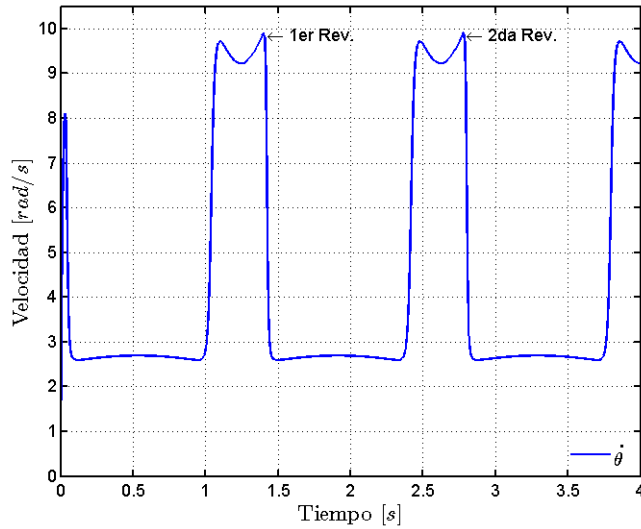


Figura 4.3: Velocidad Angular.

La Figura 4.4, es la referente a la estimación realizada por la red neuronal de la posición angular del eslabón, se observa que descartando el inicio del perfil, tiene un comportamiento similar a la Figura 4.2.

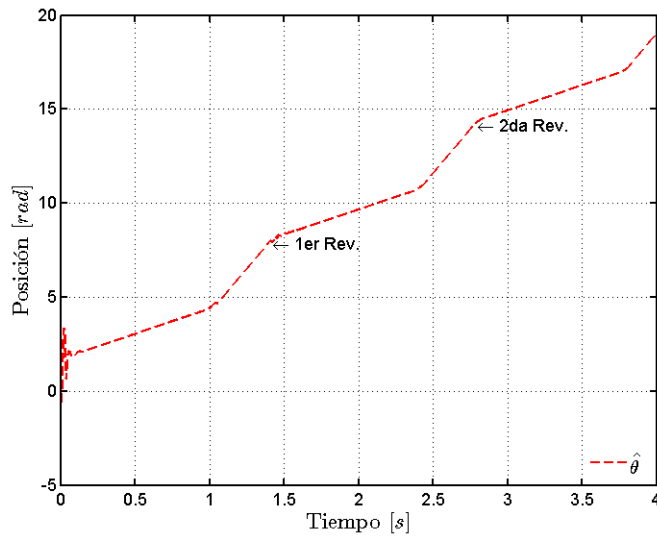


Figura 4.4: Posición Angular Estimada.

Mientras que la Figura 4.5 representa la estimación de la velocidad angular realizada por la red neuronal, en ella también se puede observar el comportamiento que refleja el cambio de ambiente que sufre el eslabón simple.

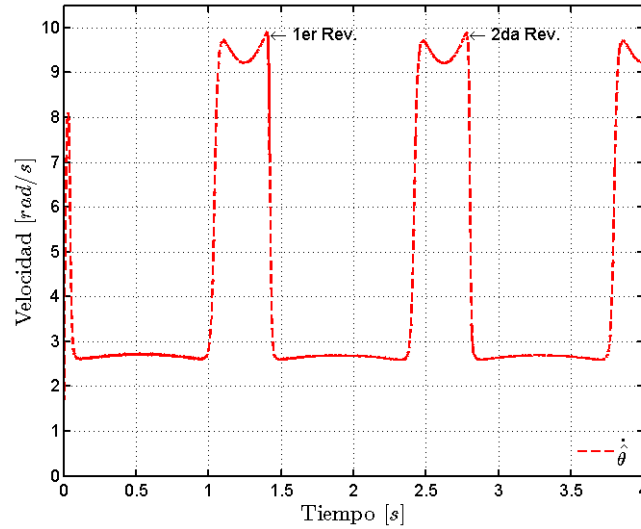


Figura 4.5: Velocidad Angular Estimada.

En la Figura 4.6 podemos observar que ambas gráficas, tanto la posición angular real y la posición angular estimada, presentan el mismo patrón de crecimiento, además de que se observa que la red neuronal arroja una aproximación bastante apegada a la posición angular del modelo.

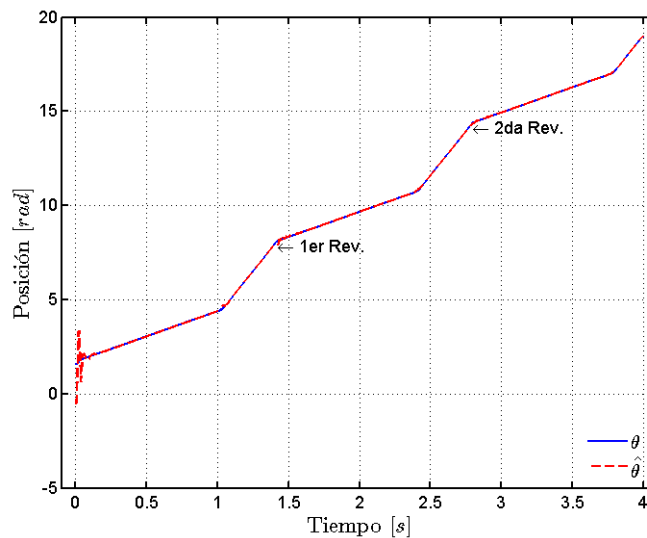


Figura 4.6: Comparativa entre la Posición Angular del Sistema y la Posición Angular Estimada.

En la Figura 4.7 se puede observar el error de estimación entre la posición angular del sistema real y la posición angular estimada, el cual es pequeño sin considerar la etapa de entrenamiento de la red, por lo que se considera una aproximación apropiada.

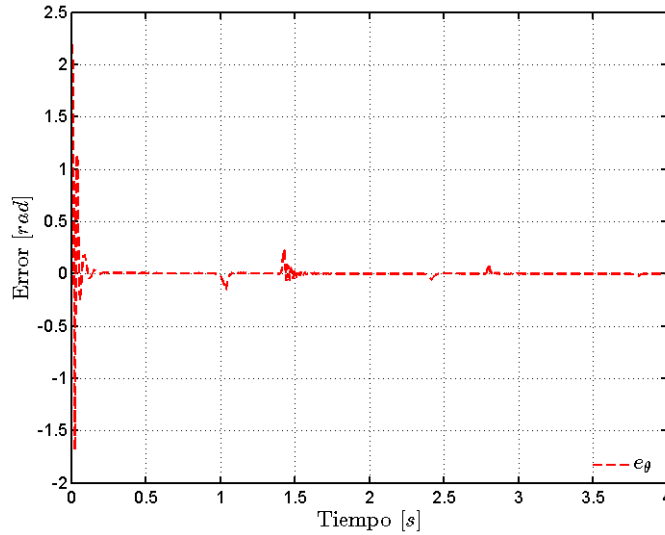


Figura 4.7: Error de Estimación de la Posición Angular.

En la Figura 4.8, puede observar la comparación entre la velocidad angular real y la velocidad angular estimada por la red neuronal.

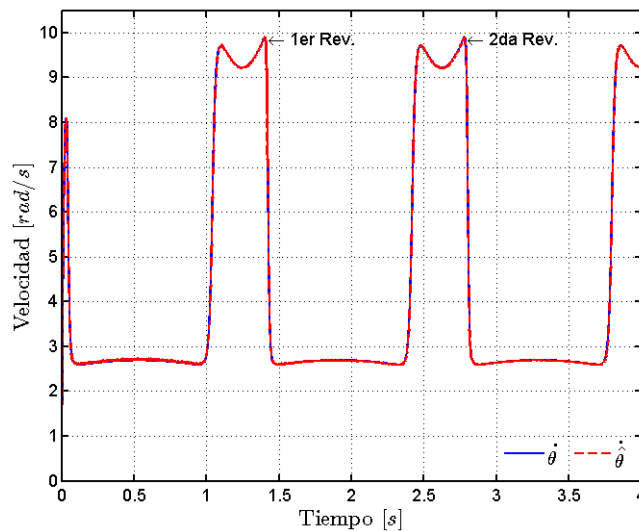


Figura 4.8: Comparativa entre la Velocidad Angular del Sistema y Velocidad Angular Estimada.

En la Figura 4.9 se puede ver que el error de estimación que existe entre la velocidad angular real y la estimada, el cual es relativamente pequeño, se puede observar que dicho error va ajustándose durante la etapa de entrenamiento de la red.

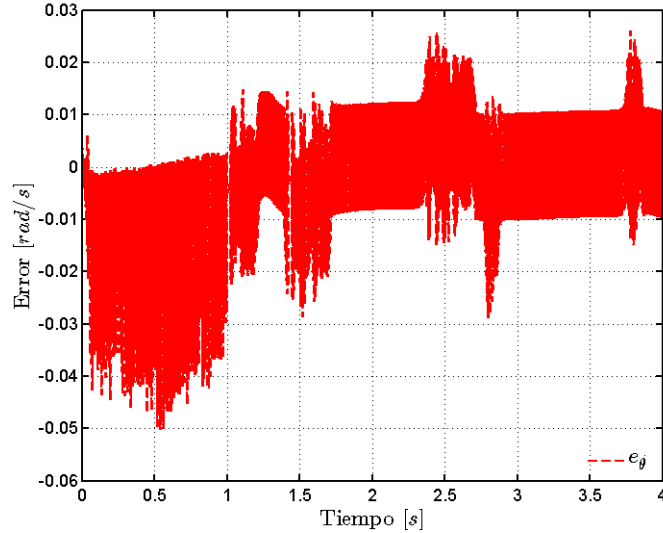


Figura 4.9: Diferencia entre la velocidad angular y la velocidad angular estimada.

Durante la obtención de los datos para realizar las Figuras anteriores, se estima la incertidumbre del sistema es decir, la fricción tanto del aire como del líquido en el que se sumerge el eslabón. Primeramente si el error de identificación fuera prácticamente 0 tendríamos que su derivada también lo sería por lo que:

$$e \cong 0 \quad , \quad (4.10)$$

$$\begin{aligned} \dot{e} &\cong \Delta_f(x) - W\varphi(\hat{x}) - \psi - Le \quad , \\ 0 &\cong \Delta_f(x) - W\varphi(\hat{x}) - \psi - Le \quad , \end{aligned} \quad (4.11)$$

entonces por las ecuaciones (3.5), (3.6) y (3.7) del teorema 3.1

$$\begin{aligned} Le &\cong 0 \quad , \\ \psi &\cong \text{sign}(0) \cdot \delta_f \cong 0 \quad , \end{aligned} \quad (4.12)$$

tenemos que la incertidumbre del sistema $\Delta_f(x)$ es aproximadamente igual a la componente neuronal $W\varphi(\hat{x})$

$$\begin{aligned} \Delta_f(x) - W\varphi(\hat{x}) &\cong 0 \quad , \\ \Delta_f(x) &\cong W\varphi(\hat{x}) \quad , \end{aligned} \quad (4.13)$$

de (4.13) tenemos que el error de estimación de la incertidumbre es igual a la resta de Δ_f menos

la componente neuronal $W\varphi(\hat{x})$, es decir

$$e_{\Delta_f} \cong \Delta_f(x) - W\varphi(\hat{x}) \ ,$$

$$e_{\Delta_f} = \begin{bmatrix} e_{\Delta_{f1}} \\ e_{\Delta_{f2}} \end{bmatrix} . \tag{4.14}$$

En la Figura 4.10 se muestra el perfil de la incertidumbre del sistema, en dicho perfil se puede ver cuando es que el eslabón cambia de ambiente

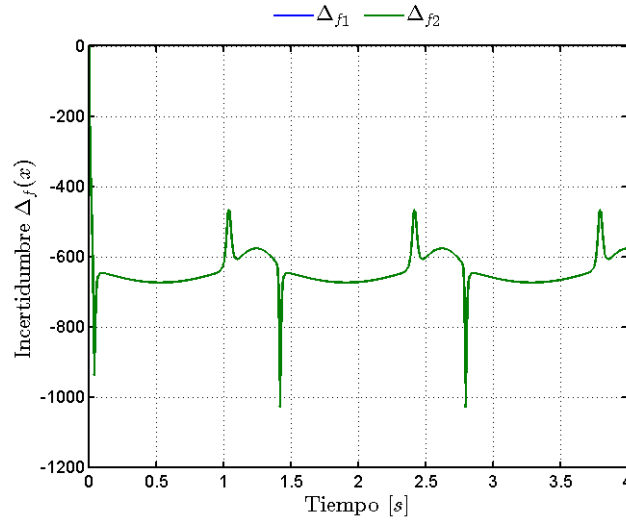


Figura 4.10: Incertidumbre del Eslabón Simple.

Mientras que en la Figura 4.11 se muestra el perfil de la componente neuronal $W\varphi(\hat{x})$ arrojado por la red neuronal dinámica.

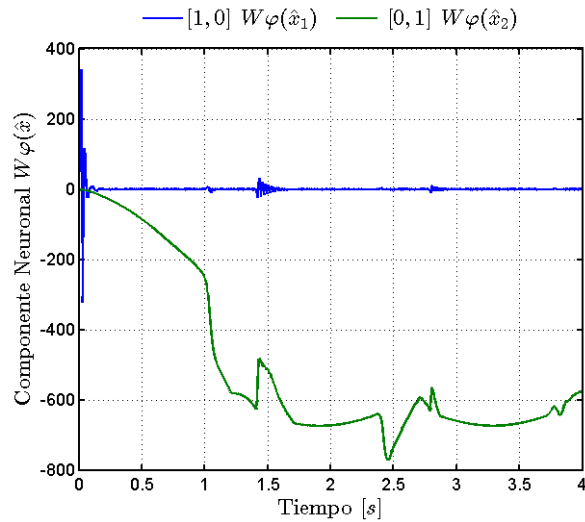


Figura 4.11: Componente Neuronal $W\varphi(\hat{x})$.

En la Figura 4.12 se muestran los perfiles de la incertidumbre del sistema Δ_f y de la componente neuronal $W\varphi(\hat{x})$; se observa que durante la etapa de entrenamiento de red la componente neuronal busca ajustarse para parecerse a la incertidumbre del sistema, además se puede hacer la deducción de que entre mayor sea el tiempo de simulación, más cercana será la aproximación.

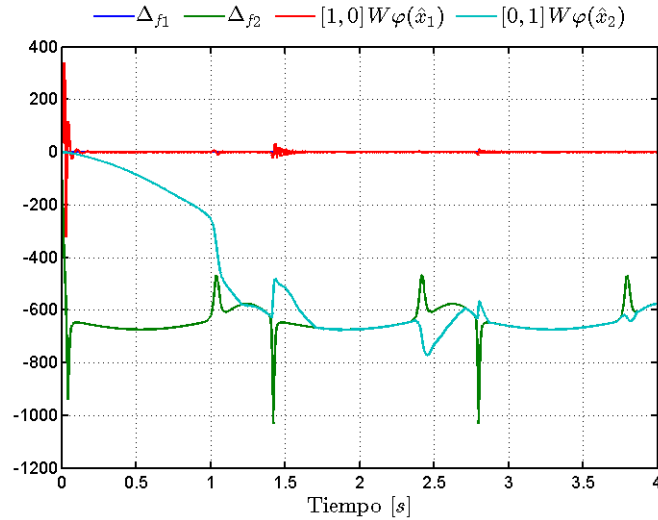
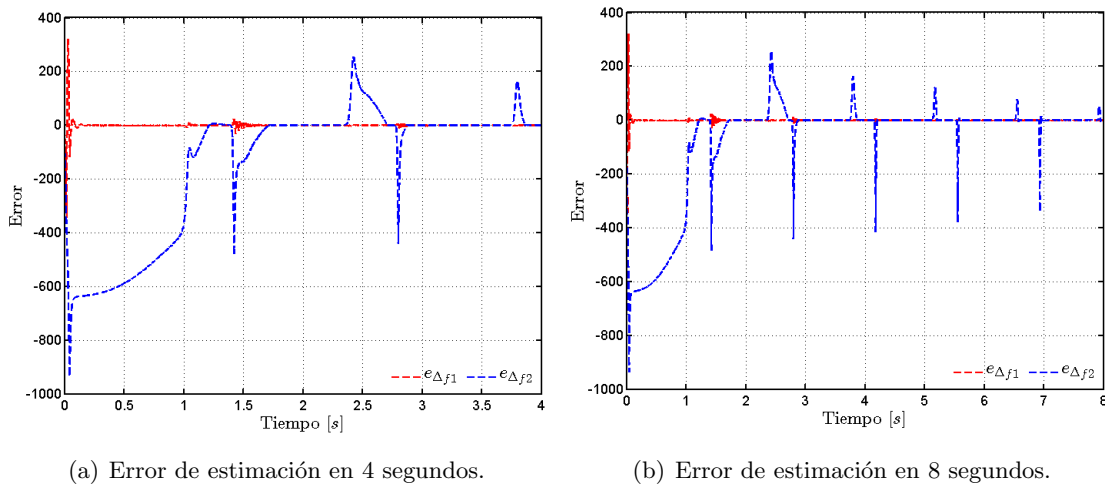


Figura 4.12: Comparación de Δ_f y $W\varphi(\hat{x})$.

Lo anterior puede ser observado en la Figura 4.13, en la cual se puede ver la convergencia del error de estimación de la incertidumbre durante dos intervalos de tiempo, el primero en 4 segundos y el segundo en 8 segundos.



(a) Error de estimación en 4 segundos.

(b) Error de estimación en 8 segundos.

Figura 4.13: Error de Estimación de la Incertidumbre del Sistema.

Es decir con las Figuras 4.12 y 4.13, se observa claramente que la componente neuronal $W\varphi(\hat{x})$ lleva a cabo la neuro-identificación de $\Delta_f(x)$.

En la Figura 4.14 se observa la evolución del error cuadrático medio a lo largo del tiempo, donde resaltan los cambios provocados por el aprendizaje de la red, al principio se observa como el error es grande debido a la etapa de entrenamiento de la red, pero luego se observa como el error cuadrático medio aumenta muy poco, finalizando con un valor de 8.5860×10^4

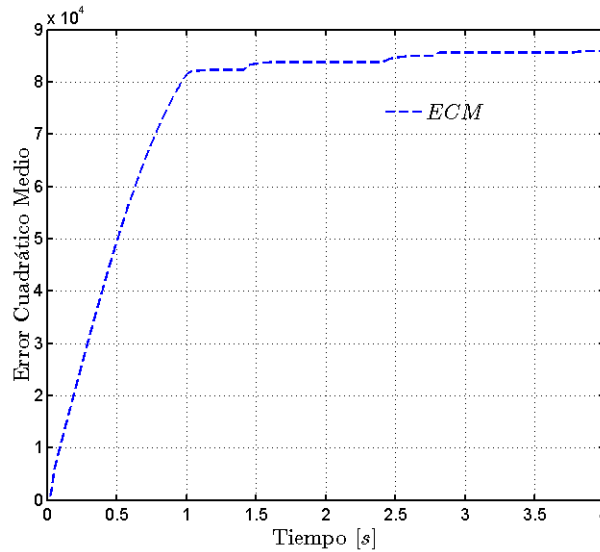


Figura 4.14: Error Cuadrático Medio a lo largo del tiempo de simulación.

En la Figura 4.15, se puede observar el comportamiento de los pesos sinápticos de la red neuronal, donde se nota el ajuste de los pesos para que la componente neuronal $W\varphi(\hat{x})$ se aproxime a la incertidumbre del sistema.

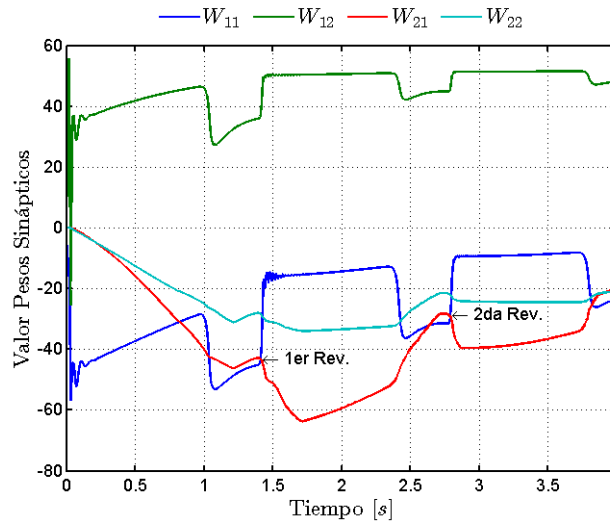


Figura 4.15: Pesos Sinápticos de la Red Neuronal Dinámica.

En la Figura 4.16 se puede observar la evolución $l \cdot \cos(\theta)$, puede observarse claramente cuando el eslabón cambia de ambiente.

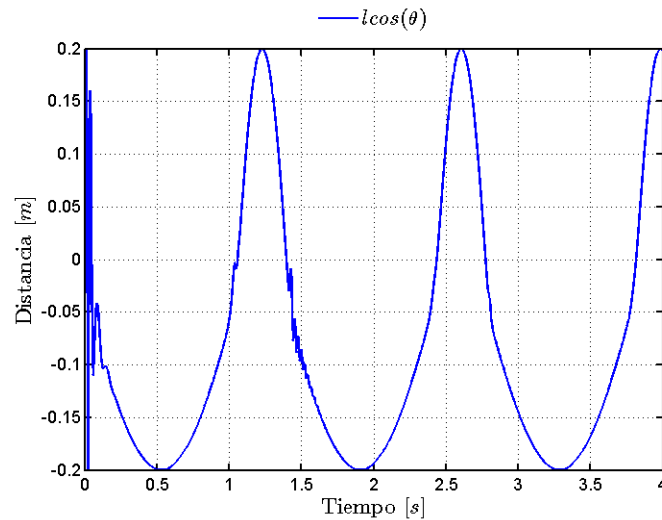


Figura 4.16: Perfil de $l \cos(\theta)$.

La Figura 4.17 representa el perfil de la fricción estimada por la red neuronal, la cual está incluida en la incertidumbre del sistema, donde puede observarse que durante la etapa de aprendizaje de la red, tiene un comportamiento anormal, luego es una especie de comportamiento periódico entre los valores de k_a y k_b .

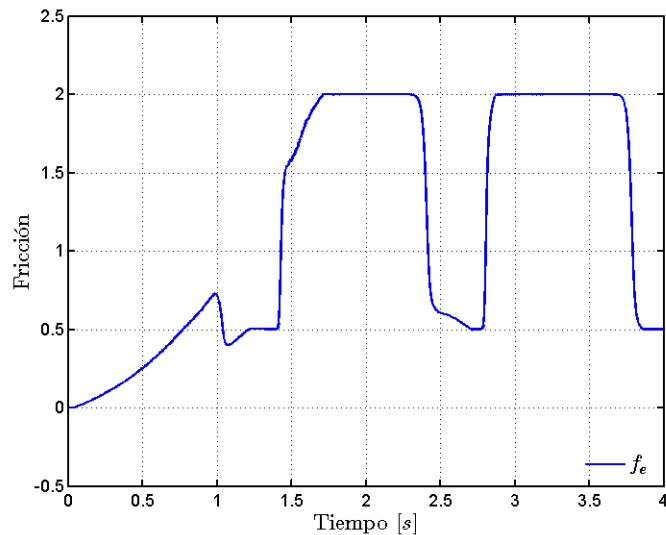


Figura 4.17: Fricción Estimada.

La Figura 4.18 muestra la localización del valor de \hat{x}_0 en la segunda revolución.

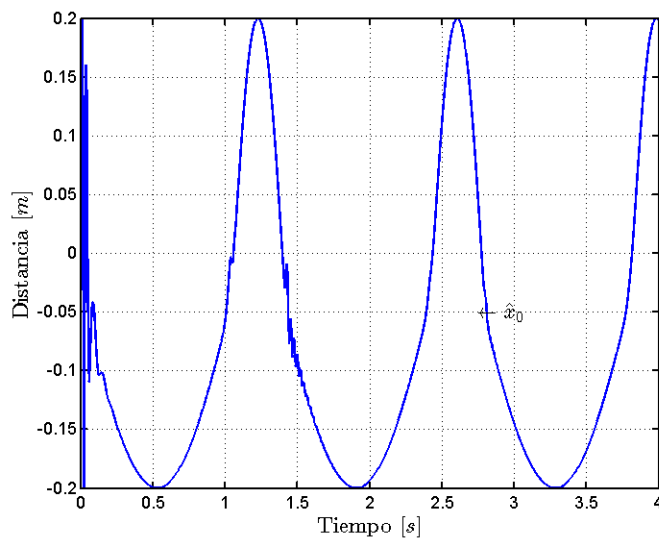


Figura 4.18: Localización de \hat{x}_0

La Figura 4.19 muestra la distancia estimada calculada a partir de $l \cos(\theta)$ y el perfil de la fricción estimada y la distancia real que existe entre el eslabón y el líquido.

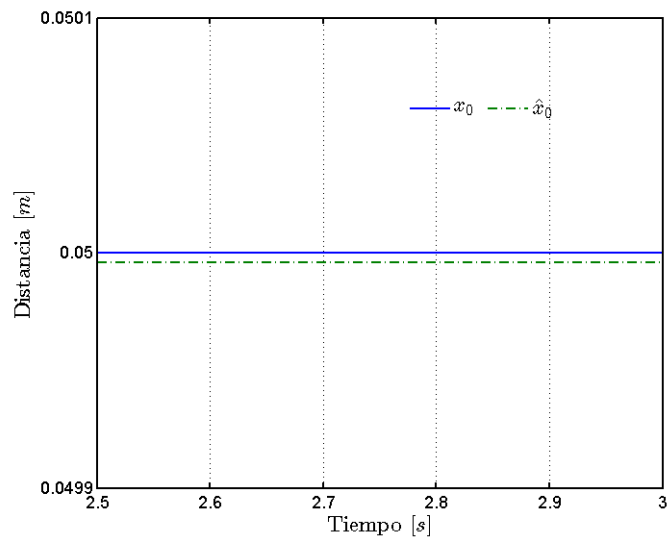


Figura 4.19: Distancia Estimada.

Con lo anterior se demuestra que a partir de la identificación de la incertidumbre se puede obtener una mayor cantidad de información del sistema, en este caso, además de poder extraer el perfil de fricción del eslabón de la incertidumbre estimada, también es posible estimar la distancia existente entre el eje del eslabón simple y el líquido donde se sumerge.

4.2. Simulación 2: Estimación de Una Fuga en un par de Tanques de Área Transversal no uniforme conectados en Cascada

4.2.1. Planteamiento del Problema

Considere las siguientes ecuaciones diferenciales

$$\sum_{S2T}: \begin{cases} \frac{d}{dt} V_1(t) = q_{i1}(t) - q_{o1}(t) \\ \frac{d}{dt} V_2(t) = q_{o1}(t) - q_O(t) \\ y = q_{o2} \end{cases} \quad (4.15)$$

donde el volumen del primer tanque es $V_1(x_1) = \int_0^{x_1} A_1(x) dx$, el volumen del segundo tanque es $V_2(x_2) = \int_0^{x_2} A_2(x) dx$, el área transversal del primer tanque a una altura del nivel del líquido x_1 es $A_1(x_1) = r_{o1} + \delta_{r1} \text{sen}(\frac{2\pi x_1}{0.7})$ y para el segundo tanque el área transversal a una altura del nivel del líquido x_2 es $A_2(x_2) = r_{o2} + \delta_{r2} \text{sen}(\frac{2\pi x_2}{0.45})$.

El flujo de entrada del segundo tanque es igual al flujo de salida del primero, es decir, $q_{i2} = q_{o1} = \frac{x_1}{R_1}$; el flujo neto de salida del segundo tanque es $q_O = q_{o2} + q_{f2} = \frac{x_2}{R}$, donde $R = \frac{1}{\frac{1}{R_2} + \frac{1}{R_{f2}}}$.

La interpretación física del sistema dinámico se muestra en la Figura 4.20

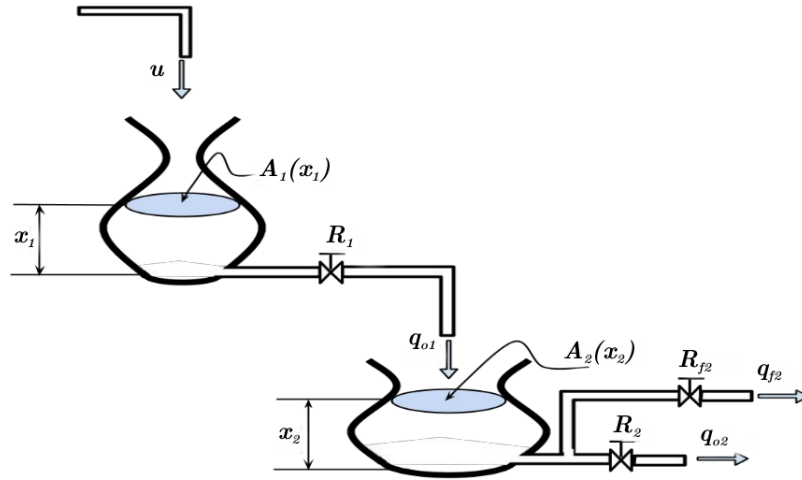


Figura 4.20: Esquema general del Sistema de tanques de área transversal no uniforme.

Se considera que la única incertidumbre del sistema es el desconocimiento del valor de R_{f2} , solo se conoce que la válvula de R_{f2} nunca está cerrada en su totalidad, así que siempre existe fuga, con lo cual la lectura del flujo de salida correspondiente a la representación de la fuga no se tiene, es decir, no se tiene lectura de q_{f2} , pero si se tiene lectura de q_{o2} .

4.2.2. Objetivos de la Simulación

La simulación consiste estimar el perfil de la fuga en el intervalo $T_{ini} \leq t \leq T_{fin}$.

Partiendo de la condición inicial $x_1(0) = x_2(0) = 0$, una señal de entrada

$$u(t) = \begin{cases} 0.25t, & 0 \leq t \leq 8 \\ 2 + 0.5\text{sen}(\frac{\pi t}{2}), & t > 8 \end{cases}, \quad (4.16)$$

se debe estimar el perfil de la fuga en un intervalo de tiempo $0 < T_{ini} \leq t \leq T_{fin} < 70$.

4.2.3. Desarrollo de la Simulación

Tomando como punto de partida el resultado principal de la presente tesis, es decir, considerando una Red Neuronal Dinámica de la forma (3.4) para realizar la identificación en línea de la incertidumbre del sistema (4.15).

El modelo del sistema utilizado para esta simulación en su representación en espacio de estados que cumple con la forma (3.3) es

$$\dot{x}(t) = \begin{bmatrix} -\frac{\frac{1}{R_1}}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} & 0 \\ \frac{\frac{1}{R_1}}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} & -\frac{1}{[r_{02} + \delta_{r2}\text{sen}(\frac{2\pi}{0.45}x_2)]R_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{[r_{02} + \delta_{r2}\text{sen}(\frac{2\pi}{0.45}x_2)]R_{f2}}x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} \\ 0 \end{bmatrix} u, \quad (4.17)$$

donde

$$\begin{aligned} f_0(x) &= \begin{bmatrix} -\frac{\frac{1}{R_1}}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} & 0 \\ \frac{\frac{1}{R_1}}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} & -\frac{1}{[r_{02} + \delta_{r2}\text{sen}(\frac{2\pi}{0.45}x_2)]R_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ \Delta_f(x) &= \begin{bmatrix} 0 \\ -\frac{1}{[r_{02} + \delta_{r2}\text{sen}(\frac{2\pi}{0.45}x_2)]R_{f2}}x_2 \end{bmatrix}, \\ g(x) &= \begin{bmatrix} \frac{1}{r_{01} + \delta_{r1}\text{sen}(\frac{2\pi}{0.7}x_1)} \\ 0 \end{bmatrix}, \end{aligned} \quad (4.18)$$

además se sabe que $\underline{R_{f2}} \leq R_{f2} \leq \overline{R_{f2}}$.

Por lo que considerando una red neuronal dinámica de la forma (3.4) tenemos que:

$$\dot{\hat{x}}(t) = \begin{bmatrix} -\frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & 0 \\ \frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & -\frac{1}{[r_{02} + \delta_{r2} \text{sen}(\frac{2\pi}{0.45} x_2)] R_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + W\varphi(\hat{x}) + \psi + \begin{bmatrix} \frac{1}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} \\ 0 \end{bmatrix} u + L(x - \hat{x}) , \quad (4.19)$$

de acuerdo con el teorema 3.1 seleccionamos al umbral neuronal ψ como

$$\psi = \text{sign}(e) \cdot \delta_f(x) , \quad (4.20)$$

con la cota superior de la norma de la incertidumbre como

$$\delta_f(x) = \left| \frac{1}{[r_{02} + \delta_{r2} \text{sen}(\frac{2\pi}{0.45} x_2)] \underline{R}_{f2}} \right| , \quad (4.21)$$

y función de activación.

$$\varphi(\hat{x}) = \hat{x} + 0.05 \cdot \tanh(0.02\hat{x}). \quad (4.22)$$

La siguiente simulación fue realizada con los datos de la Tabla 4.2.

Variable	Valor	Descripción
R_1	0.5	Resistencia de la Válvula 1
R_2	0.8	Resistencia de la Válvula 2
r_{01}	10	Constante.
r_{02}	8	Constante.
δ_{r1}	0.8	Constante.
δ_{r2}	0.5	Constante.
\underline{R}_{f2}	0.1	Resistencia Mínima de la Fuga
\overline{R}_{f2}	1	Resistencia Máxima de la Fuga

Tabla 4.2: Datos usados en la Simulación 2.

En la Figura 4.21 se observa el perfil del flujo de entrada del sistema dado por la ecuación (4.16).

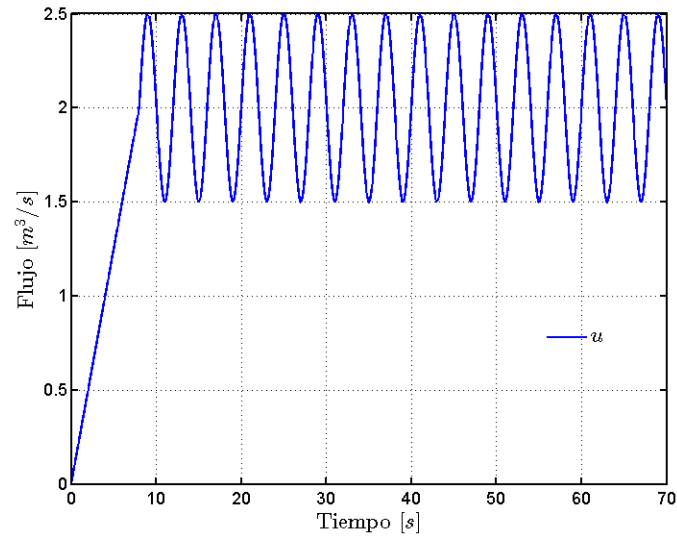


Figura 4.21: Perfil de Flujo de Entrada.

En la Figura 4.22 se muestran los niveles de los dos tanques.

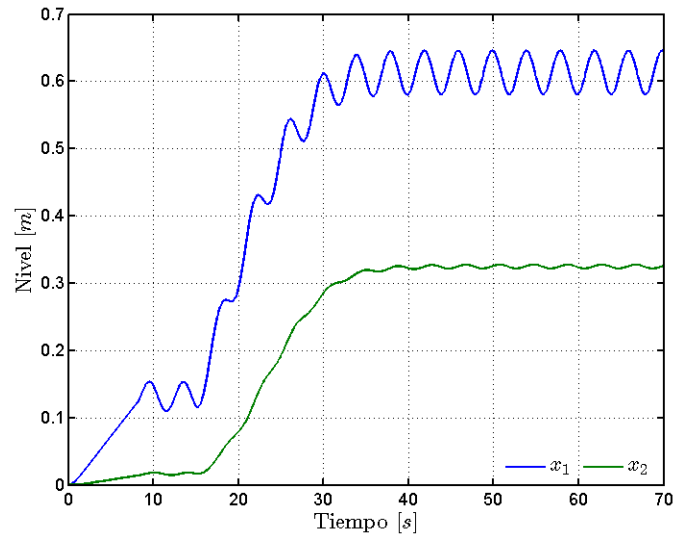


Figura 4.22: Niveles de los dos tanques.

En la Figura 4.23, se muestran los niveles estimados de los dos tanques arrojados por la red neuronal.

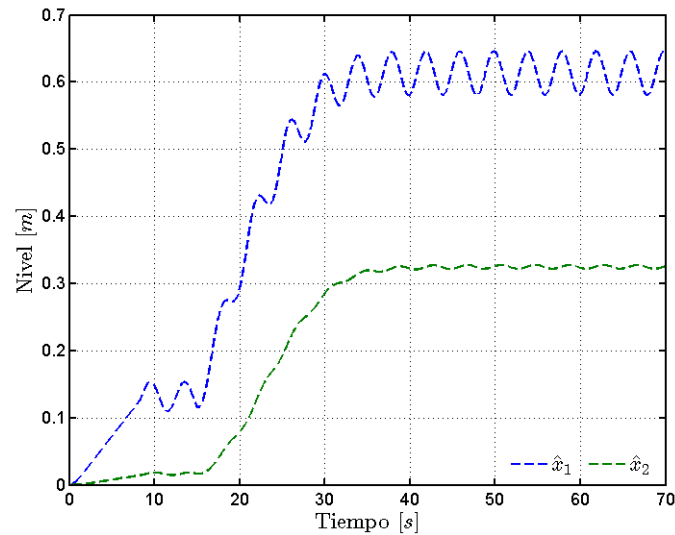


Figura 4.23: Niveles Estimados de los dos tanques.

A continuación se pueden observar una serie de gráficas comparativas entre los niveles reales y los estimados de los tanques, así como el error de estimación correspondiente a cada nivel.

La Figura 4.24 muestra la comparativa entre el nivel real y el nivel estimado para el tanque 1, se puede observar que ambos perfiles son muy similares.

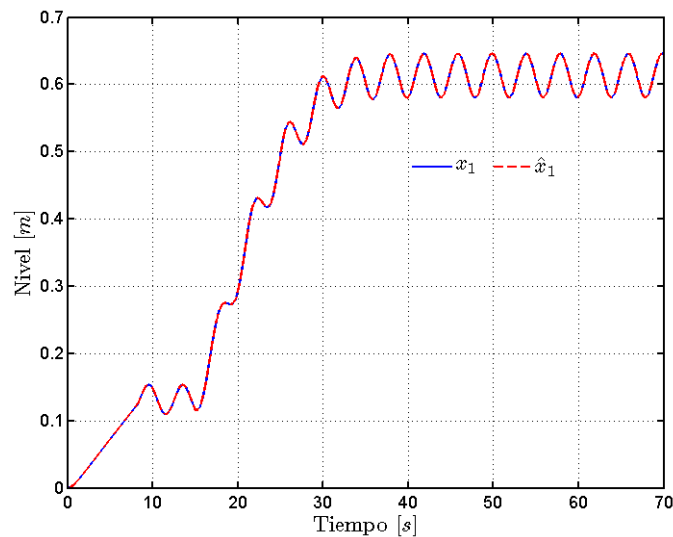


Figura 4.24: Comparativa entre Nivel y Nivel Estimado del tanque 1.

La Figura 4.25 muestra el error de estimación generado por el nivel real y el estimado del tanque 1, se puede observar que durante la etapa de entrenamiento de red, el error de estimación es mayor al generado después, además de observar la convergencia a un valor muy cercano a 0.

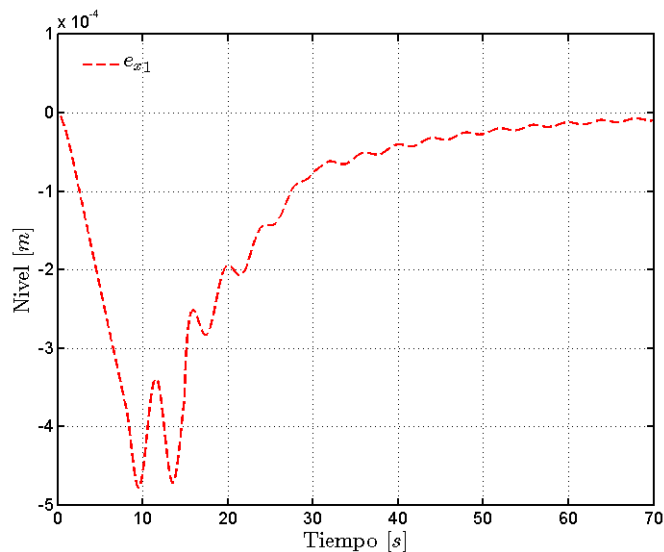


Figura 4.25: Error de Estimación del Nivel del tanque 1.

La Figura 4.26 muestra la comparativa entre el nivel real y el nivel estimado para el tanque 2, con perfiles similares, se considera una buena aproximación.

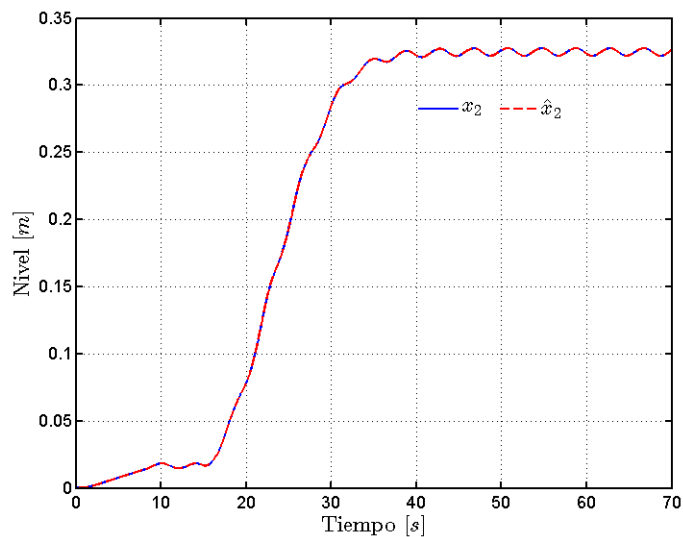


Figura 4.26: Comparativa entre Nivel y Nivel Estimado del tanque 2.

La Figura 4.27 muestra el error de estimación generado por el nivel real y el estimado del tanque 2, se puede observar la convergencia del error a un valor cercano a 0.

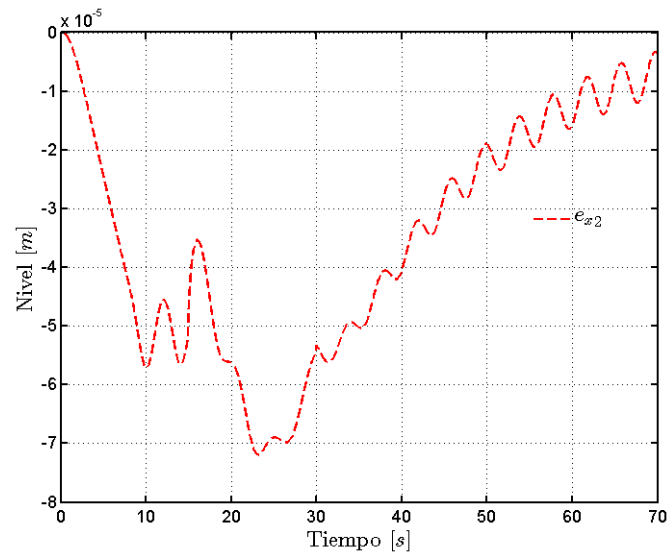


Figura 4.27: Error de Estimación del Nivel del tanque 2.

La Figura 4.28 representa el perfil del flujo de salida del tanque 2.

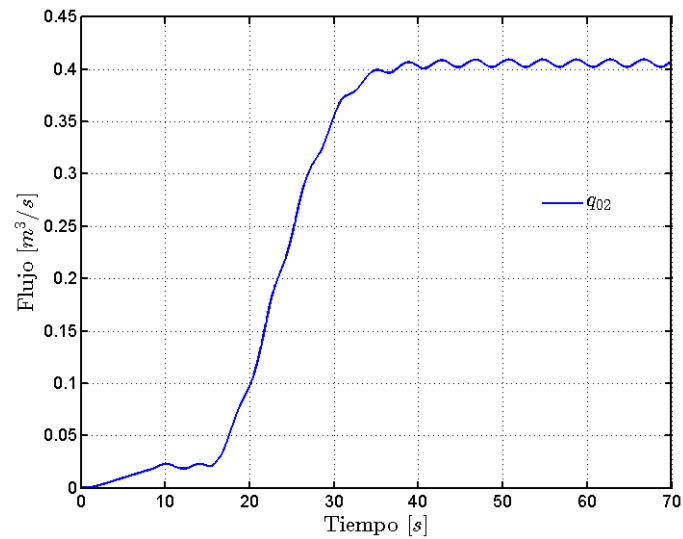


Figura 4.28: Flujo de Salida del tanque 2.

La Figura 4.29 representa el perfil del flujo de salida estimado del tanque 2.

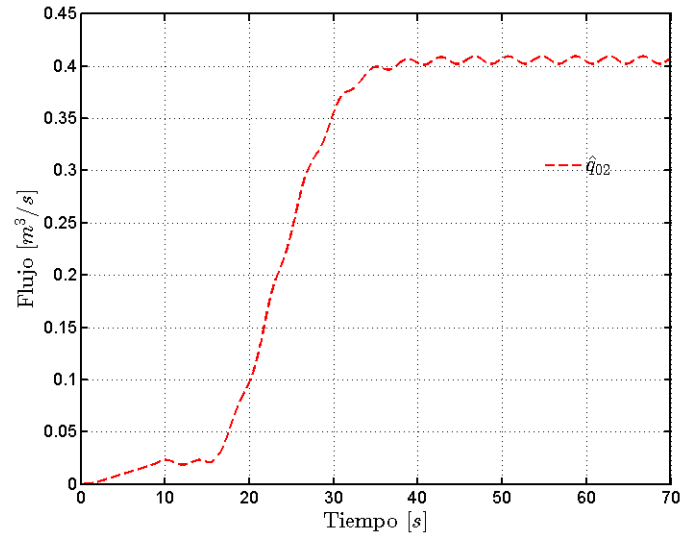


Figura 4.29: Flujo de Salida Estimado del tanque 2.

La Figura 4.30 es una comparativa entre el flujo de salida y el flujo estimado de salida del tanque 2, donde se puede observar que lo arrojado por la red neuronal se considera una buena una aproximación.

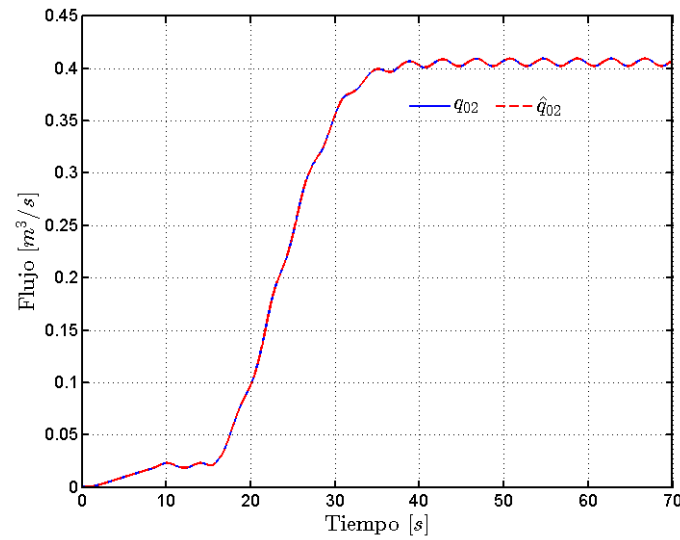


Figura 4.30: Comparativa de Flujos de Salida Real y Estimado del tanque 2.

La Figura 4.31 representa el error de estimación del perfil de salida del tanque 2, donde se observa de nueva cuenta que el error trata de converger a un valor cercano a 0.

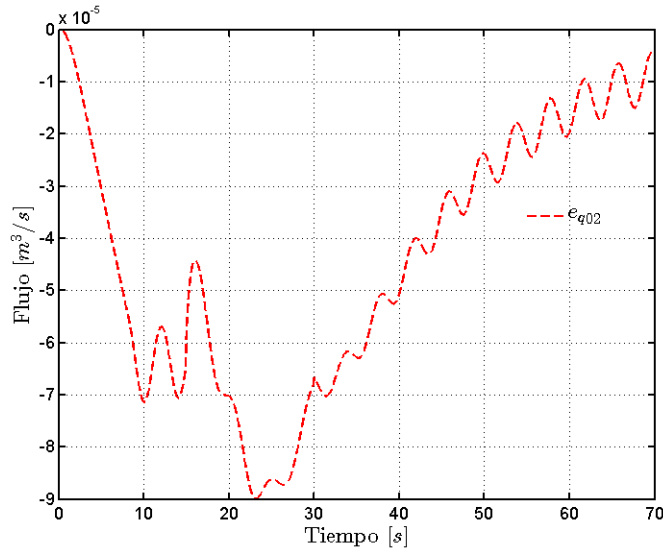


Figura 4.31: Error de Estimación del Flujo de Salida del tanque 2.

Durante la obtención de los datos para realizar las Figuras anteriores, se estima la incertidumbre del sistema es decir, el perfil de la fuga mostrado en la Figura (4.32)

Luego de la realización de la simulación, se tiene que si el error de identificación es prácticamente 0 tendríamos que su derivada también lo sería por lo que:

$$e \cong 0 \quad , \quad (4.23)$$

$$\begin{aligned} \dot{e} &\cong \Delta_f(x) - W\varphi(\hat{x}) - \psi - Le \quad , \\ 0 &\cong \Delta_f(x) - W\varphi(\hat{x}) - \psi - Le \quad , \end{aligned} \quad (4.24)$$

entonces por las ecuaciones (3.5), (3.6) y (3.7) del Teorema 3.1

$$\begin{aligned} Le &\cong 0 \quad , \\ \psi &\cong \text{sign}(0) \cdot \delta_f \cong 0 \quad , \end{aligned} \quad (4.25)$$

tenemos que la incertidumbre del sistema $\Delta_f(x)$ es aproximadamente igual a la componente neuronal $W\varphi(\hat{x})$

$$\begin{aligned} \Delta_f(x) - W\varphi(\hat{x}) &\cong 0 \quad , \\ \Delta_f(x) &\cong W\varphi(\hat{x}) \quad , \end{aligned} \quad (4.26)$$

de la ecuación (4.26) tenemos que el error de estimación de la incertidumbre es igual a la resta de Δ_f menos la componente neuronal $W\varphi(\hat{x})$, es decir

$$e_{\Delta_f} \cong \Delta_f(x) - W\varphi(\hat{x}) ,$$

$$e_{\Delta_f} = \begin{bmatrix} e_{\Delta_{f1}} \\ e_{\Delta_{f2}} \end{bmatrix} . \quad (4.27)$$

La Figura 4.32 representa el perfil de la incertidumbre del sistema, lo cual nos ayudara a calcular el flujo de la fuga del tanque 2.

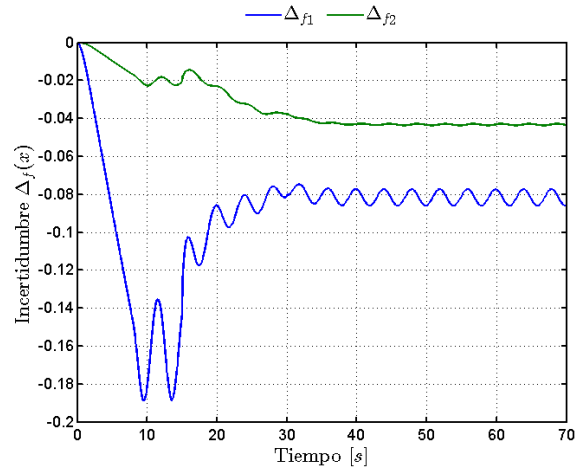


Figura 4.32: Incertidumbre del Sistema de Tanques.

La Figura 4.33 representa el perfil de la componente neuronal $W\varphi(\hat{x})$ arrojado por la red neuronal.

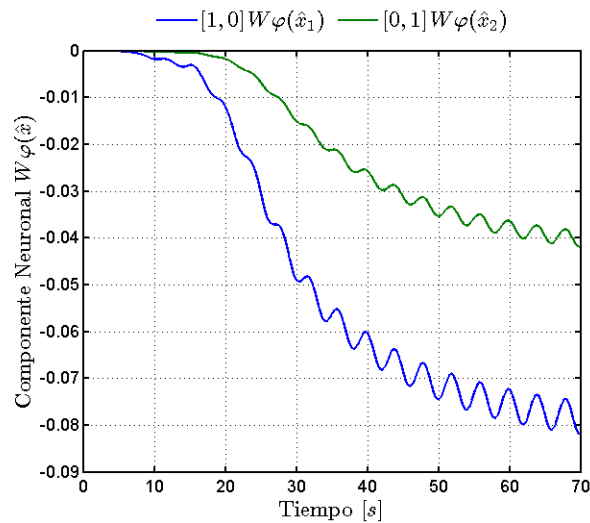


Figura 4.33: Componente Neuronal $W\varphi(\hat{x})$.

En Figura 4.34 se muestra la comparación de la Incertidumbre del sistema y la componente neuronal $W\varphi(\hat{x})$, donde se puede observar que el componente neuronal difiere bastante de la incertidumbre durante la etapa de entrenamiento de la red, pero conforme termina el entrenamiento se aproxima en mayor grado a la incertidumbre.

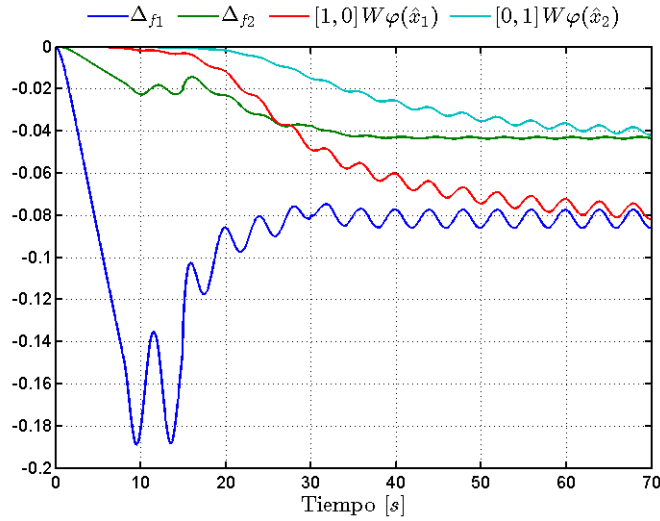


Figura 4.34: Comparación de Δ_f y $W\varphi(\hat{x})$.

La Figura 4.35 representa el error de estimación de la Incertidumbre del sistema, donde se puede observar que durante la etapa de entrenamiento el error es notable, pero conforme sale de la etapa de entrenamiento el error va convergiendo a un valor cercano a cero, por lo que se considera que la aproximación de la incertidumbre es buena conforme el tiempo avanza.

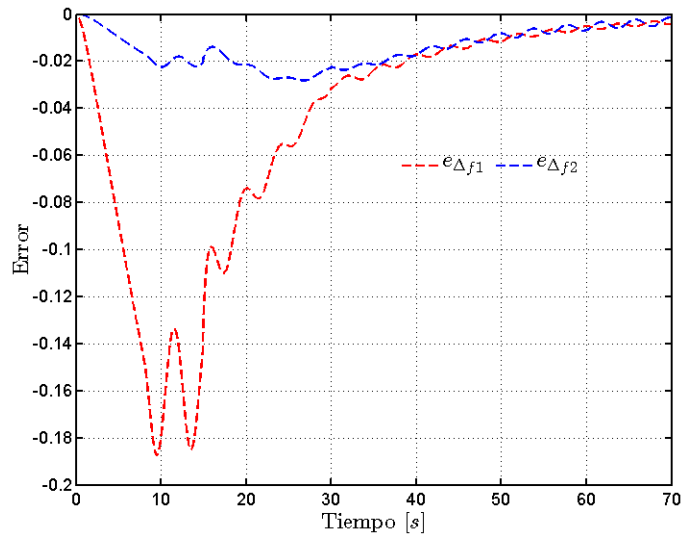


Figura 4.35: Error de Estimación la Incertidumbre del Sistema.

En la Figura 4.36 se observa la evolución del error cuadrático medio a lo largo del tiempo finalizando con un valor de 3.2662×10^{-8}

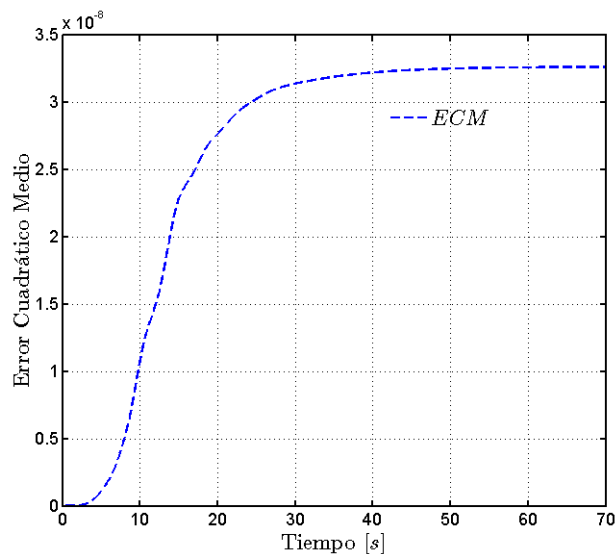


Figura 4.36: Error Cuadrático Medio a lo largo del tiempo de simulación.

En la Figura 4.37, se puede observar la evolución de los pesos sinápticos de la red neuronal.

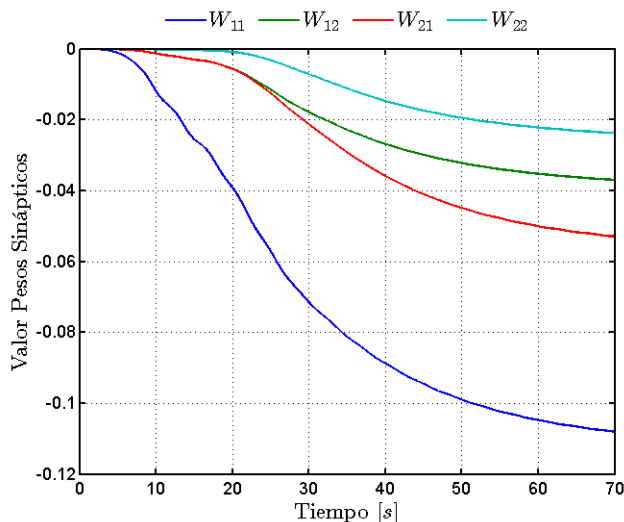


Figura 4.37: Pesos Sinápticos de la Red Neuronal Dinámica.

Con la estimación de la incertidumbre podemos calcular los perfiles de fuga real y la fuga estimada por la red neuronal, con lo que podemos observar que esta información nos puede ayudar a conocer una mayor información del sistema real.

La Figura 4.38 representa el perfil de la fuga en tanque 2, donde se puede observar claramente la variación de la válvula R_{f2} por medio de los cambios en el perfil de la fuga.

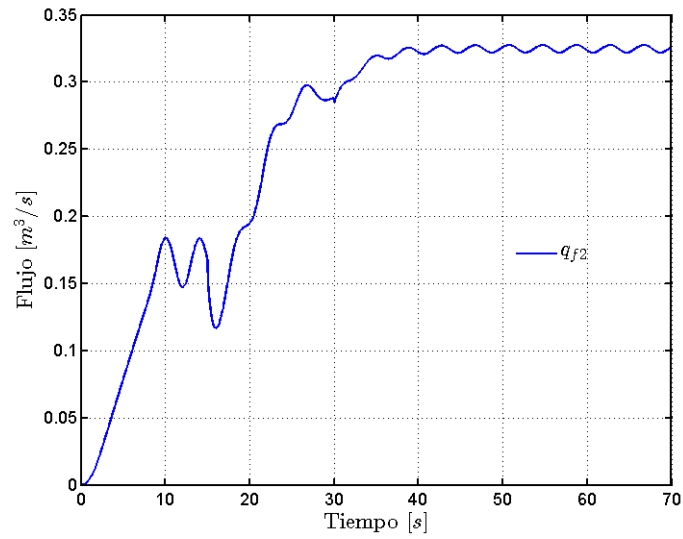


Figura 4.38: Flujo de la fuga del tanque 2.

La Figura 4.39 representa el perfil de la fuga estimado del tanque 2 arrojado por la red neuronal, donde se pueden observar de igual manera las modificaciones que sufre la válvula R_{f2} .

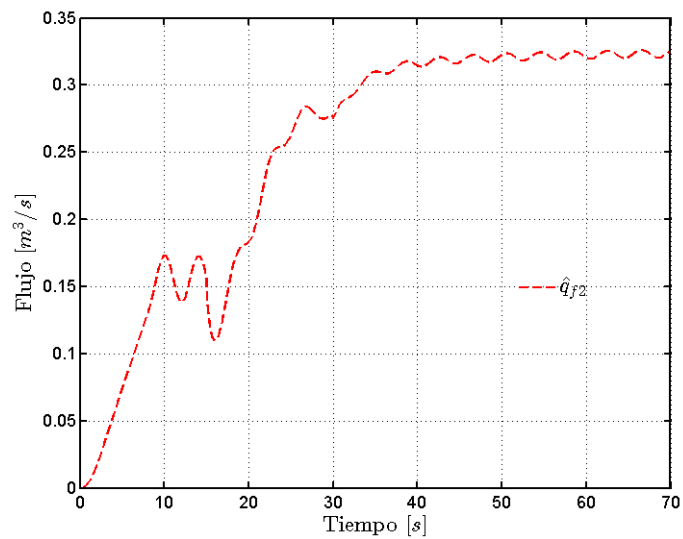


Figura 4.39: Flujo de la fuga Estimado del tanque 2.

La Figura 4.40 es una comparativa entre los perfiles de la fuga real y estimada del tanque 2, en ella se puede observar la cercanía de la estimación.

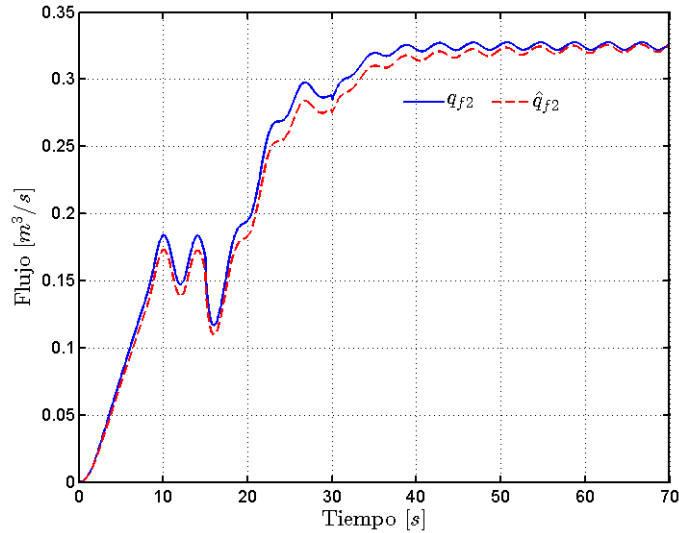


Figura 4.40: Comparativa de Flujos de Fuga Real y Estimado del tanque 2.

La Figura 4.41 representa el error de estimación del perfil de la fuga del tanque 2, donde se observa que durante la etapa de entrenamiento el error de estimación es mayor, y conforme avanza el tiempo converge a un valor cercano a cero.

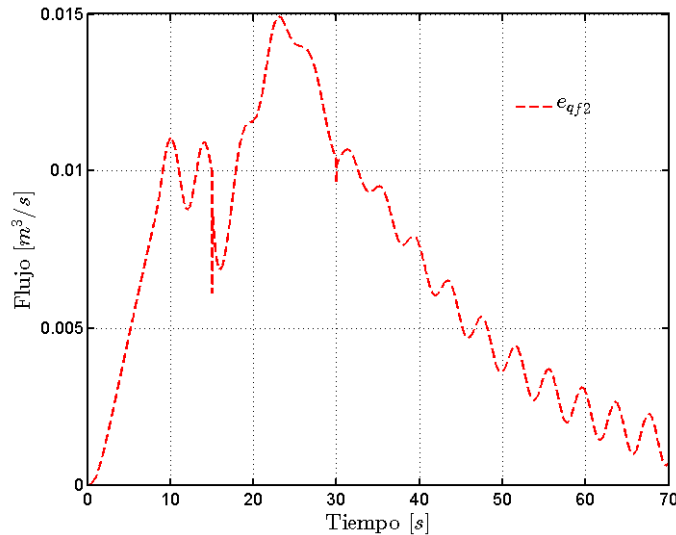


Figura 4.41: Error de Estimación del Flujo de Fuga del tanque 2.

Con lo anterior podemos calcular el flujo total de salida del tanque 2, es decir, $q_O = q_{O2} + q_{f2}$, tanto real como estimado; a continuación se muestran dichas gráficas donde podemos observar el comportamiento del flujo total de salida del tanque 2.

La Figura 4.42 representa el perfil de salida total del tanque 2, donde podemos observar que la fuga tiene un impacto directo en la forma del perfil de salida.

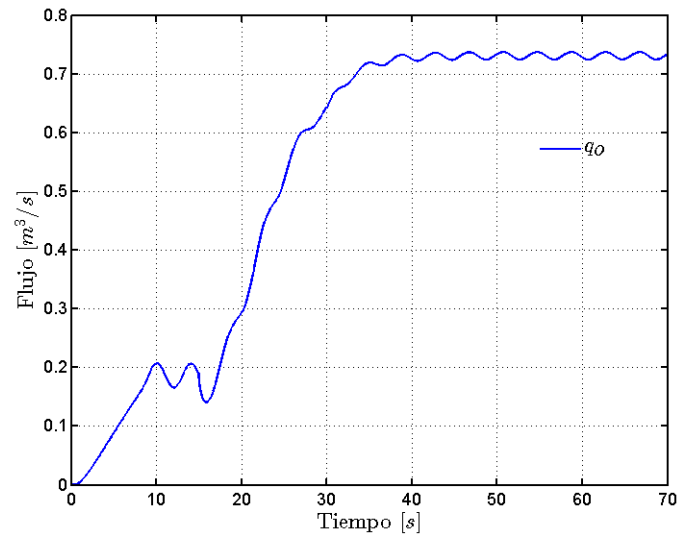


Figura 4.42: Flujo de Salida Total del tanque 2.

La Figura 4.43 representa el perfil estimado de la salida total del tanque 2, donde de igual forma se puede ver la influencia de la fuga.

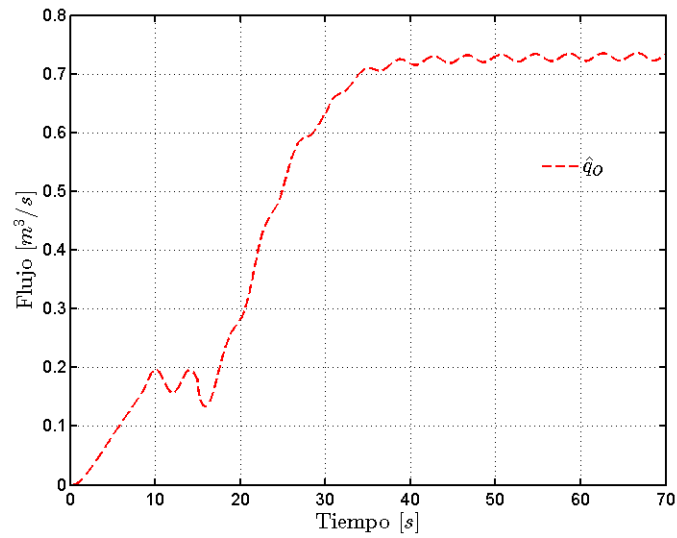


Figura 4.43: Flujo de Salida Total Estimado del tanque 2.

La Figura 4.44 es una comparativa entre los perfiles de salida total real y estimada del tanque 2, donde se puede ver que la aproximación realizada por la red es bastante parecida a la real.

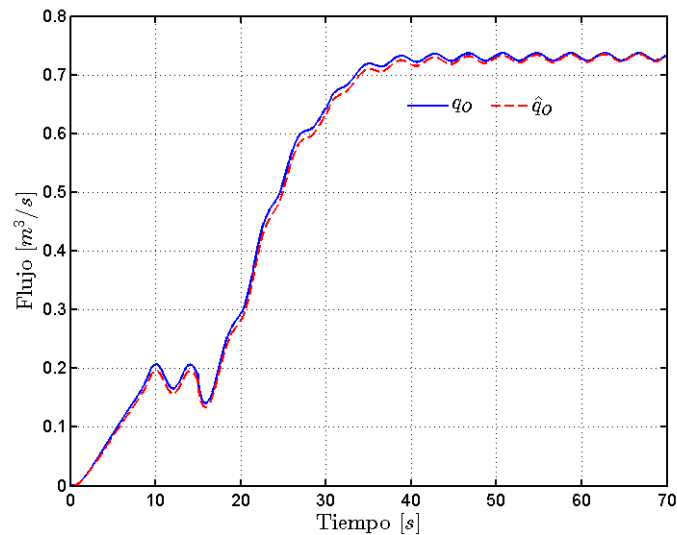


Figura 4.44: Comparativa de Flujos de Salida Total Real y Estimado del tanque 2.

La Figura 4.45 representa el error de estimación del perfil de salida total del tanque 2, donde puede observarse de igual manera que en las comparativas anteriores las diferencias entre cuando la neurona está dentro de la etapa de entrenamiento y cuando ya no se está en esa etapa.

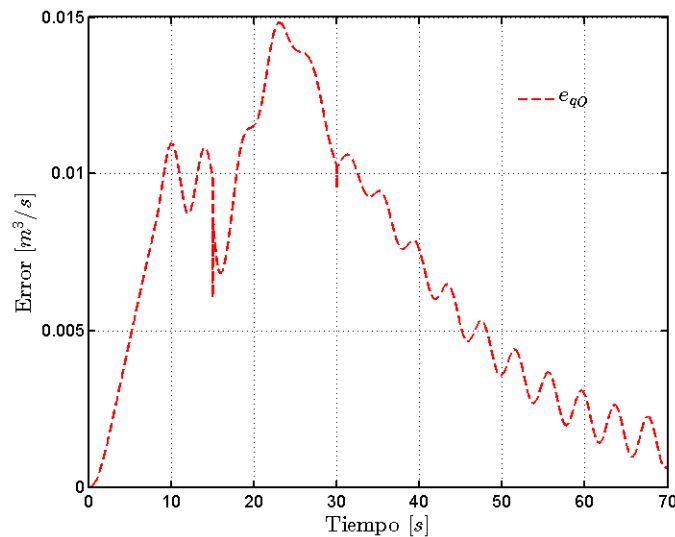


Figura 4.45: Error de Estimación del Flujo de Salida Total del tanque 2.

4.3. Simulación 3: Estimación de la competencia intraespecífica en la especie de la presa dentro de un sistema depredador-presa

4.3.1. Planteamiento del Problema

Considere las siguientes ecuaciones diferenciales

$$\begin{aligned} \frac{d}{dt}N_1(t) &= r_1N_1 - \beta N_1N_2 \\ \frac{d}{dt}N_2(t) &= -r_2N_2 + \kappa\beta N_1N_2 \end{aligned} \quad (4.28)$$

Las ecuaciones diferenciales dadas en (4.28) son conocidas como *Modelo de Lotka-Volterra para sistemas de poblaciones depredador-presa* [69, 70].

Partiendo de (4.28) e introduciendo la competencia intraespecífica en la especie de la presa [71], tenemos que

$$\begin{aligned} \frac{d}{dt}N_1(t) &= r_1N_1 - \beta N_1N_2 - \gamma N_1^2 \\ \frac{d}{dt}N_2(t) &= -r_2N_2 + \kappa\beta N_1N_2 \end{aligned} \quad (4.29)$$

donde el termino r_1 es conocido como *tasa per cápita neta de crecimiento de la población* para esta simulación se parte de que $r_1 := \underline{r}_1 + r_a u$, es decir la tasa neta de crecimiento está formada por una tasa de crecimiento fija más una tasa de crecimiento dependiente de la cantidad de alimento proporcionado al medio externamente.

Entonces el modelo a utilizar en la presente simulación es el siguiente

$$\sum_{SDP} : \begin{cases} \frac{d}{dt}N_1(t) = \underline{r}_1N_1 + r_a N_1 u - \beta N_1N_2 - \gamma N_1^2 \\ \frac{d}{dt}N_2(t) = -r_2N_2 + \kappa\beta N_1N_2 \end{cases} \quad (4.30)$$

donde \underline{r}_1 es la tasa per cápita mínima de crecimiento de la población de presas, r_a es tasa de crecimiento de la población asociada al alimento añadido, β es el factor de proporcionalidad que indica el grado de efectividad del proceso de interacción entre la presa y el depredador, γ es el coeficiente de competencia intraespecífica en el hábitat para la especie de presas, r_2 es la tasa per cápita de crecimiento de la población de la especie predatora, κ es el factor de proporcionalidad que indica el grado en que afecta el éxito en la caza en los predadores.

Se considera que el sistema (4.30) tiene como única incertidumbre el termino γN_1^2 que corresponde a la competencia intraespecífica en el hábitat para la especie de presas.

4.3.2. Objetivos de la Simulación

La simulación consiste en estimar la competencia intraespecífica para la especie de presas, además a partir de dicha estimación determinar el perfil del coeficiente estimado de γ . Partiendo de las condiciones iniciales de la población de presas de 3000 individuos, es decir, $N_1(0) = 3000$ y $N_2(0) = 1000$ y el sistema se considera para un periodo de 100 años.

4.3.3. Desarrollo de la Simulación

Considerando una Red Neuronal Dinámica de la forma (3.4) para realizar la identificación en línea de la incertidumbre del sistema (4.30).

De acuerdo con el modelo propuesto (3.3), el modelo de la simulación, es decir el sistema (4.30) queda como sigue:

$$\dot{N}(t) = \begin{bmatrix} \underline{r_1} - \gamma N_1 & -\beta N_1 \\ \kappa \beta N_2 & -r_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} + \begin{bmatrix} r_a N_1 \\ 0 \end{bmatrix} u, \quad (4.31)$$

donde

$$f_0(N) = \begin{bmatrix} \underline{r_1} & -\beta N_1 \\ \kappa \beta N_2 & -r_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}, \Delta_f(N) = \begin{bmatrix} -\gamma N_1^2 \\ 0 \end{bmatrix}, g(N) = \begin{bmatrix} r_a N_1 \\ 0 \end{bmatrix}, \quad (4.32)$$

y además se sabe que $\underline{\gamma} \leq \gamma \leq \bar{\gamma}$

La arquitectura de la Red Neuronal Dinámica de la forma (3.4) es:

$$\dot{\hat{N}}(t) = \begin{bmatrix} \underline{r_1} & -\beta N_1 \\ \kappa \beta N_2 & -r_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} + W \varphi(\hat{N}) + \psi + \begin{bmatrix} r_a N_1 \\ 0 \end{bmatrix} u + L(N - \hat{N}), \quad (4.33)$$

entonces aplicando el teorema 3.1 primero se selecciona al umbral neuronal ψ como

$$\psi = \text{sign}(e) \cdot \delta_f(x), \quad (4.34)$$

donde la cota superior de la norma de la incertidumbre del sistema se selecciona como

$$\delta_f(x) = \bar{\gamma} N_1^2, \quad (4.35)$$

y también se elige a la función de activación $\varphi(\hat{N})$ como

$$\varphi(\hat{N}) = \hat{N} + 100 \cdot \tanh(0.001\hat{N}). \quad (4.36)$$

La simulación del sistema depredador presa fue realizada con los datos de la Tabla 4.3.

Variable	Valor	Descripción
r_1	0.4	Diferencia entre las tasas de natalidad y mortalidad de la presa.
r_a	0.04	Coeficiente de tasa de crecimiento de la presa por unidad de alimento.
β	0.37	Factor de proporcionalidad que indica el grado de efectividad del proceso de interacción.
r_2	0.3	El grado de la competitividad de depredadores en que esto afecta al crecimiento de la población.
κ	$\frac{5}{37}$	Factor de proporcionalidad que indica el grado en que afecta el éxito en la caza en los depredadores.

Tabla 4.3: Datos usados en la Simulación 3.

En la Figura 4.46 se muestra el desarrollo de las poblaciones de presa (N_1) y depredador (N_2) a lo largo de 100 años, aquí se puede observar la dependencia de una especie y otra, es decir el número de presa existentes afecta proporcionalmente al número de depredadores.

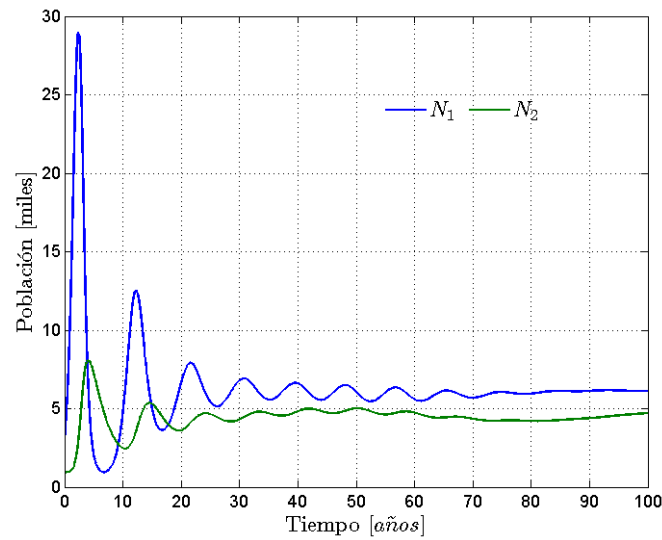


Figura 4.46: Población de las dos especies.

En la Figura 4.47, se observa el número estimado de individuos por especie que arroja la red neuronal.

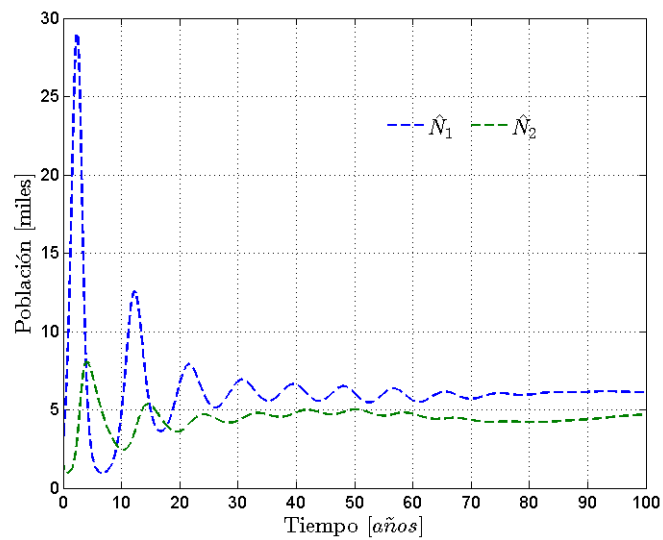


Figura 4.47: Población Estimada de las dos especies.

La Figura 4.48 presenta una comparación de la población de la especie de la presa y la población estimada de la presa arrojada por la red neuronal.

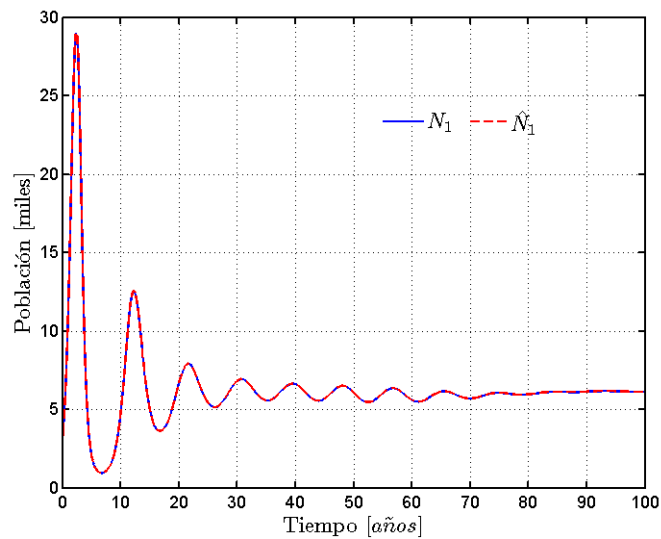


Figura 4.48: Comparativa entre Población y Población Estimada de la especie de la presa.

La Figura 4.49 representa el error de estimación entre las población real y la población estimada por la red neuronal para la especie de la presa, se puede observar que existe una diferencia muy pequeña entre ambas, por lo cual se puede considerar una buena estimación.

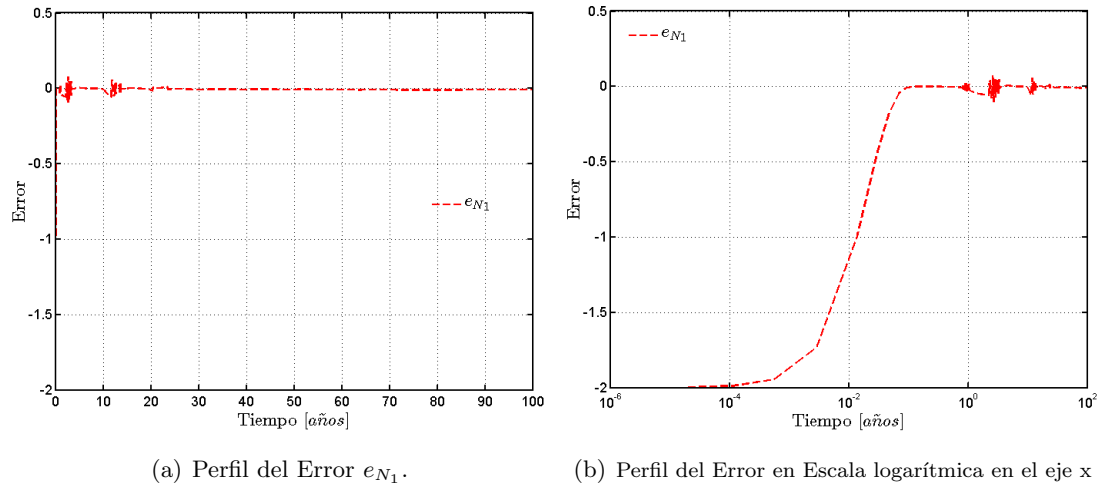


Figura 4.49: Error de Estimación de la población de la especie de la presa.

La Figura 4.50 presenta una comparación de la población de la especie del depredador y la población estimada del depredador resultante de la red neuronal.

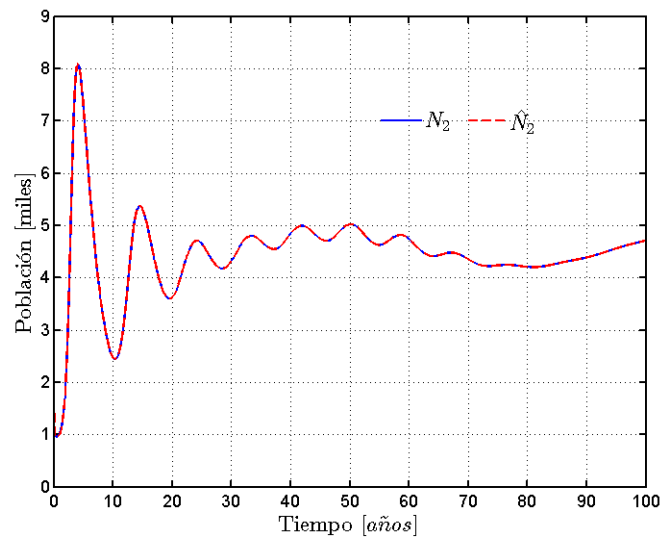


Figura 4.50: Comparativa entre la población y la población estimada en la especie del depredador.

La Figura 4.51 representa el error de estimación entre las población real y la población estimada por la red neuronal para la especie del depredador, se puede observar que existe un error de estimación pequeño, dado esto se toma como una buena estimación.

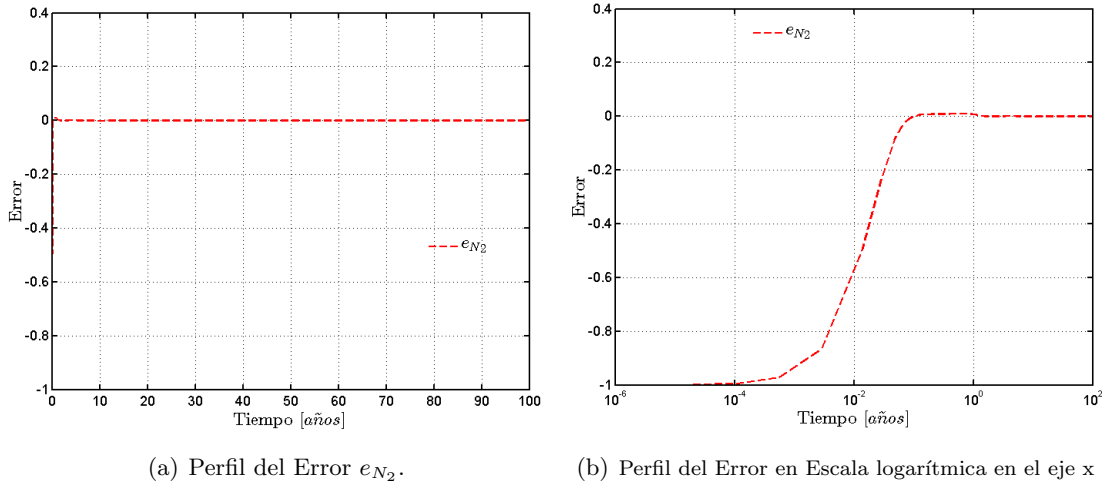


Figura 4.51: Error de Estimación de la población en la especie del depredador.

Durante la obtención de los datos para realizar las Figuras anteriores, se estima la incertidumbre del sistema

Luego de la realización de la simulación, se tiene que si el error de identificación fuera prácticamente 0 tendríamos que su derivada también lo sería por lo que:

$$e \cong 0 \quad , \quad (4.37)$$

$$\begin{aligned} \dot{e} &\cong \Delta_f(N) - W\varphi(\hat{N}) - \psi - Le \quad , \\ 0 &\cong \Delta_f(N) - W\varphi(\hat{N}) - \psi - Le \quad , \end{aligned} \quad (4.38)$$

entonces por las ecuaciones (3.5), (3.6) y (3.7) del Teorema 3.1

$$\begin{aligned} Le &\cong 0 \quad , \\ \psi &\cong \text{sign}(0) \cdot \delta_f \cong 0 \quad , \end{aligned} \quad (4.39)$$

tenemos que la incertidumbre del sistema $\Delta_f(N)$ es aproximadamente igual a la componente neuronal $W\varphi(\hat{N})$

$$\begin{aligned} \Delta_f(N) - W\varphi(\hat{N}) &\cong 0 \quad , \\ \Delta_f(N) &\cong W\varphi(\hat{N}) \quad , \end{aligned} \quad (4.40)$$

de (4.40) tenemos que el error de estimación de la incertidumbre es igual a la resta de Δ_f menos la componente neuronal $W\varphi(\hat{N})$, es decir

$$e_{\Delta_f} \cong \Delta_f(N) - W\varphi(\hat{N}) \quad ,$$

$$e_{\Delta_f} = \begin{bmatrix} e_{\Delta_{f1}} \\ e_{\Delta_{f2}} \end{bmatrix} . \quad (4.41)$$

En la Figura 4.52 se muestra el perfil de la incertidumbre del sistema, es decir, la competencia intraespecífica de la especie de la presa

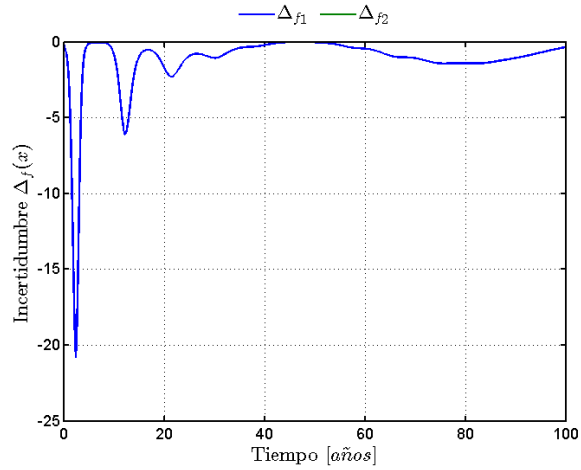


Figura 4.52: Incertidumbre del Sistema.

La Figura 4.53 representa el perfil de la componente neuronal $W\varphi(\hat{N})$ arrojado por la red neuronal.

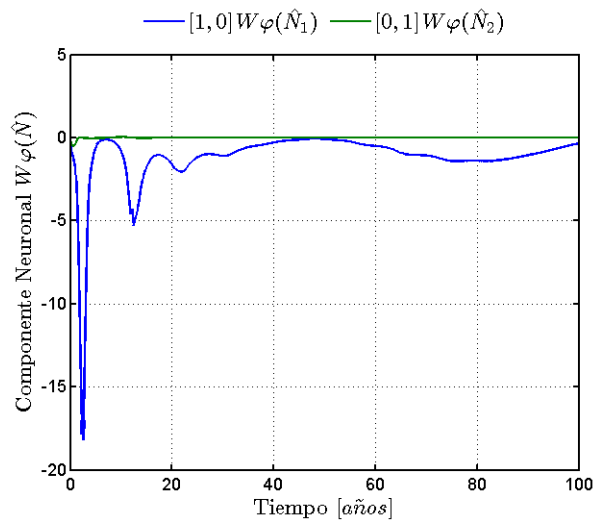


Figura 4.53: Componente Neuronal $W\varphi(\hat{N})$

La Figura 4.54 es una comparación entre la incertidumbre del sistema $\Delta_f(N)$ y la componente neuronal $W\varphi(\hat{N})$, aquí se puede observar que durante la etapa de entrenamiento difieren un poco, pero conforme va saliendo de dicha etapa el perfil de ambas es muy similar.

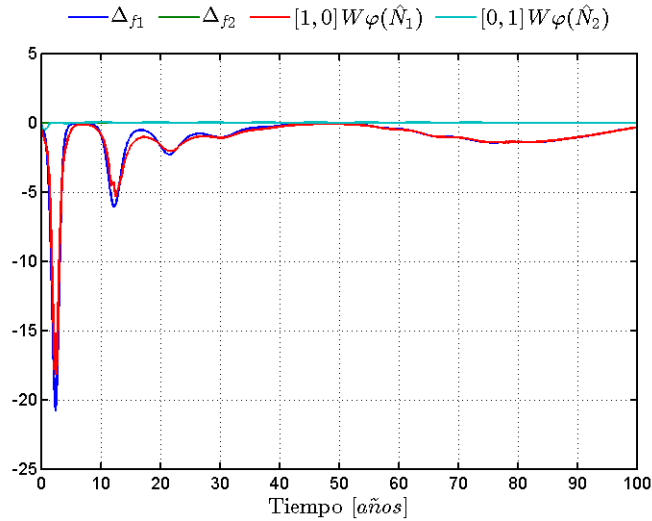


Figura 4.54: Comparación de Δ_f y $W\varphi(\hat{N})$.

La Figura 4.55 representa el error de estimación de la Incertidumbre del sistema, donde se puede observar que durante la etapa de entrenamiento el error es notable, pero conforme sale de la etapa de entrenamiento el error converge a un valor cercano a 0.

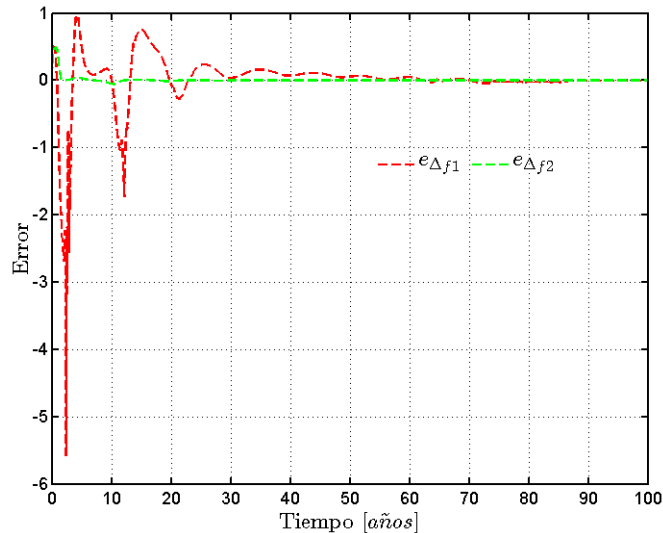


Figura 4.55: Error de Estimación la Incertidumbre del Sistema.

En la Figura 4.56 se observa la evolución del error cuadrático medio a lo largo del tiempo, donde se puede observar claramente tres cambios, finalizando con un valor de 0.2185

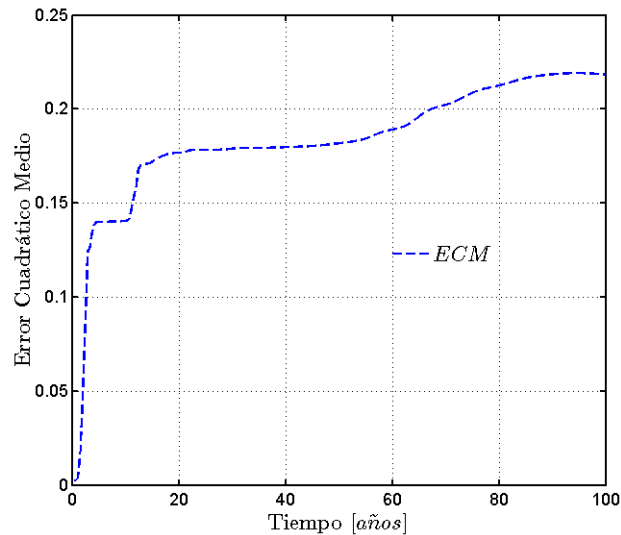


Figura 4.56: Error Cuadrático Medio a lo largo del tiempo de simulación.

En la Figura 4.57, se puede observar el comportamiento de los pesos sinápticos de la red neuronal, donde se puede observar que durante el tiempo de simulación utilizado dos de los pesos aun no convergen.

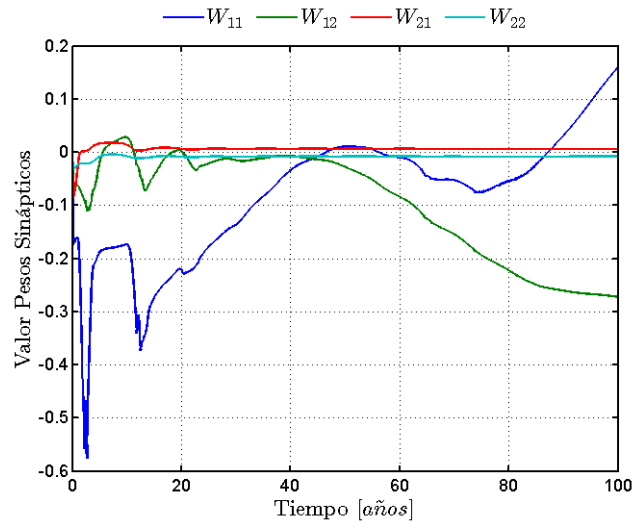


Figura 4.57: Pesos Sinápticos de la Red Neuronal Dinámica.

En el Anexo B se puede observar los resultados de esta misma simulación con el objetivo de mostrar la convergencia de los pesos sinápticos de la red neuronal dinámica.

4.4. Simulación 4: Identificación de Incertidumbre en un Péndulo Invertido

4.4.1. Planteamiento del Problema

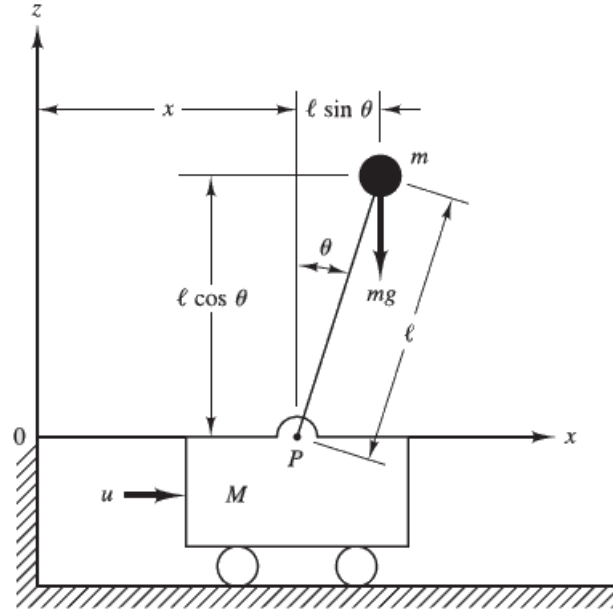


Figura 4.58: Sistema de péndulo invertido.

Considere el sistema de péndulo invertido que se muestra en la Figura 4.58. Como en este sistema la masa se concentra en lo alto de la varilla, el centro de gravedad es el centro de la bola del péndulo. El modelo del sistema de péndulo invertido está representado por las ecuaciones siguientes

$$(M + m)\ddot{x} + ml\ddot{\theta} = u, \quad (4.42)$$

$$(I + ml^2)\ddot{\theta} + ml\ddot{x} = mgl\theta. \quad (4.43)$$

Para ver la obtención de estas ecuaciones ver Anexo C.

Para este caso, el momento de inercia del péndulo respecto de su centro de gravedad es pequeño, y se supone que $I + ml^2 \simeq ml^2$. Entonces el modelo matemático para este sistema es el siguiente:

$$Ml\ddot{\theta} = (M + m)g\theta - u, \quad (4.44)$$

$$M\ddot{x} = u - mg\theta. \quad (4.45)$$

4.4.2. Objetivos de la Simulación

La presente simulación tiene como objeto la utilización del modelo linealizado del péndulo invertido del apartado anterior, cuyo desarrollo se muestra en el Anexo C, el cual es utilizado para la creación de la arquitectura de la red neuronal, para luego llevar a cabo la comparación con un modelo del péndulo invertido de distinta estructura matemática tomado de Matlab/Simulink, para así observar la capacidad de identificación de la arquitectura neuronal propuesta partiendo de modelos distintos.

La presente simulación consiste en reproducir el resultado principal de esta tesis ante un caso en que existen dinámicas no modeladas por el efecto de una linealización en un punto de equilibrio inestable.

4.4.3. Desarrollo de la Simulación

El modelo del sistema de péndulo invertido tomado de Matlab/Simulink es el que se puede observar en la Figura 4.59

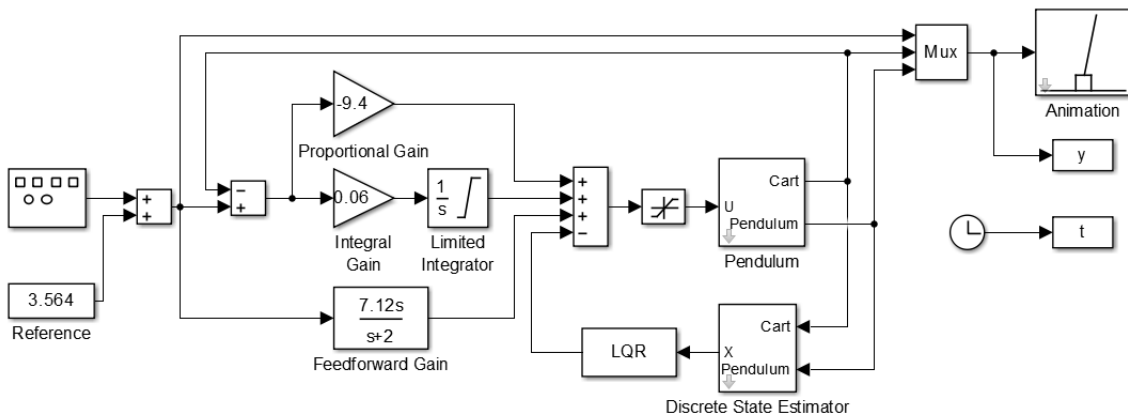


Figura 4.59: Sistema de péndulo invertido de Matlab/Simulink.

Ajustando nuestro sistema las variables de estado x_1, x_2, x_3 y x_4 siguientes:

$$\begin{aligned} x_1 &= \theta & x_3 &= x \\ x_2 &= \dot{\theta} & x_4 &= \dot{x} \end{aligned} ,$$

además se consideran disponibles todos los estados del sistema del sistema, entonces

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} ,$$

A partir de las definición de variables de estado y de las ecuaciones (4.44) y (4.45) se obtiene que

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{M+m}{Ml}gx_1 - \frac{1}{Ml}u \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{m}{M}gx_1 + \frac{1}{M}u\end{aligned}$$

El modelo del sistema del péndulo invertido utilizado para la creación de la red neuronal en su representación en espacio de estados es el siguiente:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M+m}{Ml}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m}{M}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} u, \quad (4.46)$$

Considerando una Red Neuronal Dinámica de la forma (3.4) tenemos:

$$\dot{\hat{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M+m}{Ml}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m}{M}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + W\varphi(\hat{x}) + \psi + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} u, \quad (4.47)$$

Entonces aplicando el resultado principal de esta tesis, es decir, el Teorema 3.1 primero se selecciona al umbral neuronal ψ como

$$\psi = \text{sign}(e) \cdot \delta_f(x), \quad (4.48)$$

donde la cota superior de la incertidumbre del sistema está definida como

$$\delta_f(x) = 0.0001 \text{sen}(\hat{x}), \quad (4.49)$$

y también se elige a la función de activación $\varphi(\hat{x})$ como

$$\varphi(\hat{x}) = \hat{x} + 0.01 \cdot \tanh(\hat{x}). \quad (4.50)$$

La siguiente simulación fue realizada con los datos de la Tabla 4.4.

Variable	Valor	Descripción
M	0.455	Masa del carro.
m	0.21	Masa de la bola del péndulo.
l	0.61	Longitud del péndulo.
g	9.81	Constante de gravedad.

Tabla 4.4: Datos usados en la Simulación 4.

La Figura 4.60 muestra el perfil de la señal de control que se le da al sistema del péndulo invertido.

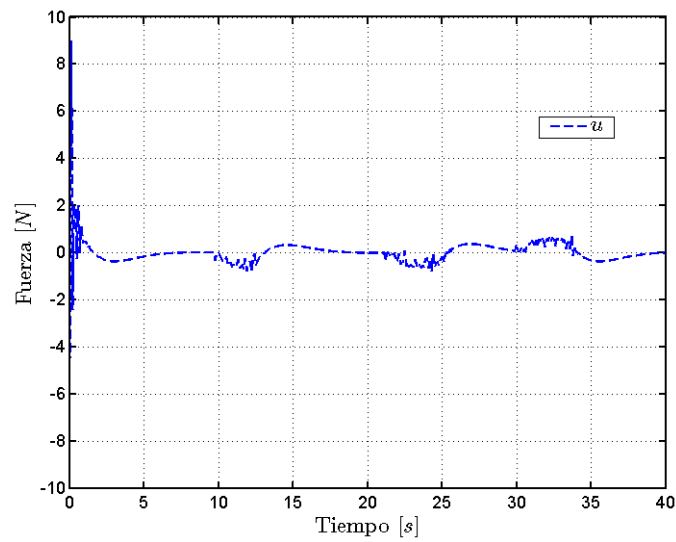


Figura 4.60: Señal de Control.

La Figura 4.61 representa el perfil de la Posición Angular del Péndulo Invertido, se puede observar las oscilaciones que tiene la barra del péndulo invertido.

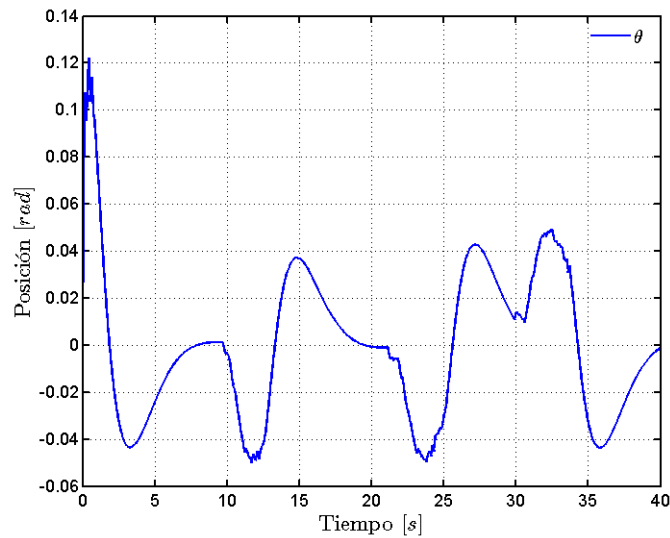


Figura 4.61: Posición Angular del Péndulo.

La Figura 4.62 muestra la posición angular estimada del péndulo invertido, es decir, la estimación realizada por la red neuronal.

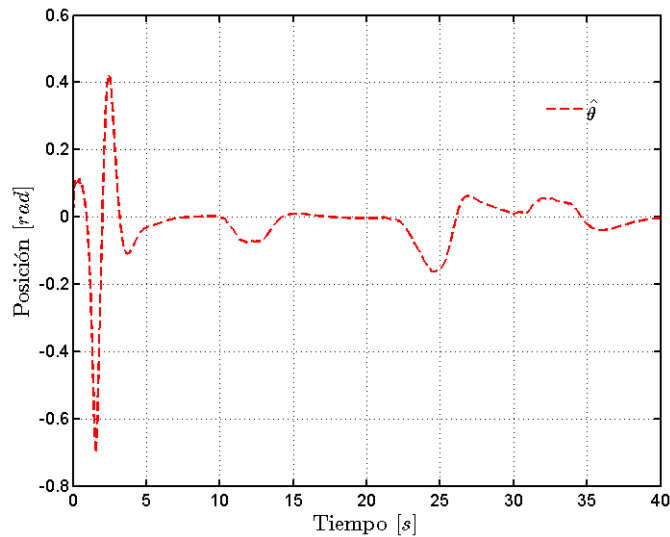


Figura 4.62: Posición Angular Estimada del Péndulo.

En la Figura 4.63 se puede observar la comparación entre los perfiles de la posición real y la estimada del péndulo invertido, además se puede observar claramente donde se encuentra la etapa de entrenamiento y cuando termina dicha etapa.

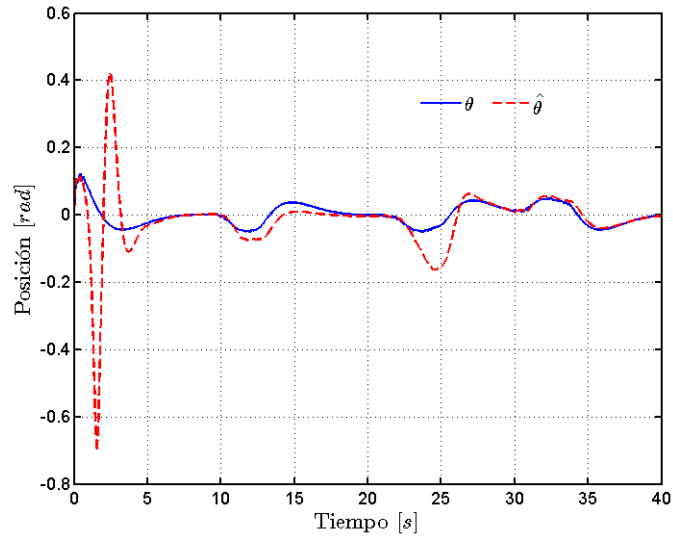


Figura 4.63: Comparativa de la Posición del Péndulo real y estimada.

En la Figura 4.64 se muestra la gráfica del error de estimación de la posición angular del péndulo invertido, el cual dejando fuera la etapa de entrenamiento es relativamente pequeño por lo cual la estimación se considera favorable.

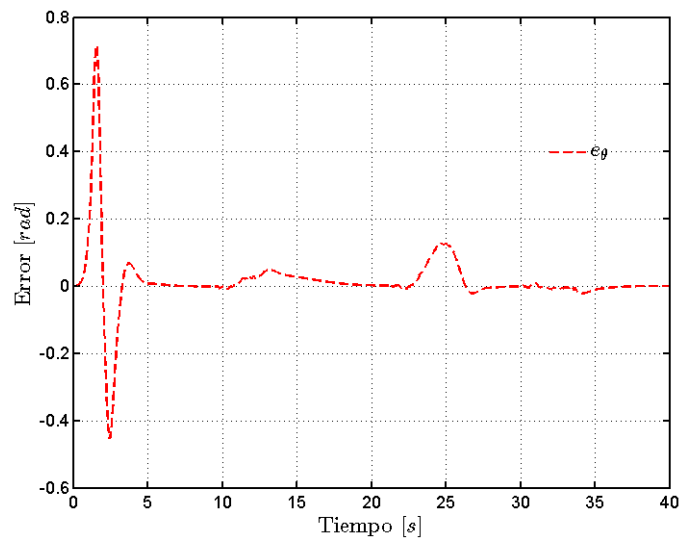


Figura 4.64: Error de Estimación de Posición Angular del Péndulo.

La Figura 4.65 representa el perfil de la velocidad del péndulo invertido.

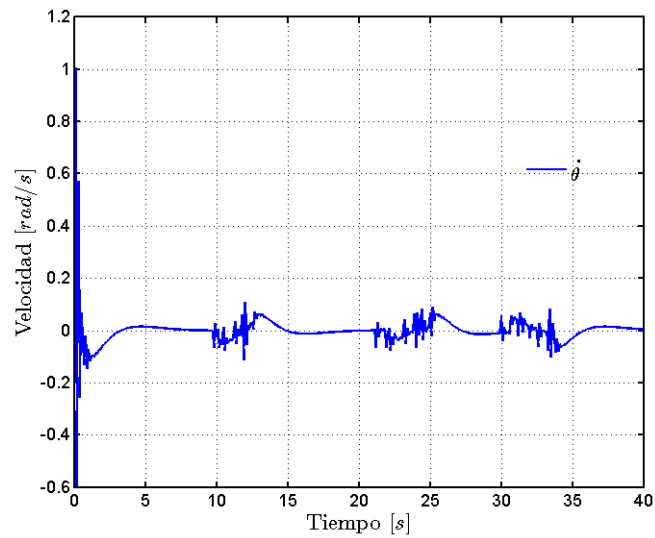


Figura 4.65: Velocidad Angular del Péndulo.

La Figura 4.66 muestra el perfil de la velocidad estimada del péndulo invertido arrojada por la red neuronal.

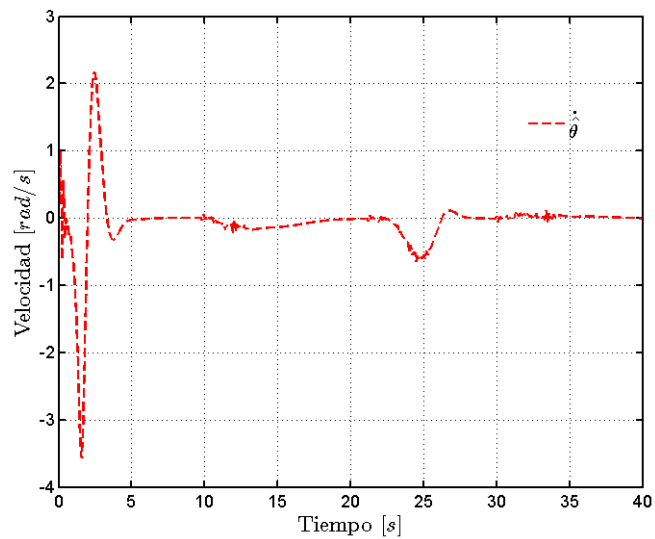


Figura 4.66: Velocidad Angular Estimada del Péndulo.

En la Figura 4.67 se observa la comparativa entre la velocidad real y estimada del péndulo invertido se puede observar que en este caso la estimación difiere de la velocidad real.

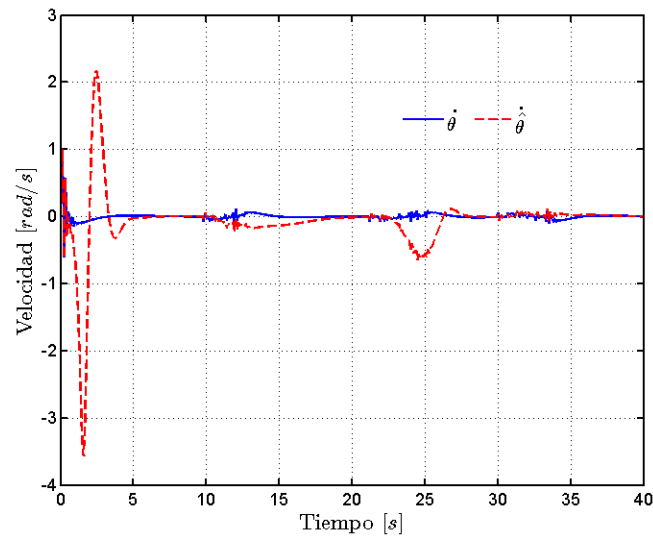


Figura 4.67: Comparativa de la Velocidad del Péndulo real y estimada.

La Figura 4.68 muestra la gráfica del error de estimación de la velocidad del péndulo, aquí se puede ver que existen instantes de tiempo donde el error es cercano a cero y esto es cuando la velocidad es cercana a cero.

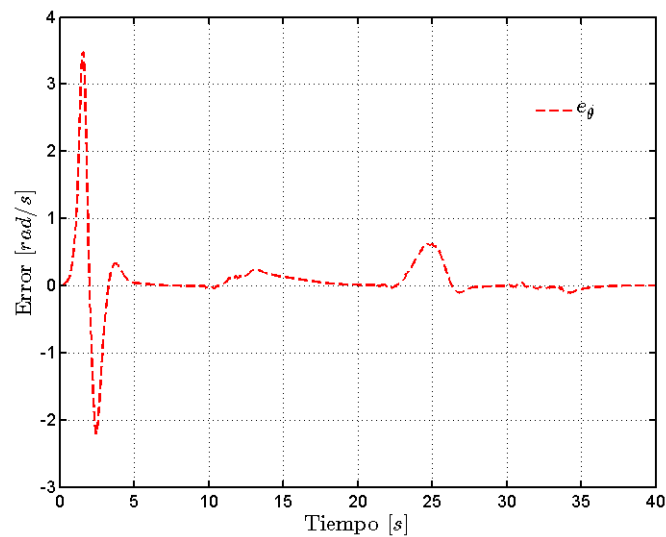


Figura 4.68: Error de Estimación de Velocidad Angular del Péndulo.

La Figura 4.69 representa el perfil de la posición del carro del péndulo invertido.

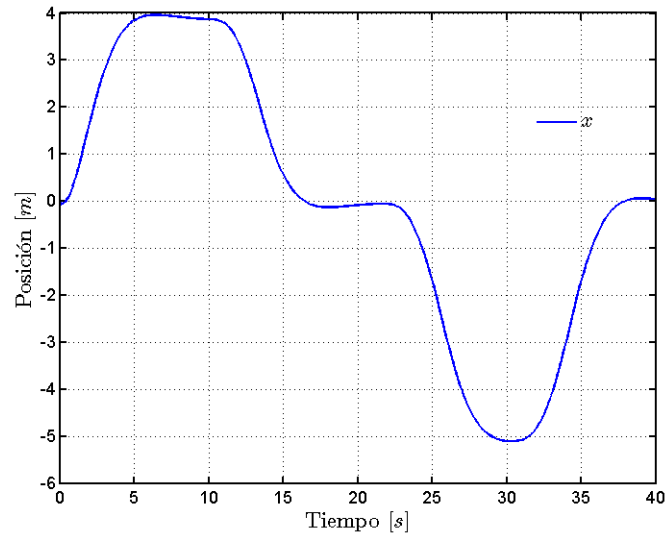


Figura 4.69: Posición del Carro.

La Figura 4.70 representa el perfil de la posición estimada del carro arrojado por la red neuronal.

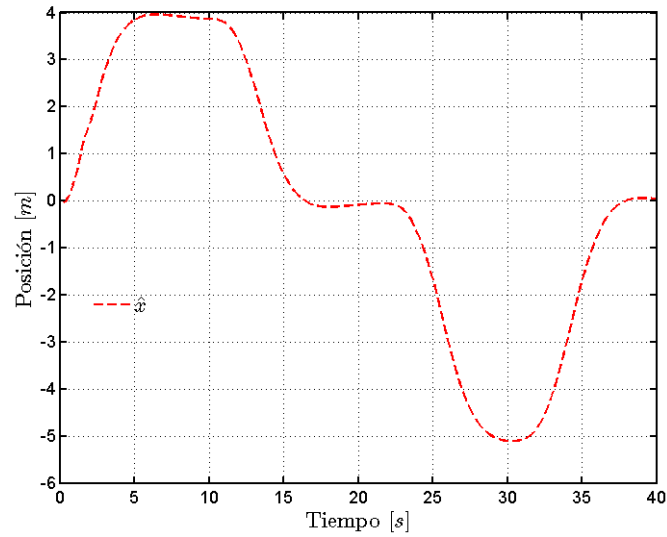


Figura 4.70: Posición Estimada del Carro.

En la Figura 4.71 se muestra la comparativa entre la posición real y estimada del carro, se puede observar que ambos tiene un perfil similar.

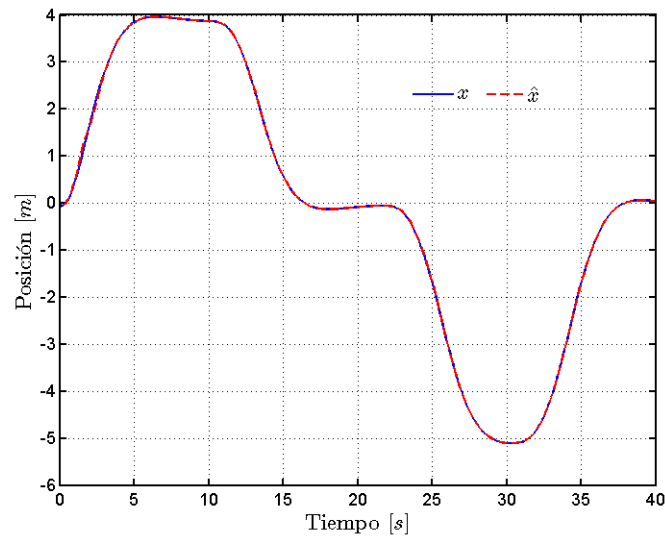


Figura 4.71: Comparativa de la Posición del Carro real y estimada.

La Figura 4.72 representa el error de estimación de la posición del carro, aquí se puede observar que la mayor diferencia existe durante la etapa de entrenamiento de la red neuronal, omitiendo esas diferencias se observa que el error de estimación es relativamente pequeño.

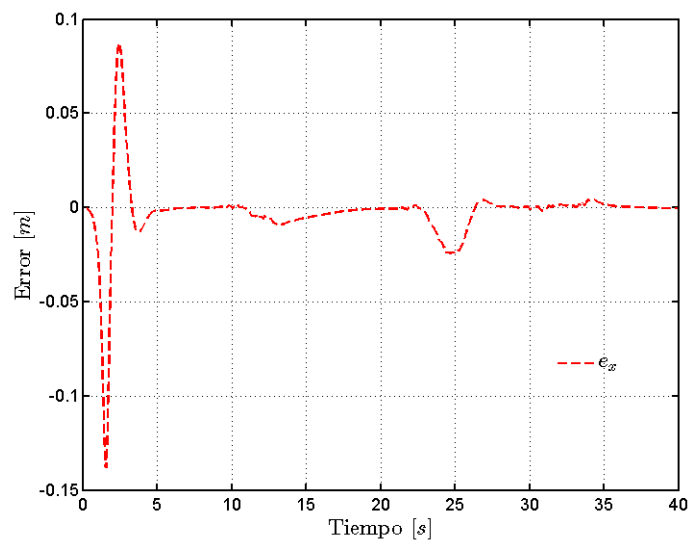


Figura 4.72: Error de Estimación de Posición del Carro.

La Figura 4.73 representa el perfil de la velocidad del carro, además se puede observar los cambios de dirección que ocasiona la señal de control en el carro.

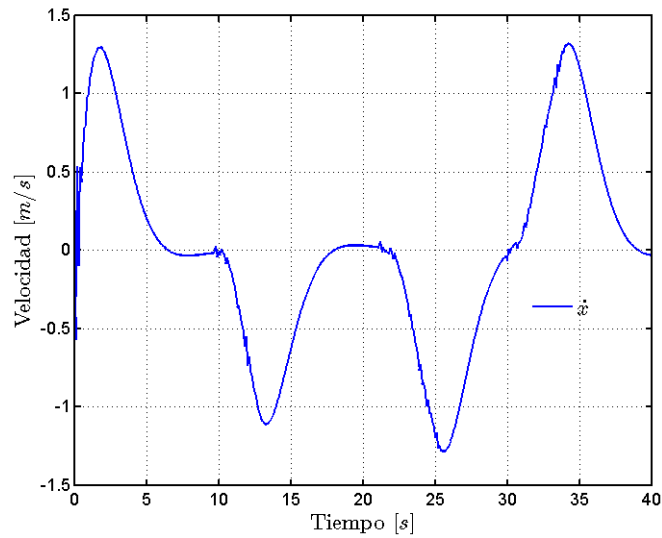


Figura 4.73: Velocidad del Carro.

La Figura 4.74 muestra el perfil de la velocidad estimada del carro arrojada por la red neuronal.

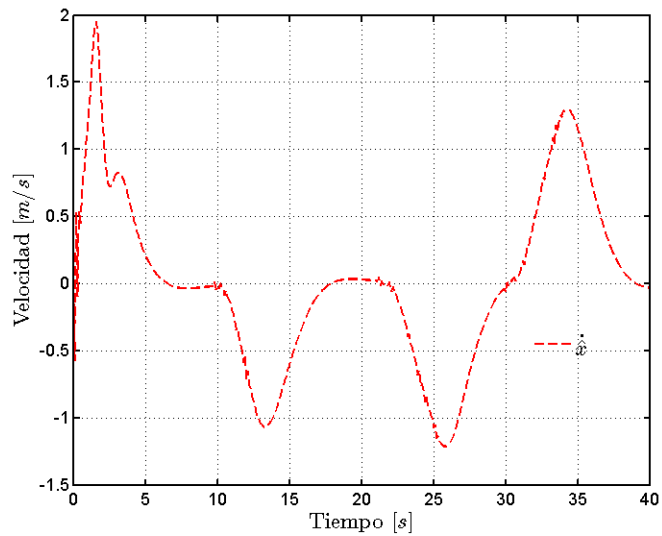


Figura 4.74: Velocidad Estimada del Carro.

En la Figura 4.8 se observa la comparativa entre los perfiles de la velocidad real y la velocidad estimada del carro, donde se puede observar que la estimación difiere de la real durante la etapa de entrenamiento, luego es prácticamente igual.

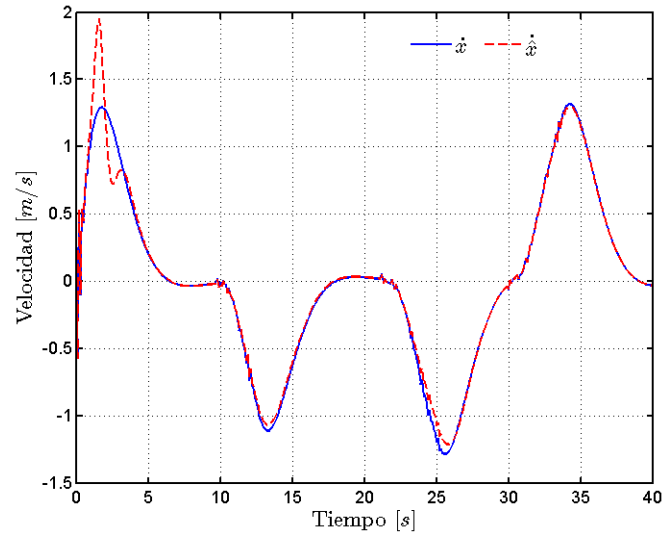


Figura 4.75: Comparativa de la Velocidad del Carro real y estimada.

La Figura 4.76 representa la gráfica del error de estimación de la Velocidad del carro, donde se observa que el error mayor corresponde a la etapa de entrenamiento de la red neuronal.

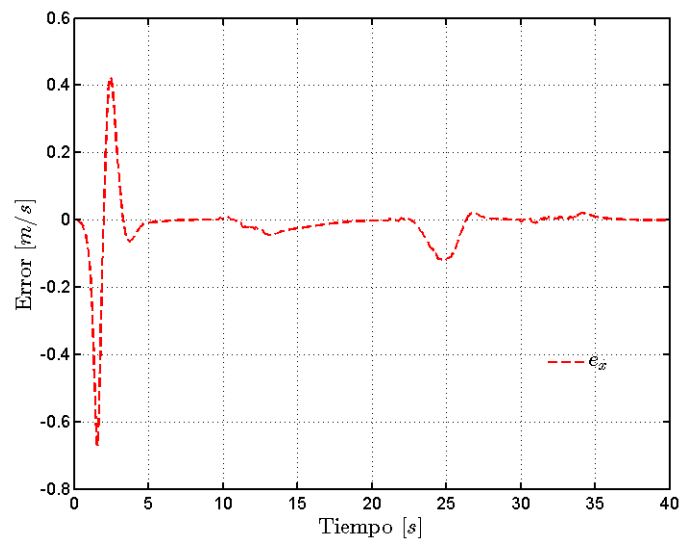


Figura 4.76: Error de Estimación de Velocidad del Carro.

La Figura 4.77 representa la incertidumbre del sistema, la cual está constituida por las dinámicas no modeladas por el efecto de una linealización en un punto de equilibrio inestable del péndulo invertido.

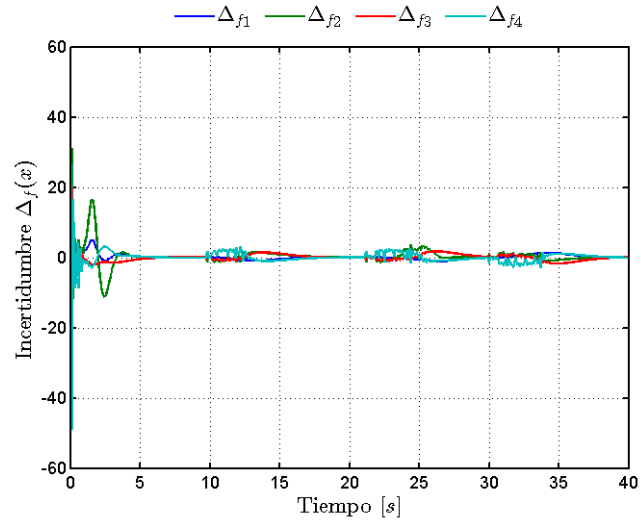


Figura 4.77: Incertidumbre del Péndulo Invertido.

La Figura 4.78 representa el perfil del componente neuronal $W\varphi(\hat{x})$ resultante de la red neuronal.

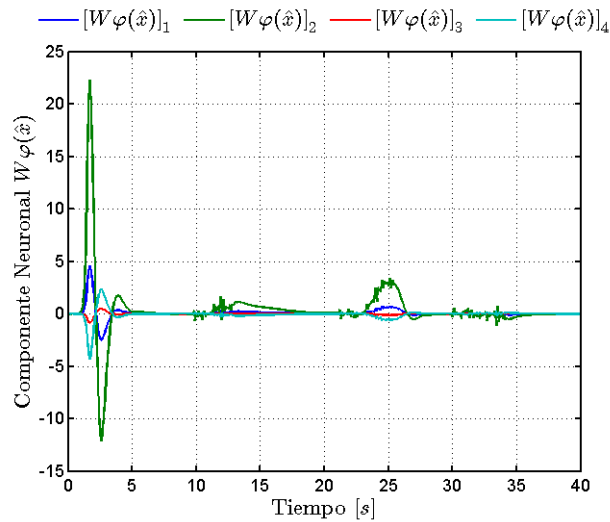


Figura 4.78: Componente Neuronal $W\varphi(\hat{x})$.

La Figura 4.79 muestra la comparación de los perfiles de Δ_f y la componente neuronal $W\varphi(\hat{x})$.

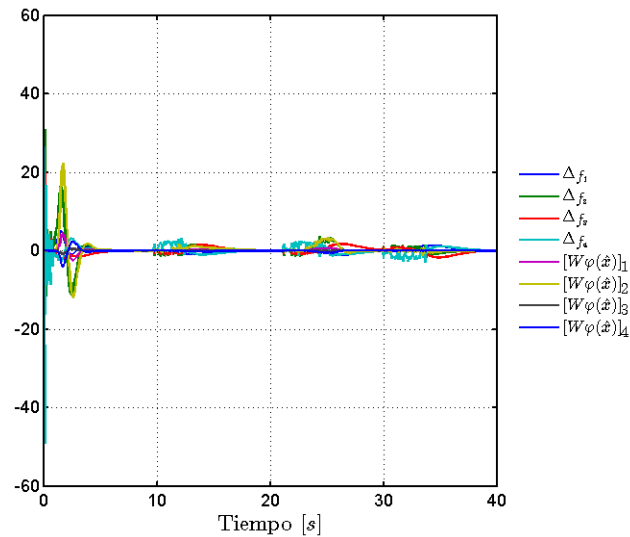


Figura 4.79: Comparación de Δ_f y $W\varphi(\hat{x})$.

Debido a la cantidad de perfiles mostrados en la Figura anterior, a continuación se muestra cada elemento de Δ_f con su correspondiente elemento de la componente neuronal $W\varphi(\hat{x})$.

La Figura 4.80 muestra la comparación del primer elemento de Δ_f con el primer elemento de $W\varphi(\hat{x})$.

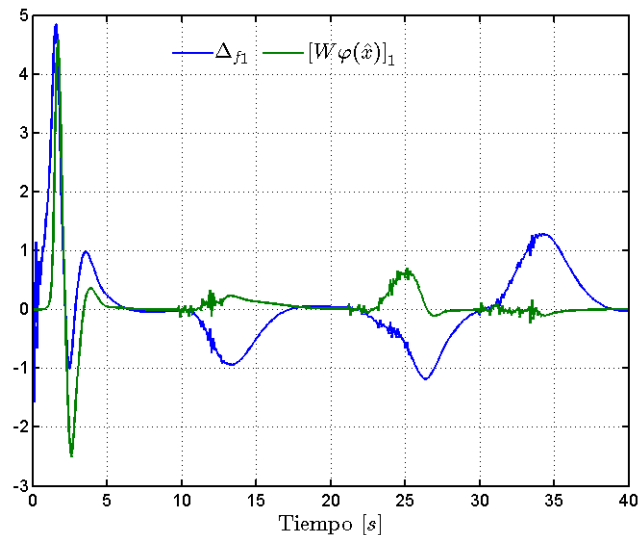


Figura 4.80: Comparación de Δ_{f_1} y $W\varphi(\hat{x})_1$.

La Figura 4.81 muestra la comparación del segundo elemento de Δ_f con el segundo elemento de $W\varphi(\hat{x})$.

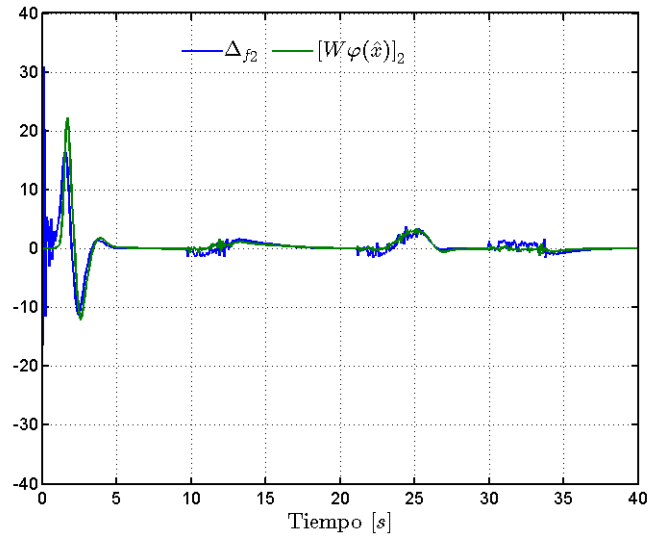


Figura 4.81: Comparación de Δ_{f2} y $W\varphi(\hat{x})_2$.

La Figura 4.82 muestra la comparación del tercer elemento de Δ_f con el tercer elemento de $W\varphi(\hat{x})$.

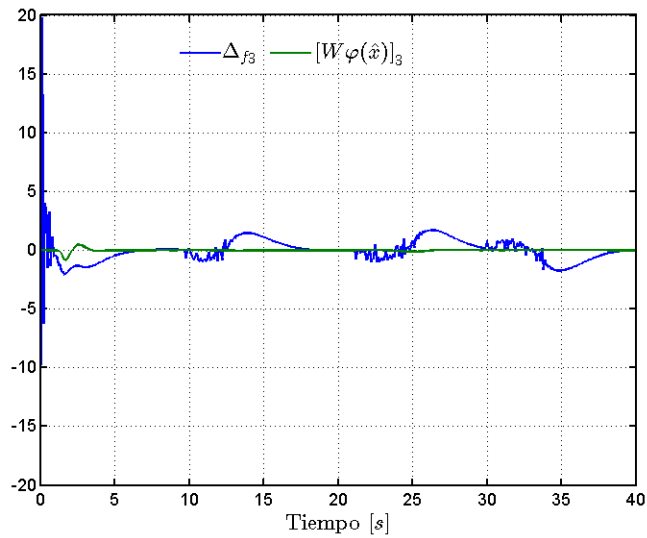


Figura 4.82: Comparación de Δ_{f3} y $W\varphi(\hat{x})_3$.

La Figura 4.83 muestra la comparación del cuarto elemento de Δ_f con el cuarto elemento de $W\varphi(\hat{x})$.

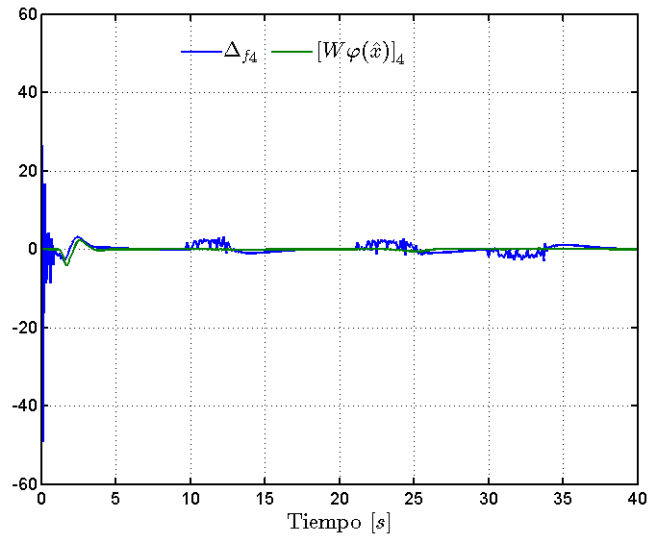


Figura 4.83: Comparación de Δ_{f4} y $W\varphi(\hat{x})_4$.

La Figura 4.84 representa el error de estimación de la incertidumbre del sistema causada por las dinámicas no modeladas.

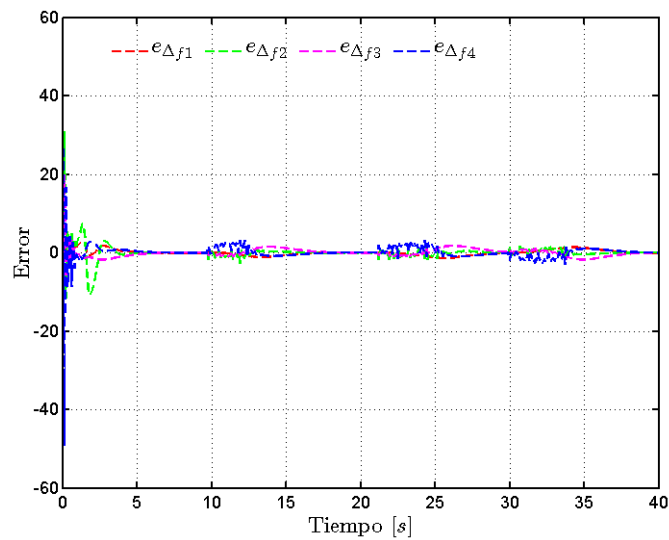


Figura 4.84: Error de estimación de la Incertidumbre en el Péndulo Invertido.

De igual manera que en la serie de figuras anteriores, para una mejor apreciación se desglosan cada uno de los errores de identificación de la incertidumbre.

La Figura 4.85 muestra el error de identificación de la incertidumbre correspondiente a la resta de los primeros elementos de los vectores de incertidumbre Δ_f y componente neuronal $W\varphi(\hat{x})$.

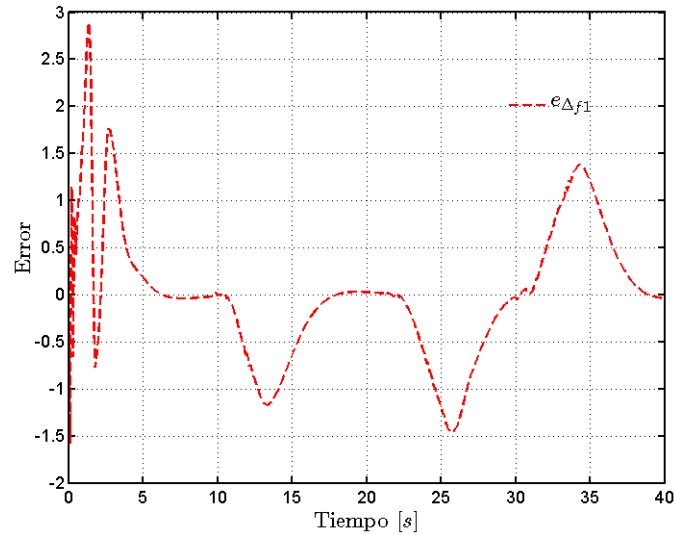


Figura 4.85: Error de estimación de la Incertidumbre $e_{\Delta_{f1}}$ en el Péndulo Invertido.

La Figura 4.86 muestra el error de identificación de la incertidumbre correspondiente a la resta de los segundos elementos de los vectores de incertidumbre Δ_f y componente neuronal $W\varphi(\hat{x})$.

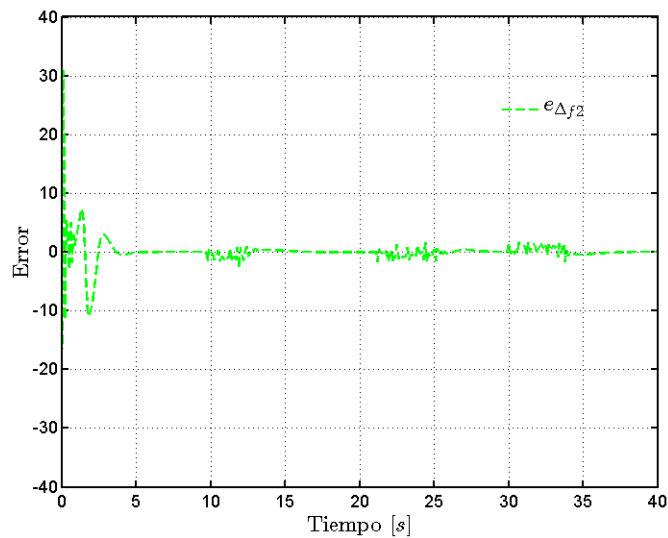


Figura 4.86: Error de estimación de la Incertidumbre $e_{\Delta_{f2}}$ en el Péndulo Invertido.

La Figura 4.87 muestra el error de identificación de la incertidumbre correspondiente a la resta de los terceros elementos de los vectores de incertidumbre Δ_f y componente neuronal $W\varphi(\hat{x})$.

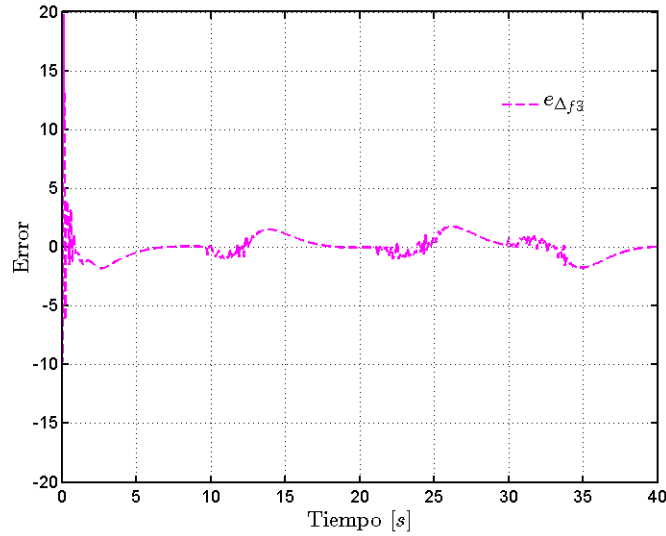


Figura 4.87: Error de estimación de la Incertidumbre $e_{\Delta_{f3}}$ en el Péndulo Invertido.

La Figura 4.85 muestra el error de identificación de la incertidumbre correspondiente a la resta de los últimos elementos de los vectores de incertidumbre Δ_f y componente neuronal $W\varphi(\hat{x})$.

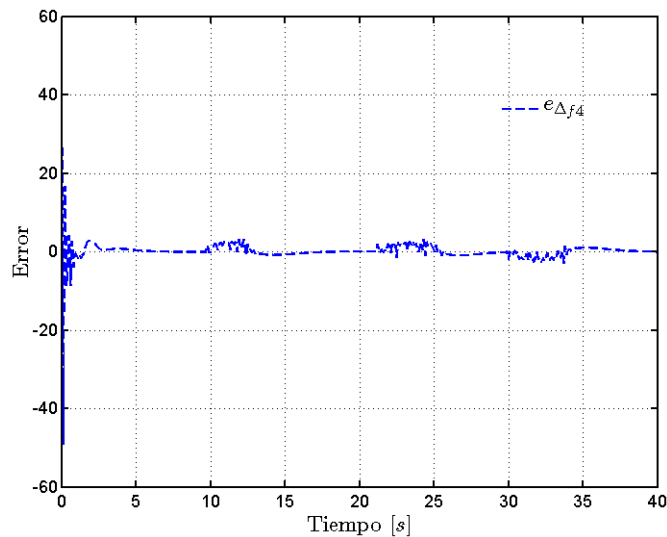


Figura 4.88: Error de estimación de la Incertidumbre $e_{\Delta_{f4}}$ en el Péndulo Invertido.

En la Figura 4.89 se muestra la evolución del error cuadrático medio a lo largo del tiempo, donde se puede observar claramente que al principio de la simulación el área bajo la curva del error cuadrático medio es grande, con el paso del tiempo el aumento en dicha área es pequeño,

lo cual es debido a la convergencia del error de identificación de la incertidumbre, finalizando el error cuadrático medio con un valor de 10.6069

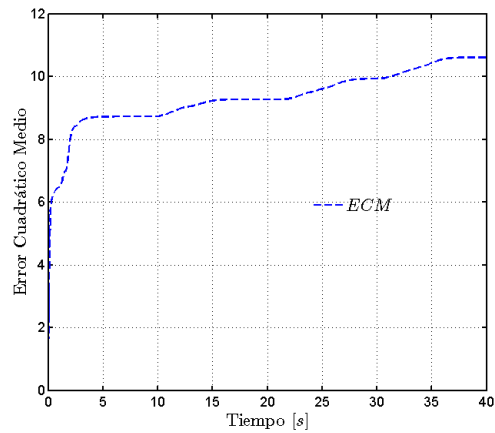


Figura 4.89: Error Cuadrático Medio a lo largo del tiempo de simulación.

La Figura 4.90 muestra la evolución de los pesos sinápticos a lo largo de la presente simulación.

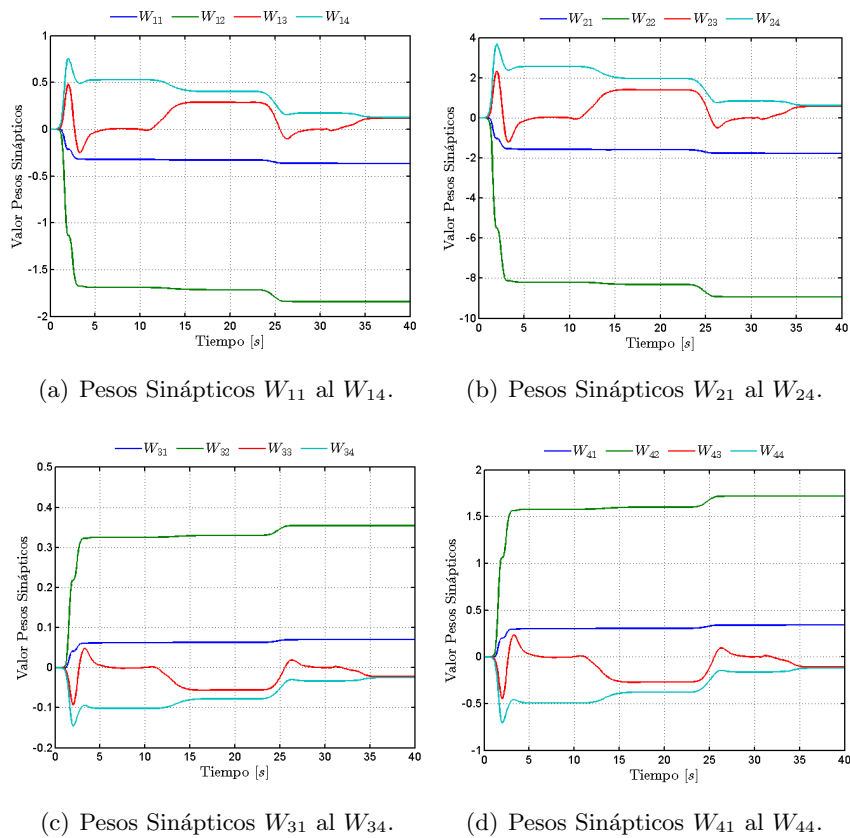


Figura 4.90: Pesos Sinápticos de la Red Neuronal Dinámica.

4.5. Implementación del Algoritmo de identificación en el Sistema Servo Modular

También se realizó la implementación del algoritmo de neuro-identificación en un sistema real, el Sistema Servo Modular (MSS por sus siglas en inglés) se encuentra en el Laboratorio de Procesos y Algoritmos de Control del Centro de Investigación en Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León.

El Sistema Servo Modular es un servomecanismo digital y un ambiente de software de arquitectura abierta para experimentos de control en tiempo real. El MSS es compatible con el diseño en tiempo real y la aplicación de métodos de control avanzados utilizando herramientas de MATLAB/Simulink. [72].

La Figura 4.91 ilustra la configuración de MSS, que consta de varios módulos montados en un riel de metal. Los módulos están colocados en el riel de tal manera que el motor de corriente continua con el módulo del Tacómetro es en la parte inicial y la caja de engranes con el disco de salida está en el extremo contrario, los módulos que lo componen son los siguientes:

- | | |
|----------------------|----------------------|
| 1. Potenciómetro | 5. Freno Magnético |
| 2. Tacómetro. | 6. Backlash. |
| 3. Motor DC de 24 V. | 7. Encoder. |
| 4. Carga Inercial. | 8. Caja de Engranes. |

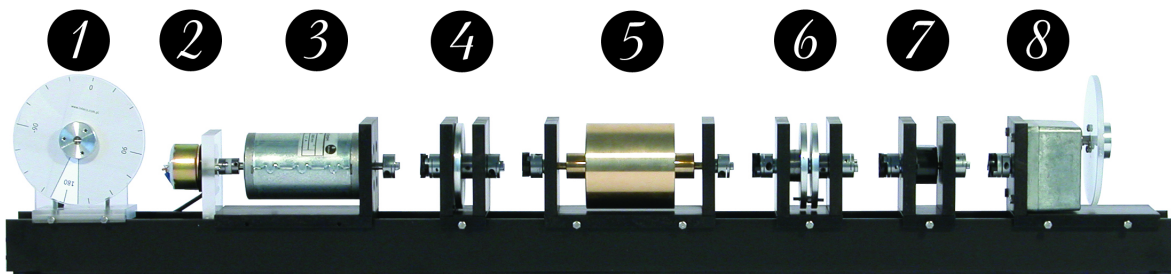


Figura 4.91: Fotografía del Sistema Servo Modular.

De acuerdo con el manual del prototipo [73], el modelo previsto solo toma en cuenta la dinámica de los módulos 3, 5 y 8, por lo que a la hora de la implementación se añade el módulo 4, con el objetivo de que la dinámica de este módulo sea la incertidumbre del sistema. Con dicho fin se diseñó una serie de experimentos, en los cuales se modifica la dinámica del sistema mediante la adición del módulo 4 con distintos perfiles de frenado, lo anterior mediante la modificación del módulo 4.

Las modificaciones realizadas al módulo 4 fueron la adición de distintas placas metálicas para modificar los perfiles del frenado del disco del módulo, dichas modificaciones permitieron la realización de 5 experimentos distintos para identificar la incertidumbre del sistema.

Los experimentos fueron:

1. Prueba sin el freno magnético.
2. Prueba con el freno magnético sin modificación.
3. Prueba con el freno magnético con un pequeño tornillo metálico cuyo efecto es frenado no lineal en una pequeña región de θ .
4. Prueba con el freno Magnético con la placa ferromagnética de forma triangular cuyo efecto hace las veces de una incertidumbre de un frenado no lineal en un segmento de θ .
5. Prueba con el freno magnético con una placa metálica que cubre en su totalidad el disco del módulo.

El modelo del sistema para la creación de la red neuronal en su representación en espacio de estados es el siguiente:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_s} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_s}{T_s} \end{bmatrix} u , \quad (4.51)$$

Cabe mencionar que el modelo (4.51) es un modelo lineal, por lo que a la incertidumbre del sistema se le añade la posibilidad de dinámicas no modeladas.

Considerando una Red Neuronal Dinámica de la forma (3.4) tenemos:

$$\dot{\hat{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_s} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + W\varphi(\hat{x}) + \psi + \begin{bmatrix} 0 \\ \frac{K_s}{T_s} \end{bmatrix} u + L(x - \hat{x}) , \quad (4.52)$$

Entonces aplicando el teorema 3.1 primero se selecciona al umbral neuronal ψ como

$$\psi = \text{sign}(e) \cdot \delta_f(x) , \quad (4.53)$$

donde la cota superior de la incertidumbre del sistema está definida como

$$\delta_f(x) = \|\Delta_f(x)\| , \quad (4.54)$$

como se tiene disponibles todos los estados del sistema a la salida, entonces podemos saber el valor aproximado de Δ_f de la siguiente forma: $\Delta_f(x) = f(x) - f_0(x)$.

y también se elige a la función de activación $\varphi(\hat{x})$ como

$$\varphi(\hat{x}) = \hat{x} + 0.05 \cdot \tanh(\hat{x}). \quad (4.55)$$

En las Figuras 4.92 - 4.96 se muestran los perfiles de la posición angular del servomotor en cada uno de los experimentos realizados.

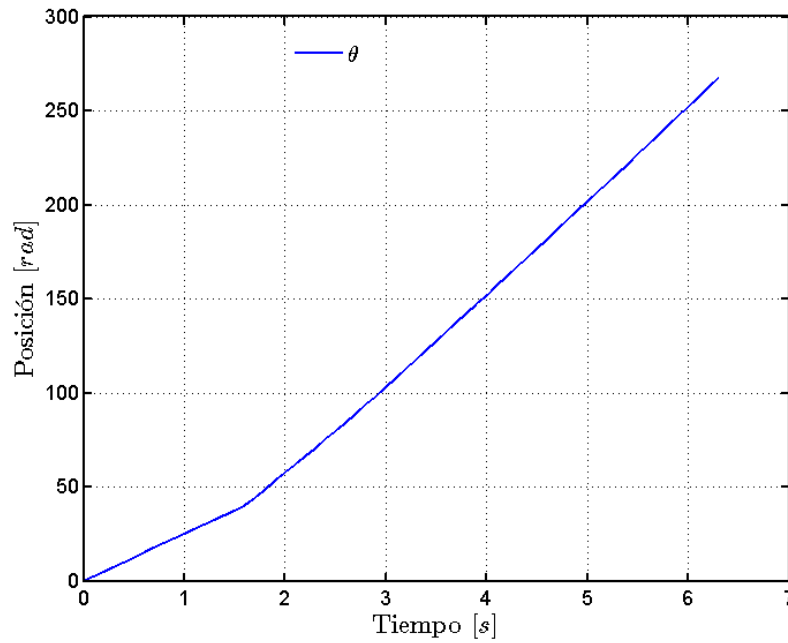


Figura 4.92: Posición Angular en Experimento 1.

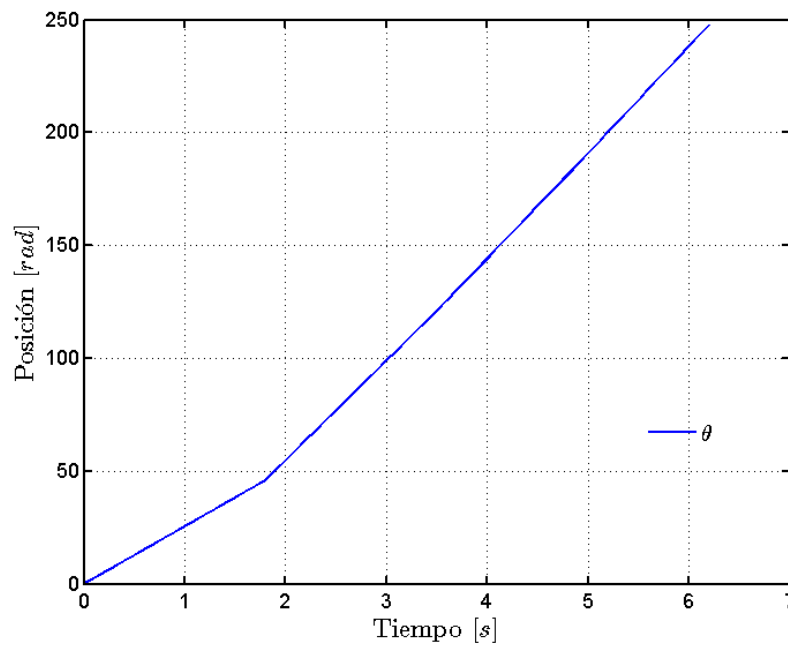


Figura 4.93: Posición Angular en Experimento 2.

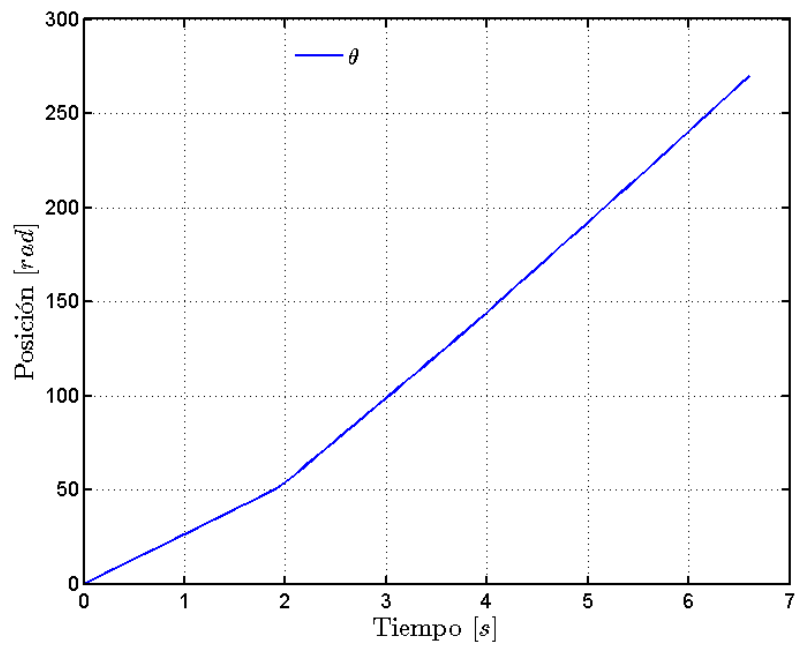


Figura 4.94: Posición Angular en Experimento 3.

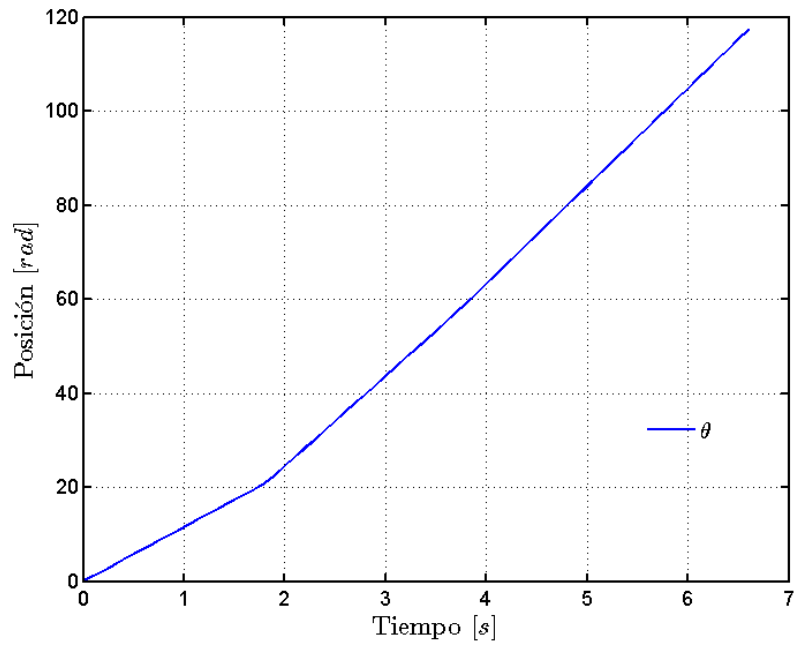


Figura 4.95: Posición Angular en Experimento 4.

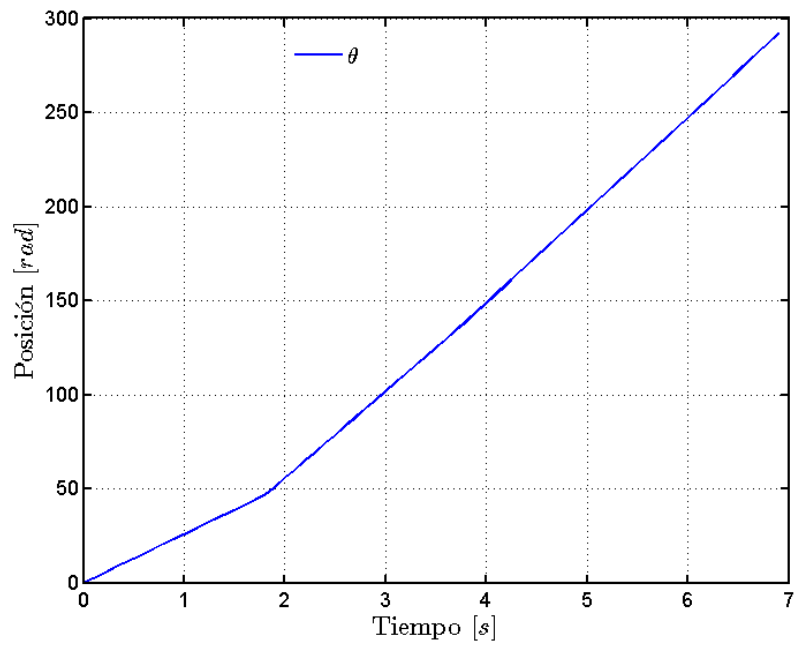


Figura 4.96: Posición Angular en Experimento 5.

En las Figuras 4.97 - 4.101 se pueden observar las gráficas referentes a la velocidad angular del servomotor en cada uno de los experimentos.

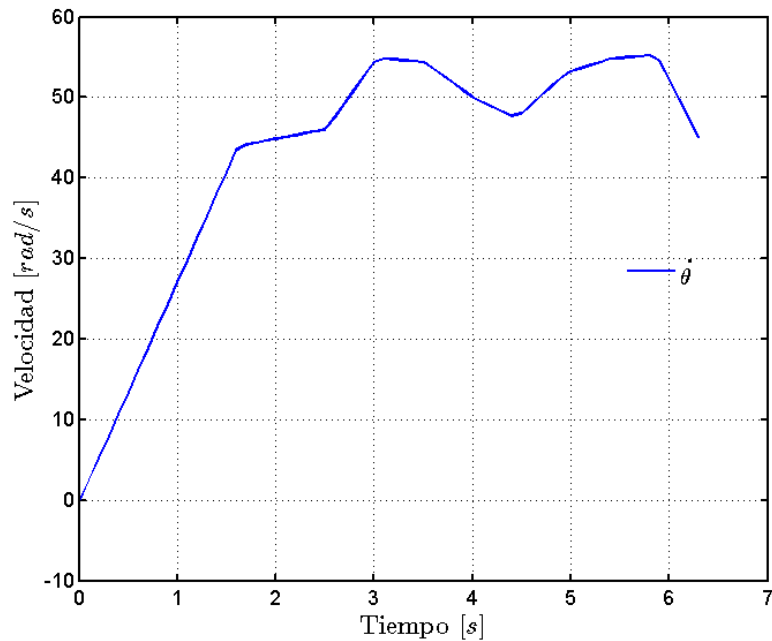


Figura 4.97: Velocidad Angular en Experimento 1.

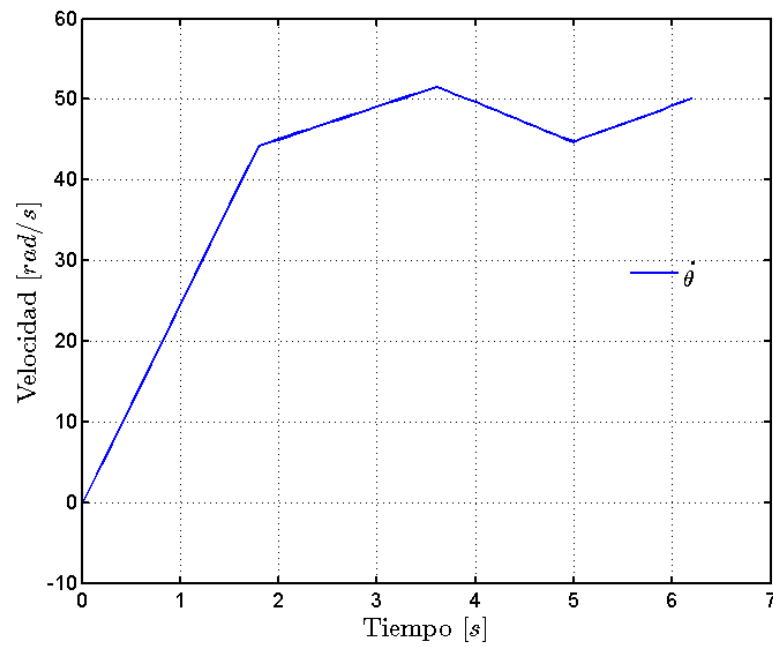


Figura 4.98: Velocidad Angular en Experimento 2.

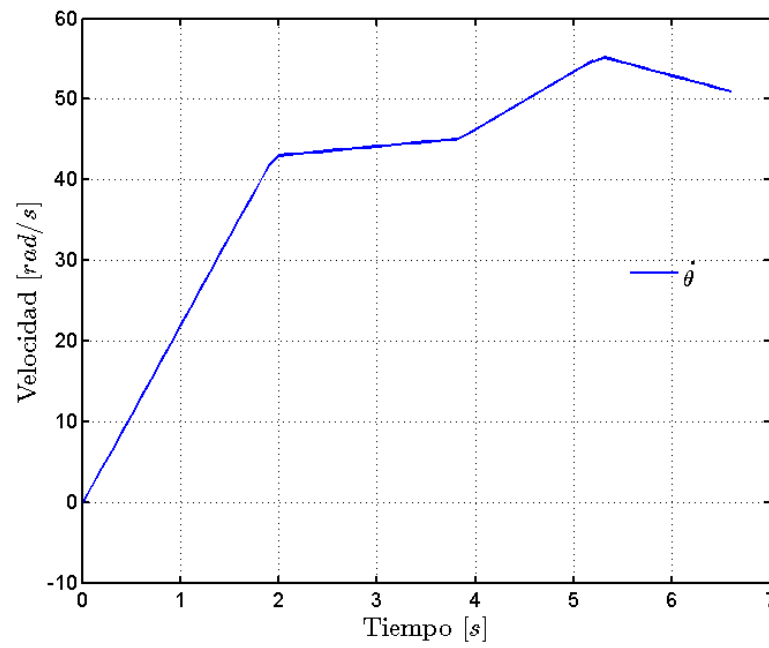


Figura 4.99: Velocidad Angular en Experimento 3.

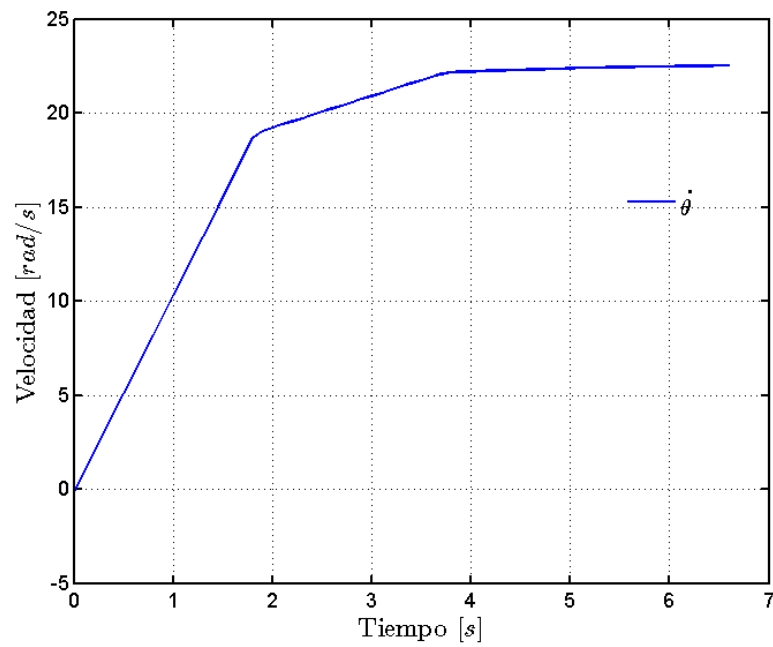


Figura 4.100: Velocidad Angular en Experimento 4.

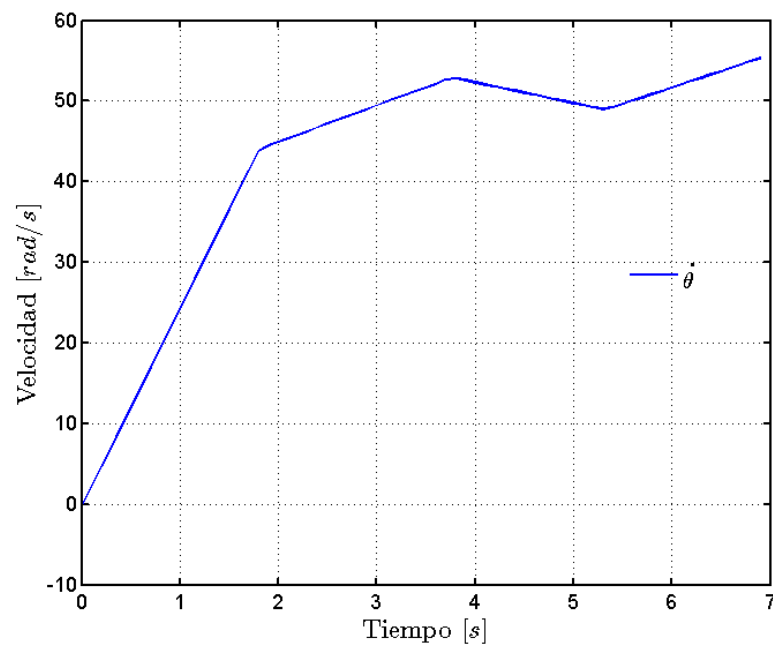


Figura 4.101: Velocidad Angular en Experimento 5.

En las Figuras 4.102 - 4.106 se muestran los perfiles de la posición angular estimada por la red neuronal en cada uno de los experimentos.

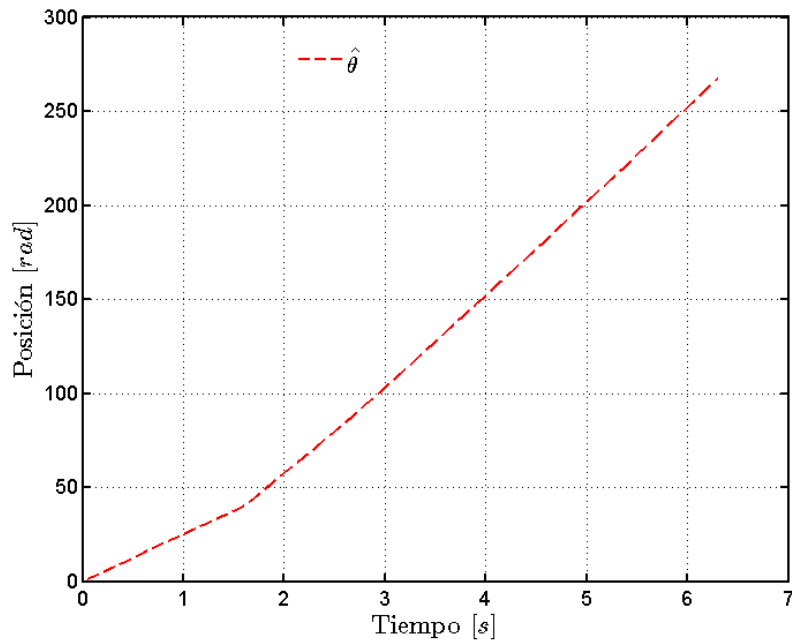


Figura 4.102: Posición Angular Estimada en Experimento 1.

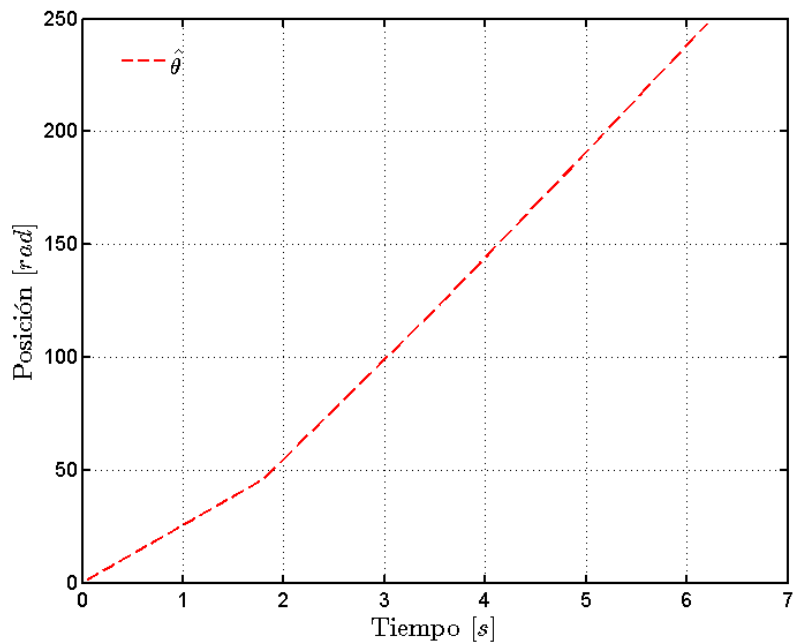


Figura 4.103: Posición Angular Estimada en Experimento 2.

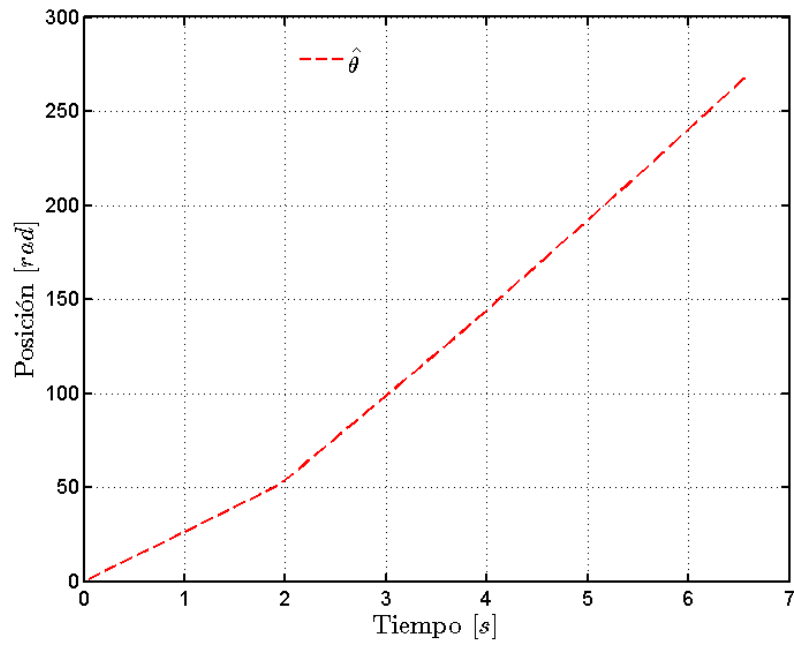


Figura 4.104: Posición Angular Estimada en Experimento 3.

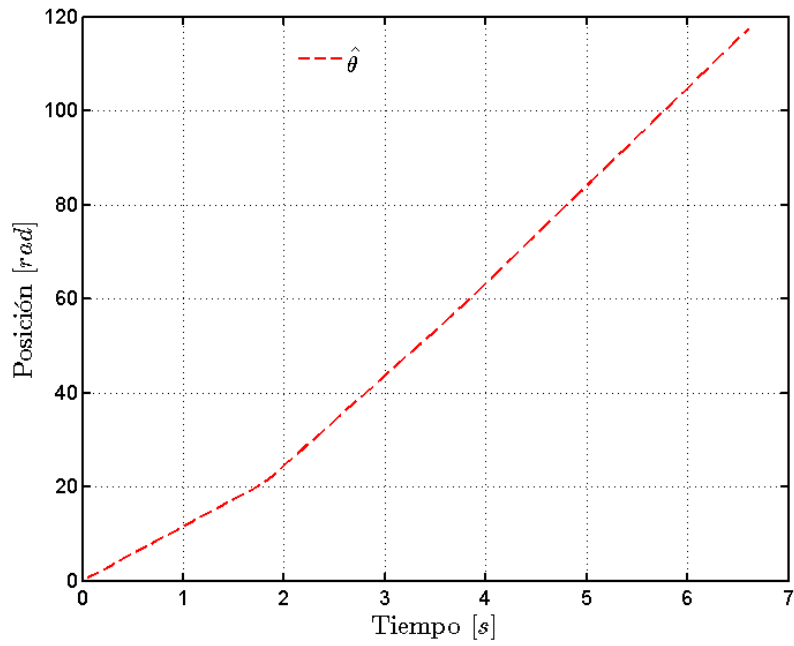


Figura 4.105: Posición Angular Estimada en Experimento 4.

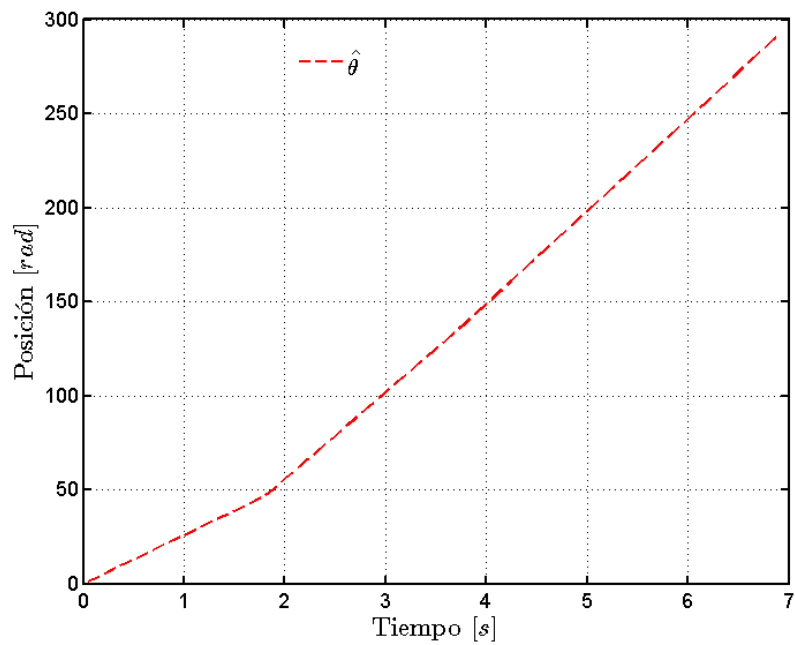


Figura 4.106: Posición Angular Estimada en Experimento 5.

En las Figuras 4.107 - 4.111 se pueden observar las gráficas referentes a la velocidad angular estimada por la red neuronal para cada uno de los experimentos.

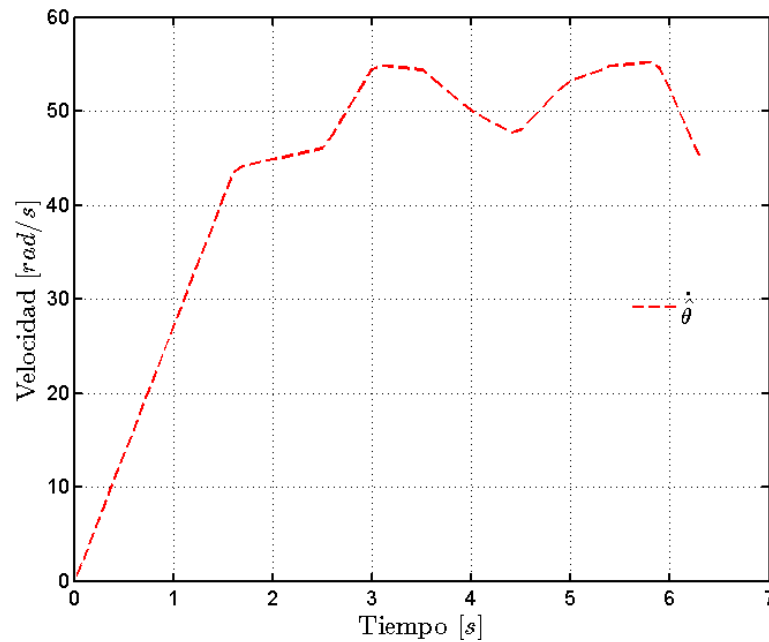


Figura 4.107: Velocidad Angular Estimada en Experimento 1.

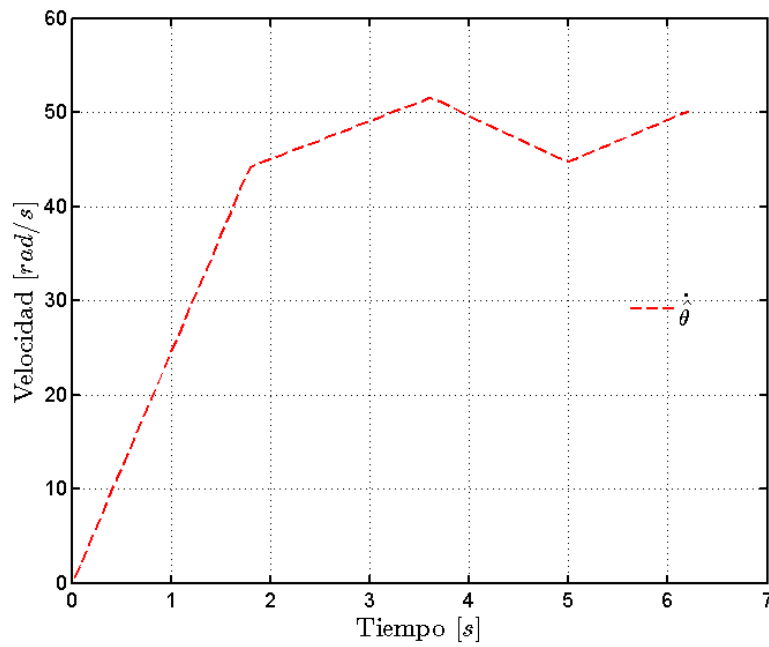


Figura 4.108: Velocidad Angular Estimada en Experimento 2.

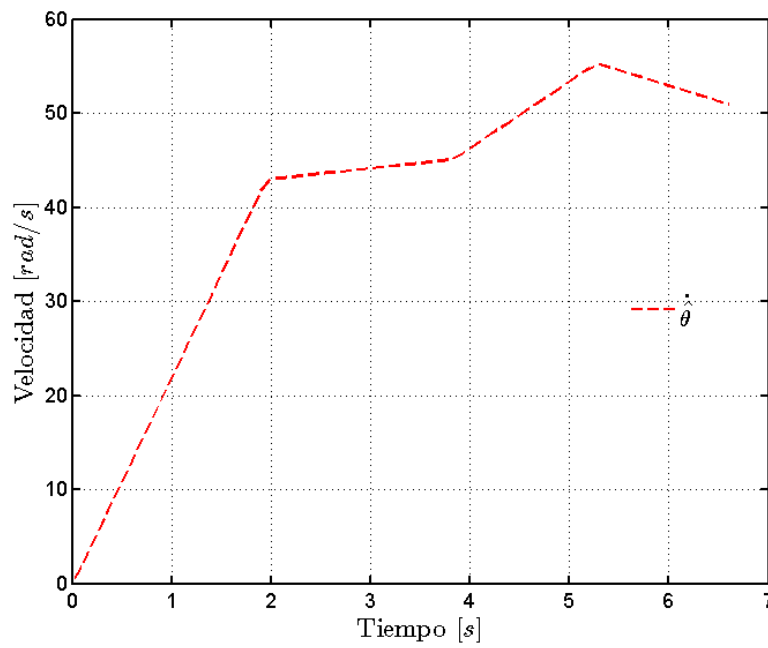


Figura 4.109: Velocidad Angular Estimada en Experimento 3.

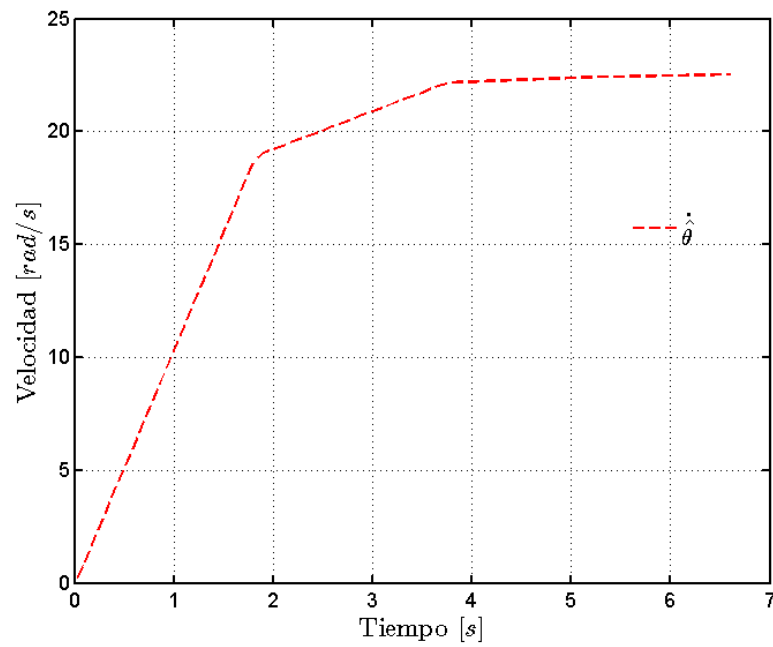


Figura 4.110: Velocidad Angular Estimada en Experimento 4.

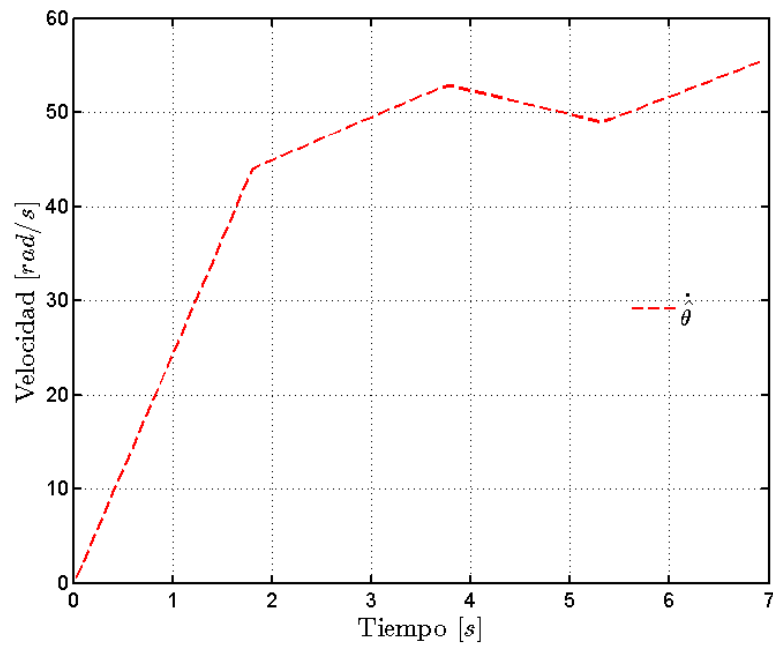


Figura 4.111: Velocidad Angular Estimada en Experimento 5.

En las Figuras 4.112 - 4.116 se muestra la comparación entre la posición angular real y estimada para cada uno de los experimentos realizados, se puede observar que para cada uno de ellos los perfiles son muy similares.

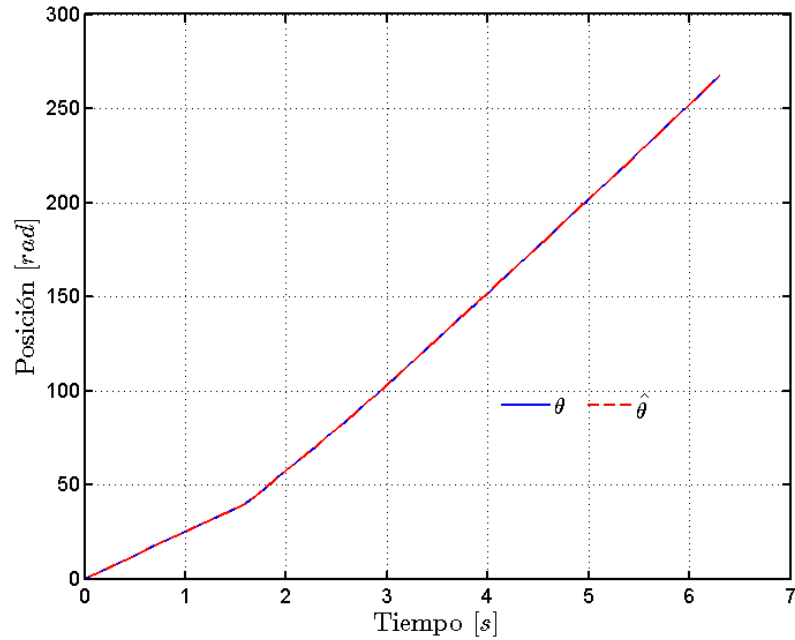


Figura 4.112: Comparación de Posición en el Experimento 1.

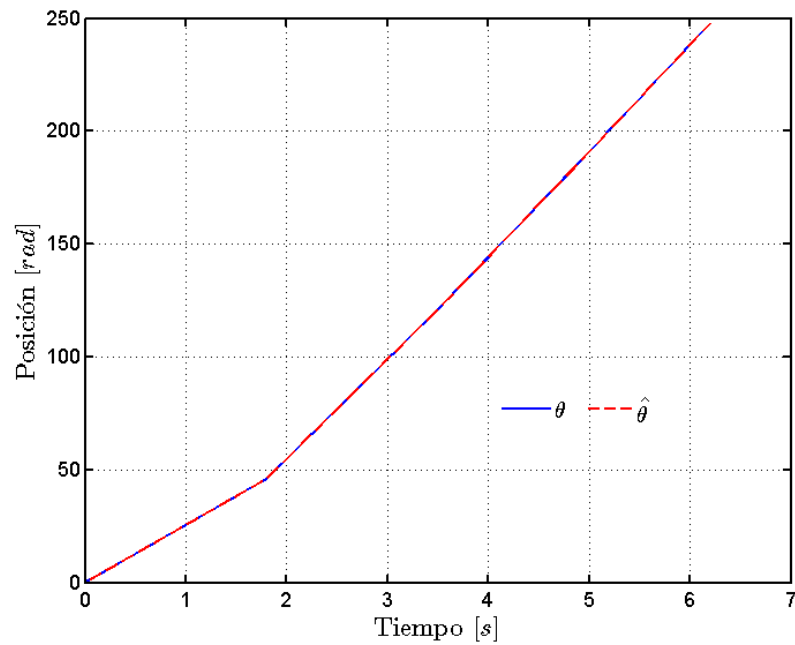


Figura 4.113: Comparación de Posición en el Experimento 2.

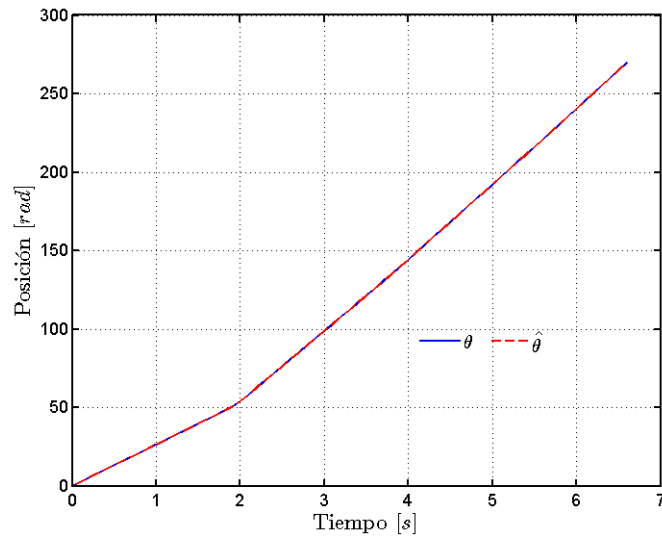


Figura 4.114: Comparación de Posición en el Experimento 3.

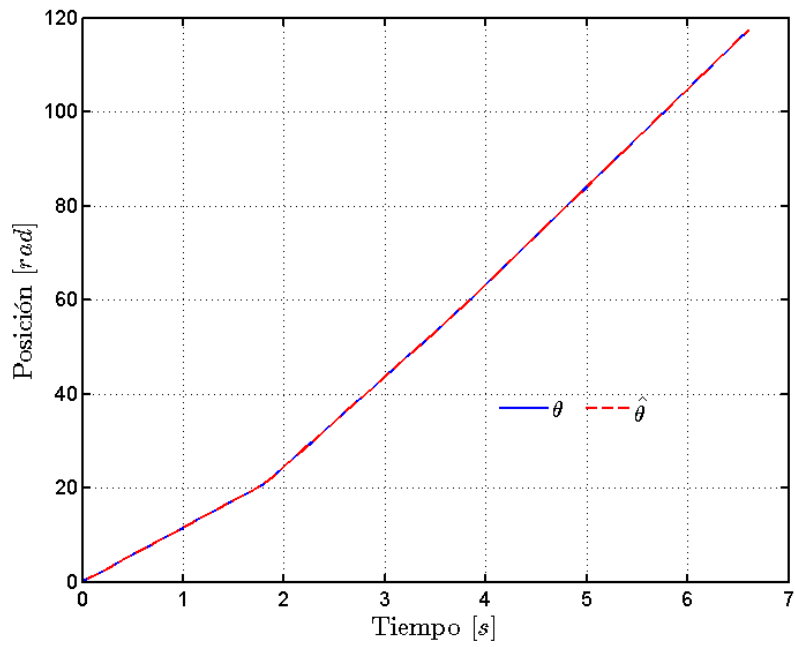


Figura 4.115: Comparación de Posición en el Experimento 4.

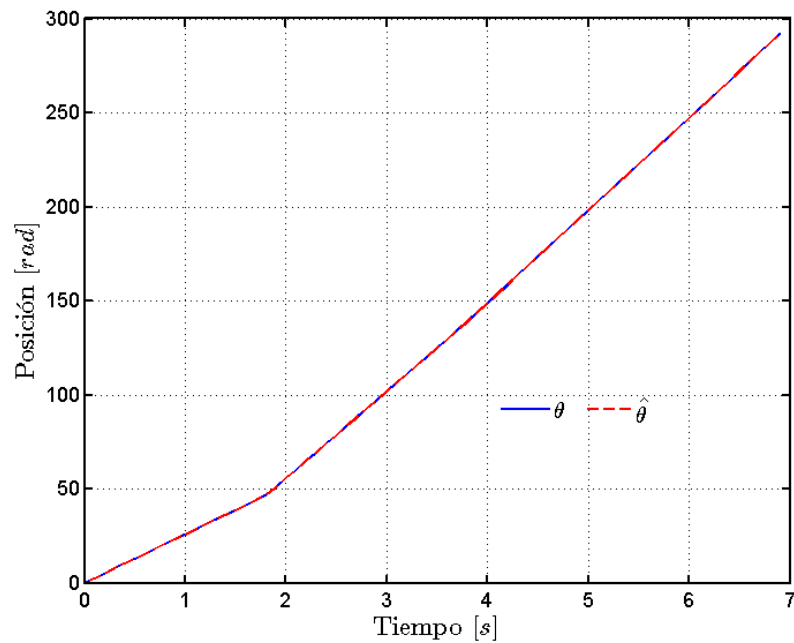
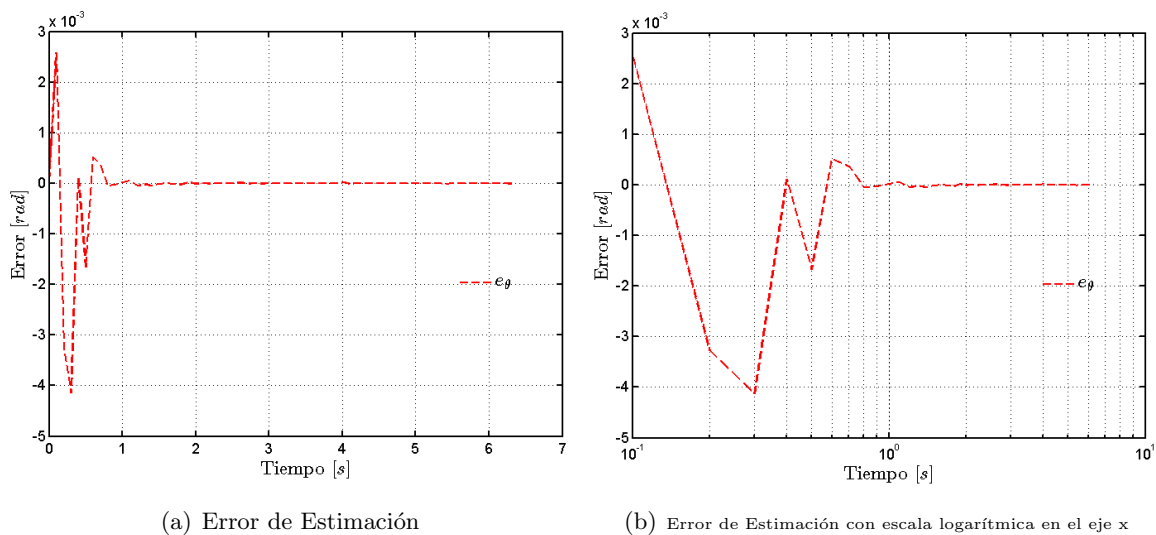


Figura 4.116: Comparación de Posición en el Experimento 5.

Las Figuras 4.117 - 4.121 se muestran los errores de estimación de la posición angular para cada uno de los experimentos, donde se observa que se tienen errores relativamente pequeños.



(a) Error de Estimación

(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.117: Error de Estimación de posición angular en el Experimento 1

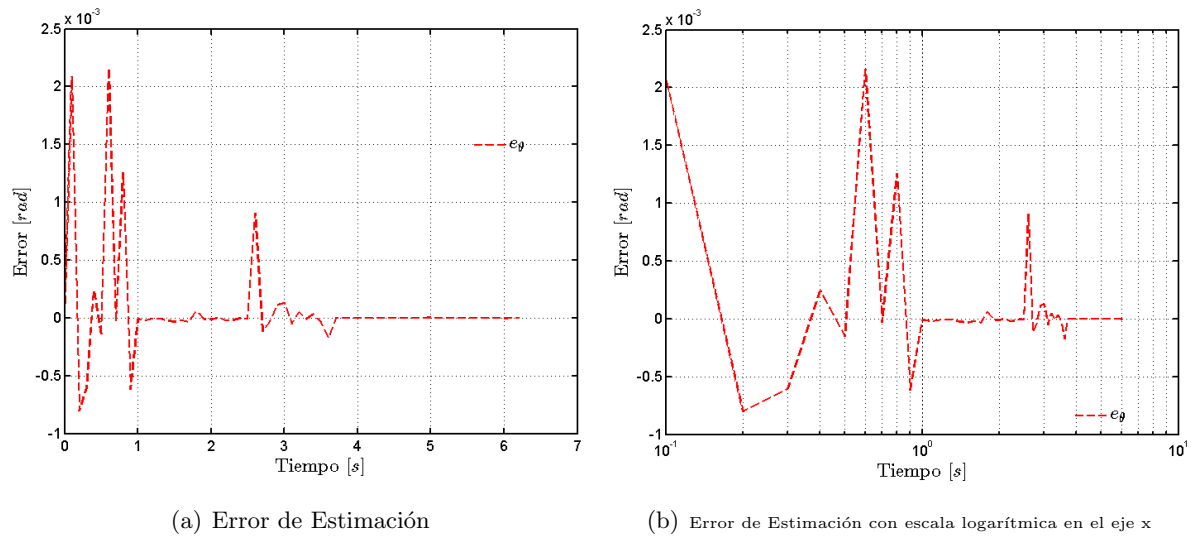


Figura 4.118: Error de Estimación de posición angular en el Experimento 2

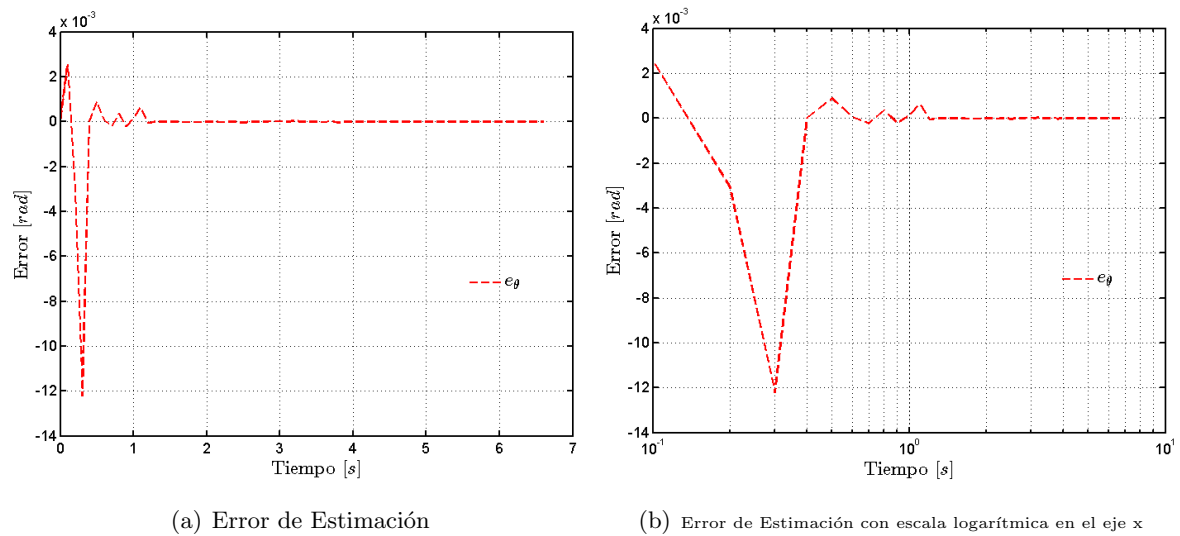
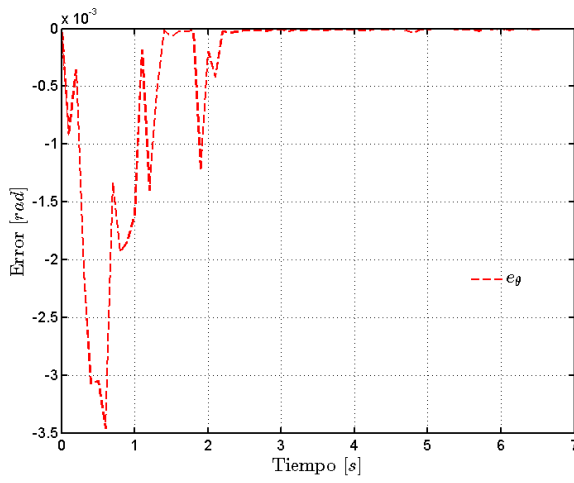
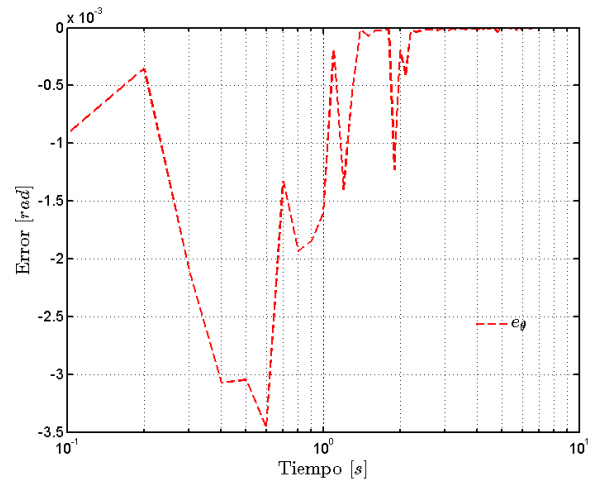


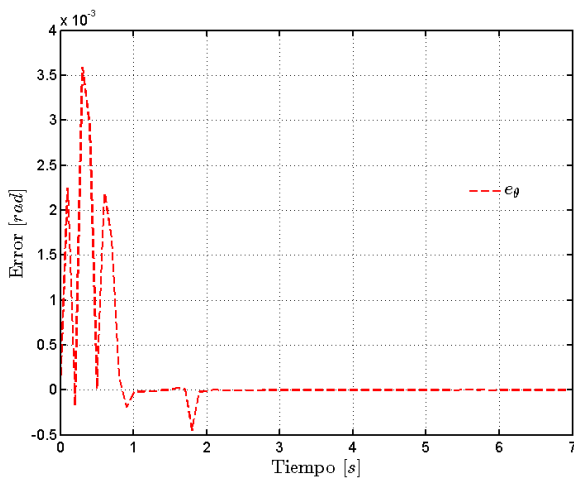
Figura 4.119: Error de Estimación de posición angular en el Experimento 3



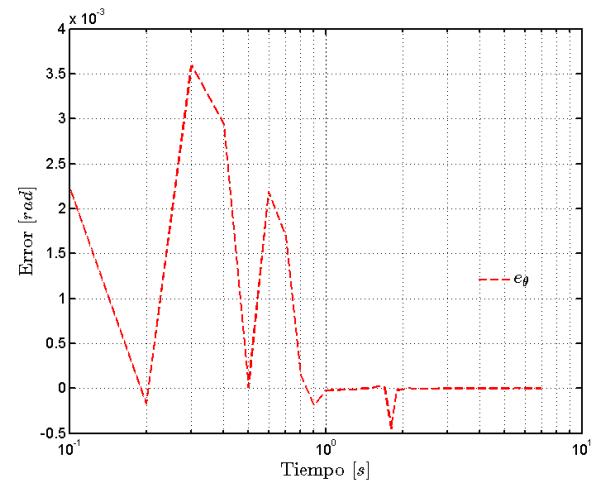
(a) Error de Estimación



(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.120: Error de Estimación de posición angular en el Experimento 4

(a) Error de Estimación



(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.121: Error de Estimación de posición angular en el Experimento 5

En las Figuras 4.122 - 4.126 se muestra la comparación entre la velocidad angular real y estimada para cada uno de los experimentos, se puede ver que para cada uno de ellos son casi idénticos.

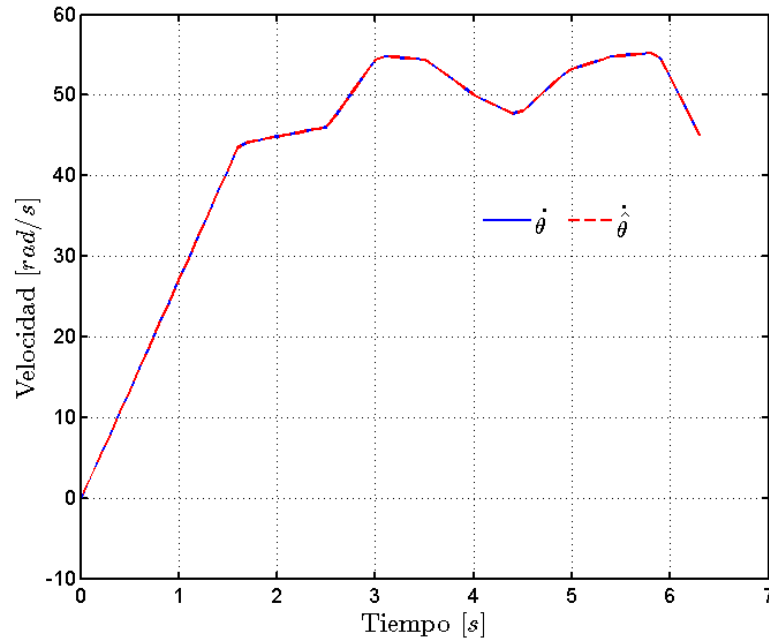


Figura 4.122: Comparación de Velocidad en el Experimento 1.

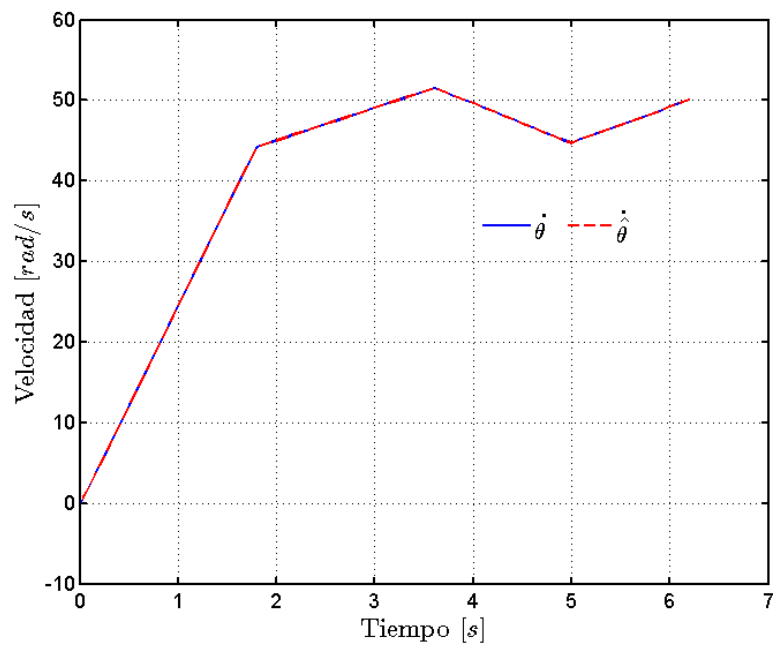


Figura 4.123: Comparación de Velocidad en el Experimento 2.

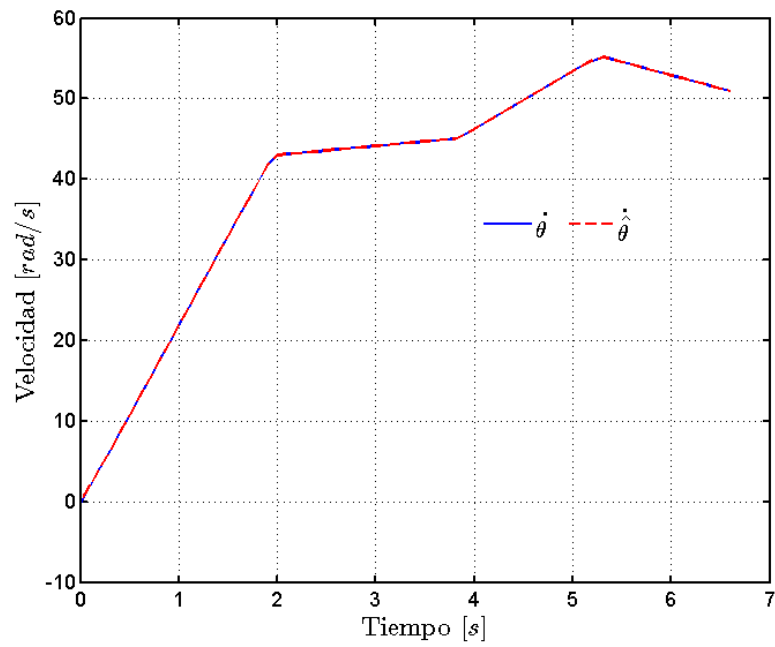


Figura 4.124: Comparación de Velocidad en el Experimento 3.

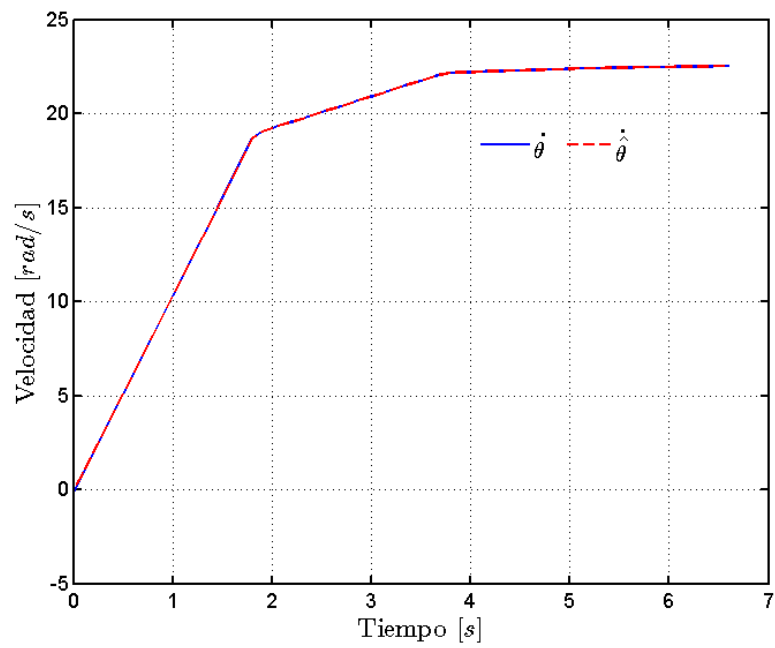


Figura 4.125: Comparación de Velocidad en el Experimento 4.

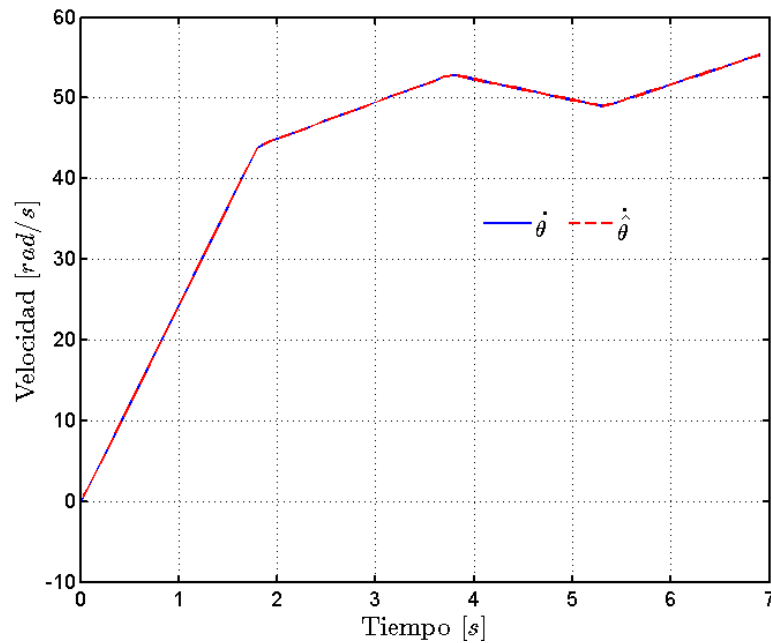
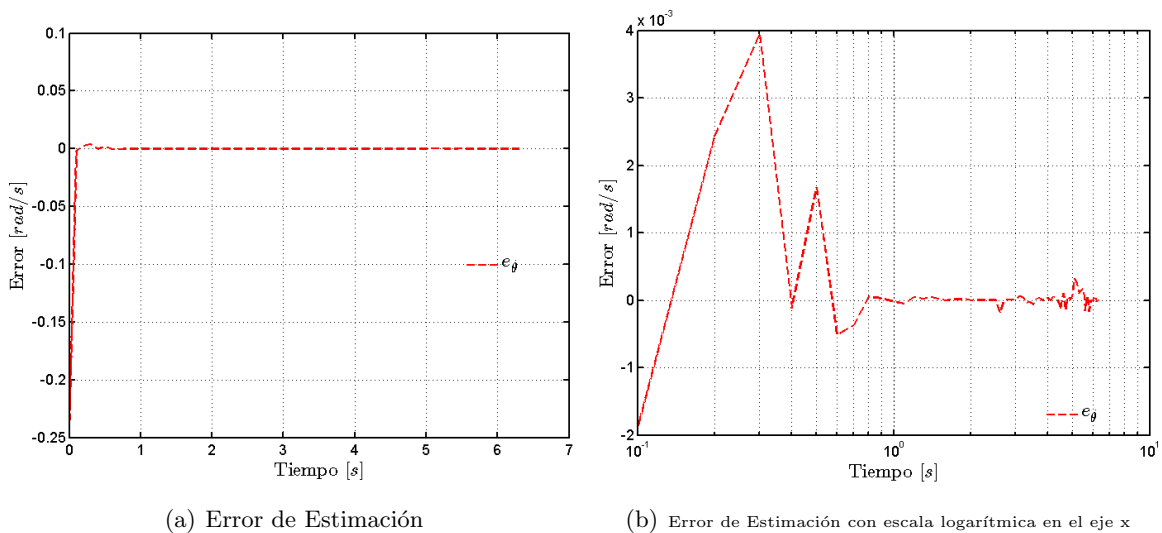


Figura 4.126: Comparación de Velocidad en el Experimento 5.

En las Figuras 4.127 - 4.131 se muestran los errores de estimación de la velocidad angular para cada uno de los experimentos, donde se puede ver que el error converge a valores muy cercanos a 0.



(a) Error de Estimación

(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.127: Error de Estimación de velocidad angular en el Experimento 1

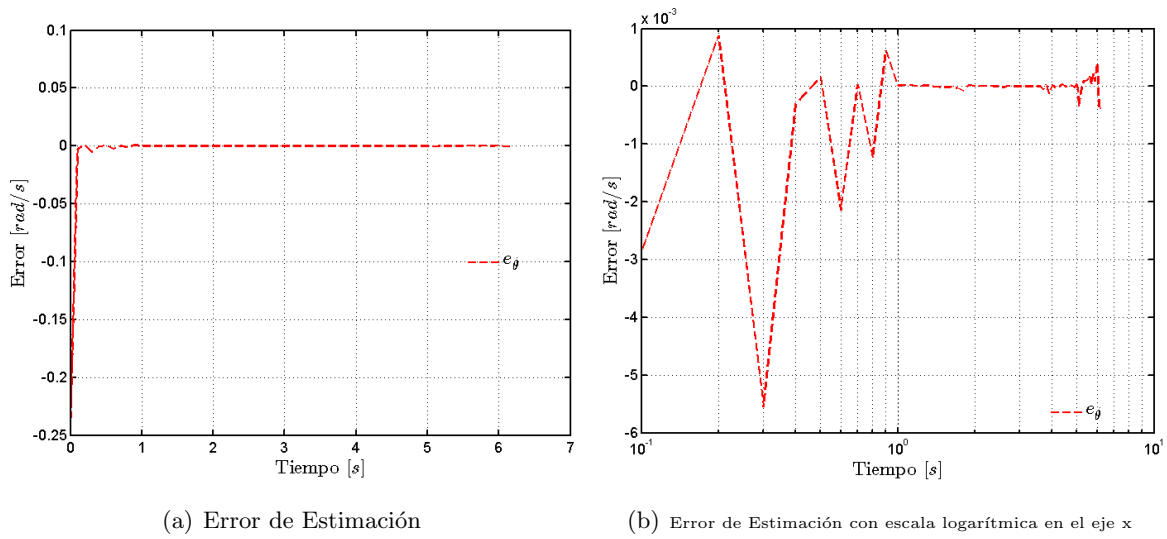


Figura 4.128: Error de Estimación de velocidad angular en el Experimento 2

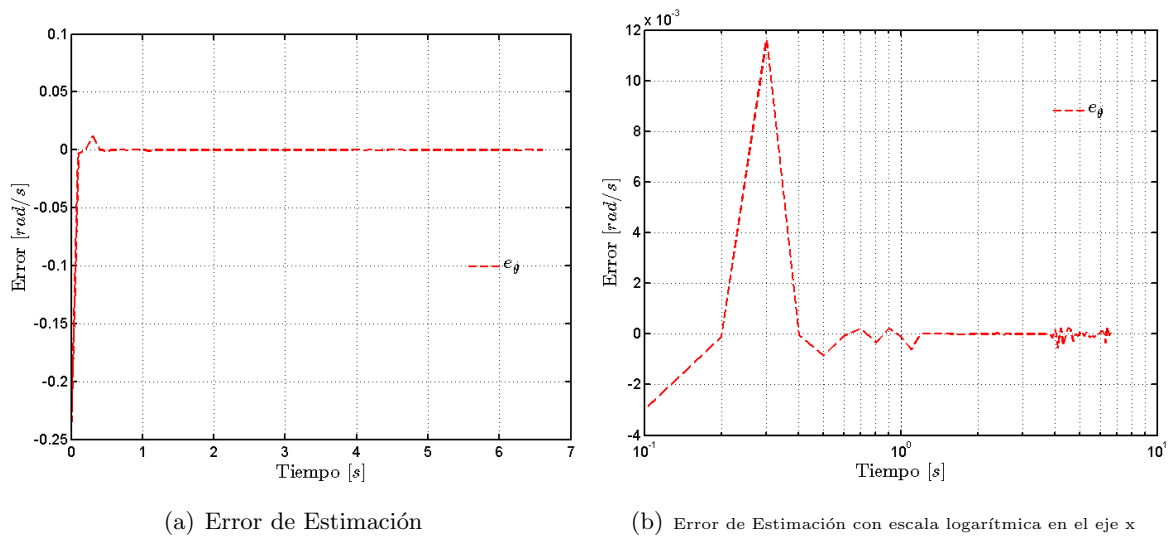


Figura 4.129: Error de Estimación de velocidad angular en el Experimento 3

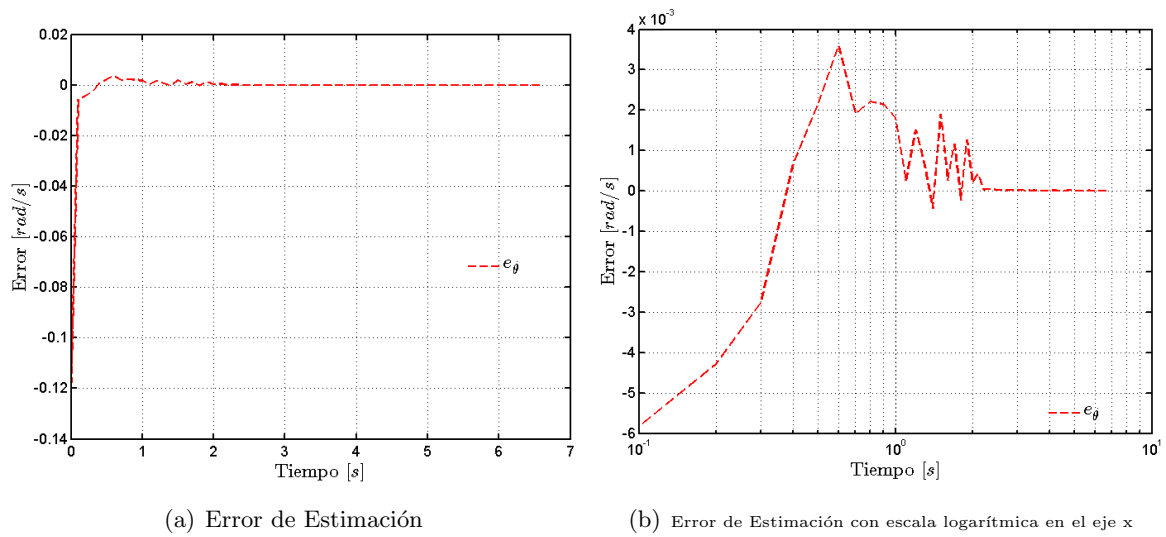


Figura 4.130: Error de Estimación de velocidad angular en el Experimento 4

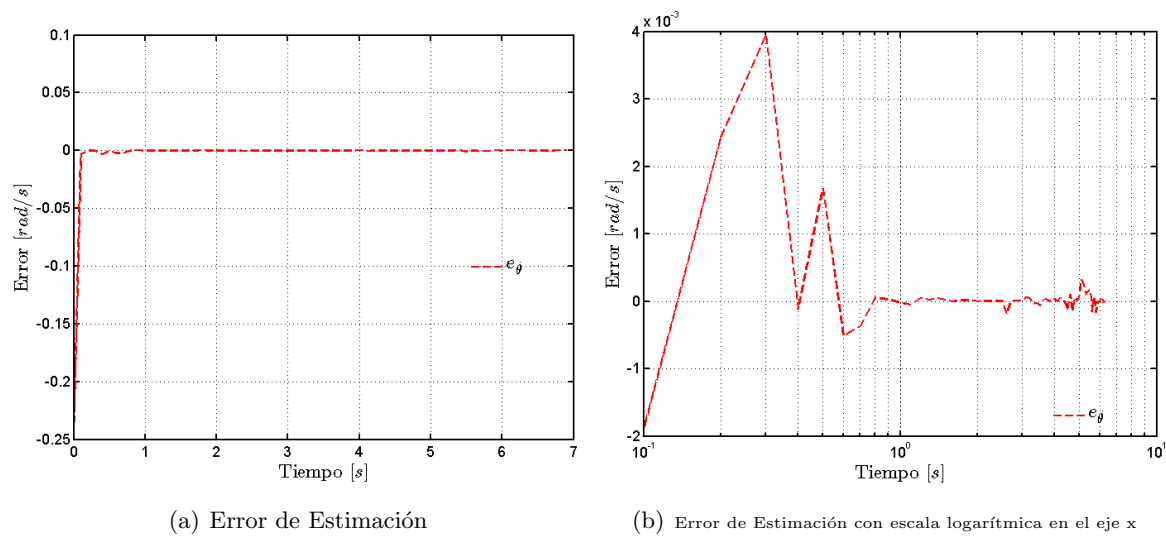


Figura 4.131: Error de Estimación de velocidad angular en el Experimento 5

El perfil de la incertidumbre en los 5 experimentos se muestra en las Figuras 4.132 - 4.136.

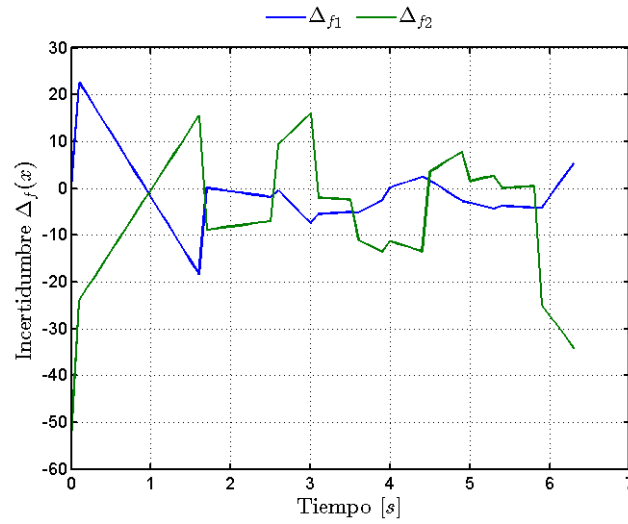


Figura 4.132: Incertidumbre del Sistema Servo Modular en el Experimento 1.

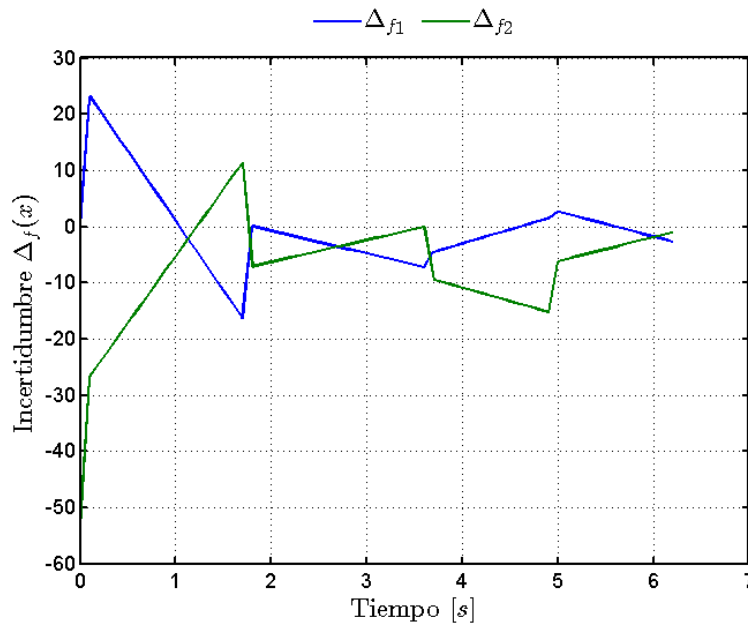


Figura 4.133: Incertidumbre del Sistema Servo Modular en el Experimento 2.

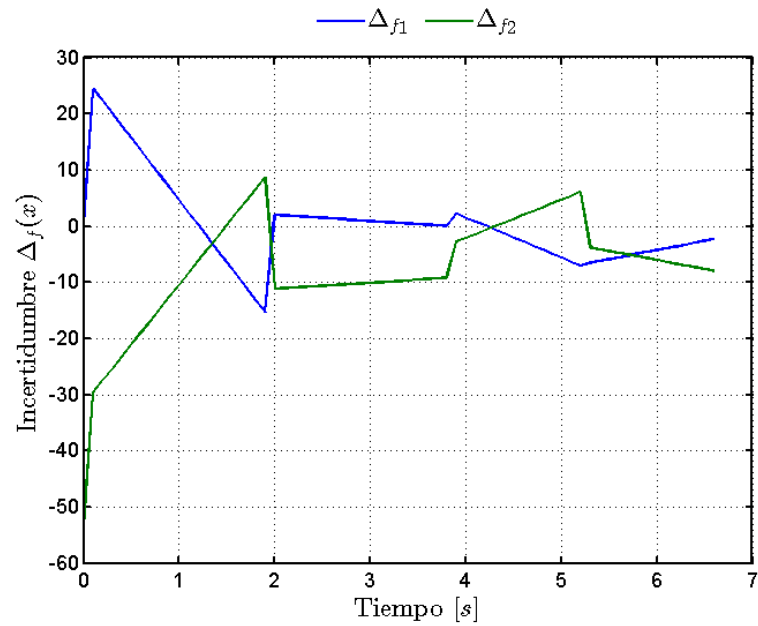


Figura 4.134: Incertidumbre del Sistema Servo Modular en el Experimento 3.

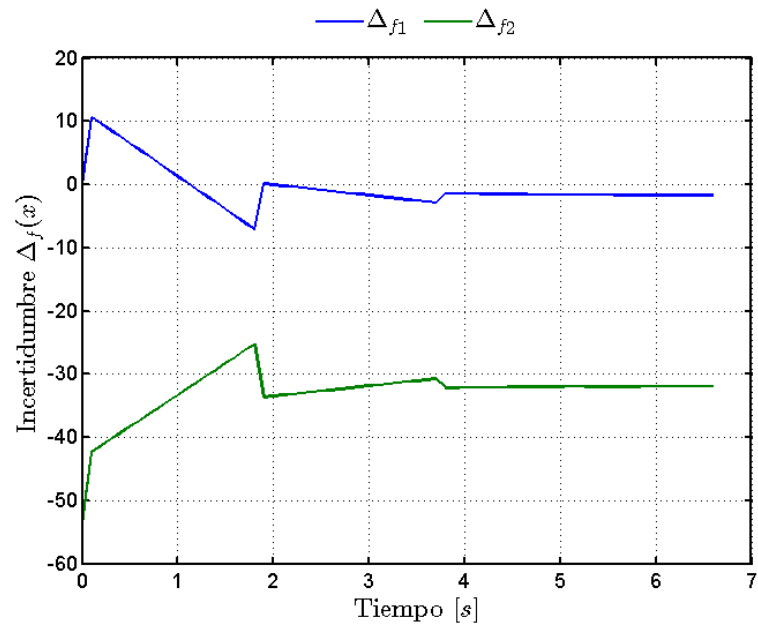


Figura 4.135: Incertidumbre del Sistema Servo Modular en el Experimento 4.

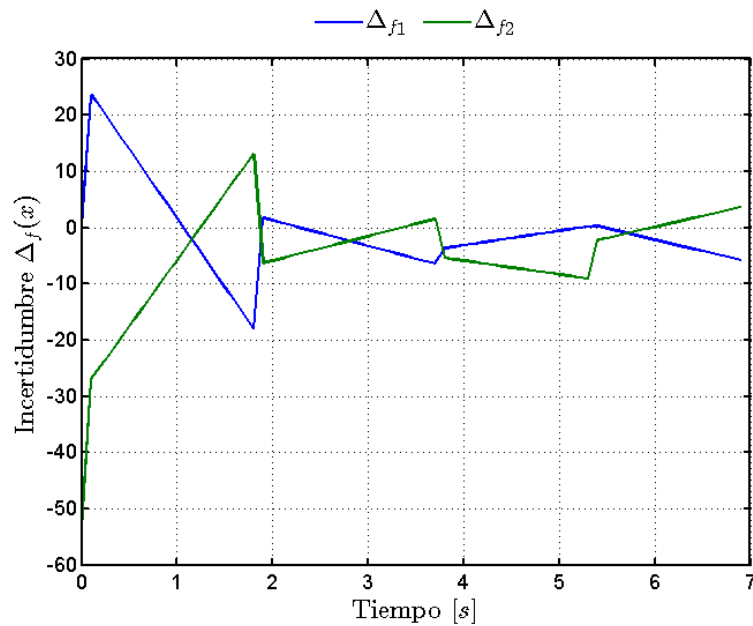


Figura 4.136: Incertidumbre del Sistema Servo Modular en el Experimento 5.

La componente neuronal $W\varphi(\hat{x})$ resultante de la Red Neuronal Dinámica de cada experimento se observa en las Figuras 4.137 - 4.141

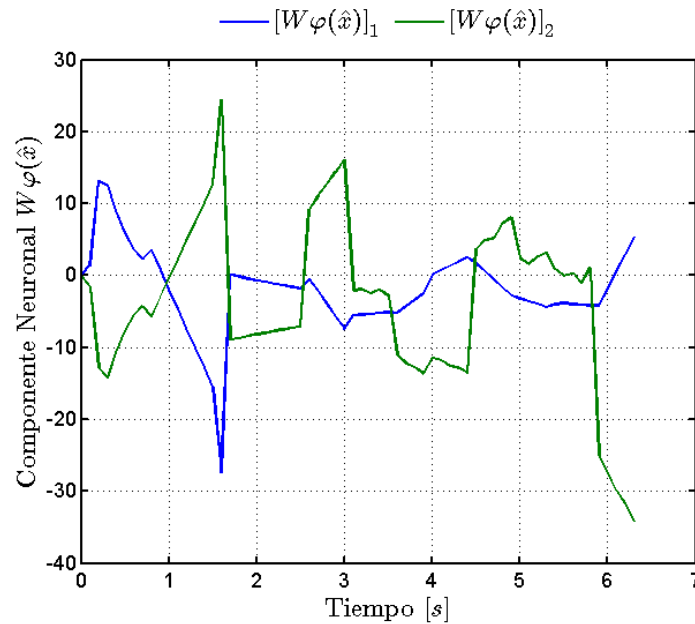
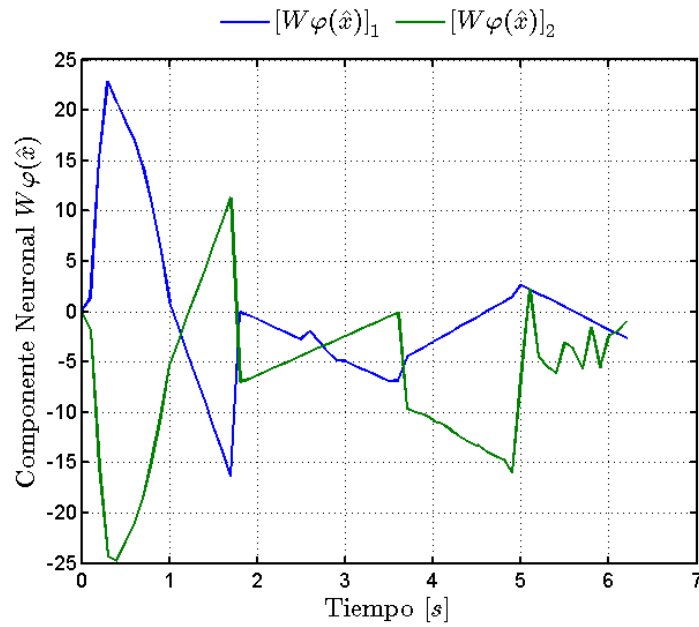
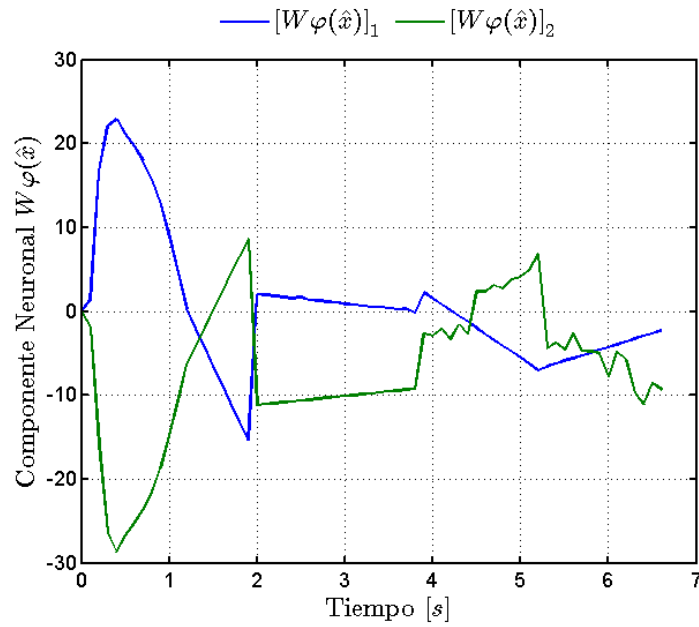


Figura 4.137: Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 1.

Figura 4.138: Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 2.Figura 4.139: Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 3.

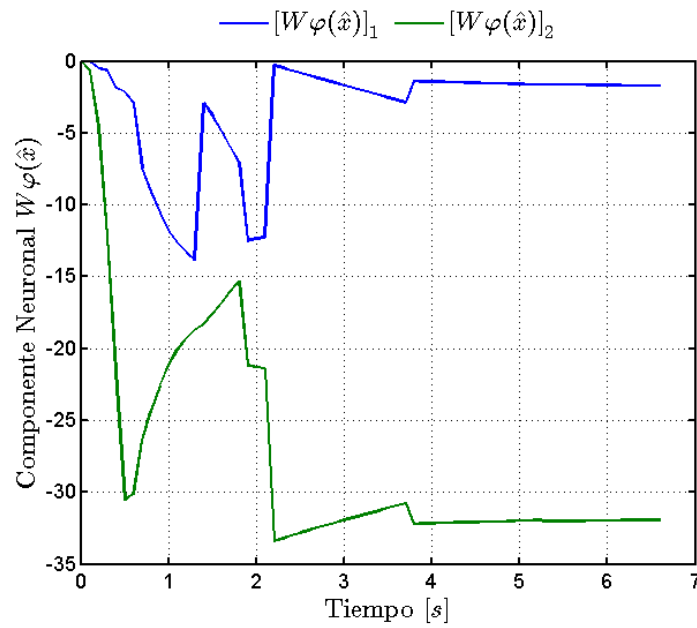


Figura 4.140: Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 4.

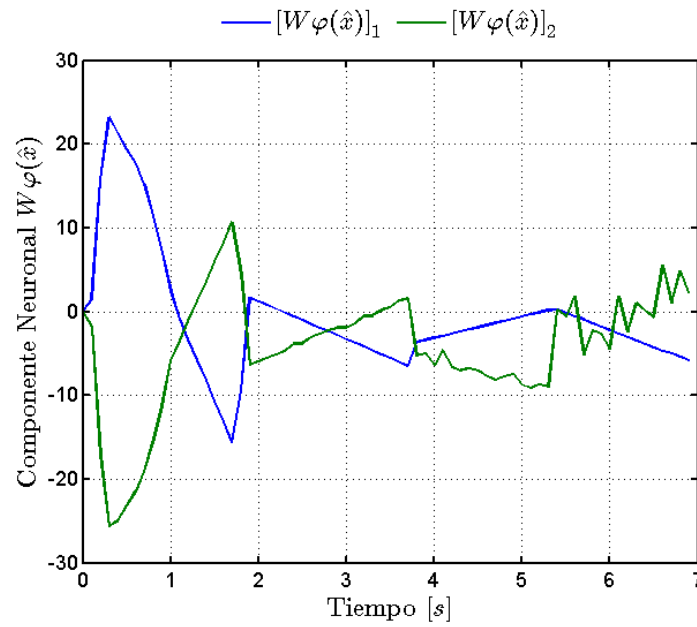


Figura 4.141: Componente Neuronal $W\varphi(\hat{x})$ en el Experimento 5.

A continuación en las Figuras 4.142 - 4.146 se muestra la comparación entre Δ_f y $W\varphi(\hat{x})$ para cada uno de los experimentos realizados.

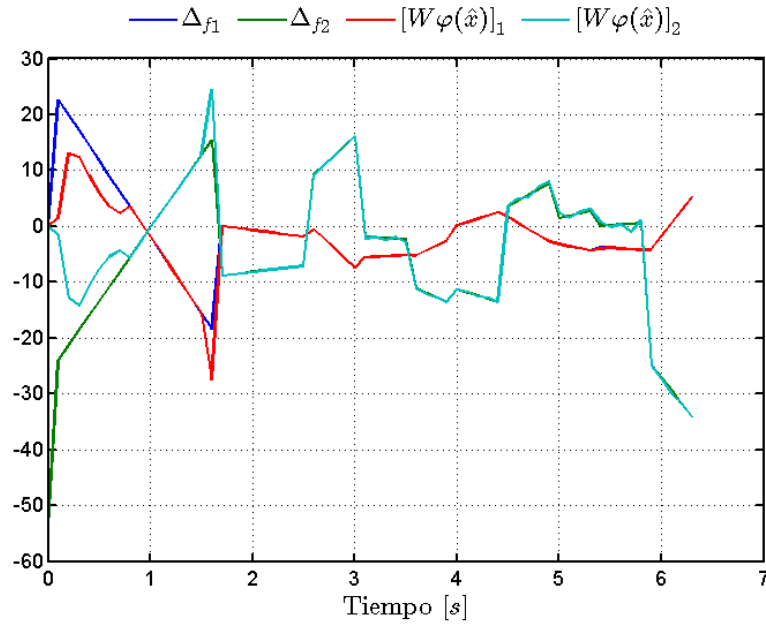


Figura 4.142: Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 1.

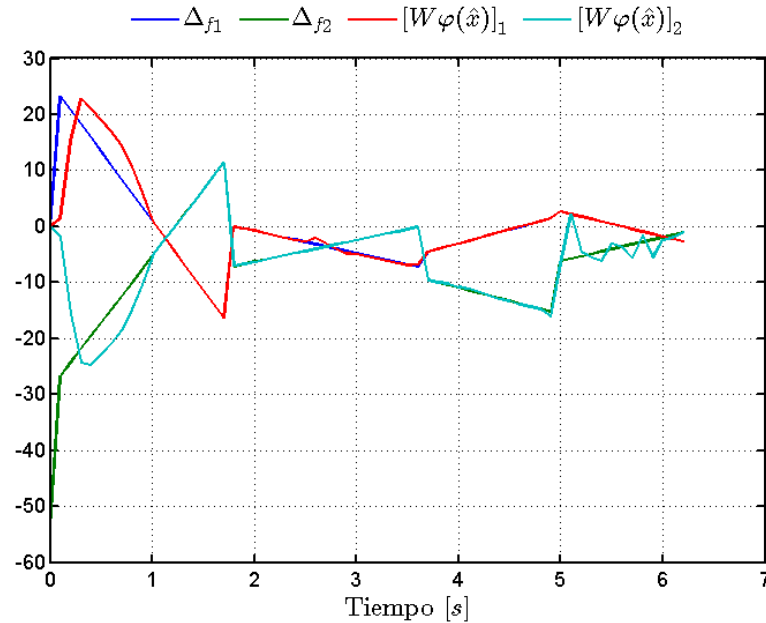


Figura 4.143: Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 2.

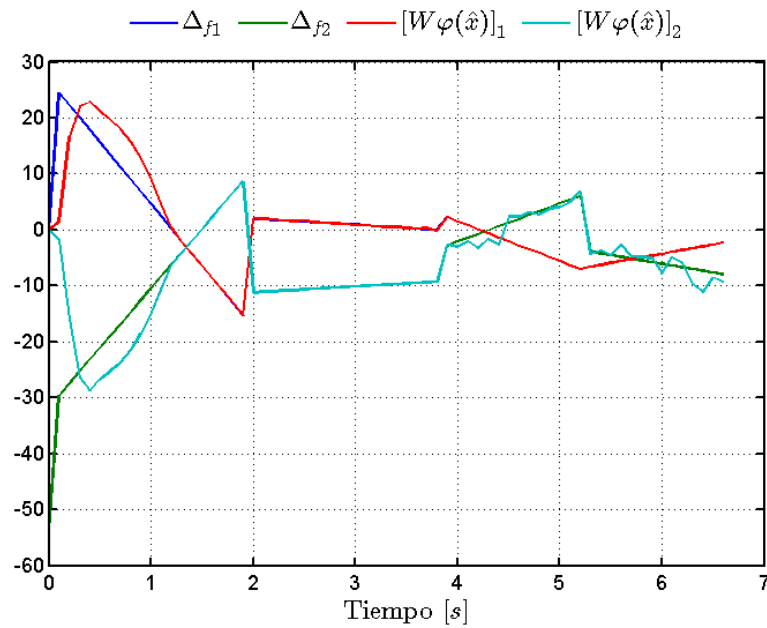


Figura 4.144: Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 3.

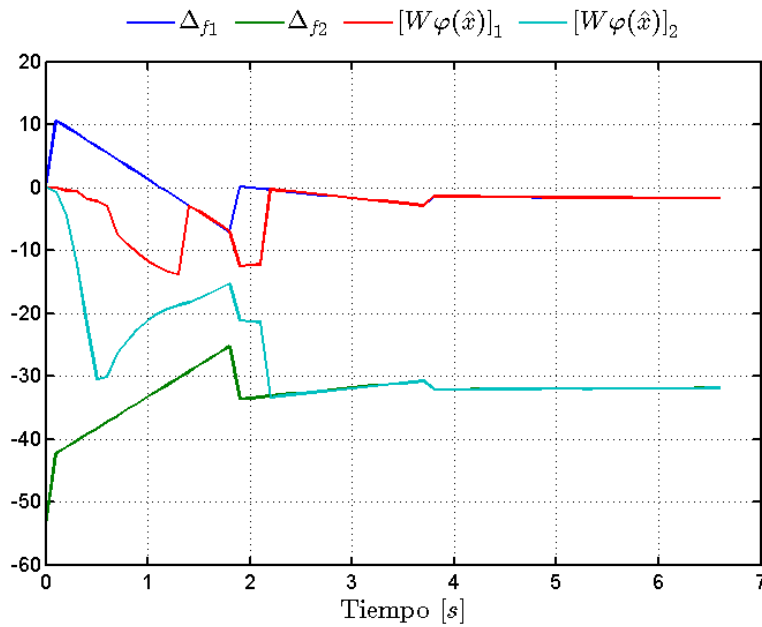


Figura 4.145: Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 4.

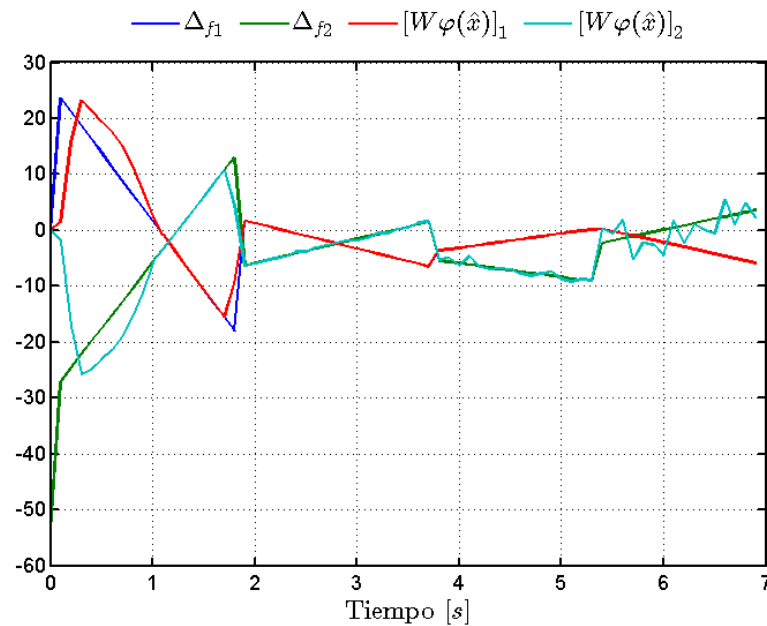
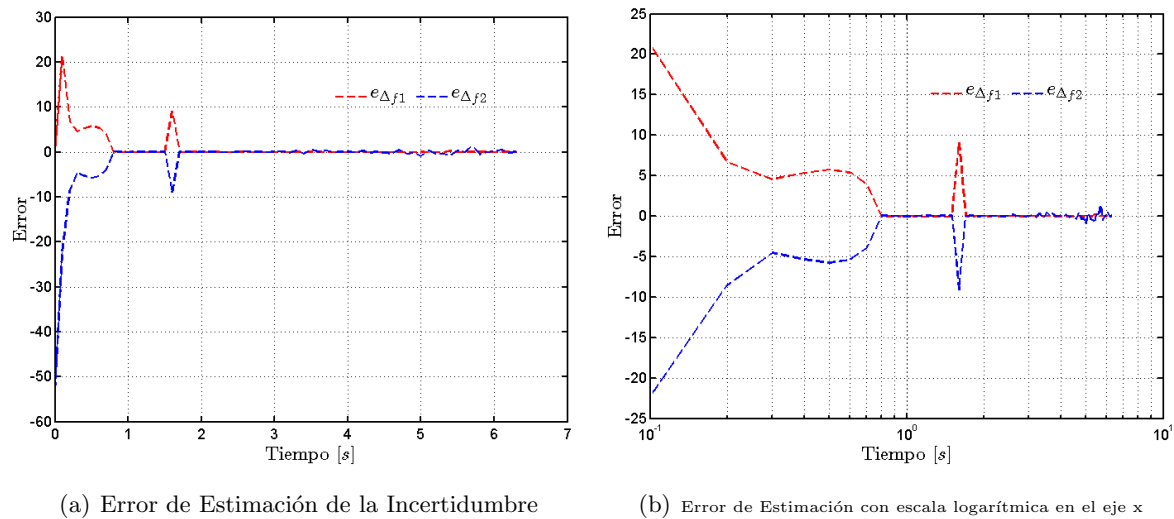


Figura 4.146: Comparación de Δ_f y $W\varphi(\hat{x})$ en el Experimento 5.

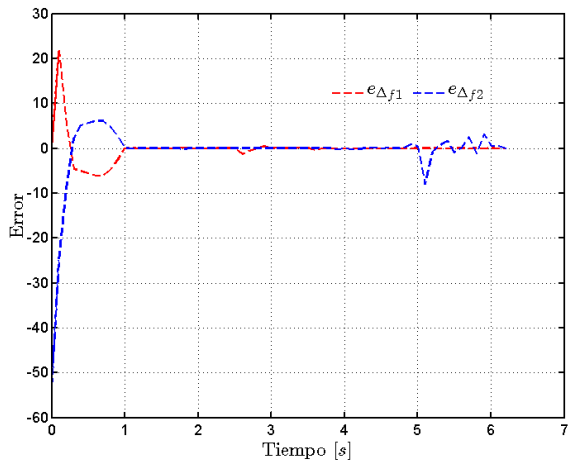
Lo anterior puede ser observado de mejor manera en las Figuras 4.147 - 4.151, en las cuales se puede apreciar la convergencia del error de estimación de la incertidumbre en cada uno de los experimentos realizados.



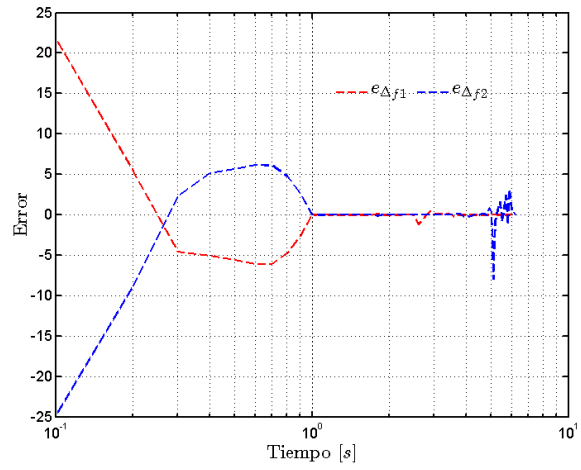
(a) Error de Estimación de la Incertidumbre

(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.147: Error de Estimación de la Incertidumbre del Sistema en el Experimento 1

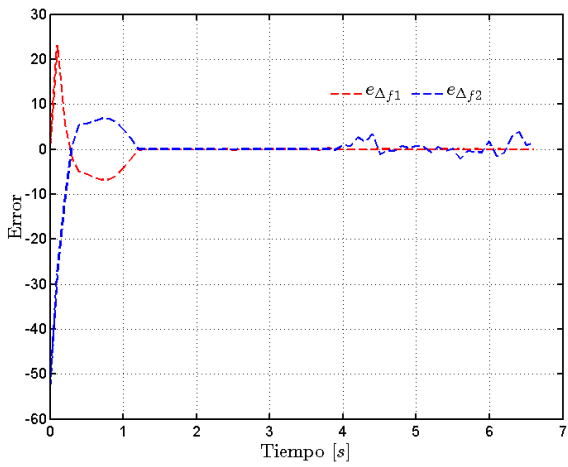


(a) Error de Estimación de la Incertidumbre

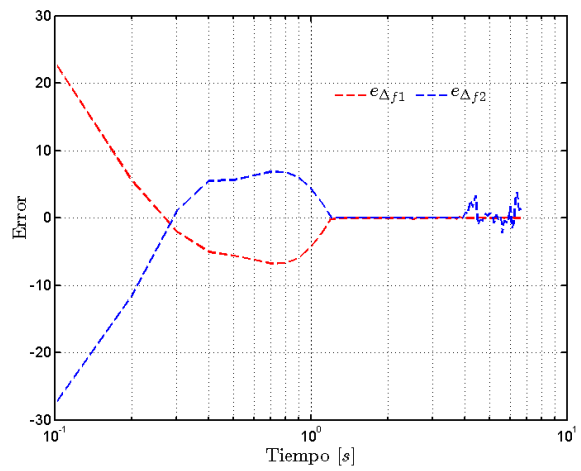


(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.148: Error de Estimación de la Incertidumbre del Sistema en el Experimento 2.



(a) Error de Estimación de la Incertidumbre



(b) Error de Estimación con escala logarítmica en el eje x

Figura 4.149: Error de Estimación de la Incertidumbre del Sistema en el Experimento 3.

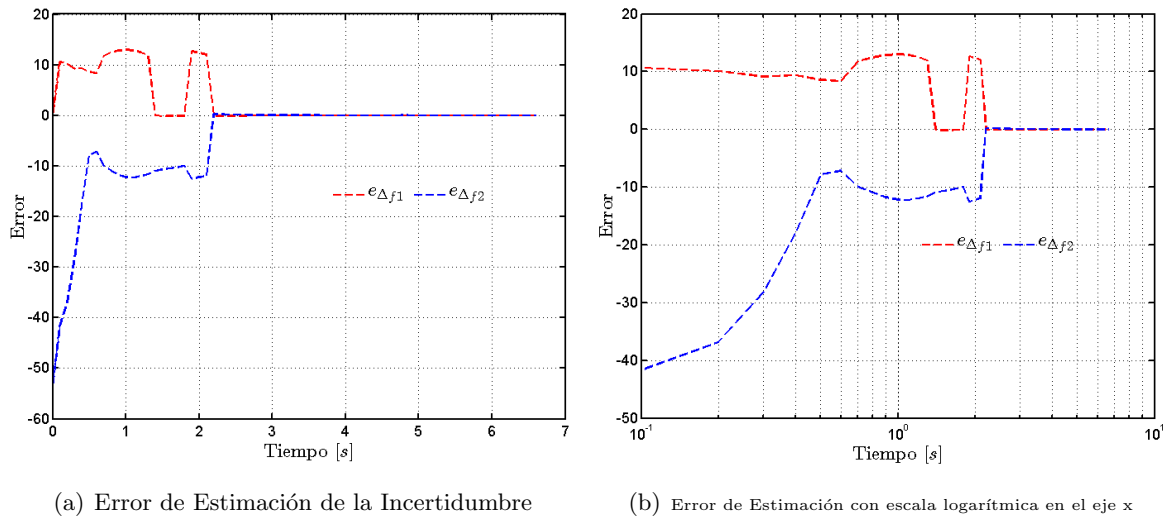


Figura 4.150: Error de Estimación de la Incertidumbre del Sistema en el Experimento 4.

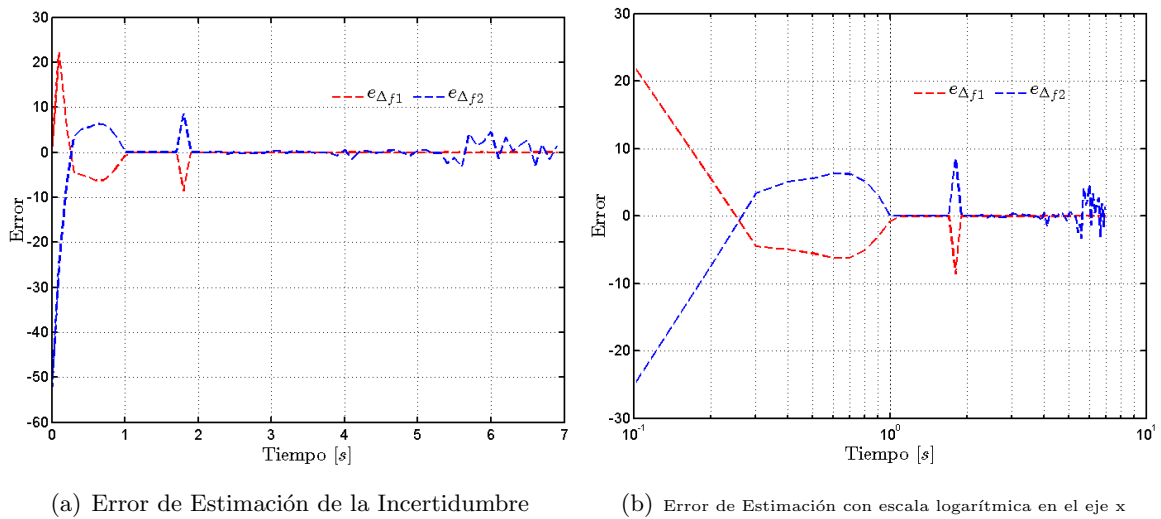


Figura 4.151: Error de Estimación de la Incertidumbre del Sistema en el Experimento 5.

En las Figuras 4.152 - 4.156 se muestra la evolución del error cuadrático medio a lo largo del tiempo para cada uno de los 5 experimentos, donde se puede observar que el área bajo la curva del error cuadrático medio es grande al principio, con el paso del tiempo el aumento en dicha área es pequeño, lo cual es debido a la convergencia del error de identificación de la incertidumbre, finalizando el error cuadrático medio con los valores de 40.3032 para el primer experimento, de 43.5570 para el segundo experimento, de 46.7339 para el tercer experimento, de 143.5968 para el cuarto experimento y de 40.7704 para el último experimento.

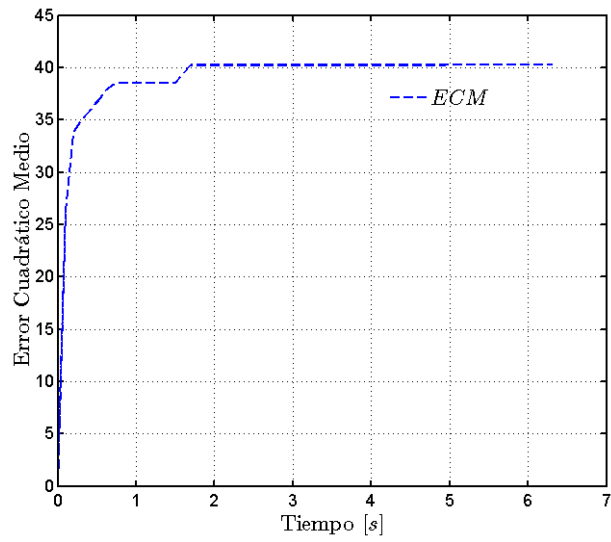


Figura 4.152: Error Cuadrático Medio en el experimento 1.

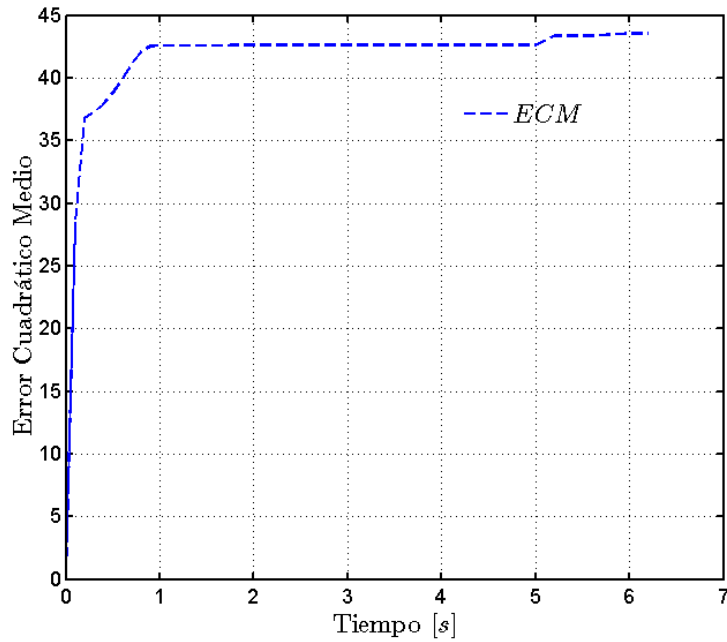


Figura 4.153: Error Cuadrático Medio en el experimento 2.

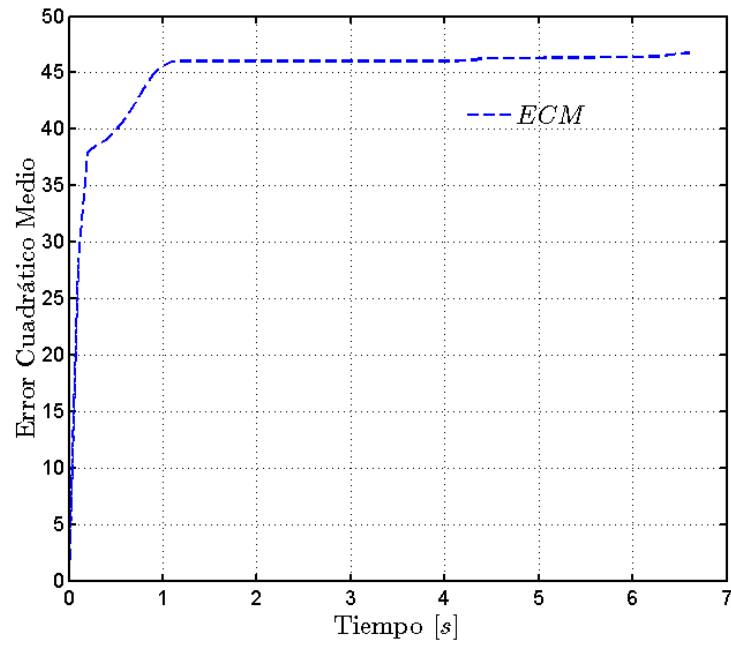


Figura 4.154: Error Cuadrático Medio en el experimento 3.

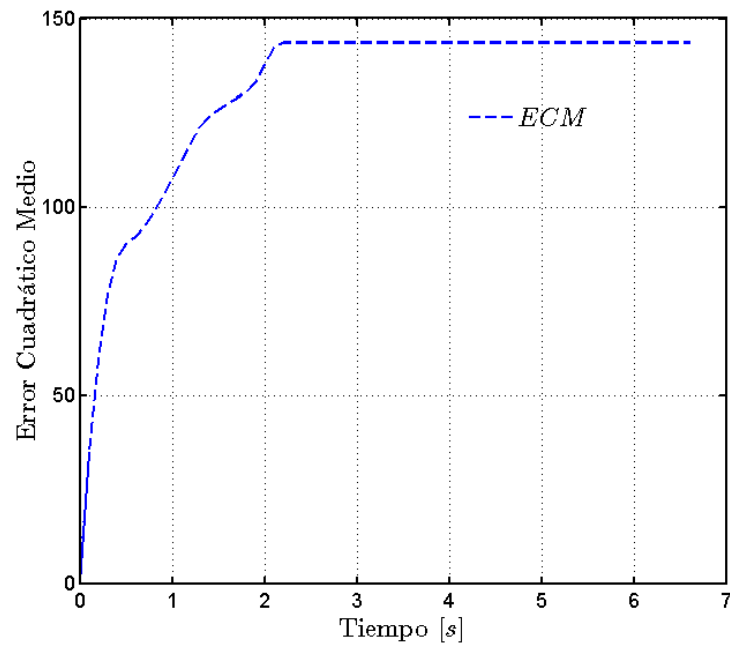


Figura 4.155: Error Cuadrático Medio en el experimento 4.

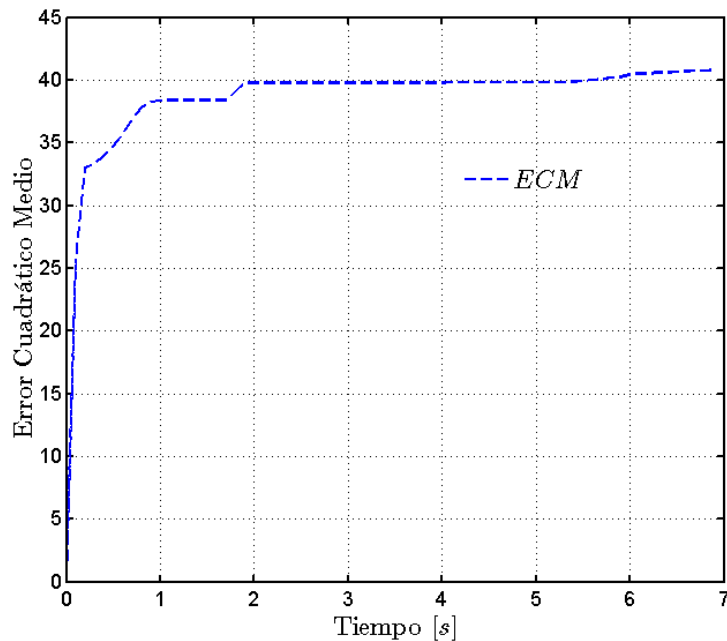


Figura 4.156: Error Cuadrático Medio en el experimento 5.

Primero que nada, el motivo de que las gráficas presentadas en el desarrollo de este apartado tengan un perfil poligonal y no suave se debe a el tiempo de muestreo de la tarjeta de adquisición de datos que proporciona el fabricante, no es fijo, se vio que se va modificando con el transcurso del tiempo, por lo que no fue posible ajustar ese tiempo para poder darle una mejor resolución a los perfiles resultantes.

Al llevar a cabo estos 5 experimentos en el sistema Servo Modular, se puede observar que el algoritmo de neuro-identificación realiza la estimación de la incertidumbre del sistema de manera satisfactoria, cumpliendo con las características buscadas en el error de identificación de la incertidumbre.

Estos Experimentos no habrían podido realizarse sin el apoyo del Dr. Manuel Alejandro Jiménez Lizárraga durante la estancia realizada en la Facultad de Ciencias Físico-Matemáticas de la Universidad Autónoma de Nuevo León.

4.6. Obtención del Modelo Neuronal Dinámico para aproximar al sistema no lineal con incertidumbre en una región de operación.

A continuación se muestran tres pruebas que se realizaron utilizando la metodología mencionada anteriormente, para las cuales se ocuparon los datos arrojados por las simulaciones.

4.6.1. Obtención del Modelo con los datos de la Simulación 1, correspondiente al sistema no lineal del Eslabón Simple con cambio de fricción

La Figura 4.157 muestra la totalidad de puntos de la simulación 1, se pueden observar una gran cantidad de puntos. Durante esta simulación se observa una gráfica en 3 dimensiones, debido a que la incertidumbre depende de los dos estados del sistema.

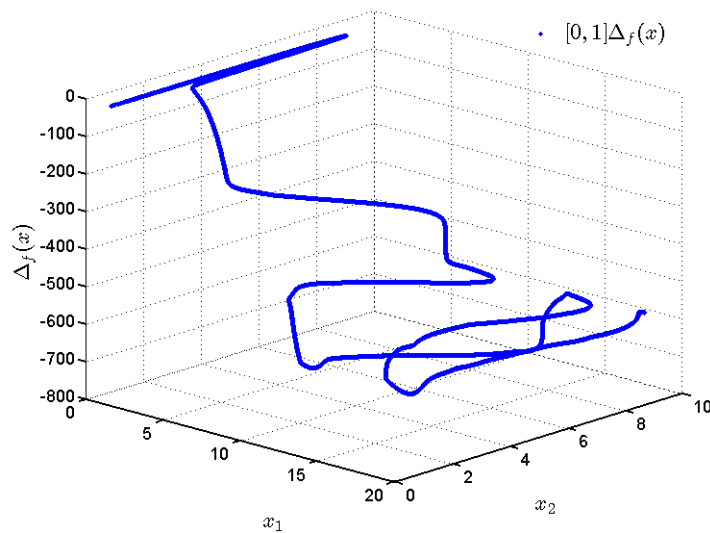


Figura 4.157: Trayectoria de la incertidumbre a lo largo de $0 \leq t \leq 4$.

La Figura 4.158 presenta la arquitectura utilizada por la red neuronal estática utilizada para los datos de la simulación 1.

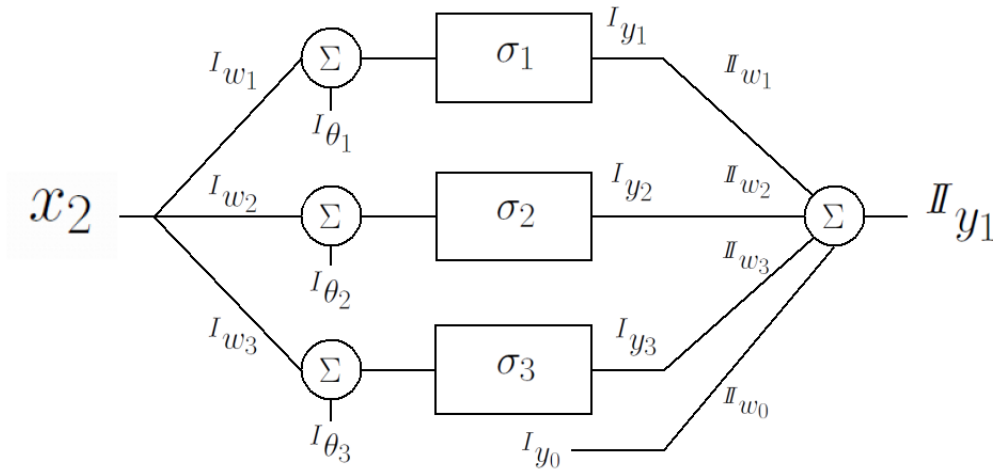


Figura 4.158: Red Neuronal Estática para la Simulación 1.

La aproximación es realizada utilizando los valores de la Tabla 4.5.

Variable	Valor	Descripción
I_{w_1}	$-\frac{3}{5}$	Peso Sináptico 1 de la primera capa.
I_{θ_1}	-70.5	Umbral Neuronal 1 de la primera capa.
I_{w_2}	$-\frac{3}{10}$	Peso Sináptico 2 de la primera capa.
I_{θ_2}	-14	Umbral Neuronal 2 de la primera capa.
I_{w_3}	$-\frac{3}{22.5}$	Peso Sináptico 3 de la primera capa.
I_{θ_3}	-22.5	Umbral Neuronal 3 de la primera capa.
σ	tanh	Función de activación utilizada

Tabla 4.5: Datos usados en la aproximación de la simulación 1.

Debido a la gran cantidad de puntos, para esta prueba se dividieron los datos en dos regiones de operación, una que contiene a los datos del eslabón cuando se encuentra fuera del líquido y la otra para cuando se encuentra dentro del líquido.

La Figura 4.159, representa a los datos que se encuentran fuera del líquido y que a su vez cumplen con las ecuaciones (3.30) y (3.31); donde el valor de α es elegido como 0.02 y el valor de β es seleccionado como 0.1.

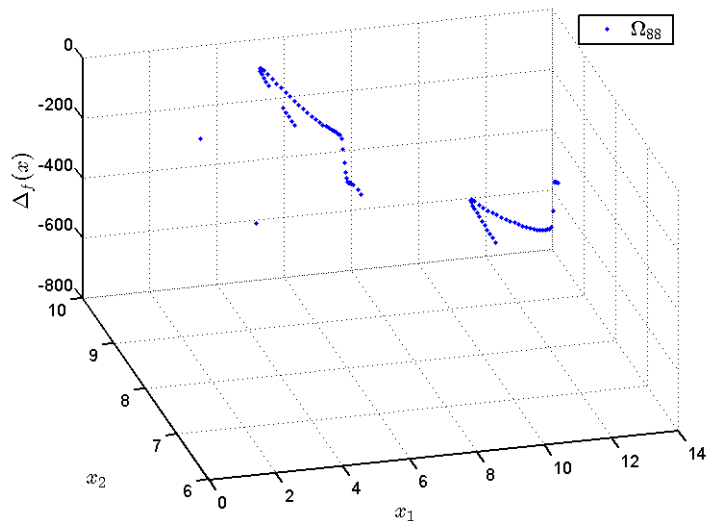


Figura 4.159: Nube de Datos del Eslabón fuera del líquido.

En la Figura 4.160 se muestra la distribución de los datos en la región de operación fuera del líquido seleccionada que va desde 5 rad hasta 7.8 rad.

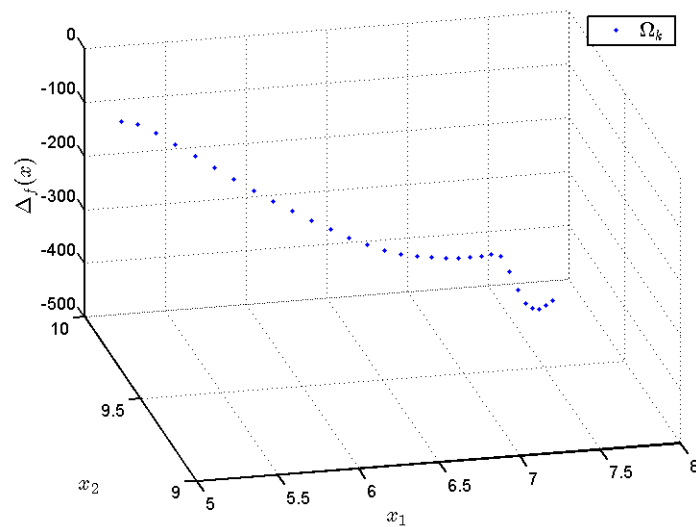


Figura 4.160: Nube de Datos del Eslabón fuera del líquido.

Luego del entrenamiento de la red neuronal estática, podemos obtener el Vector ${}^I W$.

El Vector resultante de los pesos sinápticos es:

$${}^I W^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1.4026 \times 10^3 & 2.0735 \times 10^4 & -1.8155 \times 10^4 & -1.4026 \times 10^3 \end{bmatrix}. \quad (4.56)$$

Con los valores anteriores de ${}^I W$ podemos proponer un modelo neuronal dinámico para aproximar al sistema no lineal con incertidumbre dentro de la región de operación dada

$$\dot{x}(t) \cong \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + {}^I W^* \sigma ({}^I W x + \theta) + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u, \quad (4.57)$$

es decir

$$\dot{x}(t) \cong \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1.4026 \times 10^3 & 2.0735 \times 10^4 & -1.8155 \times 10^4 & -1.4026 \times 10^3 \end{bmatrix} \cdot \begin{bmatrix} \tanh({}^I w_1 x_2 + {}^I \theta_1) \\ \tanh({}^I w_2 x_2 + {}^I \theta_2) \\ \tanh({}^I w_3 x_2 + {}^I \theta_3) \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u, \quad (4.58)$$

La Figura 4.161 muestra la aproximación a la nube de datos dada por medio del perfil representado por \hat{y} .

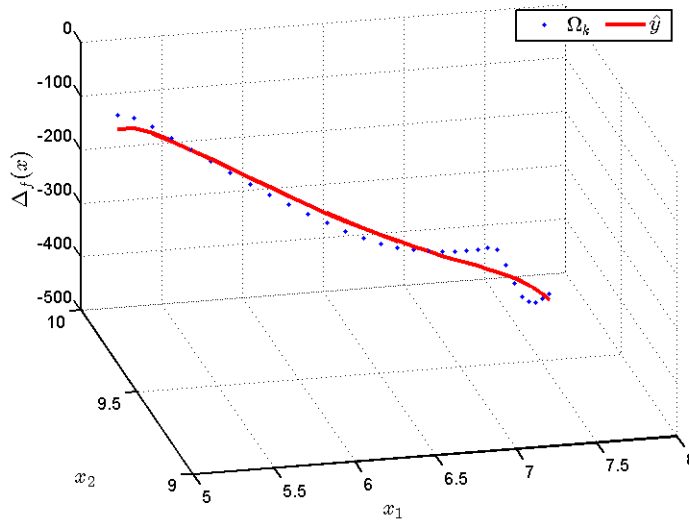


Figura 4.161: Aproximación de la Nube de Datos del Eslabón fuera del líquido.

La Figura 4.162, representa a los datos que se encuentran dentro del líquido y que a su vez cumplen con las ecuaciones (3.30) y (3.31); donde el valor de α es elegido como 0.02 y el valor de β es seleccionado como 0.1.

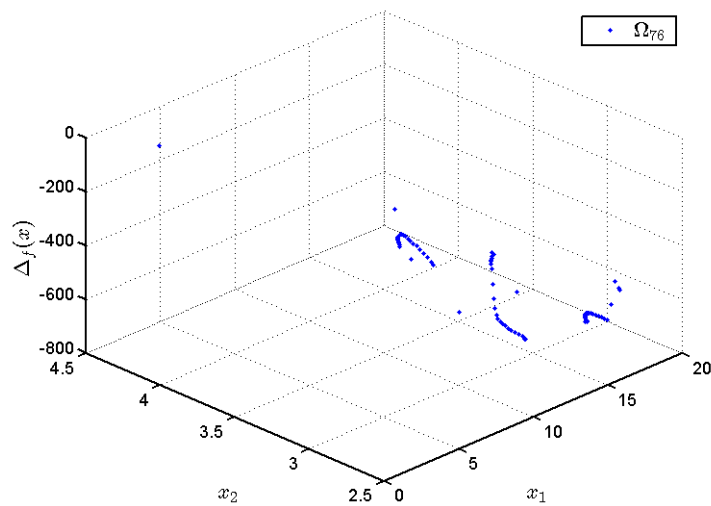


Figura 4.162: Nube de Datos del Eslabón dentro del líquido.

En la Figura 4.163 se muestra la distribución de los datos en la región de operación dentro del líquido seleccionada que va desde 8.4 rad hasta 10.6 rad.

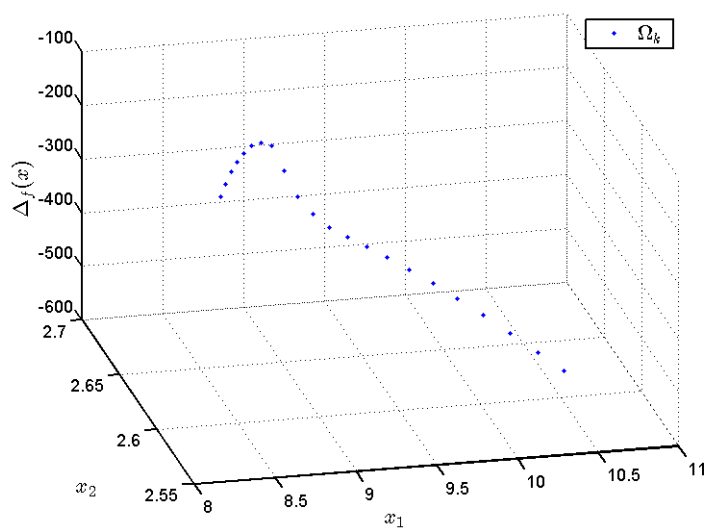


Figura 4.163: Nube de Datos del Eslabón dentro del líquido.

Luego del entrenamiento de la red neuronal estática, podemos obtener el Vector ${}^I W^*$.

El Vector resultante de los pesos sinápticos es:

$${}^I W^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -2.1716 \times 10^4 & -0.9031 \times 10^4 & 5.4189 \times 10^4 & -2.1716 \times 10^4 \end{bmatrix}. \quad (4.59)$$

Con los valores anteriores de ${}^I W$ podemos proponer un modelo neuronal dinámico para aproximar al sistema no lineal con incertidumbre dentro de la región de operación dada

$$\dot{x}(t) \cong \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + {}^I W^* \sigma ({}^I W x + \theta) + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u, \quad (4.60)$$

es decir

$$\dot{x}(t) \cong \begin{bmatrix} x_2(t) \\ \frac{-g}{l} \cos(x_1(t)) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -2.1716 \times 10^4 & -0.9031 \times 10^4 & 5.4189 \times 10^4 & -2.1716 \times 10^4 \end{bmatrix} \cdot \begin{bmatrix} \tanh({}^I w_1 x_{2_i} + {}^I \theta_1) \\ \tanh({}^I w_2 x_{2_i} + {}^I \theta_2) \\ \tanh({}^I w_3 x_{2_i} + {}^I \theta_3) \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u, \quad (4.61)$$

La Figura 4.164 muestra la aproximación a la nube de datos dada por medio del perfil representado por \hat{y} .

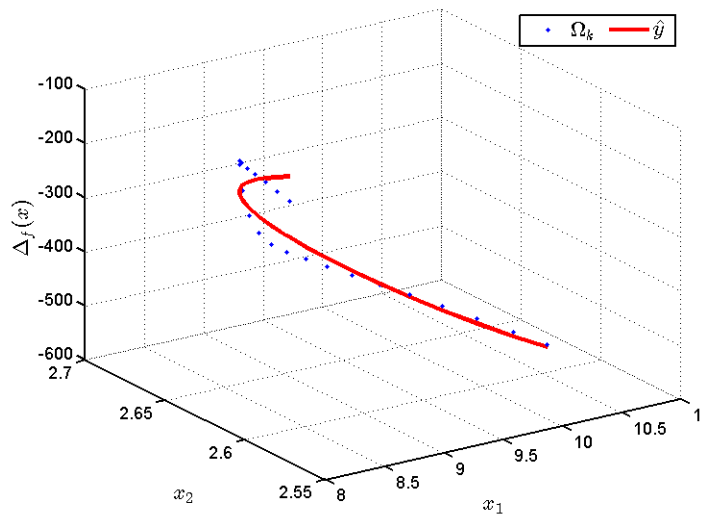


Figura 4.164: Aproximación de la Nube de Datos del Eslabón dentro del líquido.

4.6.2. Obtención del Modelo con los datos de la Simulación 2, correspondiente al sistema no lineal de tanques con una fuga

La Figura 4.165 muestra la totalidad de puntos de la simulación 2, en este caso la incertidumbre depende del segundo estado del sistema.

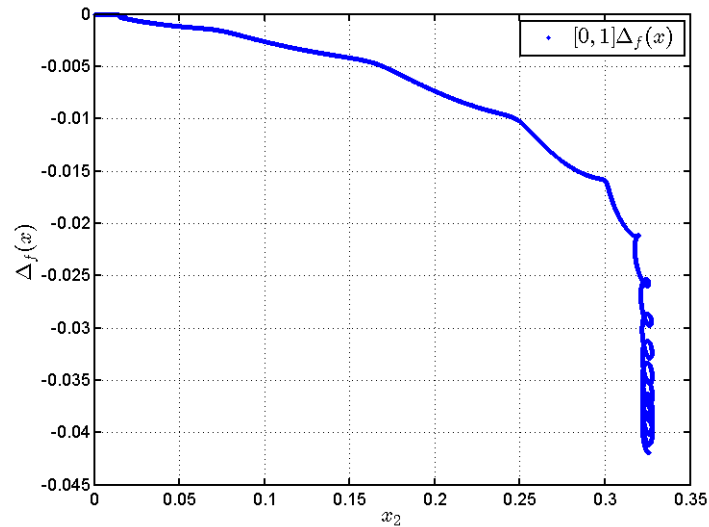


Figura 4.165: Nube de Datos del Sistema de Tanques.

La Figura 4.166, representa a los datos que se encuentran que cumplen con las ecuaciones (3.30) y (3.31); donde el valor de α es elegido como 0.01 y el valor de β es seleccionado como 0.02.

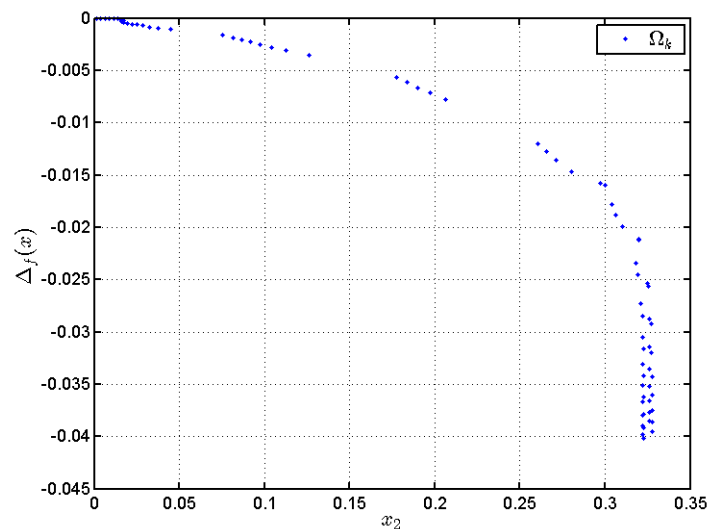


Figura 4.166: Nube de Datos de la Simulación 2.

La Figura 4.167 presenta la arquitectura utilizada por la red neuronal estática utilizada para los datos de la simulación 2.

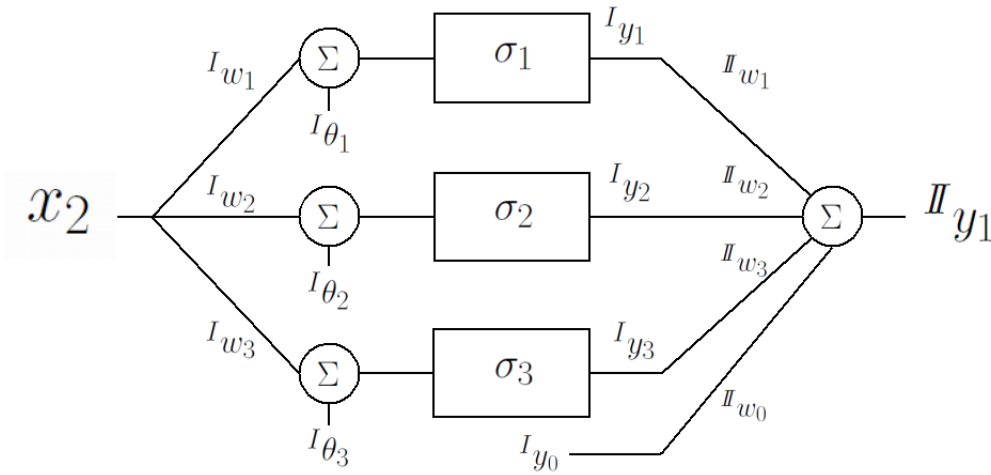


Figura 4.167: Red Neuronal Estática para la Simulación 2.

La aproximación es realizada utilizando los valores de la Tabla 4.6.

Variable	Valor	Descripción
Iw_1	$-\frac{3}{0.125}$	Peso Sináptico 1 de la primera capa.
$I\theta_1$	-0.125	Umbral Neuronal 1 de la primera capa.
Iw_2	$-\frac{3}{0.125}$	Peso Sináptico 2 de la primera capa.
$I\theta_2$	-0.425	Umbral Neuronal 2 de la primera capa.
Iw_3	$-\frac{3}{0.325}$	Peso Sináptico 3 de la primera capa.
$I\theta_3$	-0.525	Umbral Neuronal 3 de la primera capa.
σ	tanh	Función de activación utilizada

Tabla 4.6: Datos usados en la aproximación de la simulación 2.

Luego del entrenamiento de la red neuronal estática, podemos obtener el Vector ${}^{\#}W$.

El Vector resultante de los pesos sinápticos es:

$${}^{\#}W^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.0031 & 1.0701 & -0.2020 & -1.2756 \end{bmatrix}. \quad (4.62)$$

Con los valores anteriores de ${}^I W$ podemos proponer un modelo neuronal dinámico para aproximar al sistema no lineal con incertidumbre dentro de la región de operación dada

$$\dot{x}(t) \cong \begin{bmatrix} -\frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & 0 \\ \frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & -\frac{1}{[r_{02} + \delta_{r2} \text{sen}(\frac{2\pi}{0.45} x_2)] R_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + {}^I W^* \sigma ({}^I W x + \theta) + \begin{bmatrix} \frac{1}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} \\ 0 \end{bmatrix} u \quad , \quad (4.63)$$

es decir

$$\dot{x}(t) \cong \begin{bmatrix} -\frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & 0 \\ \frac{\frac{1}{R_1}}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} & -\frac{1}{[r_{02} + \delta_{r2} \text{sen}(\frac{2\pi}{0.45} x_2)] R_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.0031 & 1.0701 & -0.2020 & -1.2756 \end{bmatrix} \cdot \begin{bmatrix} \tanh({}^I w_1 x_{2_i} + {}^I \theta_1) \\ \tanh({}^I w_2 x_{2_i} + {}^I \theta_2) \\ \tanh({}^I w_3 x_{2_i} + {}^I \theta_3) \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{1}{r_{01} + \delta_{r1} \text{sen}(\frac{2\pi}{0.7} x_1)} \\ 0 \end{bmatrix} u \quad , \quad (4.64)$$

La Figura 4.168 muestra la aproximación a la nube de datos dada por medio del perfil representado por \hat{y} .

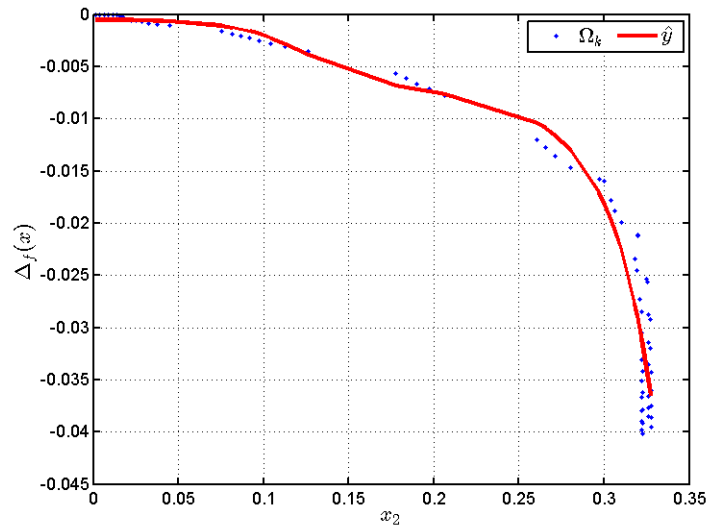


Figura 4.168: Aproximación de la Nube de Datos de la Simulación 2.

4.6.3. Obtención del Modelo con los datos de la Simulación 3, correspondiente al sistema no lineal de depredador-presa

La Figura 4.169 muestra la totalidad de puntos de la simulación 3, en este caso la incertidumbre depende del primer estado del sistema.

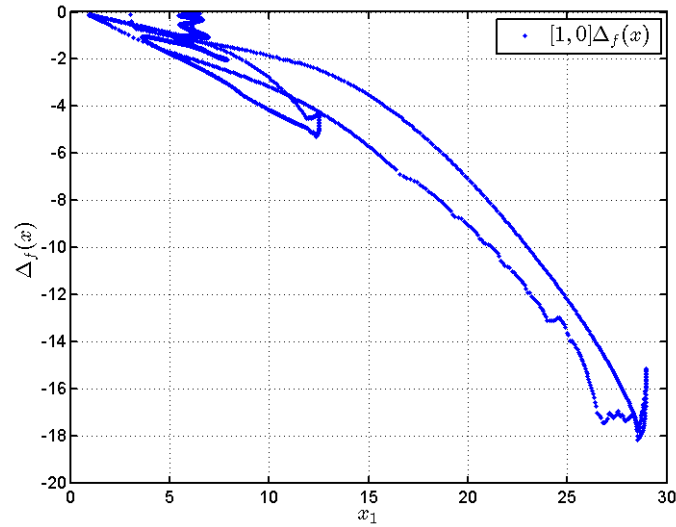


Figura 4.169: Nube de Datos del Sistema de Depredador-Presa.

La Figura 4.170, representa a los datos que se encuentran que cumplen con las ecuaciones (3.30) y (3.31); donde el valor de α es elegido como 0.2 y el valor de β es seleccionado como 0.5.

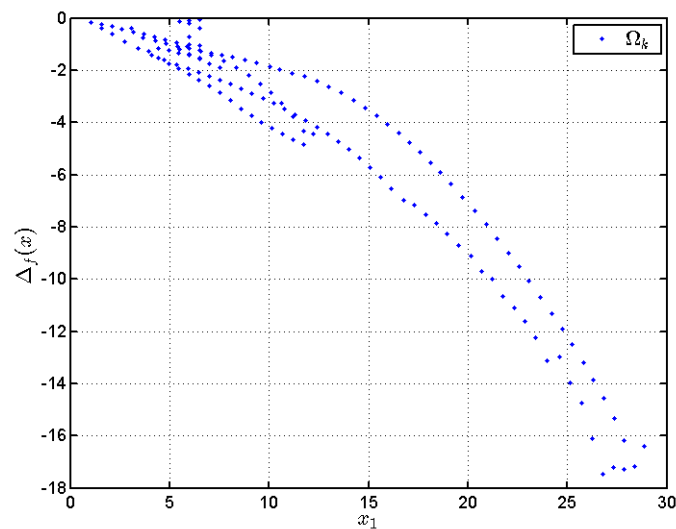


Figura 4.170: Nube de Datos de la Simulación 3.

La Figura 4.171 presenta la arquitectura utilizada por la red neuronal estática utilizada para los datos de la simulación 3.

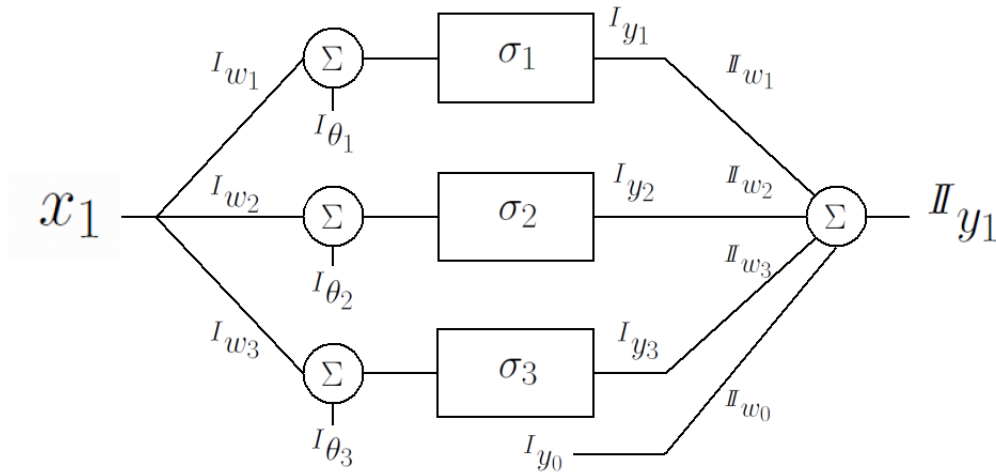


Figura 4.171: Red Neuronal Estática para la Simulación 3.

La aproximación es realizada utilizando los valores de la Tabla 4.7.

Variable	Valor	Descripción
I_{w_1}	$-\frac{3}{5}$	Peso Sináptico 1 de la primera capa.
I_{θ_1}	-7.5	Umbral Neuronal 1 de la primera capa.
I_{w_2}	$-\frac{3}{25}$	Peso Sináptico 2 de la primera capa.
I_{θ_2}	-15	Umbral Neuronal 2 de la primera capa.
I_{w_3}	$-\frac{3}{35}$	Peso Sináptico 3 de la primera capa.
I_{θ_3}	-27.5	Umbral Neuronal 3 de la primera capa.
σ	tanh	Función de activación utilizada

Tabla 4.7: Datos usados en la aproximación de la simulación 3.

Luego del entrenamiento de la red neuronal estática, podemos obtener el Vector ${}^I W$.

El Vector resultante de los pesos sinápticos es:

$${}^I W^* = \begin{bmatrix} 0.6894 & 0.0572 & 14.3484 & -15.4335 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.65)$$

Con los valores anteriores de ${}^I W^*$ podemos proponer un modelo neuronal dinámico para aproximar al sistema no lineal con incertidumbre dentro de la región de operación dada

$$\dot{x}(t) \cong \begin{bmatrix} \underline{r}_1 & -\beta N_1 \\ \kappa \beta N_2 & -r_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} + {}^I W^* \sigma({}^I W x + \theta) + \begin{bmatrix} r_a N_1 \\ 0 \end{bmatrix} u, \quad (4.66)$$

es decir

$$\begin{aligned} \dot{x}(t) \cong & \begin{bmatrix} \underline{r}_1 & -\beta N_1 \\ \kappa \beta N_2 & -r_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \\ & + \begin{bmatrix} 0.6894 & 0.0572 & 14.3484 & -15.4335 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \tanh({}^I w_1 x_{2_i} + {}^I \theta_1) \\ \tanh({}^I w_2 x_{2_i} + {}^I \theta_2) \\ \tanh({}^I w_1 x_{2_i} + {}^I \theta_3) \\ 1 \end{bmatrix} \\ & + \begin{bmatrix} r_a N_1 \\ 0 \end{bmatrix} u, \quad (4.67) \end{aligned}$$

La Figura 4.172 muestra la aproximación a la nube de datos dada por medio del perfil representado por \hat{y} .

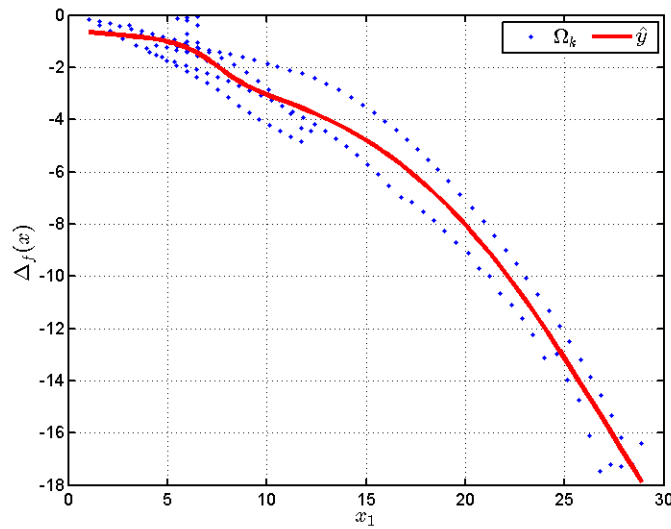


Figura 4.172: Aproximación de la Nube de Datos de la Simulación 3.

Luego de las 3 pruebas realizadas con los datos de las simulaciones se puede observar que la metodología propuesta cumple con el objetivo de aproximar la incertidumbre del sistema no lineal en una región de operación.

Las 3 pruebas se realizaron utilizando 3 neuronas en la primera capa, realizando el análisis para la creación de la red neuronal estática, se puede deducir que para para realizar una mejor aproximación pueden añadirse más neuronas, es decir, entre mayor número de neuronas, se tendrá una mejor aproximación.

V. Conclusiones y Trabajos Futuros

5.1. Conclusiones

En este trabajo de tesis se diseñó una metodología de identificación en línea mediante redes neuronales dinámicas, cuyo error de identificación sea estable, para un sistema no lineal afín con la entrada con incertidumbre parcial en el término no lineal de retroalimentación.

Para el diseño de la arquitectura de la red neuronal dinámica se llevó a cabo un análisis estilo Lyapunov de donde se obtienen la forma del umbral neuronal ψ y el algoritmo de aprendizaje \dot{W} ; además aprovechando que se tienen disponibles los estados a la salida, por la estructura del sistema no lineal, se añade un término de corrección para asegurar la convergencia del error.

Con lo anterior y el modelo de incertidumbre propuesto, se formula el teorema 3.1, el cual manifiesta las condiciones necesarias para acotar el error de identificación. Dicho teorema se aplica a los modelos de las simulaciones, arrojando resultados satisfactorios, además en una simulación se prueba la metodología utilizando modelos distintos para la arquitectura de la red neuronal y para el sistema simulado.

Dicha prueba se corroboró en la implementación en el sistema servo modular, donde se utilizó un modelo lineal propuesto por el fabricante en el manual del sistema y el sistema real.

También se propone una metodología para aproximar al sistema no lineal dentro de una región de operación, mediante la aplicación de una red neuronal estática, la cual utiliza los datos arrojados por las simulaciones referentes al teorema.

De esta última metodología propuesta se puede concluir que entre más exacta se necesite la aproximación del sistema, es necesario utilizar un mayor número de neuronas en la primera capa para obtener dicha aproximación.

5.2. Trabajos Futuros

Tomando como punto de partida el desarrollo de este trabajo de investigación pueden derivarse distintos puntos de interés para trabajos posteriores:

- Determinar el valor de la cota superior del error de estimación de la incertidumbre
- Definir criterios y efectos sobre la selección de los parámetros μ (Coeficiente de Aprendizaje) y L en la convergencia de la aproximación de la incertidumbre, o bien, incorporarlos en un algoritmo de adaptación.
- Desarrollo del algoritmo en línea que arroje el modelo neuronal para aproximar al sistema no lineal dentro de una región de operación.
- Desarrollo de la metodología para un sistema no lineal con incertidumbre en la parte afín al control.
- Desarrollo de la metodología de identificación para utilizarse en sistemas LPV afines a los parámetros.

Bibliografía

- [1] J. R. Reyes, *Pasividad Robusta de Sistemas No Lineales vía Redes Neuronales Dinámicas*. Tesis de doctorado, Departamento de Ingeniería Eléctrica CINVESTAV-IPN, Enero 2002.
- [2] C. E. González, *Estabilidad y Controlabilidad Robusta de Sistemas Lineales con Incertidumbre Múltiple*. Tesis de doctorado, Facultad de Ingeniería Mecánica y Eléctrica, UANL, Mayo 1998.
- [3] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems—a survey,” *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.
- [4] F. L. Lewis and T. Parisini, “Guest editorial: Neural network feedback control with guaranteed stability,” *Int. J. Control*, vol. 70, pp. 337–340, 1998.
- [5] M. M. Polycarpou and M. J. Mears, “Stable adaptive tracking of uncertain systems using nonlinearly parametrized on-line approximators,” *International journal of control*, vol. 70, no. 3, pp. 363–384, 1998.
- [6] N. Sadegh, “A multilayer nodal link perceptron network with least squares training algorithm,” *International Journal of Control*, vol. 70, no. 3, pp. 385–404, 1998.
- [7] H. Ohta and S. Yokota, “A regulator design of dynamical systems with nonlinear uncertainties using multilayered neural networks,” in *Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on*, vol. 2, pp. 1785–1788, IEEE, 1993.
- [8] J. H. Li and P. M. Lee, “Neural network adaptive control for a class of nonlinear systems with unknown-bound unstructured uncertainties,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, pp. 692–697, IEEE, 2004.
- [9] W. Yu, “Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms,” *Information sciences*, vol. 158, pp. 131–147, 2004.
- [10] J. de Jesús Rubio and W. Yu, “Nonlinear system identification with recurrent neural networks and dead-zone kalman filter algorithm,” *Neurocomputing*, vol. 70, no. 13, pp. 2460–2466, 2007.

-
- [11] M. Liu, “Robust H_∞ control for uncertain delayed nonlinear systems based on standard neural network models,” *Neurocomputing*, vol. 71, no. 16, pp. 3469–3492, 2008.
- [12] C.-S. Chen, “Dynamic structure adaptive neural fuzzy control for mimo uncertain nonlinear systems,” *Information Sciences*, vol. 179, no. 15, pp. 2676–2688, 2009.
- [13] G. Sun, D. Wang, Z. Peng, H. Wang, N. Wang, and W. Lan, “Robust adaptive neural network control for strict-feedback nonlinear systems with uncertainties,” in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pp. 1328–1333, IEEE, 2012.
- [14] T. Zhang, X. Shi, Q. Zhu, and Y. Yang, “Adaptive neural tracking control of pure-feedback nonlinear systems with unknown gain signs and unmodeled dynamics,” *Neurocomputing*, vol. 121, pp. 290–297, 2013.
- [15] W. Chen, S. Hua, and S. S. Ge, “Consensus-based distributed cooperative learning control for a group of discrete-time nonlinear multi-agent systems using neural networks,” *Automatica*, vol. 50, no. 9, pp. 2254–2268, 2014.
- [16] D. Wang, D. Liu, H. Li, and H. Ma, “Neural-network-based robust optimal control design for a class of uncertain nonlinear systems via adaptive dynamic programming,” *Information Sciences*, vol. 282, pp. 167–179, 2014.
- [17] Z. Miao, Y. Wang, and Y. Yang, “Robust tracking control of uncertain dynamic nonholonomic systems using recurrent neural networks,” *Neurocomputing*, vol. 142, pp. 216–227, 2014.
- [18] X. Zhang, C.-Y. Su, Y. Lin, L. Ma, and J. Wang, “Adaptive neural network dynamic surface control for a class of time-delay nonlinear systems with hysteresis inputs and dynamic uncertainties,” 2015.
- [19] W. Yu and X. Li, “Recurrent fuzzy neural networks for nonlinear system identification,” in *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*, pp. 590–595, IEEE, 2007.
- [20] J. C. Tovar and W. Yu, “Automated fuzzy neural networks for nonlinear system identification,” in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pp. 1159–1165, IEEE, 2008.
- [21] H.-W. Ge, W.-L. Du, F. Qian, and Y.-C. Liang, “Identification and control of nonlinear systems by a time-delay recurrent neural network,” *Neurocomputing*, vol. 72, no. 13, pp. 2857–2864, 2009.

-
- [22] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, “A recursive identification algorithm for switched linear/affine models,” *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 242–253, 2011.
- [23] S. Xie, Y. Zhang, C. Chen, and X. Zhang, “Identification of nonlinear hysteretic systems by artificial neural network,” *Mechanical Systems and Signal Processing*, vol. 34, no. 1, pp. 76–87, 2013.
- [24] H. M. R. Ugalde, J.-C. Carmona, V. M. Alvarado, and J. Reyes-Reyes, “Neural network design and model reduction approach for black box nonlinear system identification with reduced number of parameters,” *Neurocomputing*, vol. 101, pp. 170–180, 2013.
- [25] M. Nikravesh, A. Farell, and T. Stanford, “Model identification of nonlinear time variant processes via artificial neural network,” *Computers & chemical engineering*, vol. 20, no. 11, pp. 1277–1290, 1996.
- [26] Y. Quan, M. Zhou, and Z. Luo, “On-line robust identification of tool-wear via multi-sensor neural-network fusion,” *Engineering Applications of Artificial Intelligence*, vol. 11, no. 6, pp. 717–722, 1998.
- [27] S. Purwar, I. N. Kar, and A. N. Jha, “On-line system identification of complex systems using chebyshev neural networks,” *Applied Soft Computing*, vol. 7, no. 1, pp. 364–372, 2007.
- [28] X. Wu, J. Zhang, and Q. Zhu, “A generalized procedure in designing recurrent neural network identification and control of time-varying-delayed nonlinear dynamic systems,” *Neurocomputing*, vol. 73, no. 7, pp. 1376–1383, 2010.
- [29] K. Salahshoor and M. Hamzehnejad, “A novel online affine model identification of multivariable processes using adaptive neuro-fuzzy networks,” *Chemical Engineering Research and Design*, vol. 88, no. 2, pp. 155–169, 2010.
- [30] X. Han, W.-F. Xie, Z. Fu, and W. Luo, “Nonlinear systems identification using dynamic multi-time scale neural networks,” *Neurocomputing*, vol. 74, no. 17, pp. 3428–3439, 2011.
- [31] K. Salahshoor, M. Hamzehnejad, and S. Zakeri, “Online affine model identification of nonlinear processes using a new adaptive neuro-fuzzy approach,” *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5534–5554, 2012.
- [32] Z.-J. Fu, W.-F. Xie, and W.-D. Luo, “Robust on-line nonlinear systems identification using multilayer dynamic neural networks with two-time scales,” *Neurocomputing*, vol. 113, pp. 16–26, 2013.

- [33] L. C. Kiong, M. Rajeswari, and M. Rao, "Nonlinear dynamic system identification and control via constructivism inspired neural network," *Applied Soft Computing*, vol. 3, no. 3, pp. 237–257, 2003.
- [34] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Networks*, vol. 22, no. 5, pp. 851–860, 2009.
- [35] N. Zhang and W. Qu, "Neuroidentification and neurocontrol of dynamic systems using backpropagation technique," in *Industrial Electronics, 1992., Proceedings of the IEEE International Symposium on*, pp. 241–245, IEEE, 1992.
- [36] T. Tsuji, B. H. Xu, and M. Kaneko, "Adaptive control and identification using one neural network for a class of plants with uncertainties," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 4, pp. 496–505, 1998.
- [37] W. Yu and X. Li, "Some new results on system identification with dynamic neural networks," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 412–417, 2001.
- [38] X. Li and W. Yu, "Dynamic system identification via recurrent multilayer perceptrons," *Information sciences*, vol. 147, no. 1, pp. 45–63, 2002.
- [39] W. Yu, "Passivity analysis for dynamic neuro identifier," *Computación y Sistemas, Edición Especial*, pp. 83–91, 2002.
- [40] W. Yu, "Stability analysis of dynamic multilayer neuro identifier," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2, pp. 1770–1775, IEEE, 2002.
- [41] J. A. Vargas, K. R. Gularte, and E. M. Hemerly, "An improved on-line neuro-identification scheme," in *Control (CONTROL), 2012 UKACC International Conference on*, pp. 1088–1093, IEEE, 2012.
- [42] C. Emmanouilides and L. Petrou, "Identification and control of anaerobic digesters using adaptive, on-line trained neural networks," *Computers & chemical engineering*, vol. 21, no. 1, pp. 113–143, 1997.
- [43] S. Madan, A. Thrasher, and B. Andrews, "Fast neural system identification: application to rehabilitation engineering," in *Communications, Computers and Signal Processing, 1997. 10 Years PACRIM 1987-1997-Networking the Pacific Rim. 1997 IEEE Pacific Rim Conference on*, vol. 2, pp. 539–542, IEEE, 1997.
- [44] A. Yazdizadeh and K. Khorasani, "Identification of a turbogenerator system using adaptive time delay neural networks," in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, pp. 502–506, IEEE, 1998.

- [45] T. Zhang and M. Naka, “A hybrid neuro-inverse control approach with knowledge-based nonlinear separation for industrial nonlinear system,” *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 5, pp. 840–846, 2005.
- [46] M. Boroushaki, M. B. Ghofrani, C. Lucas, and M. J. Yazdanpanah, “Identification and control of a nuclear reactor core (vver) using recurrent neural networks and fuzzy systems,” *Nuclear Science, IEEE Transactions on*, vol. 50, no. 1, pp. 159–174, 2003.
- [47] R. P. Kalyani and G. K. Venayagamoorthy, “Neuroidentification of system parameters of the upfc in a multimachine power system,” in *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, pp. 243–248, IEEE, 2004.
- [48] Z. Dong, W. Xiang, X. Xue, S. Chen, and X. Wang, “On-line identification of thermal process using a modified ts-type neuro-fuzzy system,” in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, pp. 1282–1287, IEEE, 2011.
- [49] Y.-S. T. Hong, “Dynamic nonlinear state-space model with a neural network via improved sequential learning algorithm for an online real-time hydrological modeling,” *Journal of Hydrology*, vol. 468, pp. 11–21, 2012.
- [50] J. A. Rojas Estrada, *Análisis de estabilidad de sistemas lineales en variables de estado con incertidumbre paramétrica*. Tesis de doctorado, Facultad de Ingeniería Mecánica y Eléctrica, UANL, 1994.
- [51] P. Isasi and I. Galván, *Redes neuronales artificiales—un enfoque práctico*. Madrid: Pearson-Prentice Hall, 2004.
- [52] A. Cichocki and R. Unbehauen, *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc., 1993.
- [53] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [54] F. O. Rodríguez, *Análisis y diseño de redes neuronales CMAC para la identificación y control de sistemas no lineales*. Tesis de doctorado, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, Oct. 2008.
- [55] A. N. Kolmogorov, “On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition,” *Amer. Math. Soc. Transl*, vol. 28, pp. 55–59, 1963.

- [56] R. Hecht-Nielsen, “Kolmogorov’s mapping neural network existence theorem,” in *Proceedings of the international conference on Neural Networks*, vol. 3, pp. 11–14, New York: IEEE Press, 1987.
- [57] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” in *Neural Networks, 1989. IJCNN., International Joint Conference on*, pp. 593–605, IEEE, 1989.
- [58] V. Kurková, “Kolmogorov’s theorem and multilayer neural networks,” *Neural networks*, vol. 5, no. 3, pp. 501–506, 1992.
- [59] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [60] Y. Ito, “Approximation of functions on a compact set by finite sums of a sigmoid function without scaling,” *Neural Networks*, vol. 4, no. 6, pp. 817–826, 1991.
- [61] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [62] B. M. Seix, *Estudio de redes neuronales modulares para el modelado de sistemas dinámicos no lineales*. Tesis de doctorado, Universitat Politècnica de Catalunya, 2001.
- [63] D. R. Hush and B. Horne, “Efficient algorithms for function approximation with piecewise linear sigmoidal networks,” *Neural Networks, IEEE Transactions on*, vol. 9, no. 6, pp. 1129–1141, 1998.
- [64] M. Meltser, M. Shoham, and L. M. Manevitz, “Approximating functions by neural networks: a constructive solution in the uniform norm,” *Neural Networks*, vol. 9, no. 6, pp. 965–978, 1996.
- [65] D. Poole, *Álgebra lineal. Una introducción moderna*. Cengage Learning Editores, 2011.
- [66] I. N. Herstein and F. V. Coba, *Álgebra moderna: grupos, anillos, campos, teoría de Galois*. Trillas, 1970.
- [67] H. K. Khalil and J. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall New Jersey, 1996.
- [68] A. J. Barragán, *Síntesis de sistemas de control borroso estables por diseño*. Tesis de doctorado, Universidad de Huelva, 2009.
- [69] A. J. Lotka *et al.*, “Elements of physical biology,” 1925.
- [70] V. Volterra, “Variations and fluctuations of the number of individuals in animal species living together,” *J. Cons. Int. Explor. Mer*, vol. 3, no. 1, pp. 3–51, 1928.

-
- [71] G. Samanta and R. G. Aíza, “Modelos dinámicos de poblaciones simples y de sistemas depredador-presa,” *Miscelánea Matemática, Sociedad Matemática Mexicana*, no. 58, pp. 77–110, 2014.
- [72] A. Luo and J.-Q. Sun, *Complex systems: fractionality, time-delay and synchronization*. Springer Science & Business Media, 2011.
- [73] M. Inteco, “Modular servo system (mss),” *User’s Manual Krakow, Poland*, 2006.
- [74] O. Katsuhiko, “Ingeniería de control moderna,” *Minnesota, Pearson Education*, 1998.

Anexos

Anexo A

Manipulación Algebraica para mostrar que el Error es Acotado

Para llegar a la expresión (3.23), se toma como punto de partida (3.21) a la cual se le añade un 0, es decir,

$$\dot{V}(e(t), W(t)) \leq -e^T(t)Le(t) + e^T(t)e(t) - e^T(t)e(t), \quad (\text{A.1})$$

$$\dot{V}(e(t), W(t)) \leq -e^T(t)(L + I)e(t) - e^T(t)e(t). \quad (\text{A.2})$$

El primer termino del lado derecho de la inecuación (A.2) debe ser negativo por lo que $L + I > 0$, así que se debe elegir a $L = L^T > I$, entonces

$$\dot{V}(e(t), W(t)) \leq -e^T(t)e(t). \quad (\text{A.3})$$

Luego se procede a realizar algunas manipulaciones algebraicas:

$$e^T(t)e(t) \leq -\dot{V}(e(t), W(t)), \quad (\text{A.4})$$

$$\lim_{t \rightarrow \infty} \int_0^t e^T(\tau)e(\tau) d\tau \leq - \lim_{t \rightarrow \infty} \int_0^t \dot{V}(e(\tau), W(\tau)) d\tau, \quad (\text{A.5})$$

$$\lim_{t \rightarrow \infty} \int_0^t e^T(\tau)e(\tau) d\tau \leq - \lim_{t \rightarrow \infty} V(e(\tau), W(\tau)) \Big|_0^t, \quad (\text{A.6})$$

$$\lim_{t \rightarrow \infty} \int_0^t e^T(\tau)e(\tau) d\tau \leq - \left[\lim_{t \rightarrow \infty} V(e(\tau), W(\tau)) - V(e(0), W(0)) \right], \quad (\text{A.7})$$

$$\lim_{t \rightarrow \infty} \int_0^t e^T(\tau)e(\tau) d\tau \leq - \lim_{t \rightarrow \infty} V(e(\tau), W(\tau)) + V(e(0), W(0)), \quad (\text{A.8})$$

si se toma el lado derecho de la inecuación (A.8), y se toma como cota superior el mayor de

los términos, es decir, $V(e(0), W(0))$, entonces

$$\lim_{t \rightarrow \infty} \int_0^t \|e(\tau)\|^2 d\tau \leq V(e(0), W(0)), \quad (\text{A.9})$$

como

$$V(e(0), W(0)) < \infty, \quad (\text{A.10})$$

entonces

$$\lim_{t \rightarrow \infty} \|e(t)\|^2 < \infty, \quad (\text{A.11})$$

por lo tanto si el limite del área bajo la curva de la norma cuadrática del error de identificación cuando el tiempo tiende a infinito es acotado (A.11), por consiguiente

$$\lim_{t \rightarrow \infty} \|e(t)\| < \infty. \quad (\text{A.12})$$

Una manera de cuantificar el error de identificación, es mediante la manipulación de la ecuación (3.20) de donde tenemos que

$$\dot{V}(e, W) \leq e^T [\text{sign}(e)\delta_f(x) - \psi] - e^T L e, \quad (\text{A.13})$$

de donde se observa que el segundo termino $-e^T L e$ es negativo siempre y cuando $L = L^T > 0$, por lo cual

$$\dot{V}(e, W) \leq e^T [\text{sign}(e)\delta_f(x) - \psi], \quad (\text{A.14})$$

Si a la ecuación (A.14) se le añade un cero que contenga a la cota superior de la norma de la incertidumbre tenemos

$$\dot{V}(e, W) \leq e^T [\text{sign}(e)[\delta_f(x) + \rho\delta_f(x) - \rho\delta_f(x)] - \psi], \quad (\text{A.15})$$

donde $\rho \geq 1$. Por lo cual si agrupamos términos de la ecuación (A.15)

$$\dot{V}(e, W) \leq e^T [\text{sign}(e)(1 + \rho)\delta_f(x) - \psi] - e^T \text{sign}(e)\rho\delta_f(x), \quad (\text{A.16})$$

como por la selección de ψ el termino $e^T [\text{sign}(e)(1 + \rho)\delta_f(x) - \psi]$ es eliminado de la ecuación (A.16), por lo que

$$\dot{V}(e, W) \leq -e^T \text{sign}(e)\rho\delta_f, \quad (\text{A.17})$$

Por la propiedad de la norma representada por (2.23) se tiene que $\|e\|_1 = e^T \text{sign}(e)$ por lo tanto

$$\begin{aligned} \dot{V}(e, W) &\leq -\|e\|_1 \rho\delta_f, \\ \dot{V}(e, W) &\leq -\rho \|e\|_1 \min_x \{\delta_f\}, \\ \dot{V}(e, W) &\leq -\rho \|e\|_1 \underline{\delta}_f. \end{aligned} \quad (\text{A.18})$$

Luego de algunas manipulaciones algebraicas tenemos

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} \dot{V}(e(\tau), W(\tau)) \, d\tau \leq - \lim_{t_f \rightarrow \infty} \frac{\rho \delta_f}{t_f} \int_0^{t_f} \|e(\tau)\|_1 \, d\tau, \quad (\text{A.19})$$

entonces evaluamos la ecuación (A.19)

$$\frac{\rho}{t_f} \int_0^{t_f} \|e(\tau)\|_1 \, d\tau \leq \frac{1}{\delta_f t_f} V(e(0), W(0)). \quad (\text{A.20})$$

La ecuación A.20 representa la cuantificación del error de identificación, el cual depende de los valores iniciales de la función candidata de Lyapunov, el valor mínimo de la cota de la norma de la incertidumbre y el tiempo.

Anexo B

Convergencia de los Pesos Sinápticos de la Simulación 3

En la Figura B.1, se puede observar el comportamiento de los pesos sinápticos de la red neuronal con un mayor tiempo de simulación al planteado originalmente, donde se muestra que los pesos W_{11} y W_{12} tienen un comportamiento periódico acotado después de cierto tiempo de simulación.

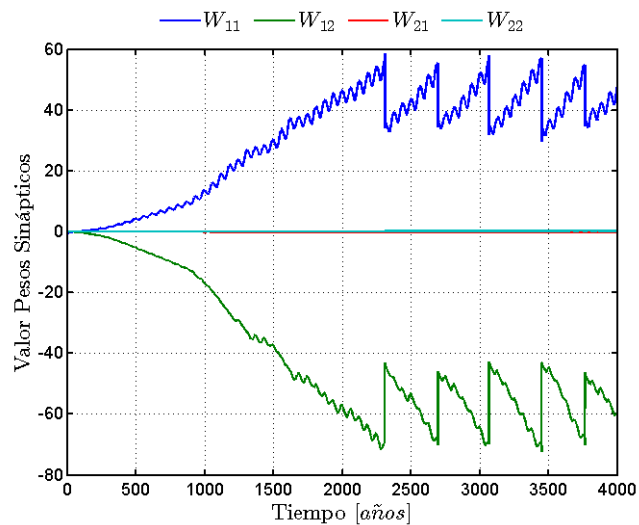


Figura B.1: Pesos Sinápticos de la Red Neuronal Dinámica de la Simulación 3.

Anexo C

Desarrollo del Sistema de Péndulo Invertido

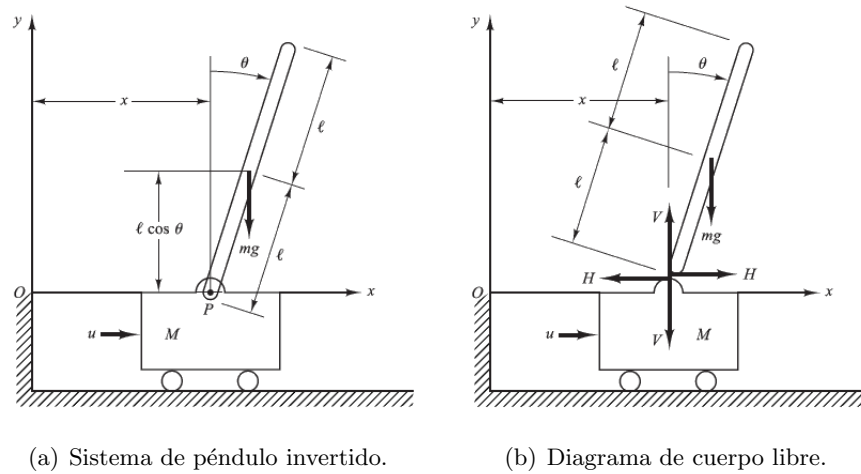


Figura C.1: Sistema de Péndulo Invertido.

Un péndulo invertido montado sobre un carro manejado por un motor aparece en la Figura C.1(a). El sistema de péndulo invertido tiene un punto de equilibrio inestable, este se presenta cuando $\theta = 0$, esto es porque puede girar en cualquier momento y en cualquier dirección, a menos que se le aplique una fuerza de control conveniente [74].

Sea θ el ángulo de la barra respecto de la línea vertical. Sean además las coordenadas (x, y) del centro de gravedad de la barra del péndulo (x_G, y_G) . De este modo

$$\begin{aligned}x_G &= x + \ell \sin \theta, \\y_G &= \ell \cos \theta.\end{aligned}$$

Para obtener las ecuaciones de movimiento para el sistema, considérese el diagrama de cuerpo libre que aparece en la Figura C.1(b). El movimiento rotacional de la barra del péndulo alrededor de su centro de gravedad se describe mediante

$$I\ddot{\theta} = Vlsen\theta - Hlcos\theta, \quad (C.1)$$

donde I es el momento de inercia de la barra alrededor de su centro de gravedad. El movimiento horizontal del centro de gravedad de la barra del péndulo se obtiene mediante

$$m\frac{d^2}{dt^2}(x + lsen\theta) = H, \quad (C.2)$$

El movimiento vertical del centro de gravedad de la barra del péndulo es

$$m\frac{d^2}{dt^2}(lcos\theta) = V - mg, \quad (C.3)$$

El movimiento horizontal del carro se describe mediante

$$M\frac{d^2x}{dt^2} = u - H. \quad (C.4)$$

Como se debe mantener el péndulo invertido en posición vertical, se puede suponer que $\theta(t)$ y $\dot{\theta}(t)$ son pequeños, de forma que $sen\theta \doteq \theta$, $cos\theta = 1$ y $\theta\dot{\theta}^2 = 0$. Entonces, las Ecuaciones (C.1) a (C.3) se linealizan del modo siguiente:

$$I\ddot{\theta} = Vl\theta - Hl, \quad (C.5)$$

$$m(\ddot{x} + l\ddot{\theta}) = H, \quad (C.6)$$

$$0 = V - mg. \quad (C.7)$$

A partir de las Ecuaciones (C.4) y (C.6), se obtiene

$$(M + m)\ddot{x} + ml\ddot{\theta} = u. \quad (C.8)$$

A partir de las Ecuaciones (C.5), (C.6) y (C.7), se obtiene

$$I\ddot{\theta} = mgl\theta - Hl = mgl\theta - l(m\ddot{x} + ml\ddot{\theta}),$$

o bien

$$(I + ml^2)\ddot{\theta} + ml\ddot{x} = mgl\theta. \quad (C.9)$$