



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD ZACATENCO**

Programa Sistemas Autónomos de Navegación Aérea y Submarina

**Control cooperativo basado en redes neuronales para  
múltiples vehículos aéreos.**

**T E S I S**

Que presenta

**M. en C. Alejandro Morfin Santana**

Para obtener el grado de

**DOCTOR EN CIENCIAS**

En

**SISTEMAS AUTÓNOMOS DE NAVEGACIÓN AÉREA Y SUBMARINA**

Directores de Tesis:

Dr. Filiberto Muñoz Palacios

Dr. Sergio Salazar Cruz

**Ciudad de México.**

**Mayo 2023**

# Índice general

---

<b>Índice de tablas</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VI</b>
<b>Acrónimos</b>	<b>VIII</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Resumen</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Estado del Arte . . . . .	3
1.2.1. Consenso utilizando técnicas inteligentes . . . . .	4
1.3. Planteamiento del Problema . . . . .	5
1.4. Justificación . . . . .	6
1.5. Contribuciones . . . . .	7
1.6. Objetivos . . . . .	8
1.6.1. Objetivo General . . . . .	8
1.6.2. Objetivos Específicos . . . . .	8
1.7. Publicaciones . . . . .	9
1.8. Estructura de la tesis . . . . .	10

<i>ÍNDICE GENERAL</i>	III
<b>2. Modelado dinámico y preliminares.</b>	<b>12</b>
2.1. Modelado matemático . . . . .	12
2.2. Modelo dinámico del sistema UAV en <i>Gazebo</i> . . . . .	15
2.3. Teoría de grafos . . . . .	18
2.4. Redes neuronales de base radial . . . . .	19
2.4.1. Estrategia de control utilizando redes de base radial . . . . .	21
2.5. Redes neuronales recurrentes de alto orden . . . . .	24
2.5.1. Estrategia de control para redes neuronales recurrentes . . . . .	26
2.6. Estimación de parámetros mejorada . . . . .	32
2.6.1. Excitación Persistente . . . . .	35
<b>3. Estrategias de control inteligente para un UAV</b>	<b>38</b>
3.1. Estructura de control para un UAV en <i>Gazebo</i> . . . . .	39
3.2. Control basado en redes neuronales de base radial . . . . .	42
3.2.1. Estrategia de control con estimación tradicional . . . . .	42
3.2.2. Estrategia de control con aprendizaje mejorado . . . . .	44
3.3. Control basado en redes neuronales recurrentes . . . . .	50
3.3.1. Estrategia de control con estimación tradicional . . . . .	50
3.3.2. Estrategia de control con aprendizaje mejorado . . . . .	52
<b>4. Control cooperativo</b>	<b>58</b>
4.1. Introducción . . . . .	59
4.1.1. Estimador de parámetros para multiagentes . . . . .	59
4.2. Diseño de controlador inteligente cooperativo . . . . .	61
4.3. Análisis de estabilidad y convergencia . . . . .	65
<b>5. Resultados</b>	<b>70</b>
5.1. Controlador inteligente usando redes neuronales de base radial para un UAV . . . . .	71
5.2. Controlador inteligente usando redes neuronales recurrentes de alto orden para un UAV . . . . .	73
5.3. Resultados para multiagentes . . . . .	77
5.3.1. Ambiente de simulación . . . . .	77

5.3.2. Escenarios de simulación . . . . .	80
<b>6. Conclusiones</b>	<b>89</b>
6.1. Trabajo Futuro . . . . .	90
<b>Bibliografía</b>	<b>91</b>

# Índice de tablas

---

2.1. Parámetros identificados para la dinámica del controlador interno. . . . .	15
5.1. Parámetros y condiciones iniciales utilizados en los resultados de la simulación.	71
5.2. Parámetros y condiciones iniciales utilizados en el control de altitud de un UAV.	74
5.3. Parámetros y condiciones iniciales utilizados en los resultados de la simulación multiagente. . . . .	82
5.4. Tabla de comparación de los índices de desempeño integral de error . . . . .	83
5.5. Parámetros y condiciones iniciales utilizados en el ejemplo horizontal. . . . .	86

# Índice de figuras

---

1.1. Grupo de vehículos multirrotores [1] . . . . .	2
2.1. Estructura de un vehículo aéreo cuadirrotor con los marcos inerciales . . . . .	13
2.2. Datos para identificación del sistema en dinámica de $\theta$ . . . . .	16
2.3. Datos para identificación del sistema en dinámica de $\phi$ . . . . .	16
2.4. Datos para identificación del sistema en dinámica de $\psi$ . . . . .	17
2.5. Experimentos para la dinámica en $z$ mediante <i>Gazebo</i> . . . . .	17
2.6. Redes Neuronales de Base Radial . . . . .	19
2.7. Redes Neuronales Recursivas de Alto Orden . . . . .	26
5.1. Resultado del proceso de seguimiento en el eje $Z$ , donde $z_d$ representa la referencia deseada, $z_{PD}$ es el seguimiento con un controlador Proporcional Derivativo, $z_{PD+NN\sigma}$ utiliza un controlador PD y el término de la red neuronal con la actualización de pesos tradicional y $z_{PD+NNR}$ controlador PD y el término de la red neuronal actualizando los pesos mediante el estimador mejorado . . . . .	72
5.2. Resultado del seguimiento en el plano $X-Y$ , donde $xy_d$ representa la referencia deseada, $xy_{PD}$ es el seguimiento con un controlador Proporcional Derivativo, $xy_{PD+NN\sigma}$ utiliza un controlador PD y el término de la red neuronal con la actualización de pesos tradicional y $xy_{PD+NNR}$ controlador PD y el término de la red neuronal actualizando los pesos mediante el estimador mejorado . . . . .	73
5.3. Comparativo del UAV en altura utilizando diferentes estrategias de control . . . . .	75
5.4. Error sobre la dinámica de altura con actualización tradicional . . . . .	76
5.5. Error sobre la dinámica de altura utilizando el estimador mejorado . . . . .	76

5.6. Grafo de comunicación propuesto para los sistemas multiagente. . . . .	80
5.7. Consenso con controlador distribuido tradicional . . . . .	83
5.8. Consenso con término adaptable . . . . .	84
5.9. Consenso con nuestro algoritmo propuesto . . . . .	84
5.10. Desempeño del consenso en el eje $X$ . . . . .	86
5.11. Desempeño del consenso en el eje $Y$ . . . . .	87
5.12. Desempeño del consenso en el plano horizontal $X - Y$ . . . . .	87

# Acrónimos

---

Nombre	Descripción
<i>MAS</i>	Multi-Agent System
<i>UAV</i>	Unmanned Aerial Vehicle
<i>UGV</i>	Unmanned Ground Vehicle
<i>UVV</i>	Unmanned Underwater Vehicle
<i>LQR</i>	Linear Quadratic Regulator
<i>GPS</i>	Global Position System
<i>UWB</i>	Ultra Wideband
<i>DOF</i>	Degrees Of Freedom
<i>GDL</i>	Grados De Libertad
<i>RBF</i>	Radial Basis Function
<i>RHONN</i>	Recurrent High-Order Neural Network

# Agradecimientos

---

Al Consejo Nacional de Ciencia y Tecnología (**CONACYT**) y al centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (**CINVESTAV-IPN**) por el apoyo económico otorgado para la realización de mis estudios de Doctorado. Al Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos 295536 por el apoyo para la realización del presente trabajo de tesis. Agradezco a los integrantes del laboratorio UMI-LAFMIA, a los profesores que compartieron sus conocimientos para trabajar en este campo de de la ciencia, también a mis compañeros con los que he convivido durante estos años. Quiero agradecer a mis tutores, a los doctores Filiberto Muñoz Palacios y Sergio Salazar Cruz por haberme guiado y corregido para lograr concluir este trabajo. Especialmente a mi hermanita Samantha Morfin Santana por ser parte del motor de mi vida, a mis padres Lorena Santana Padilla y Martin Morfin Magaña por estar presentes en cada momento en mi vida sin importar lo malo o bueno de mis acciones se que siempre podre contar con ustedes. Andrea Hernández Sánchez gracias por compartir tu vida conmigo, cada alegría, consejo, logro y hasta desafíos que hemos pasado juntos. A toda mi familia empezando por mis abuelas y abuelos, tias y tios, primas y primos, a mis grandes amigos que también han estado presente en todo este viaje, gracias por tanto. Por último y no menos importante agradezco a Killian Alexander que viene en camino, tú eres y serás la persona que aun sin conocerte ya te amo con todo mi ser.

”Knowledge, like air, is vital to life. Like air, no one should be denied it.” - V.

”But maybe we’ve spent too long trying to figure all this out with theory.” - Brand

# Resumen

---

El desarrollo de este trabajo de tesis se realizó en el laboratorio UMI-LAFMIA en CINVESTAV unidad Zacatenco, con el objetivo proponer una serie de estrategias de control inteligente para un vehículo UAV basado en redes neuronales con la estimación mejorada para los pesos de dichas redes. Además, se desarrolla una estrategia de control cooperativo líder-seguidor basada en consenso y redes neuronales de base radial para un conjunto de vehículos multirrotores sujeto a perturbaciones externas y dinámicas no modeladas, donde los pesos de la red neuronal se adaptan tomando en cuenta el error de estimación de los pesos. Obtener la prueba de estabilidad de la estrategia propuesta para garantizar las condiciones de estabilidad sobre el sistema multi-agente en lazo cerrado y verificar la efectividad de estas estrategias en el ambiente de simulación *Gazebo*, con la finalidad de abordar la problemática de las tareas cooperativas y como mediante el apoyo del control inteligente y algoritmos de adaptación robusta podemos mejorar el seguimiento para sistemas multiagentes en presencia de perturbaciones agrupadas.

La plataforma donde se probaron los diferentes algoritmos es un vehículo multirrotores de nombre *Iris*, el cual cuenta con la capacidad de desplazarse en sus seis grados de libertad  $(x, y, z, \theta, \phi$  y  $\psi)$ , mediante un autopiloto interno que permite reconfigurar las señales de entrada y realizar la comunicación entre agentes.

Este tipo de herramientas de simulación como es *Gazebo* y los modelos que se pueden introducir a este ambiente como el vehículo UAV *Iris* puede ser útil como un gemelo digital y permita el desarrollo de otras estrategias de control y obtener resultados muy cercanos a una implementación en una plataforma física de experimentación y validar en ambientes reales la aplicación de diversos algoritmos de cooperación, estimadores o como en el caso de este trabajo de tesis la comparación entre estrategias de control inteligente y consenso.

## Abstract

This thesis work was developed in the UMI-LAFMIA laboratory at the CINVESTAV Zacateco unit, proposing a series of intelligent control strategies for a UAV vehicle based on neural networks with the improved estimation for the weights of said networks. In addition, to develop a cooperative leader-follower control strategy based on consensus and radial-based neural networks for a set of multicopter vehicles subject to external disturbances and unmodeled dynamics, where the neural network weights are adapted taking into account the error of weight estimation. Obtain the stability test of the proposed strategy to guarantee stability conditions on the closed-loop multi-agent system and verify the effectiveness of these strategies in the *Gazebo* simulation environment, to address the problem of cooperative tasks and how through the support of intelligent control and robust adaptation algorithms we can improve tracking for multi-agent systems in the presence of clustered disturbances.

The platform where the different algorithms were tested is a multicopter vehicle named *Iris*, which can move in its six degrees of freedom ( $x, y, z, \theta, \phi$ , and  $\psi$ ), through an internal autopilot that allows reconfiguring the input signals and communication between agents. This type of simulation tool such as *Gazebo* and the models that can be introduced to this environment such as the UAV *Iris* vehicle can be useful as a digital twin and allow the development of other control strategies and obtain results very close to an implementation in a physical platform for experimentation and validate in real environments the application of various cooperation algorithms, estimators or, as in the case of this thesis, the comparison between intelligent control strategies and consensus.



---

# Introducción

---

Este capítulo presenta los antecedentes y la motivación de este estudio de doctorado y analiza la aplicación del vuelo cooperativo con múltiples vehículos aéreos así como los problemas asociados con este. Además, se presentan los aportes y metodología de este estudio y finalmente se presenta una noción general en la estructura de esta tesis.

## 1.1. Antecedentes

Las tareas cooperativas desde el principio de la humanidad han sido pieza clave en el desarrollo de las civilizaciones, desde el punto de vista natural existen especies que funcionan o se relacionan y subsisten en forma de grupo o enjambre de forma tal que están organizados en comunidades y corporaciones que alcanzan objetivos particulares coordinando a cada integrante. Las últimas dos décadas hemos sido testigo del avance en el control cooperativo para sistemas multiagente. Como uno de los temas principales del control cooperativo, en particular la investigación en el protocolo de consenso es muy significativa debido a su potencial aplicación en ingeniería [2], [3]. Por otro lado, la investigación y desarrollo en el campo de los vehículos aéreos no tripulados está en continuo crecimiento, principalmente en actividades militares, pero con el paso del tiempo se convirtieron en plataformas robóticas muy útiles en diferentes campos como búsqueda y rescate, vigilancia, monitoreo y mapeo de entornos, entregas y tareas de transporte, entre otras [4],[5]. Además, es ampliamente reconocido tanto en la naturaleza como en los sistemas robóticos que el desarrollo de una tarea o misión de forma cooperativa ofrece diferentes ventajas, entre las que se encuentran la

reducción de tiempos, la robustez o la tolerancia a fallos ya que si algún miembro presenta problemas, puede ser reemplazado por algún otro agente [6], [7],[8].

En este sentido, los métodos de seguimiento de formaciones para sistemas multiagente (MAS) por sus siglas en ingles (Multi-Agent System) tienen una amplia gama de aplicaciones en el campo de los sistemas autónomos no tripulados inteligentes, como los Vehículos Aéreos No Tripulados (UAVs), los Vehículos Terrestres No Tripulados (Unmanned Ground Vehicles o UGVs) y los Vehículos Submarinos No Tripulados (Unmanned Underwater Vehicles o UUV). En particular para un grupo de UAVs como el que se muestra en la Figura 1.1, se han desarrollado varios trabajos de investigación. En [9], los problemas de control de seguimiento de formaciones distribuidas en tiempo finito se resuelven para sistemas multiagente cuadirrotores con perturbaciones externas. Al aplicar las estrategias de control robustas, varios UAV multirrotores pueden lograr el seguimiento de la formación en tiempo finito. Al usar los modos deslizantes, se describe un algoritmo de seguimiento en [10], el cual usa múltiples UAVs para seguir un objetivo y permanecer en la formación deseada simultáneamente.



Figura 1.1: Grupo de vehículos multirrotores [1]

## 1.2. Estado del Arte

Los algoritmos de consenso o control distribuido para vehículos autónomos o multi-agentes han sido objeto de una enorme investigación durante las últimas dos décadas. En gran medida por lo complejo que resulta trabajar con una sola plataforma experimental, al trabajar con más de una resulta en una tarea aún más complicada, además de todos los requisitos en cuanto a componentes, sensores, sistemas embebidos o computacionales añadiendo las normas y protocolos de seguridad que se deben de tomar en cuenta para que cualquier investigación tenga un impacto considerable. Por lo que es común encontrar enfoques e ideas generalizadas para simulaciones o análisis numéricos o inclusive experimentos en ambientes controlados. La cooperación de vehículos aéreos en una red es capaz de mejorar aún más el rendimiento, cubrir una mayor área y lo más importante brinda robustez a las tareas que esté desempeñando como mencionan en [12] al existir fallas con alguno de los agentes de la red éste puede ser reemplazado por algún otro agente vecino, inclusive al existir una red de comunicación entre ellos la información se comparte y aseguramos que los datos puedan llegar en buenas condiciones a su destino.

En general, el desarrollo y la implementación de sistemas multi-agentes efectivos implica abordar un amplio espectro de suposiciones complejas y asegurar la coordinación y colaboración adecuadas entre los agentes para alcanzar los objetivos deseados. Estos sistemas son utilizados en una variedad de campos, como la robótica, la inteligencia artificial, la logística, los juegos estratégicos y muchos otros, y representan un área de investigación en constante evolución en busca de soluciones innovadoras y eficientes. , en [13] utilizan un controlador robusto denominado estrategia de control en formación completamente adaptable en tiempo finito, donde estudian el comportamiento de una flotilla de helicópteros en el cual buscan mantener la formación entre ellos y contrarrestar los efectos de las perturbaciones como el viento y si existe alguna falla con algún agente tiene la capacidad de reconfigurar la formación. También tienen la capacidad de evadir obstáculos y colisiones entre ellos como el desarrollado en [14] con el apoyo de visión artificial. En [15] analizan el comportamiento de una flotilla de cuadrirrotos con un algoritmo de optimización (LQR) y así lograr mantener la formación en estructura. La comunicación entre los multi-agentes facilita el desarrollo de las tareas cooperativas como explican en [16] donde la información del (GPS) pasa por un módulo de ultra banda ancha

(UWB) por sus siglas en inglés, el cual mejora el vuelo cooperativo entre dos UAVs al punto de mantener la formación hasta con una carga colgada entre ambas aeronaves o recuperar un aeroplano no tripulado mediante una red sostenida entre un par de multirrotores tan solo con mantener la formación [17].

Con el objetivo de solucionar el problema del control de formación, se ha prestado más atención a los controles de consenso de los sistemas multiagente, debido a su amplia aplicación en varias áreas [18], [19], [20], [21]. Principalmente, el control por consenso tiene dos estrategias de control [22], [23], [24], el consenso sin líder y el líder-seguidor. En primer lugar, los protocolos de consenso se diseñaron para sistemas lineales, incluidos sistemas integradores de primer, segundo y alto orden [25], [26]. Debido a la aplicabilidad del algoritmo de consenso, se llevaron a cabo más desarrollos. La estrategia de consenso ha sido desarrollada para sistemas no lineales [28], con retardo de tiempo [29], [30], con entrada y actuador saturados [26], [31], con técnicas de modo deslizante [32], [33], también con respecto a algoritmos que tratan incertidumbres externas dentro de sistemas dinámicos desconocidos [34]. Con el desarrollo de los recursos computacionales se han propuesto estrategias enfocadas en el aprendizaje automático, donde a partir de las observaciones tanto de los estados en sistemas multiagente como del entorno se pretende mejorar tanto el rendimiento como la comunicación dentro del consenso [35],[36], [37]. Se puede encontrar un compendio ampliado de algoritmos de consenso en [38], [39].

### **1.2.1. Consenso utilizando técnicas inteligentes**

En los últimos años, para resolver el problema de las perturbaciones agrupadas, es decir, la dinámica no modelada y las perturbaciones externas, se han desarrollado varios enfoques de consenso no lineal utilizando redes neuronales. Además, estas estrategias inteligentes son eficientes cuando los parámetros del modelo varían mientras el sistema está operando, como es el caso de los UAV que realizan tareas de logística, ya que la masa total depende de los paquetes que se envían [40]. En este sentido, se ha realizado una investigación adicional sobre protocolos de consenso basados en redes neuronales en [41]-[43], por nombrar algunos. En [41] se presenta un consenso líder-seguidor con retroalimentación de estado y salida para sistemas mul-

tiagente no lineales inciertos, donde las dinámicas no lineales inciertas se aproximan a través de redes neuronales. El problema del control de consenso neuronal adaptativo para sistemas multi-agente no lineales no afines inciertos con topologías de conmutación insuficiente y direcciones de control desconocidas se aborda en [42], donde las redes neuronales de base radial aproximan la dinámica no lineal desconocida. En [44] se presenta un control de superficie dinámico neuronal basado en un predictor para una clase de sistemas no lineales inciertos, en contraste con las metodologías existentes donde los pesos de la red neuronal se actualizan en función de los errores de seguimiento, se emplea un predictor y los errores de predicción se utilizan para actualizar los pesos neuronales. En [45] se desarrolla una sólida estrategia de consenso para sistemas multiagente no lineales de alto orden con dinámicas desconocidas, ganancias de control desconocidas y fallas de actuadores desconocidas. En este enfoque, se combina un retardo con una red neuronal para garantizar que el error de seguimiento converja en una región pequeña. En [43] se desarrolla una estrategia de control de seguimiento de formación de realimentación de salida con incertidumbres de modelado sujetas a restricciones de comunicación, donde se utilizan redes neuronales de base radial para aproximar la dinámica incierta.

### **1.3. Planteamiento del Problema**

El control cooperativo de sistemas de vehículos aéreos multirrotores es un área de investigación en constante evolución. Para lograr un control preciso y estable de estos sistemas, es necesario desarrollar estrategias de control cooperativo que permitan la coordinación efectiva de los vehículos. Sin embargo, el desarrollo de estrategias de control cooperativo efectivas sigue siendo un desafío debido a la presencia de perturbaciones externas y dinámicas no modeladas en el sistema, que pueden afectar el rendimiento del controlador. Para la navegación autónoma y en especial para la navegación colaborativa una forma de abordarlo es mediante el control adaptable y en especial utilizando redes neuronales artificiales para aproximar las funciones generadas por la dinámica de los sistemas a controlar, Además, una limitación importante es la actualización de los pesos de dichas redes la cual es la única señal disponible que refleja la diferencia entre los parámetros estimados y los parámetros reales es la señal de error de seguimiento o predicción. Para aumentar la robustez de las leyes adaptables, se mejoraron los protocolos de consenso de redes neuronales introduciendo variantes en las leyes adaptables para los pesos de

la red neuronal, como la modificación  $-\sigma$  [46] y la modificación  $-e$  [47]. Sin embargo, con estos algoritmos, los parámetros estimados pueden permanecer alrededor de los valores preestablecidos solo debido a los términos de amortiguamiento inducidos, lo que a su vez provoca un tiempo de convergencia más largo y afecta la respuesta del error de seguimiento. Como se indica en [48],[52], la rápida y precisa convergencia de estimación de pesos sinápticos de una red neuronal en los sistemas de control adaptativo es valiosa para la estabilidad y robustez del sistema en lazo cerrado. Dado que se desconocen los pesos óptimos, la diferencia entre los pesos estimados y los pesos óptimos no está disponible para la medición. Motivado por este hecho, se introdujo por primera vez un enfoque de estimación adaptativa para parámetros desconocidos en [50], donde, a diferencia de los mecanismos de adaptación convencionales, la adaptación de parámetros incluye información sobre el error de estimación del parámetro para mejorar la convergencia del seguimiento error. Motivados por el uso del error de estimación de parámetros en las leyes adaptativas, la comunidad científica llevó a cabo más investigaciones utilizando la información del error de estimación. En [53], [54] se informó sobre estrategias robustas de control adaptativo con estimación de masa en línea para sistemas robóticos. En [55], [56] se desarrolla una estimación de parámetros adaptativos para estimar la pendiente del camino y los parámetros del vehículo para los sistemas vehiculares. Además, se desarrolla un control adaptable para suspensiones activas de vehículos con no linealidades desconocidas en [57], donde se usa explícitamente una ley adaptable donde el error de estimación del parámetro se usa como un término de fuga. La nueva estrategia de estimación de parámetros también se ha utilizado en sistemas aéreos. En [58] se estiman los parámetros de inercia y masa de un vehículo aéreo cuadrirrotor introduciendo el error de estimación de parámetros. Además, siguiendo la misma metodología, los parámetros de inercia y masa de una nave espacial 6DOF son estimadas en [59].

## 1.4. Justificación

Los UAV y en específico los multirrotores son sistemas no lineales, subactuados y con una dinámica estable en una región limitada. Recientemente, además de ser utilizada para estimar parámetros desconocidos, la metodología de estimación de parámetros se utiliza para estimar los pesos óptimos en sistemas de redes neuronales [60], [61] [62] [63]. En [60] se desarrolla

un control adaptativo de manipuladores de robots que combina un controlador neuronal con un controlador robusto. Los esquemas propuestos garantizan que los pesos neuronales estimados converjan a los óptimos en un tiempo finito. También para sistemas robóticos, se desarrolla un controlador de modo deslizante combinado con una red neuronal en [61]. En este enfoque para los pesos neuronales se presenta una ley adaptativa basada en el error de estimación. Para un sistema de robot de doble brazo y el control adaptativo está diseñado en [63], donde se diseña una técnica de control de filtrado de comandos para el control de seguimiento de la trayectoria. Además, para tratar las incertidumbres dinámicas, se utiliza una red neuronal de base radial, donde la información del error estimado se utiliza en ley adaptativa para los pesos neuronales.

Se puede notar que sin la convergencia a los pesos óptimos de las redes neuronales, la compensación de la dinámica desconocida a través del sistema de redes neuronales puede no ser lo suficientemente buena (por ejemplo, convergencia lenta, gran sobreimpulso) y la propiedad de no convergencia de los pesos a sus valores ideales también puede degradar el desempeño en el seguimiento de una referencia. Además, lo anterior indica que se puede mejorar la convergencia de pesos si se puede incorporar información del error de estimación a la ley de adaptación. En este sentido, el desarrollo de un protocolo de consenso adaptativo basado en redes neuronales con la información del error se mejora el seguimiento. Hasta donde sabemos, la información del error de estimación no se ha incorporado en el desarrollo de estrategias de control de consenso basadas en redes neuronales para sistemas multiagente. Motivados por las ideas anteriores, en este trabajo de investigación, proponemos una estrategia robusta de control de consenso adaptativo para un grupo de vehículos aéreos quadrotor con perturbaciones agrupadas (es decir, no linealidades parcialmente desconocidas y perturbaciones externas). Para estimar y compensar las perturbaciones agrupadas, se introduce una red neuronal de base radial en el protocolo de consenso. Para calcular la información del error de estimación de los pesos se calculan un conjunto de variables intermedias de primer orden. Finalmente, las leyes adaptativas para los pesos de las redes neuronales incorporan la información del error de estimación.

## 1.5. Contribuciones

Las principales contribuciones de esta tesis se pueden resumir en

- El desarrollo de un algoritmo de consenso líder-seguidor mejorado para reducir el error de seguimiento, con una red neuronal de base radial para sistemas aéreos no tripulados con múltiples agentes sujetos a perturbaciones agrupadas, utilizando para actualizar los pesos de la red neuronal artificial un conjunto de filtros de primer orden y matrices auxiliares para obtener la información del error de estimación de los pesos de la red.
- Realizar un análisis de estabilidad basado en la teoría de Lyapunov para el sistema en lazo cerrado, el cual permite garantizar que todas las señales en el sistema permanecen uniformemente últimamente acotadas y se logra el objetivo de control líder-seguidor. Hasta lo mejor de nuestro conocimiento, el análisis de estabilidad para el algoritmo de control y estimación de los pesos con el enfoque presentado, no ha sido reportado en la literatura.

## 1.6. Objetivos

Tanto el objetivo general como los objetivos particulares a cumplir con el desarrollo de la presente tesis se describen a continuación.

### 1.6.1. Objetivo General

Desarrollar estrategias de control cooperativo del tipo líder-seguidor basadas en el algoritmo de consenso utilizando redes neuronales estáticas, para un sistema de vehículos aéreos multirrotores, mediante técnicas de adaptación mejorada para los pesos de las redes neuronales.

### 1.6.2. Objetivos Específicos

1. Proponer una estrategia de control para un vehículo UAV basado en redes neuronales de base radial, donde los pesos de la *RBF* se actualicen considerando la información del error de estimación.
2. Proponer una estrategia de control para un vehículo UAV basado en redes neuronales recurrentes de alto orden, donde los pesos se actualicen con la estimación mejorada mediante la información del error y lograr la convergencia del mismo a una región mas compacta.

3. Proponer una estrategia de control cooperativo líder-seguidor basada en consenso y redes neuronales de base radial para un conjunto de vehículos multirrotores sujeto a perturbaciones externas y dinámicas no modeladas, donde los pesos de la red neuronal se adapten tomando en cuenta el error de estimación de los pesos.
4. Realizar la prueba de estabilidad de la estrategia propuesta para garantizar las condiciones de estabilidad sobre el sistema multi-agente en lazo cerrado.
5. Realizar la simulación de las estrategias desarrolladas en el ambiente de Gazebo.

## 1.7. Publicaciones

Derivado de las actividades realizadas durante el desarrollo del presente trabajo de tesis, se obtuvieron las siguientes publicaciones.

### Revistas internacionales

- **Alejandro Morfin-Santana**, Filiberto Muñoz, Sergio Salazar, José Manuel Valdovinos. *Robust Neural Network Consensus for Multiagent UASs Based on Weights' Estimation Error*. *Drones*, no. 6, 300, 2022.
- José Valdovinos, Filiberto Muñoz, Jorge Cervantes Rojas, Sergio Salazar, **Alejandro Morfin-Santana**. *Leader-Follower Consensus Control for a Class of Nonlinear Multi-agent Systems Using Dynamical Neural Networks*. *Neurocomputing*, Febrero 2023 (Sometido)

### Congresos internacionales

- **Alejandro Morfin-Santana**, Filiberto Muñoz, Sergio Salazar, Iván González-Hernández, Eduardo Espinoza Quesada and Rogelio Lozano. Real-time people detection from thermal images by using an Unmanned Aerial System, *16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Ciudad de México, México, 2019.

## 1.8. Estructura de la tesis

El presente trabajo de tesis consta de 6 capítulos, la organización de cada capítulo se presenta a continuación. En el Capítulo 1 se presenta una revisión de la literatura sobre las técnicas y enfoques generalizados que se han tratado para resolver la problemática de realizar tareas cooperativas para vehículos aéreos además de presentar la justificación y objetivos del presente trabajo. En el Capítulo 2 se obtiene el modelo matemático para un vehículo multirrotor utilizando el enfoque de Euler-Lagrange, así como el modelo del vehículo tomando en cuenta los controles de orientación internos que tiene el vehículo en la plataforma de Gazebo. Se presentan los preliminares de la Teoría de Grafos necesarios para el desarrollo de las estrategias de control cooperativas. También, se presentan las redes neuronales estáticas y recurrentes que serán utilizadas y finalmente el estimador de parámetros que nos permitirá obtener una estimación mejorada dentro de la aproximación en las funciones de las redes neuronales. Las estrategias de control para un sólo vehículo utilizando las redes neuronales estáticas y recurrentes ya con el aprendizaje mejorado para los pesos de la red neuronal son presentadas en el Capítulo 3. El Capítulo 4 presenta la parte central de este tema de tesis, el desarrollo de los algoritmos de control cooperativo del tipo líder-seguidor, basado en redes neuronales de base radial. Para el algoritmo basado en redes neuronales de base radial, se propone la ley de adaptación mejorada para cada uno de los agentes seguidores. También se presenta un análisis de estabilidad basado en la teoría de Lyapunov para el sistema multi-agente en lazo cerrado. Los resultados de simulación obtenidos en el ambiente de Gazebo son presentados en el Capítulo 5. Se muestran los resultados para un vehículo multirrotor aplicando los dos tipos de red neuronal considerados en este trabajo, también se presentan los resultados en simulación para el control cooperativo basado en redes neuronales de base radial. Finalmente, las conclusiones del trabajo de tesis y las direcciones de investigación futuras son abordadas en el Capítulo 6.



---

# Modelado dinámico y preliminares.

---

En este capítulo se presenta el modelo matemático para un vehículo multirrotor utilizando el enfoque de Euler-Lagrange en la Sección 2.1, los cuales funcionan como plataformas robóticas aéreas con características notables, tales como la capacidad de mantener su posición en algún punto en el aire, despegar y aterrizar verticalmente. El vehículo multirrotor es un sistema subactuado de múltiples entradas y múltiples salidas que cuenta con 6 GDL (grados de libertad) y una dinámica no lineal y fuertemente acoplada. Además del modelado del vehículo multirrotor, en la Sección 2.2 se presenta el modelo de este mismo vehículo utilizado en la plataforma de simulación de *Gazebo*, debido a que en esta plataforma el vehículo ya cuenta internamente con controles de orientación y altura. En la Sección 2.3 se presenta la teoría de grafos que será utilizada posteriormente para el desarrollo de las estrategias de control cooperativo. Las estructuras de las redes neuronales de base radial y las redes neuronales dinámicas son presentadas en la Sección 2.4 y 2.5, respectivamente. Además, se presenta el diseño de una estrategia de control para un sistema de segundo orden, utilizando cada una de las redes neuronales. Finalmente, en la Sección 2.6 se presenta un algoritmo de estimación de parámetros mejorado basado en un conjunto de filtros de primer orden, el cual permite introducir la información del error de estimación en la ley de adaptación de los parámetros.

## 2.1. Modelado matemático

Un cuadrirrotor puede ser considerado como un cuerpo rígido de 6 GDL pero solo cuatro de estos son actuados. La obtención del modelo dinámico del vehículo es basada en el desarrollo

presentado en [66]. Considere el vehículo presentado en la Figura 2.1, se asume un marco fijo al cuerpo  $\Gamma_B = \{X_B, Y_B, Z_B\}$  el cual está localizado en el centro de gravedad del UAV y un marco inercial  $\Gamma_I = \{X_I, Y_I, Z_I\}$ . Sean  $\eta = [\phi \ \theta \ \psi]^T$  y  $\xi = [x, y, z]$  el vector de posición angular y de posición de traslación definidos en el marco inercial, respectivamente, donde  $\psi$ ,  $\theta$  y  $\phi$  se denominan ángulos de Euler que representan la guiñada, cabeceo y balanceo del UAV, respectivamente. Los vectores de velocidad angular y de traslación definidos en el marco del cuerpo, están dados por  $\dot{\eta}_B = [p, q, r]$  y  $\dot{\xi}_B = [u, v, w]$ , respectivamente.

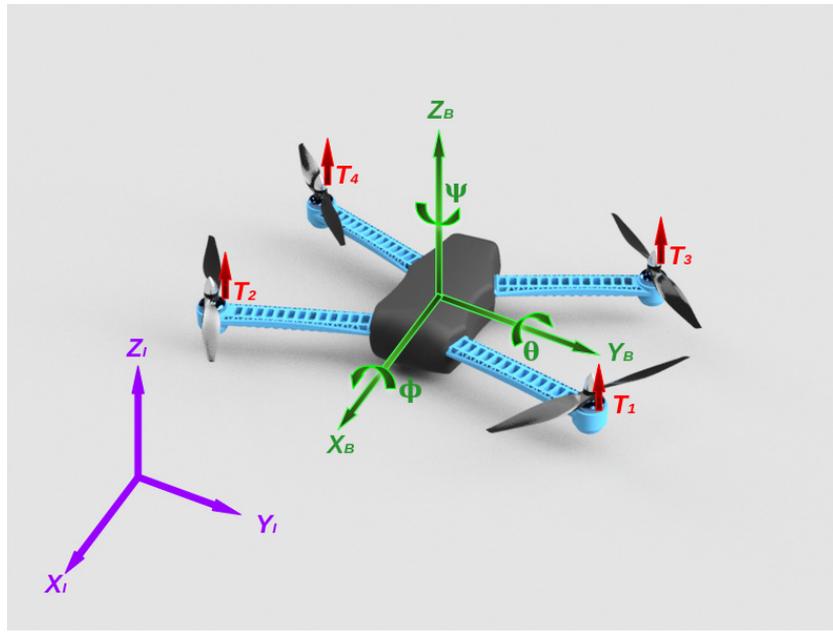


Figura 2.1: Estructura de un vehículo aéreo cuadrirrotor con los marcos inerciales

La matriz de rotación  $R(\eta)$  que transforma un vector del marco del cuerpo al marco inercial, está dada por

$$R(\eta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.1)$$

donde  $s(\cdot)$ ,  $c(\cdot)$  representan  $\sin(\cdot)$  y  $\cos(\cdot)$ , respectivamente. Sean  $\dot{\xi}$  y  $\dot{\eta}$  las velocidades lineal y angular del vehículo con respecto al marco inercial. Las velocidades angulares del cuerpo están relacionadas con las velocidades angulares del marco inercial mediante  $\dot{\eta} = W_\eta \dot{\eta}_B$ , donde

la matriz  $W_\eta$  se define como

$$W_\eta = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad (2.2)$$

Usando el enfoque de Euler-Lagrange, la dinámica del UAV expresada en el marco inercial, está dada de la siguiente manera

$$m\ddot{\xi} = R(\eta)v_z u_1 - mgv_z \quad (2.3a)$$

$$J(\eta)\ddot{\eta} = \tilde{\tau} - C(\eta, \dot{\eta})\dot{\eta} \quad (2.3b)$$

donde  $m \in \mathbb{R}$  es la masa en el vehículo y  $g \in \mathbb{R}$  es la constante de aceleración gravitatoria.  $J(\eta) = W_\eta^\top I_B W_\eta$  actúa como matriz de inercia para la rotación completa del vehículo expresada directamente en términos de los ángulos en las coordenadas generalizadas de  $\eta$ ,  $v_z = [0, 0, 1]^\top$ , la fuerza de empuje está dada por  $u_1 = F_1 + F_2 + F_3 + F_4$ , y la matriz de Coriolis está dada por  $C(\eta, \dot{\eta})$  la cual puede consultarse en [67]. Se propone una entrada de control  $\tilde{\tau}$  definida como

$$\tilde{\tau} = J(\eta) I_B^{-1} \tau + C(\eta, \dot{\eta})\dot{\eta} \quad (2.4)$$

donde el vector de entrada auxiliar está definido como  $\tilde{\tau} = [u_2, u_3, u_4]^\top$ .

Por ultimo, usando (2.3)–(2.4), el modelo dinámico del UAV puede ser representado por

$$\ddot{x} = \frac{u_1}{m} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \quad (2.5a)$$

$$\ddot{y} = \frac{u_1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \quad (2.5b)$$

$$\ddot{z} = \frac{u_1}{m} (\cos \theta \cos \phi) - g \quad (2.5c)$$

$$\ddot{\theta} = \frac{u_2}{I_x} \quad (2.5d)$$

$$\ddot{\phi} = \frac{u_3}{I_y} \quad (2.5e)$$

$$\ddot{\psi} = \frac{u_4}{I_z} \quad (2.5f)$$

## 2.2. Modelo dinámico del sistema UAV en *Gazebo*

Para las pruebas en simulación que se describen a partir del cuarto capítulo, utilizamos el UAV que está disponible en el software de simulación de *Gazebo*, el cual incluye el firmware del piloto automático *PX4*. Este firmware incluye un controlador de orientación y un control de altura. Por lo tanto, es necesario identificar las dinámicas del UAV dentro de la plataforma de simulación, con la finalidad de obtener un modelo dinámico que incluya las dinámicas de los controladores internos.

Para llevar a cabo la identificación de las dinámicas internas para los controladores de orientación y altura, se llevaron a cabo una recopilación de datos en el ambiente de *Gazebo*, donde se utilizaron diferentes señales de referencia. Con los datos obtenidos y utilizando el método de mínimos cuadrados en *MATLAB*, se estimaron los parámetros de un modelo de segundo orden lineal para las dinámicas del controlador interno. La función de transferencia propuesta para el modelo de segundo orden está dada por

$$F(s) = \frac{\alpha_3}{s^2 + \alpha_1 s + \alpha_2} \quad (2.6)$$

donde los coeficientes obtenidos para las funciones de transferencia se resumen en la Tabla 2.1. Los coeficientes  $a_i$  son para la dinámica de altura, con respecto a la dinámica de orientación en roll/alabeo y pitch/cabeceo corresponden los coeficientes  $b_i$  y  $c_i$  respectivamente, y los coeficientes  $d_i$  para la dinámica de yaw/guiñada, donde  $i = 1, 2, 3$  y el coeficiente del subíndice 3 afecta directamente a la entrada y los demás se relacionan a la salida como la posición y velocidad de su respectiva dinámica.

Tabla 2.1: Parámetros identificados para la dinámica del controlador interno.

$a_1 = 0.1862$	$b_1 = 179.523$	$c_1 = 170.635$	$d_1 = 9.769$
$a_2 = 0.1093$	$b_2 = 25.353$	$c_2 = 22.747$	$d_2 = 29.289$
$a_3 = 1.2737$	$b_3 = 177.246$	$c_3 = 167.685$	$d_3 = 29.283$

Podemos observar en las Figuras 2.2, 2.3 y 2.4 una serie de pruebas con los datos obtenidos en *Gazebo* (los cuales pueden observarse en la línea punteada) y los datos del modelo de segundo orden identificado (línea continua azul) a partir de la señal de entrada (línea continua verde).

Se puede observar que para los tres ángulos, la respuesta del vehículo sigue satisfactoriamente las señales de referencia la cual denominamos  $u_i$  donde  $i = \theta, \phi, \psi$ .

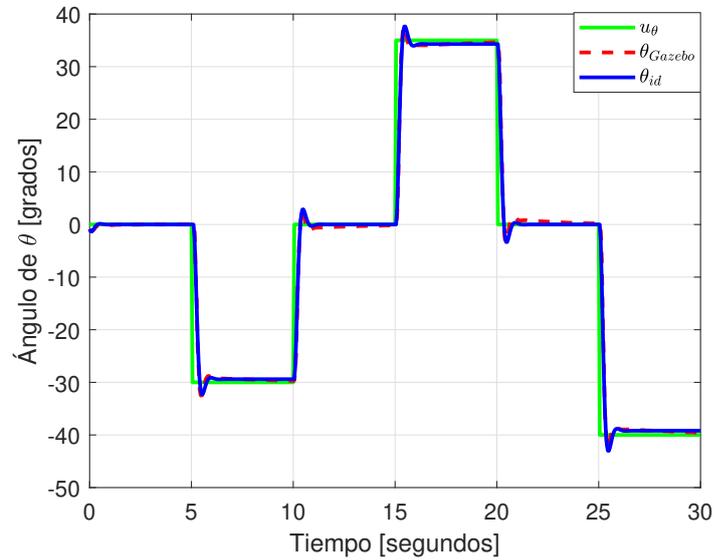


Figura 2.2: Datos para identificación del sistema en dinámica de  $\theta$ .

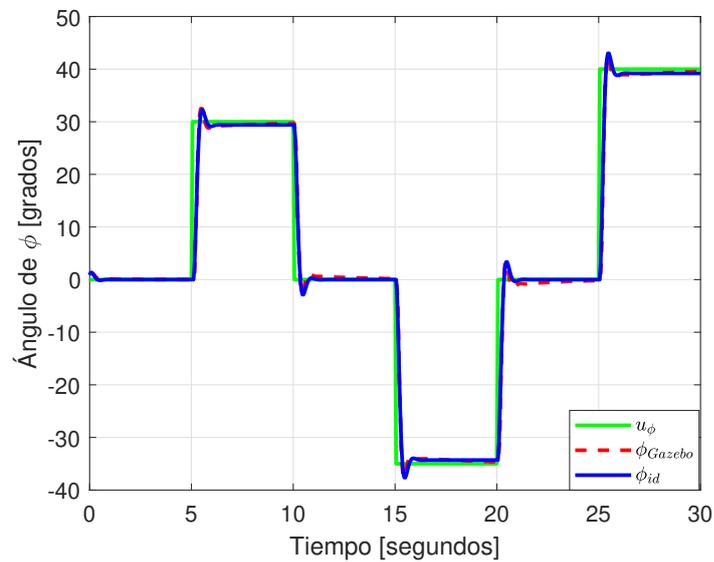


Figura 2.3: Datos para identificación del sistema en dinámica de  $\phi$ .

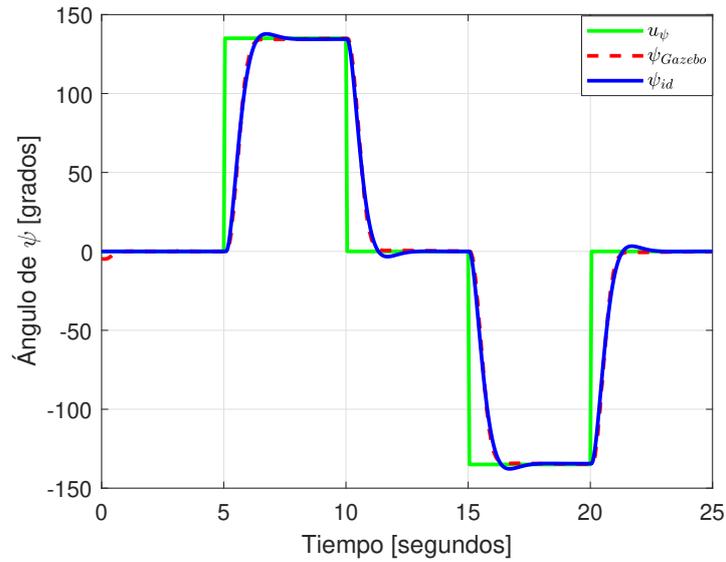


Figura 2.4: Datos para identificación del sistema en dinámica de  $\psi$ .

Para los coeficientes del sistema de altura se utilizó como entrada de control la función escalón llegando a un valor de 0.75 ya que éste sería el valor medio para el rango de valores entre 0.0 y 1.0 tomando en cuenta que el controlador interno considera 0.5 como el valor de referencia para comenzar a vencer la gravedad del vehículo entero. Podemos observar en la Figura 2.5 la salida de la prueba realizada.

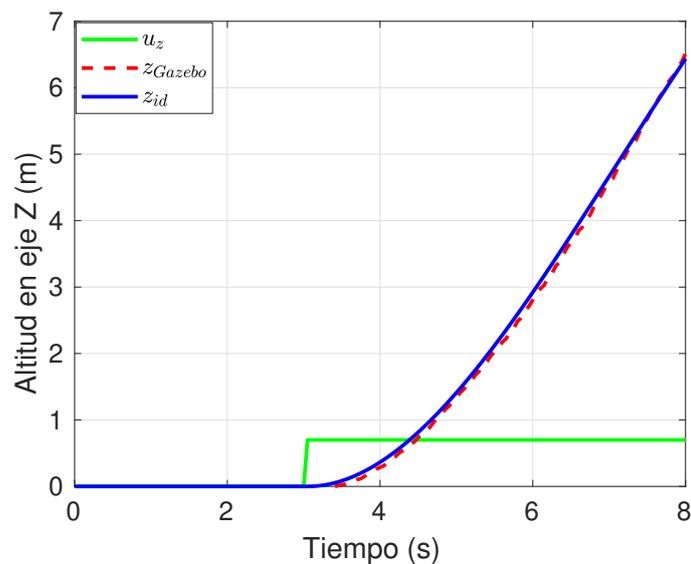


Figura 2.5: Experimentos para la dinámica en  $z$  mediante *Gazebo*

Finalmente, el modelo dinámico general presentado en la ecuación (2.5), para el vehículo

*Iris* Drone del ambiente de *Gazebo*, está dada por

$$\ddot{x} = \frac{u_1}{m} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \quad (2.7a)$$

$$\ddot{y} = \frac{u_1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \quad (2.7b)$$

$$\ddot{z} = -a_1 \dot{z} - a_2 z + a_3 u_z \quad (2.7c)$$

$$\ddot{\theta} = -b_1 \dot{\theta} - b_2 \theta + b_3 u_\theta \quad (2.7d)$$

$$\ddot{\phi} = -c_1 \dot{\phi} - c_2 \phi + c_3 u_\phi \quad (2.7e)$$

$$\ddot{\psi} = -d_1 \dot{\psi} - d_2 \psi + d_3 u_\psi \quad (2.7f)$$

donde  $u_z$ ,  $u_\theta$ ,  $u_\phi$  y  $u_\psi$  indican las entradas de control al vehículo aéreo cuadrirrotor en el firmware PX4. Cabe destacar que las ecuaciones para las dinámicas de orientación y altura fueron sustituidas por los modelos lineales de segundo orden identificados.

## 2.3. Teoría de grafos

Considere un sistema multiagente que consta de  $N$  agentes y un líder. Si cada agente se considera como un nodo, la comunicación entre los agentes se puede describir mediante un grafo  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , donde  $\mathcal{V} = \{n_1, \dots, n_N\}$  es un conjunto de nodos y  $\mathcal{E} = \{(n_i, n_j) \in \mathcal{V} \times \mathcal{V}\}$  es un conjunto de enlaces o aristas con el elemento  $(n_i, n_j)$  que describe la comunicación del nodo  $i$  a  $j$ . El conjunto vecino del nodo  $i$  se denota por  $\mathcal{N}_i = \{j | (n_j, n_i) \in \mathcal{E}\}$ .

Se define la matriz de adyacencia  $\mathcal{A} = (a_{ij}) \in \mathbb{R}^{N \times N}$ , donde  $a_{ij} = 1$ , si  $(n_j, n_i) \in \mathcal{E}$ , y  $a_{ij} = 0$ , de lo contrario. Se define la matriz de grados de entrada como una matriz diagonal  $\mathcal{D} = \text{diag}(d_i) \in \mathbb{R}^{N \times N}$  con  $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$  para el nodo  $i$ . La matriz laplaciana  $\mathcal{L} = (l_{ij}) \in \mathbb{R}^{N \times N}$  asociada con el grafo  $\mathcal{G}$  se define como  $\mathcal{L} = \mathcal{D} - \mathcal{A}$ . Si  $a_{ij} = a_{ji}$ , para  $i, j = 1, \dots, N$ , entonces el grafo  $\mathcal{G}$  se vuelve no dirigido. Una ruta en un grafo es una secuencia ordenada de nodos de manera que hay un enlace entre dos nodos consecutivos en la secuencia. Un grafo no dirigido esta conectado si existe un camino entre cada par de nodos. Finalmente, defina la matriz de adyacencia líder como  $\mathcal{A}_0 = \text{diag}(a_{i0}) \in \mathbb{R}^{N \times N}$ , donde  $a_{i0} > 0$  si y solo si el  $i$ -ésimo agente tiene acceso a la información del líder; de lo contrario  $a_{i0} = 0$ . Si el grafo no dirigido

está conectado y al menos un agente seguidor tiene acceso al líder, entonces la matriz  $\mathcal{H}$  es simétrica y definida positiva, donde  $\mathcal{H} = \mathcal{L} + \mathcal{A}_0 \in \mathbb{R}^{N \times N}$  [68].

## 2.4. Redes neuronales de base radial

A partir de que las redes neuronales artificiales se proponen para el control adaptable de sistemas dinámicos no lineales [69], las redes neuronales de base radial (o por sus siglas en inglés RBF) han llamado la atención debido a su buena capacidad de generalización y una estructura de red simple que evita cálculos innecesarios y extensos en comparación con la redes continuas multicapas. Investigaciones previas de los teoremas de aproximación universal en redes neuronales RBF han demostrado que cualquier función no lineal sobre un conjunto compacto con precisión arbitraria puede aproximarse mediante una red neuronal RBF [70].

Este tipo de red neuronal RBF cuenta con tres capas: la capa de entrada, la capa oculta y la capa de salida, como se puede observar en la Figura 2.6. Las neuronas de la capa oculta se activan mediante una función de base radial.

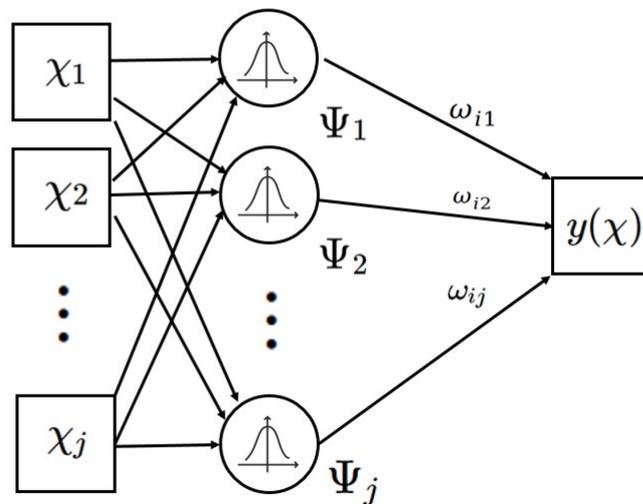


Figura 2.6: Redes Neuronales de Base Radial

Cada neurona de la capa oculta contiene un vector central  $c_j$  que es un vector de parámetros de la misma dimensión que el vector de entrada  $\chi$ ; la distancia euclidiana entre el centro y el vector de entrada de la red está definida por  $\|\chi(t) - c_j(t)\|$ . La salida de cada una de las neuronas de la capa oculta se obtiene a través de una función de activación no lineal  $\Psi_j(t)$  como se describe a continuación:

$$\Psi_j(t) = e^{-\frac{\|\chi(t) - c_j(t)\|^2}{2b_j^2}} \quad j = 1, \dots, s \quad (2.8)$$

donde  $b_j$  es un escalar positivo que determina la amplitud de la función de base radial y  $s$  representa el número de neuronas ocultas.

La capa de salida es una combinación ponderada lineal de la siguiente manera

$$y(t) = \sum_{j=1}^s w_j \Psi_j(t), \quad j = 1, \dots, s \quad (2.9)$$

donde  $y(t)$  representa la salida de la red,  $w_j$  son los pesos de la capa de salida. Como podemos notar este tipo de redes neuronales están gobernadas por un vector de pesos constantes y la función de activación mapea los valores de entrada a una región constante delimitada por los valores de centroides  $c_j$  y el ancho  $b_j$  por lo que podemos clasificar estas redes neuronales como estáticas. Ahora si suponemos que esta salida en (2.9) puede aproximar linealmente cualquier función continua como se menciona en [71], podemos representar cualquier perturbación agrupada  $d_\chi$  mediante la siguiente ecuación

$$d_\chi = W^{*\top} \Psi_\chi + \epsilon \quad \forall \quad \chi \in \mathbf{D} \quad (2.10)$$

donde  $W^{*\top}$  son los pesos óptimos,  $\Psi_\chi$  representa el vector de funciones de activación y  $\epsilon$  es el error de aproximación que satisface  $|\epsilon| \leq \epsilon_M$  con  $\epsilon_M$  también como una constante positiva, y  $\mathbf{D} \subset \mathbb{R}^n$  es un dominio suficientemente amplio.

### 2.4.1. Estrategia de control utilizando redes de base radial

Considere el siguiente sistema de segundo orden

$$\ddot{\chi} = f(\chi, \dot{\chi}) + bu_{\chi} + d_{\chi} \quad (2.11)$$

donde todos los términos de (2.11) son escalares. Las dinámicas conocidas del sistema están contenidas en  $f(\chi, \dot{\chi})$ ,  $b$  es un parámetro constante conocido diferente de cero,  $u_{\chi}$  es la entrada de control, el término  $d_{\chi}$  contiene las perturbaciones agrupadas desconocidas (dinámicas no modeladas y perturbaciones externas). El error de seguimiento se define como  $e_{\chi} = \chi - \chi_d$ , donde la señal de referencia está dada por  $\chi_d$ . El término  $d_{\chi}$  se aproxima mediante una red neuronal RBF, como en la ecuación (2.10). Por lo tanto, la dinámica del error de seguimiento está dada por

$$\ddot{e}_{\chi} = \ddot{\chi} - \ddot{\chi}_d = f(\chi, \dot{\chi}) + bu_{\chi} + W_{\chi}^{*\top} \Psi_{\chi}(\chi, \dot{\chi}) + \epsilon_{\chi} - \ddot{\chi}_d \quad (2.12)$$

Considere la entrada de control  $u_{\chi}$  definida como

$$u_{\chi} = \frac{-k_{1_{\chi}} e_{\chi} - k_{2_{\chi}} \dot{e}_{\chi} - \hat{W}_{\chi}^{\top} \Psi_{\chi}(\chi, \dot{\chi}) - f(\chi, \dot{\chi}) + \ddot{\chi}_d}{b} \quad (2.13)$$

donde  $\hat{W}_{\chi}$  son la estimación de los pesos óptimos  $W_{\chi}^*$ . Sustituyendo la señal de control (2.13) dentro de (2.12) obtenemos

$$\ddot{e}_{\chi} = -k_{1_{\chi}} e_{\chi} - k_{2_{\chi}} \dot{e}_{\chi} + \tilde{W}_{\chi}^{\top} \Psi_{\chi}(\chi, \dot{\chi}) + \epsilon_{\chi} \quad (2.14)$$

donde  $\tilde{W}_{\chi} = W_{\chi}^* - \hat{W}_{\chi}$  representa el error de estimación entre los pesos sinápticos ideales y los pesos estimados, se asume que los pesos ideales están acotados por  $\|W_{\chi}^*\|^2 \leq W_{\chi}^+$ , con  $W_{\chi}^+$  una constante positiva. También, las ganancias  $k_{1_{\chi}}$  y  $k_{2_{\chi}}$  son positivas.

Definiendo el vector  $E_{\chi} = [e_{\chi} \quad \dot{e}_{\chi}]^{\top}$ , la dinámica del error de seguimiento (2.14), representada en espacio de estados, está dada por

$$\dot{E}_{\chi} = AE_{\chi} + B \left( \tilde{W}_{\chi}^{\top} \Psi_{\chi}(\chi, \dot{\chi}) + \epsilon_{\chi} \right) \quad (2.15)$$

donde

$$A = \begin{bmatrix} 0 & e_x \\ -k_{1_x} & -k_{2_x} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.16)$$

La ley de actualización de los pesos de la red neuronal, utilizando la modificación- $\sigma$ , se propone como

$$\dot{\hat{W}}_x = \Gamma_x \left( \Psi_x(\chi, \dot{\chi}) B^\top P_x E_x - \kappa_x \hat{W}_x \right) \quad (2.17)$$

donde  $\kappa_x \in \mathbb{R}$  es una constante positiva,  $\Gamma_x = \text{diag}\{\Gamma_{x1}, \dots, \Gamma_{xs}\} \in \mathbb{R}^{s \times s}$ , con los términos  $\Gamma_{x_i}$  siendo escalares positivos,  $0 < P_x = P_x^\top \in \mathbb{R}^{2 \times 2}$ .

**Teorema 2.1.** *Considere el sistema de segundo orden dado por la ecuación (2.11) sujeto a las perturbaciones agrupadas  $d_x$ . Si la ley de control está dada por (2.13) y la ley de actualización de los pesos dada por (2.17), entonces el error de seguimiento  $e_x$  converge a una región acotada centrada en el origen.*

*Demostración.* Considere la siguiente función candidata de Lyapunov

$$V_1(E_x, \tilde{W}_x) = \frac{1}{2} E_x^\top P_x E_x + \frac{1}{2} \tilde{W}_x^\top \Gamma_x^{-1} \tilde{W}_x \quad (2.18)$$

Derivando  $V_1$  a través de todas sus trayectorias obtenemos

$$\begin{aligned} \dot{V}_1 &= E_x^\top P_x \dot{E}_x + \tilde{W}_x^\top \Gamma_x^{-1} \dot{\tilde{W}}_x \\ &= E_x^\top P_x \left( A E_x + B \left( \tilde{W}_x^\top \Psi_x(\chi, \dot{\chi}) + \epsilon_x \right) \right) + \tilde{W}_x^\top \Gamma_x^{-1} \dot{\tilde{W}}_x \\ &= E_x^\top P_x A E_x + E_x^\top P_x B \tilde{W}_x^\top \Psi_x(\chi, \dot{\chi}) + E_x^\top P_x B \epsilon_x + \tilde{W}_x^\top \Gamma_x^{-1} \dot{\tilde{W}}_x \\ &= \frac{1}{2} E_x^\top (A^\top P_x + P_x A) E_x - \tilde{W}_x^\top \left( \Psi_x(\chi, \dot{\chi}) E_x^\top P_x B - \kappa_x \hat{W}_x \right) + E_x^\top P_x B \epsilon_x \end{aligned} \quad (2.19)$$

sustituyendo (2.17) y aplicando la desigualdad de Young al término  $E_x^\top P_x B \epsilon_x$  se obtiene

$$\begin{aligned} \dot{V}_1 &\leq \frac{1}{2} E_x^\top (A^\top P_x + P_x A + P_x B B^\top P_x^\top) E_x + \kappa_x \tilde{W}_x^\top \hat{W}_x + \frac{1}{2} \epsilon_x^2 \\ &\leq \frac{1}{2} E_x^\top (A^\top P_x + P_x A + P_x B B^\top P_x^\top) E_x - \kappa_x \tilde{W}_x^\top \tilde{W}_x + \kappa_x \tilde{W}_x^\top W_x^* + \frac{1}{2} \epsilon_x^2 \end{aligned} \quad (2.20)$$

Completando el cuadrado para el término  $\tilde{W}_x^\top W_x^*$  se tiene que  $\tilde{W}_x^\top W_x^* \leq \frac{1}{2} \tilde{W}_x^\top \tilde{W}_x + \frac{1}{2} W_x^{*\top} W_x^*$ ,

con lo cual se obtiene

$$\begin{aligned} \dot{V}_1 \leq & \frac{1}{2} E_x^T (A^T P_x + P_x A + P_x B B^T P_x^T + Q_x) E_x - \frac{1}{2} E_x^T Q_x E_x \\ & - \frac{\kappa_x}{2} \tilde{W}_x^T \tilde{W}_x + \frac{\kappa_x}{2} W_x^{*T} W_x^* + \frac{1}{2} \epsilon_x^2 \end{aligned} \quad (2.21)$$

donde  $0 < Q_x = Q_x^T \in \mathbb{R}^{2 \times 2}$ . Las condiciones para que exista la solución de la ecuación algebraica de Riccati  $A^T P_x + P_x A + P_x B B^T P_x^T + Q_x = 0$  pueden encontrarse en [74]. Con lo cual se obtiene lo siguiente

$$\dot{V}_1 \leq -\frac{1}{2} E_x^T Q_x E_x - \frac{\kappa_x}{2} \tilde{W}_x^T \tilde{W}_x + \beta_x \quad (2.22)$$

donde  $\beta_x = \frac{\kappa_x}{2} W_x^+ + \frac{1}{2} \epsilon_x^2$ . Notar que, a partir de la función candidata de Lyapunov (2.18) y la desigualdad de Rayleigh–Ritz, tenemos

$$\dot{V}_1 \leq -\frac{1}{2} \lambda_{\min}(P_x) \|E_x\|^2 - \frac{\kappa_x}{2} \lambda_{\min}(\Gamma_x^{-1}) \|\tilde{W}_x\|^2 + \beta_x \quad (2.23)$$

Reorganizando la expresión obtenemos

$$\dot{V}_1 \leq \frac{\lambda_{\min}(P_x)}{\lambda_{\max}(P_x)} \left( -\frac{1}{2} \lambda_{\max}(P_x) \|E_x\|^2 \right) + \frac{\kappa_x \lambda_{\min}(\Gamma_x^{-1})}{\lambda_{\max}(\Gamma_x^{-1})} \left( -\frac{1}{2} \lambda_{\max}(\Gamma_x^{-1}) \|\tilde{W}_x\|^2 \right) + \beta_x \quad (2.24)$$

donde  $\lambda_{\min}(\cdot)$  y  $\lambda_{\max}(\cdot)$  denotan el eigenvalor mínimo y máximo de la matriz  $(\cdot)$  respectivamente, entonces podemos reescribir (2.24) como

$$\dot{V}_1 \leq -\alpha V_1 + \beta_x \quad (2.25)$$

donde

$$\alpha = \min \left( \frac{\lambda_{\min}(P_x)}{\lambda_{\max}(P_x)}, \frac{\kappa_x \lambda_{\min}(\Gamma_x^{-1})}{\lambda_{\max}(\Gamma_x^{-1})} \right) \quad (2.26)$$

La solución de la ecuación diferencial (2.25) es

$$0 \leq V_1(t) \leq \left( V_1(0) - \frac{\beta_x}{\alpha} \right) e^{-\alpha t} + \frac{\beta_x}{\alpha} \quad (2.27)$$

por lo tanto,

$$\lim_{t \rightarrow \infty} V_1(t) \leq \frac{\beta_X}{\alpha} \quad (2.28)$$

Entonces  $E_X$ ,  $\tilde{W}_X$  están últimamente acotados uniformemente.

Sabemos que  $\frac{1}{2}\|E_X\|^2 \leq \frac{1}{2}E_X^\top P_X E_X \leq V_1(t) \leq \left(V_1(0) - \frac{\beta_X}{\alpha}\right) e^{-\alpha t} + \frac{\beta_X}{\alpha}$ , con lo siguiente para el error de seguimiento.

$$\|E_X\|^2 \leq \left\{ \frac{2}{\lambda_{\min}(P_X)} \left[ \left(V_1(0) - \frac{\beta_X}{\alpha}\right) e^{-\alpha t} + \frac{\beta_X}{\alpha} \right] \right\} \quad (2.29)$$

Por tanto, el error de seguimiento  $E_X$  converge exponencialmente a un conjunto acotado definido como

$$\Omega_{E_X} = \left\{ E_X \mid \|E_X\| \leq \sqrt{\left( \frac{2}{\lambda_{\min}(P_X)} \frac{\beta_X}{\alpha} \right)} \right\} \quad (2.30)$$

Asimismo, el vector de error de estimación  $\tilde{W}_X$  queda acotado por

$$\Omega_{\tilde{W}_X} = \left\{ \tilde{W}_X \mid \|\tilde{W}_X\| \leq \sqrt{\left( \frac{2}{\lambda_{\min}(\Gamma_X^{-1})} \frac{\beta_X}{\alpha} \right)} \right\} \quad (2.31)$$

□

## 2.5. Redes neuronales recurrentes de alto orden

Las redes neuronales recurrentes, a diferencia de las redes neuronales estáticas, es un sistema dinámico que involucra información de realimentación de las salidas de las neuronas. Este tipo de redes neuronales recurrentes han sido ampliamente utilizadas en la identificación y control de sistemas, debido a que como se mencionó en la sección anterior cuentan con una habilidad de aproximación importante por lo que nos permite identificar la dinámica del sistema aún con variaciones en las perturbaciones agrupadas y facilitando el diseño de la señal de control. En las redes neuronales recurrentes de alto orden (RHONN Recurrent High-Order Neural Network, por sus siglas en ingles), la dinámica de cada neurona está representada por una ecuación diferencial que tiene la siguiente forma

$$\dot{\hat{\chi}}_i(t) = -a_i \hat{\chi}_i(t) + b_i \left[ \sum_{k=1}^L W_{ik} \prod_{j \in I_k} \varphi_j^{d_j^{(k)}}(t) \right], \quad i = 1, \dots, n \quad (2.32)$$

donde  $\hat{\chi}_i(t)$  representa el estado de la  $i$ -ésima neurona,  $a_i > 0$  y  $b_i$  son números reales,  $W_{ik}$  son los pesos de la red neuronal,  $\{I_1, I_2, \dots, I_L\}$  es una colección de  $L$  subconjuntos no ordenados de  $\{1, 2, \dots, m + s\}$  donde  $m$  es el número de entradas y  $s$  es el número de neuronas de la RHONN,  $d_j^k$  son enteros mayores o iguales a cero. El vector de entradas de la RHONN se puede describir como

$$\varphi(t) = \begin{bmatrix} \varphi_1(t) \\ \vdots \\ \varphi_s(t) \\ \varphi_{s+1}(t) \\ \varphi_{s+2}(t) \\ \vdots \\ \varphi_{m+s}(t) \end{bmatrix} = \begin{bmatrix} \Psi(x_1(t)) \\ \vdots \\ \Psi(x_s(t)) \\ u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix} \quad (2.33)$$

donde  $\Psi(\cdot)$  es una función tangente hiperbólica dada por

$$\Psi(x) = \frac{2}{1 + e^{-2x}} + 1 \quad (2.34)$$

También, se define un vector que contiene los términos de alto orden entre las entradas y los estados de la RHONN, descrito a continuación

$$\Psi_i = \begin{bmatrix} \Psi_{i_1} \\ \Psi_{i_2} \\ \vdots \\ \Psi_{i_L} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} \varphi_j^{d_j^{(1)}}(t) \\ \prod_{j \in I_2} \varphi_j^{d_j^{(2)}}(t) \\ \vdots \\ \prod_{j \in I_L} \varphi_j^{d_j^{(L)}}(t) \end{bmatrix} \quad (2.35)$$

y un vector de pesos sinápticos de la forma  $\hat{W}_i = b_i [W_{i1} \quad W_{i2} \quad \dots \quad W_{iL}]^\top \in \mathbb{R}^L$ . Con estas definiciones, la dinámica para la  $i$ -ésima neurona de la RHONN se reescribe como

$$\dot{\hat{\chi}}_i(t) = -a_i \hat{\chi}_i(t) + \hat{W}_i^\top \Psi_i \quad (2.36)$$

La estructura de la RHONN se puede observar en la Figura 2.7

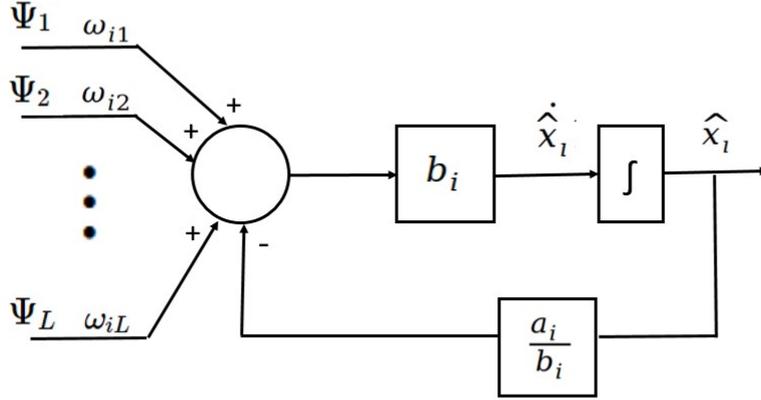


Figura 2.7: Redes Neuronales Recursivas de Alto Orden

### 2.5.1. Estrategia de control para redes neuronales recurrentes

En [72] se muestra que si se cuenta con suficientes conexiones de alto orden que sean procesadas en la RHONN, entonces se puede aproximar con cualquier grado de precisión los sistemas dinámicos no-lineales. Para llevar a cabo la tarea de identificación y control del sistema de segundo orden (2.11), con el cambio de variable  $\chi_1 = \chi$  y  $\chi_2 = \dot{\chi}$ , se propone que el sistema de segundo orden tiene la siguiente estructura

$$\begin{aligned}\dot{\chi}_1 &= -a_1\chi_1 + W_1^{*\top}\Psi_1 + \chi_2 + \Delta_1 \\ \dot{\chi}_2 &= -a_2\chi_2 + W_2^{*\top}\Psi_2 + u + \Delta_2\end{aligned}\tag{2.37}$$

donde  $a_i > 1$ ,  $i = 1, 2$  son constantes de diseño positivas,  $W_1^* \in \mathbb{R}^L$  y  $W_2^* \in \mathbb{R}^L$  son los pesos óptimos que mejor aproximan al sistema (2.11), los cuales están acotados mediante  $\|\tilde{W}_i^*\|^2 \leq W_i^+$ ,  $i = 1, 2$ , donde  $W_1^+$  y  $W_2^+$  son constantes escalares positivas.  $\Psi_1 \in \mathbb{R}^L$  y  $\Psi_2 \in \mathbb{R}^L$  son los vectores que contienen los términos de alto orden de la RHONN, definidos en (2.35),  $\Delta_1$  y  $\Delta_2$  son los errores de estimación para cada variable de estado, los cuales están acotados por  $\Delta_1^2 \leq \Delta_1^+$  y  $\Delta_2^2 \leq \Delta_2^+$ , donde  $\Delta_1^+$  y  $\Delta_2^+$  son constantes escalares positivas.

Para identificar el sistema de segundo orden, se propone una red neuronal recurrente de alto orden que tiene la siguiente forma

$$\begin{aligned}\dot{\hat{\chi}}_1 &= -a_1\hat{\chi}_1 + \hat{W}_1^\top \Psi_1 + \hat{\chi}_2 \\ \dot{\hat{\chi}}_2 &= -a_2\hat{\chi}_2 + \hat{W}_2^\top \Psi_2 + u\end{aligned}\tag{2.38}$$

donde  $\hat{W}_1$  y  $\hat{W}_2$  son los estimados de los pesos óptimos. Considerando la estructura de las ecuaciones diferenciales definidas, se proponen los errores de identificación como

$$\begin{aligned}\tilde{\chi}_1 &= \chi_1 - \hat{\chi}_1 \\ \tilde{\chi}_2 &= \chi_2 - \hat{\chi}_2\end{aligned}\tag{2.39}$$

y el error de seguimiento como

$$\begin{aligned}e_1 &= \hat{\chi}_1 - \chi_{1d} \\ e_2 &= \hat{\chi}_2 - \chi_{2d}\end{aligned}\tag{2.40}$$

donde las señales de referencia están dadas como  $\chi_{1d}$  y  $\chi_{2d} = \dot{\chi}_{1d}$ . Derivando ambos errores, tanto de identificación como de seguimiento, se obtienen las dinámicas descritas a continuación

$$\begin{aligned}\dot{\tilde{\chi}}_1 &= -a_1\chi_1 + W_1^{*\top} \Psi_1 + \chi_2 + \Delta_1 - (-a_1\hat{\chi}_1 + \hat{W}_1^\top \Psi_1 + \hat{\chi}_2) \\ &= -a_1\tilde{\chi}_1 + \tilde{W}_1^\top \Psi_1 + \tilde{\chi}_2 + \Delta_1\end{aligned}\tag{2.41a}$$

$$\begin{aligned}\dot{\tilde{\chi}}_2 &= -a_2\chi_2 + W_2^{*\top} \Psi_2 + u + \Delta_2 - (-a_2\hat{\chi}_2 + \hat{W}_2^\top \Psi_2 + u) \\ &= -a_2\tilde{\chi}_2 + \tilde{W}_2^\top \Psi_2 + \Delta_2\end{aligned}\tag{2.41b}$$

donde  $\tilde{W}_1 = W_1^* - \hat{W}_1$  y  $\tilde{W}_2 = W_2^* - \hat{W}_2$  representan los errores de estimación de los pesos de la red.

La dinámica del error de seguimiento queda descrita como

$$\begin{aligned}
\dot{e}_1 &= -a_1\hat{\chi}_1 + \hat{W}_1^\top \Psi_1 + \hat{\chi}_2 - \dot{\chi}_{1d} \\
&= -a_1(e_1 + \chi_{1d}) + \hat{W}_1^\top \Psi_1 + (e_2 + \chi_{2d}) - \dot{\chi}_{2d} \\
&= -a_1e_1 - a_1\chi_{1d} + \hat{W}_1^\top \Psi_1 + e_2
\end{aligned} \tag{2.42a}$$

$$\begin{aligned}
\dot{e}_2 &= -a_2\hat{\chi}_2 + \hat{W}_2^\top \Psi_2 + u - \dot{\chi}_{2d} \\
&= -a_2(e_2 + \chi_{2d}) + \hat{W}_2^\top \Psi_2 + u - \dot{\chi}_{2d} \\
&= -a_2e_2 - a_2\chi_{2d} + \hat{W}_2^\top \Psi_2 + u - \dot{\chi}_{2d}
\end{aligned} \tag{2.42b}$$

Para lograr estabilizar las dinámicas de los errores de seguimiento  $e_1$  y  $e_2$ , así como los errores de identificación, se procede con una metodología del tipo *backstepping*. Proponiendo una entrada de control virtual  $\alpha_2$  como

$$\alpha_2 = a_1\chi_{1d} - \hat{W}_1^\top \Psi_1 - g_1e_1 \tag{2.43}$$

donde  $g_1 > 0$ , la dinámica del error de seguimiento para  $e_1$  queda reescrito como

$$\dot{e}_1 = -(a_1 + g_1)e_1 \tag{2.44}$$

Definiendo el error de seguimiento entre  $e_2$  y la entrada de control virtual  $\alpha_2$  se tiene

$$\zeta_2 = e_2 - \alpha_2 \tag{2.45}$$

substituyendo (2.43) en (2.45) y despejando para  $e_2$  obtenemos

$$\begin{aligned}
\zeta_2 &= e_2 - a_1\chi_{1d} + \hat{W}_1^\top \Psi_1 + g_1e_1 \\
e_2 &= \zeta_2 + a_1\chi_{1d} - \hat{W}_1^\top \Psi_1 - g_1e_1
\end{aligned} \tag{2.46}$$

ahora sustituimos  $e_2 = \zeta_2 + \alpha_2$  en la dinámica del error de seguimiento  $e_1$ , obteniendo

$$\dot{e}_1 = -(a_1 + g_1)e_1 + \zeta_2 \tag{2.47}$$

La dinámica de la entrada de control virtual  $\alpha_2$  está dada como

$$\dot{\alpha}_2 = a_1 \dot{\chi}_{1_d} - \dot{\hat{W}}_1^\top \Psi_1 - \dot{\hat{\Psi}}_1^\top \hat{W}_1 - g_1 \dot{e}_1 \quad (2.48)$$

a partir de (2.42b), (2.47) y (2.48), la dinámica para el error de la entrada de control virtual  $\zeta_2$  está dado por

$$\begin{aligned} \dot{\zeta}_2 &= \dot{e}_2 - \dot{\alpha}_2 \\ &= -a_2 e_2 - a_2 \chi_{2_d} + \dot{\hat{W}}_2^\top \Psi_2 + u - \dot{\chi}_{2_d} - (a_1 \dot{\chi}_{1_d} - \dot{\hat{W}}_1^\top \Psi_1 - \dot{\hat{\Psi}}_1^\top \hat{W}_1 - g_1 \dot{e}_1) \\ &= -a_2 e_2 - a_2 \chi_{2_d} + \dot{\hat{W}}_2^\top \Psi_2 + u - \dot{\chi}_{2_d} - a_1 \dot{\chi}_{1_d} + \dot{\hat{W}}_1^\top \Psi_1 + \dot{\hat{\Psi}}_1^\top \hat{W}_1 - (a_1 g_1 + g_1^2) e_1 \end{aligned} \quad (2.49)$$

A partir de (2.49), se propone la entrada de control  $u$  de la siguiente forma

$$u = a_2 e_2 + a_2 \chi_{2_d} - \dot{\hat{W}}_2^\top \Psi_2 + \dot{\chi}_{2_d} + a_1 \dot{\chi}_{1_d} - \dot{\hat{W}}_1^\top \Psi_1 - \dot{\hat{\Psi}}_1^\top \hat{W}_1 + (a_1 g_1 + g_1^2) e_1 - g_2 \zeta_2 - e_1 \quad (2.50)$$

con lo cual, la dinámica del error de la entrada de control virtual está dado por

$$\dot{\zeta}_2 = -g_2 \zeta_2 - e_1 \quad (2.51)$$

obteniéndose las dinámicas para el sistema  $e_1$ - $\zeta_2$  de la siguiente forma

$$\begin{aligned} \dot{e}_1 &= -(a_1 + g_1) e_1 + \zeta_2 \\ \dot{\zeta}_2 &= -g_2 \zeta_2 - e_1 \end{aligned} \quad (2.52)$$

Las leyes de actualización de los pesos para las redes neuronales recurrentes de alto orden están dados por

$$\dot{\hat{W}}_i = \Gamma_i (\Psi_i \tilde{\chi}_i - \kappa_i \hat{W}_i), \quad i = 1, 2 \quad (2.53)$$

donde  $0 < \Gamma_i^\top = \Gamma_i = \text{diag}(\Gamma_{i_1}, \Gamma_{i_2}, \dots, \Gamma_{i_L}) \in \mathbb{R}^{L \times L}$  y  $\kappa_i$  son ganancias escalares positivas.

**Teorema 2.2.** *Considere el sistema de segundo orden dado por la ecuación (2.11) representado en la forma (2.37). Considere el error de identificación y error de seguimiento dados (2.39) y (2.40) respectivamente, así como sus dinámicas dadas por (2.41) y (2.42). Definiendo una entrada de control virtual como en (2.43), las dinámicas del error de la entrada de control virtual*

como en (2.49), la entrada de control dada por (2.50) y las leyes de actualización para los pesos de la red neuronal recurrente de alto orden como (2.53). Entonces, todas las señales del sistema son uniformemente últimamente acotadas y convergen a una región acotada centrada en el origen.

*Demostración.* Considere la función candidata de Lyapunov siguiente

$$V_2 \left( \tilde{\chi}_1, \tilde{\chi}_2, \tilde{W}_1, \tilde{W}_2, e_1, \zeta_2 \right) = \frac{1}{2} \sum_{i=1}^2 \tilde{\chi}_i^2 + \frac{1}{2} \sum_{i=1}^2 \tilde{W}_i^\top \Gamma_i^{-1} \tilde{W}_i + \frac{1}{2} e_1^2 + \frac{1}{2} \zeta_2^2 \quad (2.54)$$

Derivando la ecuación (2.54) en el tiempo a través de sus trayectorias obtenemos

$$\dot{V}_2 = \sum_{i=1}^2 \tilde{\chi}_i \dot{\tilde{\chi}}_i + \sum_{i=1}^2 \tilde{W}_i^\top \Gamma_i^{-1} \dot{\tilde{W}}_i + e_1 \dot{e}_1 + \zeta_2 \dot{\zeta}_2 \quad (2.55)$$

Sustituyendo (2.41) y (2.52) obtenemos

$$\begin{aligned} \dot{V}_2 &= -a_1 \tilde{\chi}_1^2 + \tilde{W}_1^\top \Psi_1 \tilde{\chi}_1 + \tilde{\chi}_1 \tilde{\chi}_2 + \tilde{\chi}_1 \Delta_1 - a_2 \tilde{\chi}_2^2 + \tilde{W}_2^\top \Psi_2 \tilde{\chi}_2 + \tilde{\chi}_2 \Delta_2 \\ &\quad + \tilde{W}_1^\top \Gamma_1^{-1} \dot{\tilde{W}}_1 + \tilde{W}_2^\top \Gamma_2^{-1} \dot{\tilde{W}}_2 - (a_1 + g_1) e_1^2 + e_1 \zeta_2 - g_2 \zeta_2^2 - e_1 \zeta_2 \\ &= -a_1 \tilde{\chi}_1^2 - a_2 \tilde{\chi}_2^2 - (a_1 + g_1) e_1^2 - g_2 \zeta_2^2 + \tilde{W}_1^\top \left( \Psi_1 \tilde{\chi}_1 + \kappa_1 \dot{\tilde{W}}_1 \right) \\ &\quad + \tilde{W}_2^\top \left( \Psi_2 \tilde{\chi}_2 + \kappa_2 \dot{\tilde{W}}_2 \right) + \tilde{\chi}_1 \tilde{\chi}_2 + \tilde{\chi}_1 \Delta_1 + \tilde{\chi}_2 \Delta_2 \end{aligned} \quad (2.56)$$

Aplicando la desigualdad de Young para dos escalares  $ab \leq \frac{a^2}{2} + \frac{b^2}{2}$  a los últimos tres términos de (2.56) y sustituyendo las leyes de actualización (2.53), se obtiene

$$\begin{aligned} \dot{V}_2 &\leq - (a_1 - 1) \tilde{\chi}_1^2 - (a_2 - 1) \tilde{\chi}_2^2 - g_1 e_1^2 - g_2 \zeta_2^2 + \kappa_1 \tilde{W}_1^\top \hat{W}_1 + \kappa_2 \tilde{W}_2^\top \hat{W}_2 \\ &\quad + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+ \end{aligned} \quad (2.57)$$

A partir de las definiciones de los errores de los pesos de la RHONN, se obtiene

$$\begin{aligned} \dot{V}_2 &\leq - (a_1 - 1) \tilde{\chi}_1^2 - (a_2 - 1) \tilde{\chi}_2^2 - g_1 e_1^2 - g_2 \zeta_2^2 - \kappa_1 \tilde{W}_1^\top \tilde{W}_1 - \kappa_2 \tilde{W}_2^\top \tilde{W}_2 \\ &\quad + \kappa_1 \tilde{W}_1^\top W_1^* + \kappa_2 \tilde{W}_2^\top W_2^* + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+ \end{aligned} \quad (2.58)$$

aplicando nuevamente la desigualdad de Young obtenemos

$$\begin{aligned}
\dot{V}_2 &\leq -(a_1 - 1)\tilde{\chi}_1^2 - (a_2 - 1)\tilde{\chi}_2^2 - g_1 e_1^2 - g_2 \zeta_2^2 - \kappa_1 \tilde{W}_1^\top \tilde{W}_1 - \kappa_2 \tilde{W}_2^\top \tilde{W}_2 \\
&\quad + \frac{\kappa_1}{2} \tilde{W}_1^\top \tilde{W}_1 + \frac{\kappa_1}{2} W_1^{*\top} W_1^* + \frac{\kappa_2}{2} \tilde{W}_2^\top \tilde{W}_2 + \frac{\kappa_2}{2} W_2^{*\top} W_2^* + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+ \\
&\leq -(a_1 - 1)\tilde{\chi}_1^2 - (a_2 - 1)\tilde{\chi}_2^2 - g_1 e_1^2 - g_2 \zeta_2^2 - \frac{\kappa_1}{2} \tilde{W}_1^\top \tilde{W}_1 - \frac{\kappa_2}{2} \tilde{W}_2^\top \tilde{W}_2 \\
&\quad + \frac{\kappa_1}{2} W_1^+ + \frac{\kappa_2}{2} W_2^+ + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+
\end{aligned} \tag{2.59}$$

Si reescribimos  $\bar{a}_1 = (a_1 - 1)$  y  $\bar{a}_2 = (a_2 - 1)$ . Notar que, a partir de la función candidata de Lyapunov (2.54) y la desigualdad de Rayleigh–Ritz, tenemos

$$\begin{aligned}
\dot{V}_2 &\leq -\bar{a}_1 |\tilde{\chi}_1|^2 - \bar{a}_2 |\tilde{\chi}_2|^2 - \frac{\kappa_1}{2} \lambda_{\min}(\Gamma_1^{-1}) \|\tilde{W}_1\|^2 - \frac{\kappa_2}{2} \lambda_{\min}(\Gamma_2^{-1}) \|\tilde{W}_2\|^2 \\
&\quad - g_1 |e_1|^2 - g_2 |\zeta_2|^2 + \beta_W
\end{aligned} \tag{2.60}$$

donde  $\beta_W = +\frac{\kappa_1}{2} W_1^+ + \frac{\kappa_2}{2} W_2^+ + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+$ . Reorganizando la expresión obtenemos

$$\begin{aligned}
\dot{V}_2 &\leq 2\bar{a}_1 \left( -\frac{1}{2} |\tilde{\chi}_1|^2 \right) + 2\bar{a}_2 \left( -\frac{1}{2} |\tilde{\chi}_2|^2 \right) + g_1 \left( -\frac{1}{2} |e_1|^2 \right) + g_2 \left( -\frac{1}{2} |\zeta_2|^2 \right) \\
&\quad + \frac{\kappa_1 \lambda_{\min}(\Gamma_1^{-1})}{\lambda_{\max}(\Gamma_1^{-1})} \left( -\frac{1}{2} \lambda_{\max}(\Gamma_1^{-1}) \|\tilde{W}_1\|^2 \right) + \frac{\kappa_2 \lambda_{\min}(\Gamma_2^{-1})}{\lambda_{\max}(\Gamma_2^{-1})} \left( -\frac{1}{2} \lambda_{\max}(\Gamma_2^{-1}) \|\tilde{W}_2\|^2 \right) \\
&\quad + \beta_W
\end{aligned} \tag{2.61}$$

entonces podemos reescribir (2.61) como

$$\dot{V}_2 \leq -\mu V_2 + \beta_W \tag{2.62}$$

donde

$$\mu = \min \left( 2\bar{a}_1, 2\bar{a}_2, \frac{\kappa_1 \lambda_{\min}(\Gamma_1^{-1})}{\lambda_{\max}(\Gamma_1^{-1})}, \frac{\kappa_2 \lambda_{\min}(\Gamma_2^{-1})}{\lambda_{\max}(\Gamma_2^{-1})}, g_1, g_2 \right) \tag{2.63}$$

La solución de la ecuación diferencial (2.61) es

$$0 \leq V_2(t) \leq \left( V_2(0) - \frac{\beta_W}{\mu} \right) e^{-\mu t} + \frac{\beta_W}{\mu} \tag{2.64}$$

por lo tanto,

$$\lim_{t \rightarrow \infty} V_2(t) \leq \frac{\beta_W}{\mu} \quad (2.65)$$

Entonces tanto los errores de identificación  $\tilde{\chi}_i$ , el error de seguimiento  $e_1$ , el error de estimación  $\tilde{W}_i$  y la entrada de control virtual  $\zeta_2$  están últimamente acotados uniformemente. Sabemos que  $|e_1|^2 \leq \frac{1}{2}e_1^2 \leq V_2(t) \leq \left(V_2(0) - \frac{\beta_W}{\mu}\right)e^{-\mu t} + \frac{\beta_W}{\mu}$  obtenemos lo siguiente para el error de seguimiento.

$$|e_1|^2 \leq \left\{ \frac{2}{g_1} \left[ \left( V_2(0) - \frac{\beta_W}{\mu} \right) e^{-\mu t} + \frac{\beta_W}{\mu} \right] \right\} \quad (2.66)$$

Por tanto, el error de seguimiento  $e_1$  converge exponencialmente a un conjunto acotado definido como

$$\Omega_{e_1} = \left\{ e_1 \mid |e_1| \leq \sqrt{\left( \frac{2}{g_1} \frac{\beta_W}{\mu} \right)} \right\} \quad (2.67)$$

Asimismo, el vector de error de estimación  $\tilde{W}_i$  queda acotado por

$$\Omega_{\tilde{W}_i} = \left\{ \tilde{W}_i \mid \|\tilde{W}_i\| \leq \sqrt{\left( \frac{2}{\lambda_{\min}(\Gamma_i^{-1})} \frac{\beta_W}{\mu} \right)} \right\} \quad (2.68)$$

□

## 2.6. Estimación de parámetros mejorada

Para la estimación de parámetros desconocidos, se han desarrollado diferentes estrategias, entre las que se encuentran los métodos convencionales de gradiente descendiente, los métodos de mínimos cuadrados (Least Squares LS por sus siglas en ingles) y los métodos de proyección [73],[74],[75] por mencionar algunos. Estas estrategias de estimación de parámetros han sido utilizadas en diferentes técnicas de control adaptable, donde en algunos casos se garantiza la convergencia del error de seguimiento, pero no se garantiza la convergencia del error de estimación. Mejoras en las leyes de adaptación para la estimación de parámetros han sido propuestas, tales como lo son el algoritmo de modificación  $-\sigma$  o modificación  $-e$  [90], [91]. Parte esencial en el desarrollo de esta tesis se encuentra en el algoritmo para la estimación de parámetros, el cual se encuentra basado en la información del error de los parámetros a estimar.

En el caso de estudio de la presente tesis, se utiliza una estrategia de estimación de parámetros reportada primeramente en [76], la cual genera y utiliza la información del error de estimación de los parámetros para mejorar los resultados de seguimiento de trayectoria. En [51] se desarrolla la estrategia de control con tres variantes del mismo para un sistema robótico, Se desarrolla e integra el enfoque del estimador en tiempo finito para una mesa de de trabajo en manufactura [52] y en [62] se utiliza una red neuronal junto con el algoritmo adaptable para controlar un servo-motor. De forma general, para aplicar esta estrategia mejorada, es necesario realizar los siguientes pasos

1. Parametrización
2. Filtrado
3. Matrices auxiliares
4. Ley de actualización

Primeramente, el sistema que contiene un conjunto de parámetros desconocidos a estimar se parametriza de la siguiente forma

$$\dot{\chi} = F(\chi, u) + \Lambda(\chi) W_{\chi} \quad (2.69)$$

donde  $\chi \in \mathbb{R}^2$  es el vector de estado, el vector de funciones conocidas está denotado por  $F(\chi, u) \in \mathbb{R}^2$ , la matriz  $\Lambda(\chi) \in \mathbb{R}^{2 \times s}$  también es conocida como "regresor" y  $W_{\chi} \in \mathbb{R}^s$  contiene los parámetros desconocidos del sistema. Para llevar a cabo la estimación de los parámetros, se asume que el estado del sistema y la entrada de control  $u$  son medibles y acotados.

Para continuar con la estimación de los parámetros desconocidos, se aplica un filtro de primer orden a  $\chi$ ,  $F(\chi, u)$  y  $\Lambda(\chi)$  obteniéndose  $\chi_f$ ,  $F_f$  y  $\Lambda_f$ , respectivamente. Los filtros de primer orden están dados por

$$\begin{aligned}
k\dot{\chi}_f + \chi_f &= \chi, & \chi_f(0) &= 0 \\
k\dot{F}_f + F_f &= F, & F_f(0) &= 0 \\
k\dot{\Lambda}_f + \Lambda_f &= \Lambda, & \Lambda_f(0) &= 0
\end{aligned} \tag{2.70}$$

donde  $k > 0$  es la constante de tiempo del filtro. Entonces, a partir de (2.69) y (2.70) se obtiene

$$\dot{\chi}_f = \frac{\chi - \chi_f}{k} = F_f + \Lambda_f W_\chi \tag{2.71}$$

Posteriormente, se propone una matriz y un vector auxiliar  $P_W \in \mathbb{R}^{s \times s}$  y  $Q_W \in \mathbb{R}^s$ , respectivamente, de la siguiente forma

$$\begin{aligned}
\dot{P}_W &= -lP_W + \Lambda_f^T \Lambda_f, & P_W(0) &= 0 \\
\dot{Q}_W &= -lQ_W + \Lambda_f^T \left[ \frac{\chi - \chi_f}{k} - F_f \right], & Q_W(0) &= 0
\end{aligned} \tag{2.72}$$

donde  $l > 0$  es otro parámetro de diseño. La solución de (2.72) se obtiene como

$$\begin{aligned}
P_W(t) &= \int_0^t e^{-l(t-r)} \Lambda_f^T(r) \Lambda_f(r) dr \\
Q_W(t) &= \int_0^t e^{-l(t-r)} \Lambda_f^T(r) \left[ \frac{\chi(r) - \chi_f(r)}{k} - F_f(r) \right] dr
\end{aligned} \tag{2.73}$$

Con base en las soluciones de  $P_W(t)$  y  $Q_W(t)$ , definimos un vector auxiliar  $N_W$  como

$$N_W = P_W \hat{W} - Q_W \tag{2.74}$$

donde  $\hat{W}_\chi$  es el vector de parámetros estimados. A partir de las ecuaciones de filtrado (2.70) y los vectores y matrices auxiliares (2.72) podemos verificar el hecho que  $Q_W = P_W W_\chi$ . Sustituyendo  $Q_W$  en (2.74), obtenemos

$$\begin{aligned}
N_W &= P_W \hat{W}_\chi - Q_W \\
&= P_W \hat{W}_\chi - P_W W_\chi \\
&= -P_W \tilde{W}_\chi
\end{aligned} \tag{2.75}$$

donde  $\tilde{W}_x = W_x - \hat{W}_x$  representa el error de los parámetros del error. Es decir, el término  $N_W$ , que puede ser calculado a partir de las matrices auxiliares propuestas, contiene de forma implícita la información del error de estimación de los parámetros. Por lo tanto, este término puede ser utilizado en las leyes de actualización propuestas en la literatura (gradiente descendiente, modificación- $e$ , modificación- $\sigma$ ).

En este trabajo de tesis, las leyes de actualización de los pesos de las redes neuronales (de base radial y recurrentes de alto orden) serán modificadas para incluir  $N_W$  con las dimensiones adecuadas, quedando de la siguiente forma para (2.17) y (2.53), respectivamente

$$\dot{\hat{W}}_x = \Gamma_x (\Psi_x(\chi, \dot{\chi}) B^\top P_x E_x - \kappa_x N_{W_x}) \quad (2.76)$$

$$\dot{\hat{W}}_i = \Gamma_i (\Psi_i \tilde{\chi}_i - \kappa_i N_{W_i}), \quad i = 1, 2 \quad (2.77)$$

se puede observar que el último término de (2.76) y (2.77) contienen a  $N_{(\cdot)}$  en lugar de los pesos estimados.

### 2.6.1. Excitación Persistente

La excitación persistente es una condición crucial en la teoría de control adaptable, ya que garantiza la convergencia de los algoritmos de adaptación y la estabilidad del sistema. En general, un sistema posee excitación persistente (PE, por sus siglas en inglés) si las entradas del sistema cambian lo suficientemente rápido y en una amplia gama de frecuencias para que todas las características del sistema puedan ser observadas y estimadas por el algoritmo de adaptación.

Para el caso del algoritmo de estimación de parámetros presentado en la sección 2.6, una condición necesaria y suficiente para garantizar que el regresor  $\Lambda(\chi)$  es PE es establecida en el siguiente lema

**Lema 1.** *Si el vector regresor  $\Lambda_f(\chi)$  definido en (2.70) es PE, La matriz  $P_W$  en (2.73) es definida positiva y por lo tanto el valor propio mínimo de  $P_W$  cumple con  $\lambda_{\min}(P_W) > \sigma > 0$  para una constante positiva  $\sigma$ . Además, si  $P_W$  es definida positiva entonces el regresor  $\Lambda$  es PE.*

*Demostración.* La condición PE convencional del vector regresor  $\Lambda(\chi)$  es suficiente para garantizar que  $P_w$  sea definida positiva. Por lo tanto, de manera similar a otra literatura de iden-

tificación de sistemas y estimación de parámetros [49],  $P_w$  puede mantenerse definido como positivo al proponer una señal de entrada  $u$  adecuada y/o una señal *dither* PE en el sistema dinámico. Sin embargo, la validación en línea directa de la condición PE es difícil, en particular para un sistema no lineal. Por lo tanto, se demostrará que la prueba de la definición positiva de  $P_w$  permite verificar numéricamente las condiciones de convergencia de los algoritmos adaptables sugeridos en [50]. Aquí en [51] tomamos la demostración de la segunda parte. Cuando  $P_W$  es definida positiva y la solución es  $P_W = \int_0^t e^{-l(t-\tau)} \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau \geq \sigma I$  se cumple si consideramos

$$\begin{aligned} \int_0^t e^{-l(t-\tau)} \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau &= \int_0^{t-T} e^{-l(t-\tau)} \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau + \int_{t-T}^t e^{-l(t-\tau)} \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau \\ &\leq \frac{e^{-lT}}{l} \|\Lambda_f\|_\infty^2 I + \int_{t-T}^t \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau \end{aligned} \quad (2.78)$$

la última parte de esta ecuación se obtiene de las desigualdades  $\int_0^{t-T} e^{-l(t-\tau)} d\tau \leq \frac{e^{-lT}}{l}$  y  $0 < e^{-l(t-T)} \leq 1$  para  $\tau \in [t-T, t]$ . Por lo tanto podemos verificar que

$$\int_{t-T}^t \Lambda_f^\top(\tau) \Lambda_f(\tau) d\tau \geq \sigma^* I, \text{ para } t \geq T \quad (2.79)$$

donde  $\sigma^* = \sigma - \frac{e^{-lT}}{l} \|\Lambda_f\|_\infty^2 > 0$  entonces es positiva cuando establecemos  $l$  y  $T$  lo suficientemente grande. La ecuación (2.79) indica que  $\Lambda_f$  es PE. Entonces debido a que el filtro de primer orden  $\frac{1}{ks+1}$  en (2.70) es estable, entonces el regresor  $\Lambda(\chi)$  es PE.  $\square$

En resumen el Lema 1 establece una condición para verificar en línea que el regresor sea PE, por lo que, la red neuronal es una arquitectura adecuada para implementar sistemas adaptativos de control inteligente, y la condición PE se puede verificar en línea mediante el cálculo del valor propio mínimo de la matriz  $P_W$ .



---

# Estrategias de control inteligente para un UAV

---

Antes de presentar el desarrollo de las estrategias de consenso en tareas cooperativas para un sistema multiagente de vehículos multirrotores, en este capítulo se presenta la estrategia de control para un solo vehículo UAV desarrollando las estrategias control inteligente con redes neuronales estáticas de base radial y las redes neuronales recurrentes, en ambos casos utilizando el estimador de parámetros basado en el error de estimación. En la sección 3.1 definimos la estructura del subsistema a controlar para el vehículo multirrotor dentro del ambiente de simulación. A partir de la Sección 3.2 la estrategia de control basado en las redes neuronales estáticas se encuentra dividido en dos subsecciones, la primera utilizando el algoritmo de modificación- $\sigma$  y en la segunda aplicando el estimador mejorado. En la parte final de este capítulo desarrollamos en la Sección 3.3 la estrategia de control inteligente para las redes dinámicas, dividiendo en subsecciones para la estimación tradicional y por último la estimación mejorada.

### 3.1. Estructura de control para un UAV en *Gazebo*

Para presentar la estrategia de control utilizada para un sólo vehículo, primeramente se re-toma el modelo matemático para un vehículo multirrotor en el ambiente de *Gazebo*, el cual está dado en la ecuación (2.7) y se reescribe a continuación

$$\ddot{x} = \frac{u_1}{m} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \quad (3.1a)$$

$$\ddot{y} = \frac{u_1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \quad (3.1b)$$

$$\ddot{z} = -a_1 \dot{z} - a_2 z + a_3 u_z \quad (3.1c)$$

$$\ddot{\theta} = -b_1 \dot{\theta} - b_2 \theta + b_3 u_\theta \quad (3.1d)$$

$$\ddot{\phi} = -c_1 \dot{\phi} - c_2 \phi + c_3 u_\phi \quad (3.1e)$$

$$\ddot{\psi} = -d_1 \dot{\psi} - d_2 \psi + d_3 u_\psi \quad (3.1f)$$

En la ecuación (3.1) se puede observar la presencia de  $u_1$ , la cual en el sistema de *Gazebo* ya no representa una entrada de control, por lo tanto es necesario expresar  $u_1$  en términos de  $u_z$ , para lo cual se procede de la siguiente forma a partir del modelado de Euler-Lagrange dada por (2.5c) con la dinámica de altura identificada en *Gazebo* dada por (3.1c), se tiene

$$\frac{u_1}{m} (\cos \theta \cos \phi) - g = -a_1 \dot{z} - a_2 z + a_3 u_z \quad (3.2)$$

a partir de lo cual obtenemos para la señal  $u_1$  lo siguiente

$$u_1 = \frac{m (-a_1 \dot{z} - a_2 z + a_3 u_z + g)}{\cos \theta \cos \phi} \quad (3.3)$$

Sustituyendo la señal  $u_1$  dada por (3.3) en (3.1a) y (3.1b), obtenemos las dinámicas de traslación

del vehículo en el ambiente de *Gazebo*, las cuales están dadas como

$$\ddot{x} = f(z, u_z) \left( \cos \psi \tan \theta + \frac{\sin \psi}{\cos \theta} \tan \phi \right) + d_x \quad (3.4a)$$

$$\ddot{y} = f(z, u_z) \left( \sin \psi \tan \theta - \frac{\cos \psi}{\cos \theta} \tan \phi \right) + d_y \quad (3.4b)$$

$$\ddot{z} = -a_1 \dot{z} - a_2 z + a_3 u_z + d_z \quad (3.4c)$$

donde  $f(z, u_z) = (-a_1 \dot{z} - a_2 z + a_3 u_z + g)$ , y los términos  $d_x$ ,  $d_y$  y  $d_z$  se agregan para considerar las incertidumbres agrupadas, los cuales contienen las dinámicas no modeladas y perturbaciones externas. De las ecuaciones de traslación (3.4) se realizan las siguientes observaciones

- Las dinámicas de traslación en los tres ejes se ven afectadas por la entrada de control  $u_z$ , la cual es la entrada de control para el UAV en el ambiente de *Gazebo*. La señal de entrada  $u_1$  obtenida por el modelado de Euler-Lagrange ya no aparece de forma explícita en las ecuaciones (3.4).
- Subactuado, dado que el número de señales de entrada es menor al número de señales de salida de este sistema.
- Las dinámicas de orientación ya no son incluidas, debido a que en el ambiente de *Gazebo* ya cuentan con un controlador interno para cada una de estas dinámicas

Para desarrollar la estrategia de control para el sistema de traslación (3.4), se definen las entradas de control virtual  $u_x$  y  $u_y$  de la siguiente forma

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = (-a_1 \dot{z} - a_2 z + a_3 u_z + g) \begin{bmatrix} \tan \theta & \frac{\tan \phi}{\cos \theta} \\ -\frac{\tan \phi}{\cos \theta} & \tan \theta \end{bmatrix} \begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix} \quad (3.5)$$

Con las definiciones de  $u_x$  y  $u_y$ , la dinámica de traslación para el vehículo cuadrirrotor viene dada por

$$\ddot{x} = u_x + d_x \quad (3.6a)$$

$$\ddot{y} = u_y + d_y \quad (3.6b)$$

$$\ddot{z} = -a_1 \dot{z} - a_2 z + a_3 u_z + d_z \quad (3.6c)$$

Para las dinámicas de traslación (3.6) se realizan las siguientes observaciones

- El sistema (3.6) se puede abordar como un sistema actuado debido a la presencia de las entradas de control virtual  $u_x$  y  $u_y$  y la entrada de control  $u_z$  para la dinámica de altura.
- El sistema (3.6) presenta una estructura lineal debido a las señales de control virtual. Sin embargo, estas señales de control virtual contienen una estructura no-lineal, y en esencia el sistema (3.6) es no-lineal.
- Para las entradas de control virtual  $u_x$  y  $u_y$  así como para  $u_z$ , se puede proponer cualquier estrategia de control.
- Dado que  $u_x$  y  $u_y$  son virtuales, es necesario encontrar la relación existente entre los ángulo de roll y pitch y dichas señales de control virtual.

A partir de las señales de control virtuales dadas en (3.5), se obtienen los ángulos de referencia deseados  $\phi_d$  y  $\theta_d$  que permiten al sistema seguir las posiciones deseadas  $x_d$  y  $y_d$ , los cuales están dados por

$$\phi_d = \arctan \left[ \frac{(u_x \cos \psi + u_y \sin \psi)}{f(z, u_z)} \right] \quad (3.7a)$$

$$\theta_d = \arctan \left[ -\frac{(u_x \sin \psi + u_y \cos \psi) \cos \phi_d}{f(z, u_z)} \right] \quad (3.7b)$$

Cabe destacar a partir de la ecuación (3.7) que los ángulos deseados están en función de las entradas las entradas de control  $u_x$ ,  $u_y$  y  $u_z$  se calcularán a través de la estrategia de control presentadas en los siguientes secciones. Los controladores propuestos se sintonizarán de tal manera que los valores de  $\phi_d$  y  $\theta_d$  se mantengan dentro de la región de identificación para los modelos lineales de roll y pitch presentados en las Figuras 2.2 y 2.3. Finalmente, los ángulos deseados  $\phi_d$  y  $\theta_d$  serán las entradas para el controlador interno de del vehículo aéreo en el entorno de ROS y están delimitados de la siguiente manera: ángulo de alabeo  $\theta_d$  por  $(-\pi/2 < \theta_d < \pi/2)$ , ángulo de cabeceo  $\phi_d$  por  $(-\pi/2 < \phi_d < \pi/2)$  y el ángulo de guiñada  $\psi_d$  por  $(-\pi < \psi_d < \pi)$ .

## 3.2. Control basado en redes neuronales de base radial

Ahora, dividimos en dos los enfoques utilizados para las estrategias de control con una red neuronal de base radial para un vehículo UAV donde en primer lugar la ley de actualización de pesos se obtiene mediante el método clásico con modificación  $-\sigma$  y después se desarrolla la misma estrategia introduciendo el algoritmo mejorado de aprendizaje basado en el error de estimación.

### 3.2.1. Estrategia de control con estimación tradicional

Analizamos el subsistema de altitud dado en la ecuación (3.6c) para obtener la señal de control de entrada  $u_z$  seguido de la ley de actualización con la técnica de estimación de parámetros clásica que se encuentra basada en la modificación  $-\sigma$ . Por otro lado, para el plano horizontal, dado por las ecuaciones (3.6a) y (3.6b) obtenemos las señales de control virtual  $u_x$  y  $u_y$  y la ley de actualización para dichas dinámicas, las cuales repercuten en el cálculo de los ángulos deseados para el subsistema de orientación del vehículo.

#### Control de altura para un UAV

La dinámica de altura en un vehículo multirrotores resulta en un sistema más complejo ya que como podemos notar este afecta al resto de las dinámicas de traslación además de contener términos que se ven afectados por la fuerza de gravedad. Aproximando la perturbación agrupada  $d_z$  como  $d_z = W_z^* \Psi(z) + \epsilon_z$  como se definió en el Capítulo 2 en la Ecuación (2.10), por lo que a partir de la ecuación (3.6c) podemos reescribirla de la siguiente forma:

$$\ddot{z} = -a_1 \dot{z} - a_2 z + a_3 u_z + W_z^{*\top} \Psi(z) + \epsilon_z \quad (3.8)$$

procedemos a definir el error de seguimiento como  $e_z = z - z_d$ , donde la dinámica de este error puede ser descrito de la siguiente forma:

$$\ddot{e}_z = \ddot{z} - \ddot{z}_d = -a_1 \dot{z} - a_2 z + a_3 u_z + W_z^{*\top} \Psi(z) + \epsilon_z - \ddot{z}_d \quad (3.9)$$

Con base en esta dinámica, definimos la señal de control  $u_z$  como

$$u_z = \frac{-Kp_z e_z - Kd_z \dot{e}_z - \hat{W}_z^\top \Psi_z + a_1 \dot{z} + a_2 z + \ddot{z}_d}{a_3} \quad (3.10)$$

Sustituyendo la señal de control (3.10) dentro de (3.9) obtenemos

$$\ddot{e}_z = -Kp_z e_z - Kd_z \dot{e}_z + \tilde{W}_z^\top \Psi(z) + \epsilon_z \quad (3.11)$$

donde  $\tilde{W} = W^* - \hat{W}$  representa el error de estimación entre los pesos sinápticos ideales y los pesos estimados, las ganancias  $Kp_z$  y  $Kd_z$  son positivas y si representamos la dinámica del error de seguimiento en espacio de estados  $E_z = [e_z \quad \dot{e}_z]^\top$  obtenemos:

$$\dot{E}_z = AE_z + B(\tilde{W}_z^\top \Psi(z) + \epsilon_z) \quad (3.12)$$

donde las matrices  $A$  y  $B$  tienen la misma estructura que en la Ecuación (2.16), entonces definimos la ley de actualización con la técnica clásica de adaptación quedando definida como

$$\dot{\hat{W}}_z = \Gamma(\Psi_z B^\top P E_z - \kappa_z \hat{W}_z) \quad (3.13)$$

donde  $\Gamma_z > 0$  y  $\kappa_z > 0$  son ganancias.

A partir del Teorema 2.1, podemos concluir que tanto el error de seguimiento en la dinámica de  $z$  del vehículo UAV como el error de estimación con la técnica clásica de adaptación convergen a una pequeña zona compacta, la cual depende en gran medida de la buena aproximación de la red neuronal de base radial.

### Control en el plano horizontal $X - Y$

Para analizar el controlador con redes neuronales de base radial para un vehículo en las dinámicas de  $X$  y  $Y$  podemos observar en la ecuación (3.6) que dichas dinámicas son más sencillas para desarrollar y podemos escribirlas como:

$$\ddot{x} = u_x + W_x^* \Psi(x) + \epsilon_x \quad (3.14)$$

y

$$\ddot{y} = u_y + W_y^* \Psi(y) + \epsilon_y \quad (3.15)$$

Por lo tanto, utilizando el mismo método de la sección anterior definimos el error de seguimiento para la dinámica en el eje  $X$  y  $Y$  como  $e_x = x - x_d$  y  $e_y = y - y_d$  respectivamente y definimos nuestra señal de control para el movimiento en el plano horizontal  $X - Y$  como

$$u_x = -Kp_x e_x - Kd_x \dot{e}_x + \hat{W}_x^\top \Psi(x) \quad (3.16)$$

y

$$u_y = -Kp_y e_y - Kd_y \dot{e}_y + \hat{W}_y^\top \Psi(y) \quad (3.17)$$

donde las ganancias de retroalimentación  $Kp_i$  y  $Kd_i$  con  $(i = x, y)$  son positivas. Definimos las siguientes leyes de actualización

$$\dot{\hat{W}}_x = \Gamma(\Psi_x B^\top P E_x - \kappa_x \hat{W}_x) \quad (3.18)$$

y

$$\dot{\hat{W}}_y = \Gamma(\Psi_y B^\top P E_y - \kappa_y \hat{W}_y) \quad (3.19)$$

de la misma forma que para la ley de actualización en la dinámica de altitud las ganancias  $\kappa_{x,y} > 0$  y  $\Gamma_{x,y} > 0$ , de la misma forma que con la dinámica de altura, a partir del Teorema 2.1 podemos asegurar la convergencia tanto del error de seguimiento con el error de estimación a una región pequeña, que dependerá en su mayoría del error de aproximación  $\epsilon$ .

### 3.2.2. Estrategia de control con aprendizaje mejorado

En este apartado desarrollamos la estrategia de control para el subsistema de traslación para un vehículo UAV con el objetivo de desarrollar el procedimiento de adaptación para los pesos de la red RBF, inspirados en el enfoque presentado en [76], [51]. Como se presentó en la la Sección 2.6, se parametriza el sistema, seguido de un conjunto de filtros de primer orden y la introducción de una serie de matrices auxiliares. Vale la pena mencionar que con este enfoque, podemos utilizar información sobre el error de estimación de los pesos de la red neuronal, en contraste con los mecanismos de adaptación clásicos. Para analizar esta estrategia, comenzaremos con la dinámica en la altura del vehículo y después mostraremos para la dinámica en el plano horizontal.

## Control de altura para un UAV con estimador robusto

Primeramente, la dinámica de altura dada por la ecuación (3.8), donde la perturbación agrupada se aproxima mediante una RBF, se reescribe de forma similar a (2.69), quedando de la siguiente forma

$$\begin{aligned}\ddot{z} &= \underbrace{-a_1\dot{z} - a_2z + a_3u_z}_{F(z, u_z)} + \overbrace{\Psi(z)^\top}^{\Lambda_z} \underbrace{W_z^*}_{W_z} + \epsilon_z \\ &= F(z, u_z) + \Lambda_z W_z + \epsilon_z\end{aligned}\quad (3.20)$$

Se aplica un filtro de primer orden a  $z$ ,  $F(z, u_z)$  y  $\Lambda_z$  obteniéndose  $z_f$ ,  $F_{zf}$  y  $\Lambda_{zf}$ , respectivamente. Los filtros de primer orden están dados por

$$\begin{aligned}k\dot{z}_f + z_f &= z, & z_f(0) &= 0 \\ k\dot{F}_{zf} + F_{zf} &= F_z, & F_{zf}(0) &= 0 \\ k\dot{\Lambda}_{zf} + \Lambda_{zf} &= \Lambda_z, & \Lambda_{zf}(0) &= 0 \\ k\dot{\epsilon}_{zf} + \epsilon_{zf} &= \epsilon_z, & \epsilon_{zf}(0) &= 0\end{aligned}\quad (3.21)$$

donde  $k > 0$  es la constante de tiempo del filtro. Se obtiene

$$\dot{z}_f = \frac{z - z_f}{k} = F_f + \Lambda_f W_z + \epsilon_{zf}\quad (3.22)$$

Se propone una matriz y un vector auxiliar  $P_{W_z} \in \mathbb{R}^{s \times s}$  y  $Q_{W_z} \in \mathbb{R}^s$ , respectivamente, de la siguiente forma

$$\begin{aligned}\dot{P}_{W_z} &= -lP_{W_z} + \Lambda_{zf}^T \Lambda_{zf}, & P_{W_z}(0) &= 0 \\ \dot{Q}_{W_z} &= -lQ_{W_z} + \Lambda_{zf}^T \left[ \frac{z - z_f}{k} - F_{zf} - \epsilon_{zf} \right], & Q_{W_z}(0) &= 0\end{aligned}\quad (3.23)$$

Definimos un vector auxiliar  $N_{W_z}$  como

$$N_{W_z} = P_{W_z} \hat{W}_z - Q_{W_z}\quad (3.24)$$

donde  $\hat{W}_z$  es el vector de parámetros estimados. A partir de las ecuaciones de filtrado (3.21) y los vectores y matrices auxiliares (3.23) podemos verificar el hecho que  $Q_{W_z} = P_{W_z}W_z - \varrho_z$ . Sustituyendo  $Q_{W_z}$  en (3.24), obtenemos

$$\begin{aligned} N_{W_z} &= P_{W_z}\hat{W}_z z_z - Q_{W_z} \\ &= P_{W_z}\hat{W}_z z_z - (P_{W_z}W_z - \varrho_z) \\ &= -P_{W_z}\tilde{W}_z + \varrho_z \end{aligned} \quad (3.25)$$

Ahora, podemos definir a la señal de control  $u_z$  como

$$u_z = \frac{-Kp_z e_z - Kd_z \dot{e}_z - \hat{W}_z \psi_z + a_1 \dot{z} + a_2 z + \ddot{z}_d}{a_3} \quad (3.26)$$

donde la ley de actualización quedaría definida como

$$\dot{\hat{W}}_z = \Gamma(\psi_z B^\top P E_z - \kappa_z N_{W_z}) \quad (3.27)$$

En este sentido si sustituimos la señal de control  $u_z$  dentro de su respectiva dinámica del error de seguimiento  $e_z = z - z_d$  y reescribiendo estas dinámicas en espacios de estado.

$$\dot{E}_z = A E_z + B(\tilde{W}_z^\top \Psi(z) + \epsilon_z) \quad (3.28)$$

donde  $\tilde{W} = W^* - \hat{W}$  representa el error de estimación entre los pesos sinápticos ideales y los pesos estimados y sus respectivas matrices  $A$  y  $B$  mantienen la estructura, por lo que si consideramos el siguiente Teorema:

**Teorema 3.1.** *Considere el sistema de segundo orden parametrizado dado por la ecuación (3.20) sujeto a las perturbaciones agrupadas  $d_z$ . Si la ley de control está dada por (3.26) y la ley de actualización de los pesos dada por (3.27), entonces el error de seguimiento  $e_z$  converge a una región acotada centrada en el origen  $\Omega_{E_z}$  y otra región acotada  $\Omega_{\tilde{W}_z}$  donde converge el error de estimación  $\tilde{W}_z$ .*

*Demostración.* Considere la siguiente función candidata

$$V_z(E_z, \tilde{W}_z) = \frac{1}{2} E_z^\top P_z E_z + \frac{1}{2} \tilde{W}_z^\top \Gamma_z^{-1} \tilde{W}_z \quad (3.29)$$

Derivando  $V_z$  a través de todas sus trayectorias obtenemos

$$\begin{aligned}
\dot{V}_z &= E_z^T P_z \dot{E}_z + \tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z \\
&= E_z^T P_z \left( A E_z + B \left( \tilde{W}_z^T \Psi_z(z, \dot{z}) + \epsilon_z \right) \right) + \tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z \\
&= E_z^T P_z A E_z + E_z^T P_z B \tilde{W}_z^T \Psi_z(z, \dot{z}) + E_z^T P_z B \epsilon_z + \tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z \\
&= \frac{1}{2} E_z^T (A^T P_z + P_z A) E_z - \tilde{W}_z^T (\Psi_z(z, \dot{z}) E_z^T P_z B - \kappa_z N_{W_z}) + E_z^T P_z B \epsilon_z
\end{aligned} \tag{3.30}$$

recordando (3.25) y aplicando la desigualdad de Young al término  $E_z^T P_z B \epsilon_z$  se obtiene

$$\begin{aligned}
\dot{V}_z &\leq \frac{1}{2} E_z^T (A^T P_z + P_z A + P_z B B^T P_z) E_z + \kappa_z \tilde{W}_z^T N_{W_z} + \frac{1}{2} \epsilon_z^2 \\
&\leq \frac{1}{2} E_z^T (A^T P_z + P_z A + P_z B B^T P_z) E_z - \kappa_z \tilde{W}_z^T (P_{W_z} \tilde{W}_z + \varrho_z) + \frac{1}{2} \epsilon_z^2
\end{aligned} \tag{3.31}$$

$$\begin{aligned}
\dot{V}_z &\leq \frac{1}{2} E_z^T (A^T P_z + P_z A + P_z B B^T P_z + Q_z) E_z - \frac{1}{2} E_z^T Q_z E_z \\
&\quad - \kappa_z \tilde{W}_z^T (P_{W_z} \tilde{W}_z + \varrho_z) + \frac{1}{2} \epsilon_z^2
\end{aligned} \tag{3.32}$$

donde  $0 < Q_z = Q_z^T \in \mathbb{R}^{2 \times 2}$ . Las condiciones para que exista la solución de la ecuación algebraica de Riccati  $A^T P_z + P_z A + P_z B B^T P_z + Q_z = 0$ . Con lo cual se obtiene lo siguiente

$$\dot{V}_z \leq -\frac{1}{2} E_z^T Q_z E_z - \kappa_z \tilde{W}_z^T P_{W_z} \tilde{W}_z + \beta_z \tag{3.33}$$

donde  $\beta_z = \frac{1}{2} \epsilon_z^2$ . Notar que, a partir de la función candidata de Lyapunov (3.29) y la desigualdad de Rayleigh–Ritz, tenemos

$$\dot{V}_z \leq -\frac{1}{2} \lambda_{\min}(P_z) \|E_z\|^2 - \kappa_z \lambda_{\min}(\Gamma_z^{-1}) \lambda_{\min}(P_{W_z}) \|\tilde{W}_z\|^2 + \beta_z \tag{3.34}$$

Reorganizando la expresión obetnemos

$$\dot{V}_z \leq \frac{\lambda_{\min}(P_z)}{\lambda_{\max}(P_z)} \left( -\frac{1}{2} \lambda_{\max}(P_z) \|E_z\|^2 \right) + \frac{\kappa_z \lambda_{\min}(\Gamma_z^{-1}, P_{W_z})}{\lambda_{\max}(\Gamma_z^{-1}, P_{W_z})} \left( -\lambda_{\max}(\Gamma_z^{-1}, P_{W_z}) \|\tilde{W}_z\|^2 \right) + \beta_z \tag{3.35}$$

entonces podemos reescribir (3.35) como

$$\dot{V}_z \leq -\alpha_z V_z + \beta_z \tag{3.36}$$

donde

$$\alpha_z = \min \left( \frac{\lambda_{\min}(P_z)}{\lambda_{\max}(P_z)}, \frac{\kappa_z \lambda_{\min}(\Gamma_z^{-1}) \lambda_{\min}(P_{W_z})}{\lambda_{\max}(\Gamma_z^{-1}) \lambda_{\max}(P_{W_z})} \right) \quad (3.37)$$

La solución de la ecuación diferencial (3.35) es

$$0 \leq V_z(t) \leq \left( V_z(0) - \frac{\beta_z}{\alpha_z} \right) e^{-\alpha_z t} + \frac{\beta_z}{\alpha_z} \quad (3.38)$$

por lo tanto,

$$\lim_{t \rightarrow \infty} V_z(t) \leq \frac{\beta_z}{\alpha_z} \quad (3.39)$$

Entonces  $E_z, \tilde{W}_z$  están últimamente acotados uniformemente.

Sabemos que  $\frac{1}{2} \lambda_{\min}(P_z) \|E_z\|^2 \leq \frac{1}{2} E_z^\top P_z E_z \leq V_z(t) \leq \left( V_z(0) - \frac{\beta_z}{\alpha_z} \right) e^{-\alpha_z t} + \frac{\beta_z}{\alpha_z}$ , con lo cual obtenemos lo siguiente para el error de seguimiento.

$$\|E_z\|^2 \leq \left\{ \frac{2}{\lambda_{\min}(P_z)} \left[ \left( V_z(0) - \frac{\beta_z}{\alpha_z} \right) e^{-\alpha_z t} + \frac{\beta_z}{\alpha_z} \right] \right\} \quad (3.40)$$

Por tanto, el error de seguimiento  $E_z$  converge exponencialmente a un conjunto acotado definido como

$$\Omega_{E_z} = \left\{ E_z \mid \|E_z\| \leq \sqrt{\left( \frac{2}{\lambda_{\min}(P_z)} \frac{\beta_z}{\alpha_z} \right)} \right\} \quad (3.41)$$

Asimismo, el vector de error de estimación  $\tilde{W}_z$  queda acotado por

$$\Omega_{\tilde{W}_z} = \left\{ \tilde{W}_z \mid \|\tilde{W}_z\| \leq \sqrt{\left( \frac{\beta_z}{\lambda_{\min}(\Gamma_z^{-1}) \lambda_{\min}(P_{W_z}) \alpha_z} \right)} \right\} \quad (3.42)$$

□

### Control en el plano horizontal $X - Y$ con estimador robusto

Recordemos que los sistemas dinámicos de  $X$  y  $Y$  descritos en la ecuación (3.6) son menos complicados de parametrizar apoyándonos en la ecuación de aproximación de la red neuronal (2.10).

$$\ddot{x} = u_x + \Psi(x)^\top W_x^* + \epsilon_x \quad (3.43)$$

y

$$\ddot{y} = u_y + \Psi(y)^\top W_y^* + \epsilon_y \quad (3.44)$$

Entonces podemos identificar por un lado las funciones conocidas  $F(x, u)$ , la matriz "regresora"  $\Lambda(x, y)$  y el vector de parámetros a ser estimados  $W$  como se describe a continuación

$$\begin{aligned} \ddot{x} &= \underbrace{u_x}_{F(x, u_x)} + \overbrace{\Psi(x)^\top}^{\Lambda_x} \underbrace{W_x^*}_{W_x} + \epsilon_x \\ &= F(x, u_x) + \Lambda_x W_x + \epsilon_x \end{aligned} \quad (3.45)$$

y de la misma forma para la dinámica en en el eje  $Y$

$$\begin{aligned} \ddot{y} &= \underbrace{u_y}_{F(y, u_y)} + \overbrace{\Psi(y)^\top}^{\Lambda_y} \underbrace{W_y^*}_{W_y} + \epsilon_y \\ &= F(y, u_y) + \Lambda_y W_y + \epsilon_y \end{aligned} \quad (3.46)$$

Por lo tanto, empleando el mismo método utilizado para la dinámica de altura, de la sección anterior definimos el error para la dinámica en el eje  $X$  y  $Y$  como  $e_x = x - x_d$  y  $e_y = y - y_d$  respectivamente y definimos nuestra señal de control para el movimiento en el plano horizontal  $X - Y$  como

$$\mu_x = -Kp_x e_x - Kd_x \dot{e}_x - \hat{W}_x \psi_x + \ddot{x}_d \quad (3.47)$$

y

$$\mu_y = -Kp_y e_y - Kd_y \dot{e}_y - \hat{W}_y \psi_y + \ddot{y}_d \quad (3.48)$$

donde la ganancia de retroalimentación  $Kp_i$  y  $Kd_i$  son positivas. Realizamos el filtrado de los términos conocidos tanto de los estados  $x$  y  $y$  como de las funciones de activación de base radial contenidas en  $\psi_x$  y  $\psi_y$  como del termino  $F_x$  y  $F_y$  para construir las matrices y vectores  $P_{W_{x,y}}$  y  $Q_{W_{x,y}}$  para sus respectivas dinámicas y finalmente formar los vectores auxiliares  $N_{W_x}$  y  $N_{W_y}$  con la información del error en los parámetros para cada eje y mediante las siguientes leyes de actualización

$$\dot{W}_x = \Gamma(\psi_x B^\top P E_x - \kappa_x N_{W_x}) \quad (3.49)$$

y

$$\dot{W}_y = \Gamma(\psi_y B^\top P E_y - \kappa_y N_{W_y}) \quad (3.50)$$

En este sentido si sustituimos las señales de control  $u_x$  y  $u_y$  dentro de sus respectivos errores de seguimiento y reescribiendo estas dinámicas en espacios de estado.

$$\dot{E}_x = A E_x + B(\tilde{W}_x^\top \Psi(x) + \epsilon_x) \quad (3.51)$$

y

$$\dot{E}_y = A E_y + B(\tilde{W}_y^\top \Psi(y) + \epsilon_y) \quad (3.52)$$

### 3.3. Control basado en redes neuronales recurrentes

En esta sección, definimos la estrategia de control basada en el desarrollo matemático de una red neuronal recurrente para un sistema de segundo orden ahora considerando el subsistema de traslación de un vehículo UAV.

#### 3.3.1. Estrategia de control con estimación tradicional

Se propone un RHONN descentralizado entrenado mediante el algoritmo tradicional con ley de actualización de pesos modificación- $\sigma$  para la identificación y control de un multi-rotor UAV. El problema de identificación neuronal para todo el subsistema (3.6) se puede simplificar el sistema dinámico de segundo orden  $z = [z_1 \quad z_2]^\top$  el cual se describe de la siguiente manera:

$$\dot{z} = A_z z + W^{*\top} \Psi(z) + B u_z + \Delta \quad (3.53)$$

donde  $A_z = \begin{bmatrix} -a_{1z} & 1 \\ 0 & -a_{2z} \end{bmatrix}$  corresponden a los coeficientes de la dinámica, también  $W^* \in \mathbf{R}^{s \times 2}$  representa los pesos óptimos,  $\Psi(\xi) \in \mathbf{R}^{s \times 1}$  el vector que contiene las funciones de activación a ser definidas,  $B \in \mathbf{R}^{2 \times 1}$  y  $\Delta \in \mathbf{R}^{2 \times 1}$  describe el vector de error de aproximación.

Ahora definimos la red neuronal recurrente de alto orden y para este subsistema.

$$\dot{\hat{z}} = A_z \hat{z} + \hat{W}^\top \Psi(z) + B u_z \quad (3.54)$$

Aquí los estados y pesos estimados se describen como  $\hat{z}$  y  $\hat{W}$  respectivamente,  $\Psi(z)$  concentra el vector de funciones de activación con respecto a las mediciones locales de posición y velocidad del sistema dinámico, y  $u_z$  representa la entrada de control para cada estado del subsistema. Debe notarse que este esquema de control tiene la forma de un sistema de retroalimentación estricta, entonces, el uso del enfoque *backstepping* resulta adecuado para el diseño del controlador por redes neuronales recurrentes. En este caso, para cada  $i$ -ésimo sistema, el error de identificación entre el identificador neuronal y estado se define como

$$\tilde{z}_1 = z_1 - \hat{z}_1 \quad (3.55)$$

para la posición y

$$\tilde{z}_2 = z_2 - \hat{z}_2 \quad (3.56)$$

para la velocidad. Para actualizar en línea los pesos sinápticos, las leyes de aprendizaje adaptativo están dadas por

$$\dot{\hat{W}}_i = \Gamma_i (\Psi_i(z) \tilde{z}_i^\top P - \kappa \hat{W}_i) \quad (3.57)$$

donde  $i = 1, 2$  corresponde a los pesos estimados  $\hat{W}$  y vector de funciones de activación tanto para la posición con el subíndice 1 como para la velocidad con el subíndice 2.

Considerando la estructura del sistema el error de seguimiento queda definido como

$$e_i = \hat{z}_i - z_{id} \quad (3.58)$$

y retomando la ecuación (2.50) proponemos la entrada de control para cada estado  $z$  como:

$$u_z = a_2 e_2 + a_2 z_{2d} - \tilde{W}_2^\top \Psi_2(z) + \dot{z}_{2d} + a_1 \dot{z}_{1d} - \hat{W}_1^\top \Psi_1(z) - \dot{\Psi}_1(z)^\top \hat{W}_1 + (a_1 g_1 + g_1^2) e_1 - g_2 \zeta_2 \quad (3.59)$$

Como se demostró en el Capítulo 2 con el Teorema 2.2 la ley de control (3.59) y ley de actualización (3.57) el cual, utiliza el método de modificación- $\sigma$  de adaptación, estabiliza uni-

formemente los subsistemas concluyendo en que tanto el error de identificación como el error de seguimiento convergen a una región compacta.

### 3.3.2. Estrategia de control con aprendizaje mejorado

El diseño del controlador inteligente basado en redes recurrentes aplicando un estimador de parámetros permite mejorar el aprendizaje, pero dificulta en cierta medida el planteamiento del sistema ya que para este caso como primer paso debemos parametrizarlo por lo que las dimensiones de los vectores y matrices dentro de la red recurrente se tratan de forma diferente a la mencionada en la sección previa así que el nuevo subsistema (3.6) se puede desarrollar de la siguiente manera:

$$\begin{aligned}\dot{z}_1 &= \underbrace{-a_{1z}z_1 + z_2}_{F_1(z)} + \overbrace{\Psi_1^\top}^{\Lambda_{z1}} \underbrace{W_{z2}^*}_{W_{z2}^*} + \Delta_1 \\ \dot{z}_2 &= \underbrace{-a_{2z}z_2 + u_z}_{F_2(z, u_z)} + \overbrace{\Psi_2^\top}^{\Lambda_{z2}} \underbrace{W_{z1}^*}_{W_{z1}^*} + \Delta_2\end{aligned}\tag{3.60}$$

por lo que el sistema parametrizado queda descrito como:

$$\begin{aligned}\dot{z}_1 &= F_1(z, u_z) + \Lambda_{z1}W_{z1}^* + \Delta_1 \\ \dot{z}_2 &= F_2(z, u_z) + \Lambda_{z2}W_{z2}^* + \Delta_2\end{aligned}\tag{3.61}$$

donde  $\Lambda_z \in \mathbf{R}^{1 \times s}$  es la matriz que contiene las funciones de activación a ser definidas, además  $W_{z1, z2}^* \in \mathbf{R}^{s \times 1}$  representa el vector de los pesos ideales y  $\Delta_{1,2} \in \mathbf{R}^1$  describe el vector de error de aproximación.

Para actualizar en línea los pesos sinápticos basado en la estimación del error definimos unas variables filtradas  $z_f, \Lambda_{z1, z2}, F_{1f, 2f}, \Delta_{1f, 2f}$  y  $\Psi_{1f, 2f}$  a partir de la parametrización.

$$\begin{aligned}
k_r \dot{z}_{1f,2f} + z_{1f,2f} &= z, & z_{1f,2f}(0) &= 0 \\
k_r \dot{F}_{1f,2f} + F_{1f,2f} &= F_{1,2}, & F_{1f,2f}(0) &= 0 \\
k_r \dot{\Lambda}_{1f,2f} + \Lambda_{1f,2f} &= \Lambda_{z1,z2}, & \Lambda_{1f,2f}(0) &= 0 \\
k_r \dot{\Delta}_{1f,2f} + \Delta_{1f,2f} &= \Delta_{1,2}, & \Delta_{1f,2f}(0) &= 0
\end{aligned} \tag{3.62}$$

donde  $k_r > 0$  es un parámetro del filtro. Entonces, podemos obtener desde la ecuación (3.61) y (3.62) que:

$$\dot{z}_{1f,2f} = \frac{z_{1,2} - z_{1f,2f}}{k_r} = F_{1,2}(z, u_z) + \Lambda_{1f,2f} W_{z1,z2}^* + \Delta_{1f,2f} \tag{3.63}$$

Definimos una matriz regresora auxiliar, filtrada e "integrada" llamada  $P_w \in \mathbb{R}^{s \times s}$  y un vector  $Q_w \in \mathbb{R}^s$  como:

$$\begin{aligned}
\dot{P}_{w1,w2} &= -l_r P_{w1,w2} + \Lambda_{1f,2f}^T \Lambda_{1f,2f}, & P_{w1,w2}(0) &= 0 \\
\dot{Q}_{w1,w2} &= -l_r Q_{w1,w2} + \Lambda_{1f,2f}^T \left[ \frac{z_{1,2} - z_{1f,2f}}{k_r} - \Delta_{1f,2f} \right], & Q_{w1,w2}(0) &= 0
\end{aligned} \tag{3.64}$$

donde  $l_r > 0$  es otro parámetro de diseño. La solución de (3.64) se obtiene como:

$$\begin{aligned}
P_{w1,w2}(t) &= \int_0^t e^{-l_r(t-r)} \Lambda_{1f,2f}^T(r) \Lambda_{1f,2f}(r) dr \\
Q_{w1,w2}(t) &= \int_0^t e^{-l_r(t-r)} \Lambda_{1f,2f}^T(r) \left[ \frac{z_{1,2}(r) - z_{1f,2f}(r)}{k_r} - \Delta_{1f,2f}(r) \right] dr
\end{aligned} \tag{3.65}$$

Basados en estas soluciones  $P_{w1,w2}(t)$  y  $Q_{w1,w2}(t)$  definimos un vector auxiliar  $N_{w1,w2}$  de la siguiente forma:

$$\begin{aligned}
N_{w1} &= P_{w1}(t) \hat{W}_1 - Q_{w1}(t) \\
N_{w2} &= P_{w2}(t) \hat{W}_2 - Q_{w2}(t)
\end{aligned} \tag{3.66}$$

Entonces, podemos notar que con la ecuación (3.65) y estableciendo el error de estimación de parámetros  $\tilde{W}_{1,2} = W_{1,2}^* - \hat{W}_{1,2}$ , podemos verificar que  $Q_{w1,w2} = P_{w1,w2} W_{1,2}^*$ , sustituimos

esto dentro del vector auxiliar en la ecuación (3.66) y obtenemos:

$$\begin{aligned}
N_{w1,w2} &= P_{w1,w2}(t)\hat{W}_{1,2} - Q_{w1,w2}(t) \\
&= P_{w1,w2}(t)\hat{W}_{1,2} - P_{w1,w2}(t)W_{1,2}^* \\
&= -P_{w1,w2}(t)\tilde{W}_{1,2}
\end{aligned} \tag{3.67}$$

Debido a que la matriz regresora  $\Lambda_{1,2}$  está acotado, entonces las versiones filtradas  $\Lambda_{1f,2f}$   $\Delta_{1f,2f}$  también se encuentran acotadas.

Podemos utilizar la matriz auxiliar  $N_w$  para integrarla a las leyes de aprendizaje adaptable que están definidas por

$$\begin{aligned}
\dot{\hat{W}}_1 &= \Gamma_1(\Lambda_1(z)\tilde{z}_1^\top P - \kappa_1 N_{w1}) \\
\dot{\hat{W}}_2 &= \Gamma_2(\Lambda_2(z)\tilde{z}_2^\top P - \kappa_2 N_{w2})
\end{aligned} \tag{3.68}$$

donde  $i = 1, 2$  corresponde a los pesos estimados  $\hat{W}$  y vector de funciones de activación tanto para la posición con el subíndice 1 como para la velocidad con el subíndice 2. De la misma forma que en el apartado previo definimos la red neuronal recurrente de alto orden para este sistema.

$$\begin{aligned}
\dot{\hat{z}}_1 &= F_1(\hat{z}) + \Lambda_1 \hat{W}_1 \\
\dot{\hat{z}}_2 &= F_2(\hat{z}, u_z) + \Lambda_2 \hat{W}_2
\end{aligned} \tag{3.69}$$

Aquí los estados y pesos estimados se describen como  $\hat{z}_{1,2}$  y  $\hat{W}_{1,2}$  respectivamente,  $\Lambda_{1,2}$  concentra la matriz de funciones de activación con respecto a las mediciones locales de posición y velocidad del sistema de traslación. y  $u_z$  representa la entrada de control para la red neuronal. Debe recordarse que este esquema de control tiene la forma de un sistema de retroalimentación estricta, entonces, el uso del enfoque *backstepping* resulta adecuado para el diseño del controlador por redes neuronales recurrentes. En este caso, para cada  $i$ -ésimo sistema, el error de identificación entre el identificador neuronal y estado se define como

$$\tilde{z}_1 = z_1 - \hat{z}_1 \tag{3.70}$$

$$\tilde{z}_2 = z_2 - \hat{z}_2 \quad (3.71)$$

tanto para la posición y la velocidad respectivamente, por lo tanto nuestro vector de error de identificación parametrizado queda definido como

$$\begin{aligned} \dot{\tilde{z}}_1 &= F_1(z_1) + \Lambda_1 \tilde{W}_1 + \Delta_1 \\ \dot{\tilde{z}}_2 &= F_2(z_2, u_z) + \Lambda_2 \tilde{W}_2 + \Delta_2 \end{aligned} \quad (3.72)$$

Considerando la estructura del sistema el error de seguimiento para el subsistema de traslación del vehículo definimos

$$\begin{aligned} e_1 &= \hat{z}_1 - z_{1d} \\ e_2 &= \hat{z}_2 - z_{2d} \end{aligned} \quad (3.73)$$

y planteando la entrada de control  $u_z$  como:

$$u_z = a_{z2}e_2 + a_{z2}z_{2d} - \tilde{W}_2^\top \Lambda_2(z) + \dot{z}_{2d} + a_{z1}\dot{z}_{1d} - \hat{W}_1^\top \Lambda_1(z) - \dot{\Lambda}_1(z)^\top \hat{W}_1 + (a_{z1}g_{z1} + g_{z1}^2)e_1 - g_{z2}\zeta_2 \quad (3.74)$$

Podemos enunciar el siguiente Teorema:

**Teorema 3.2.** *Considere el sistema de segundo orden parametrizado dado por la ecuación (3.61) que esta sujeto a dinámicas y perturbaciones desconocidas. Si la ley de control está dada por (3.74) y la ley de actualización de los pesos dada por (3.68), entonces el error de seguimiento  $e_1$  converge a una región acotada  $\Omega_{e_1}$  centrada en el origen y otra región acotada  $\Omega_{\tilde{W}_i}$  centrada en el origen donde converge el error de estimación  $\tilde{W}_i$ .*

*Demostración.* Basada en las dinámicas en el error de identificación  $\tilde{z}_{1,2}$  y de seguimiento  $e_1$  así como en la entrada de control virtual  $\zeta_2$  y el error de estimación  $\tilde{W}_{1,2}$ , consideramos la siguiente función candidata de Lyapunov.

$$V_z \left( \tilde{z}_1, \tilde{z}_2, \tilde{W}_1, \tilde{W}_2, e_1, \zeta_2 \right) = \frac{1}{2} \sum_{i=1}^2 \tilde{z}_i^2 + \frac{1}{2} \sum_{i=1}^2 \tilde{W}_i^\top \Gamma_i^{-1} \tilde{W}_i + \frac{1}{2} e_1^2 + \frac{1}{2} \zeta_2^2 \quad (3.75)$$

donde  $\Gamma_i = \text{diag}(\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in})$  corresponde a una matriz definida positiva y  $P$  a un matriz

simétrica y definida positiva, la cual satisface la ecuación de Lyapunov  $A_z^\top P + PA_z = -Q$ .

Si derivamos la ecuación (3.75) en el tiempo a través de sus trayectorias obtenemos

$$\dot{V}_z = \sum_{i=1}^2 \tilde{z}_i \dot{\tilde{z}}_i + \sum_{i=1}^2 \tilde{W}_i^\top \Gamma_i^{-1} \dot{\tilde{W}}_i + e_1 \dot{e}_1 + \zeta_2 \dot{\zeta}_2 \quad (3.76)$$

Substituimos las dinámicas correspondientes la derivada de la Función de Lyapunov.

$$\begin{aligned} \dot{V}_z &= -a_1 \tilde{z}_1^2 + \tilde{W}_1^\top A_1 \tilde{z}_1 + \tilde{z}_1 \tilde{z}_2 + \tilde{z}_1 \Delta_1 - a_2 \tilde{z}_2^2 + \tilde{W}_2^\top A_2 \tilde{z}_2 + \tilde{z}_2 \Delta_2 \\ &\quad + \tilde{W}_1^\top \Gamma_1^{-1} \dot{\tilde{W}}_1 + \tilde{W}_2^\top \Gamma_2^{-1} \dot{\tilde{W}}_2 - (a_1 + g_1) e_1^2 + e_1 \zeta_2 - g_2 \zeta_2^2 - e_1 \zeta_2 \\ &= -a_1 \tilde{z}_1^2 - a_2 \tilde{z}_2^2 - (a_1 + g_1) e_1^2 - g_2 \zeta_2^2 + \tilde{W}_1^\top (A_1 \tilde{z}_1 + \kappa_1 N_{W1}) \\ &\quad + \tilde{W}_2^\top (A_2 \tilde{z}_2 + \kappa_2 N_{W2}) + \tilde{z}_1 \tilde{z}_2 + \tilde{z}_1 \Delta_1 + \tilde{z}_2 \Delta_2 \end{aligned} \quad (3.77)$$

reordenando términos y recordando que  $N_{w1,w2} = -P_{w1,w2}(t) \tilde{W}_{1,2}$  obtenemos

$$\begin{aligned} \dot{V}_z &\leq - (a_1 - 1) \tilde{z}_1^2 - (a_2 - 1) \tilde{z}_2^2 - g_1 e_1^2 - g_2 \zeta_2^2 \\ &\quad - \kappa_1 \tilde{W}_1^\top P_{w1} \tilde{W}_1 - \kappa_2 \tilde{W}_2^\top P_{w2} \tilde{W}_2 + \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+ \end{aligned} \quad (3.78)$$

Si reescribimos  $\bar{a}_1 = (a_1 - 1)$  y  $\bar{a}_2 = (a_2 - 1)$ . Utilizando la desigualdad de Rayleigh–Ritz, tenemos

$$\dot{V}_z \leq -\bar{a}_1 |\tilde{z}_1|^2 - \bar{a}_2 |\tilde{z}_2|^2 - \kappa_1 \lambda_{\min}(P_{w1}) \|\tilde{W}_1\|^2 - \kappa_2 \lambda_{\min}(P_{w2}) \|\tilde{W}_2\|^2 - g_1 |e_1|^2 - g_2 |\zeta_2|^2 + \beta_W \quad (3.79)$$

donde  $\beta_W = \frac{1}{2} \Delta_1^+ + \frac{1}{2} \Delta_2^+$ . Reorganizando la expresión obtenemos

$$\begin{aligned} \dot{V}_z &\leq 2\bar{a}_1 \left( -\frac{1}{2} |\tilde{z}_1|^2 \right) + 2\bar{a}_2 \left( -\frac{1}{2} |\tilde{z}_2|^2 \right) + g_1 \left( -\frac{1}{2} |e_1|^2 \right) + g_2 \left( -\frac{1}{2} |\zeta_2|^2 \right) \\ &\quad + \frac{\kappa_1 \lambda_{\min}(P_{w1})}{\lambda_{\max}(P_{w1})} \left( -\lambda_{\max}(P_{w1}) \|\tilde{W}_1\|^2 \right) + \frac{\kappa_2 \lambda_{\min}(P_{w2})}{\lambda_{\max}(P_{w2})} \left( -\lambda_{\max}(P_{w2}) \|\tilde{W}_2\|^2 \right) \\ &\quad + \beta_W \end{aligned} \quad (3.80)$$

entonces podemos reescribir (3.80) como

$$\dot{V}_z \leq -\mu V_z + \beta_W \quad (3.81)$$

donde

$$\mu = \min \left( 2\bar{a}_1, 2\bar{a}_2, \frac{\kappa_1 \lambda_{\min}(P_{w1})}{\lambda_{\max}(P_{w1})}, \frac{\kappa_2 \lambda_{\min}(P_{w2})}{\lambda_{\max}(P_{w2})}, g_1, g_2 \right) \quad (3.82)$$

La solución de la ecuación diferencial (3.81) es

$$0 \leq V_z(t) \leq \left( V_z(0) - \frac{\beta_W}{\mu} \right) e^{-\mu t} + \frac{\beta_W}{\mu} \quad (3.83)$$

por lo tanto,

$$\lim_{t \rightarrow \infty} V_z(t) = \frac{\beta_W}{\mu} \quad (3.84)$$

Entonces tanto los errores de identificación  $\tilde{z}_1$  y  $\tilde{z}_2$ , el error de seguimiento  $e_1$ , los errores de estimación  $\tilde{W}_1, \tilde{W}_2$  y la entrada de control virtual  $\zeta_2$  están acotados uniformemente. Sabemos que  $|e_1|^2 \leq \frac{1}{2}e_1^2 \leq V_z(t) \leq \left( V_z(0) - \frac{\beta_W}{\mu} \right) e^{-\mu t} + \frac{\beta_W}{\mu}$  obtenemos lo siguiente para el error de seguimiento.

$$|e_1|^2 \leq \left\{ \frac{2}{g_1} \left[ \left( V_z(0) - \frac{\beta_W}{\mu} \right) e^{-\mu t} + \frac{\beta_W}{\mu} \right] \right\} \quad (3.85)$$

Por tanto, el error de seguimiento  $e_1$  converge exponencialmente a un conjunto acotado definido como

$$\Omega_{e_1} = \left\{ e_1 \mid |e_1| \leq \sqrt{\left( \frac{2}{g_1} \frac{\beta_W}{\mu} \right)} \right\} \quad (3.86)$$

Asimismo, el vector de error de estimación  $\tilde{W}_i$  queda acotado por

$$\Omega_{\tilde{W}_{1,2}} = \left\{ \tilde{W}_{1,2} \mid \|\tilde{W}_{1,2}\| \leq \sqrt{\left( \frac{\beta_W}{\lambda_{\min}(P_{w1,w2})\mu} \right)} \right\} \quad (3.87)$$

□

Las simulaciones de esta ultima demostración nos permite verificar que no requiere de un modelado preciso o con gran detalle para la dinámica en altura o el movimiento horizontal, sin embargo la identificación y control del sistema depende en su mayoría de la cantidad de neuronas que se utiliza para aproximar la función de la dinámica de traslación junto con los términos agrupados de perturbaciones externas o incertidumbres endógenas.

---

## Control cooperativo

---

En este Capítulo se analizará la estrategia cooperativa mediante una ley de consenso inteligente para lidiar con las oscilaciones y perturbaciones desconocidas como son ráfagas de viento u otras fuerzas restrictivas (por ejemplo, el peso de la carga) como se menciona en el Capítulo previo para de esta forma lograr estabilizar la posición a los agentes, algunos ejemplos de esta aplicación podemos enfocarla al desarrollo tareas cooperativas para transportar una carga o realizar la vigilancia o monitoreo de alguna zona, incluso el combate de incendios donde se necesita una coordinación adecuada de los equipos de respuesta para lograr mitigar el conato de fuego. En la literatura especializada existe una cantidad de enfoques y aproximaciones para lograr este objetivo, entre las que destacan las propuestas por Jackson et al [78], Shirani et al [79] y Klausen [80]. por mencionar algunos. En el presente capítulo contaremos con la sección especializada en el diseño de la estrategia de control cooperativo utilizando redes estáticas, seguido del estimador de parámetros robusto para multiagentes, En la sección 4.2 abordaremos el diseño del Control de consenso basado en redes neuronales para la dinámica en altura y el movimiento en el plano horizontal y en la sección 4.3 encontraremos el análisis de estabilidad y convergencia para el sistema multiagente utilizando múltiples vehículos UAVs.

## 4.1. Introducción

Considerando la cantidad de algoritmos y modelos matemáticos tendremos que revisar e identificar las leyes de control necesarias tanto para el agente como para el conjunto de ellos en un sistema multiagente, por lo tanto comenzamos describiendo las dinámicas del agente líder y el agente seguidor, el estimador de pesos para la red neuronal utilizado así como la estrategia de consenso para la dinámica de altura como para las dinámicas en los ejes  $X - Y$ . Establecemos el  $i$ -ésimo agente seguidor como

$$\dot{x}_i = Ax_i + B(u_{ix} + d_{ix}) \quad (4.1)$$

donde  $x_i = [x_{i1} \ x_{i2}]^\top$  representa los estados del vehículo como es la posición y velocidad respectivamente,  $A = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 0 \\ a_3 \end{bmatrix}$  y  $d_{ix}$  es la incertidumbre o perturbación en el sistema dinámico. Además, Podemos definir de la misma forma el agente líder como:

$$\dot{x}_0 = Ax_0 + Br(t) \quad (4.2)$$

donde  $r(t)$  es la referencia a seguir por parte del agente líder y con las dinámicas de los agentes podemos definir el error de seguimiento  $\delta_{ix} = x_i - x_0$

### 4.1.1. Estimador de parámetros para multiagentes

Obtenemos la dinámica del error de seguimiento  $\delta_i$  y lo definimos como:

$$\begin{aligned} \dot{\delta}_i &= \dot{x}_i - \dot{x}_0 \\ \dot{\delta}_i &= Ax_i + B(u_i + W_i^{*\top} \Lambda_i(x_i) + \epsilon_i) - Ax_0 - Br(t) \\ &= A\delta_i + B(u_i + W_i^{*\top} \Lambda_i(x_i) + \epsilon_i - r(t)) \end{aligned} \quad (4.3)$$

Entonces procedemos a parametrizar este sistema de la siguiente forma

$$\begin{aligned}\dot{\delta}_i &= \underbrace{A\delta_i + Bu_i}_{\varphi_i(\delta_i, u_i)} + \overbrace{B\Lambda_i(x_i)^T}^{\Psi_i} W_i^* + \underbrace{B(\epsilon_i - r(t))}_{\varepsilon_i} \\ &= \varphi_i(\delta_i, u_i) + \Psi_i W_i^* + \varepsilon_i\end{aligned}\quad (4.4)$$

donde  $\delta_i \in \mathbb{R}^n$  es el estado del sistema,  $\varphi_i(\delta_i, u_i) \in \mathbb{R}^n$  es un vector función conocido y  $\Psi_i \in \mathbb{R}^{n \times s}$  sera conocida como la matriz regresora. Además, el estado  $\delta_i$  y la entrada de control  $u_i$  son requeridos, pero sus derivadas  $\dot{\delta}_i$  no son utilizadas en nuestro esquema de estimación. Finalmente tanto error de aproximación  $\epsilon_i$  y la entrada acotada  $r(t)$  en el agente líder podemos considerarlos como un termino desconocido para el algoritmo de estimación. Ahora definimos unas variables filtradas  $\delta_{if}$ ,  $\varphi_{if}$  y  $\Psi_{if}$  a partir de la parametrización.

$$\begin{aligned}k\dot{\delta}_{if} + \delta_{if} &= \delta_i, & \delta_{if}(0) &= 0 \\ k\dot{\Psi}_{if} + \Psi_{if} &= \Psi_i, & \Psi_{if}(0) &= 0 \\ k\dot{\varphi}_{if} + \varphi_{if} &= \varphi, & \varphi_{if}(0) &= 0\end{aligned}\quad (4.5)$$

donde  $k > 0$  es un parámetro del filtro. Entonces, podemos obtener desde la ecuación (4.4) y (4.5) que:

$$\dot{\delta}_{if} = \frac{\delta_i - \delta_{if}}{k} = \varphi_{if} + \Psi_{if} W_i^* + \varepsilon_i \quad (4.6)$$

Definimos una matriz regresora auxiliar, filtrada e "integrada" llamada  $P_w \in \mathbb{R}^{s \times s}$  y un vector  $Q_w \in \mathbb{R}^s$  como:

$$\begin{aligned}\dot{P}_w &= -lP_w + \Psi_{if}^T \Psi_{if}, & P_w(0) &= 0 \\ \dot{Q}_w &= -lQ_w + \Psi_{if}^T \left[ \frac{\delta_i - \delta_{if}}{k} - \varphi_{if} \right], & Q_w(0) &= 0\end{aligned}\quad (4.7)$$

donde  $l > 0$  es otro parámetro de diseño. La solución de (4.7) se obtiene como:

$$\begin{aligned}
P_w(t) &= \int_0^t e^{-l(t-r)} \Psi_{if}^T(r) \Psi_{if}(r) dr \\
Q_w(t) &= \int_0^t e^{-l(t-r)} \Psi_{if}^T(r) \left[ \frac{\delta_i(r) - \delta_{if}(r)}{k} - \varphi_{if}(r) \right] dr
\end{aligned} \tag{4.8}$$

Basados en estas soluciones  $P_w(t)$  y  $Q_w(t)$  definimos un vector auxiliar  $H_w$  de la siguiente forma:

$$H_w = P_w(t) \hat{W}_i - Q_w(t) \tag{4.9}$$

Entonces, podemos notar que con la ecuación (4.8) y estableciendo el error de estimación de parámetros  $\tilde{W}_i = W_i - \hat{W}_i$ , podemos verificar que  $Q_w = P_w W_i - \varrho_i$ , donde  $\varrho_i \in \mathbb{R}^s$ , y substituímos esto dentro del vector auxiliar en la ecuación (4.9) y obtenemos:

$$\begin{aligned}
H_w &= P_w(t) \hat{W}_i - Q_w(t) \\
&= P_w(t) \hat{W}_i - P_w(t) W_i^* + \varrho_i \\
&= -P_w(t) \tilde{W}_i + \varrho_i
\end{aligned} \tag{4.10}$$

Debido a que la matriz regresora  $\Psi_i$  y el término  $\varepsilon_i$  están todos acotados, entonces las versiones filtradas  $\tilde{\Psi}_{if} \varepsilon_{if}$  también se encuentran acotadas.

Por otro lado,  $\varrho_i = -\int_0^t e^{-l(t-r)} \tilde{\Psi}_{if}(r) \varepsilon_{if}(r) dr$  está acotado por una constante positiva  $\varepsilon_{Nf} > 0$ .

El desarrollo previo del estimador de parámetros robusto es importante para el diseño y análisis de estabilidad de la ley de consenso cooperativo, así como la obtención de la ley de actualización de los parámetros estimados, por lo que resta de este capítulo diseñamos y comprobamos la estabilidad del algoritmo propuesto en cada una de las dinámicas de posición para los vehículos multirrotores.

## 4.2. Diseño de controlador inteligente cooperativo

La estrategia propuesta se basa en una ley de consenso distribuida con un estimador de parámetros robusto para la parte adaptable de una red neuronal que permite contrarrestar per-

turbaciones o incertidumbres generadas por la carga mientras es transportada además de reducir los efectos del viento en cada una de las dinámicas de posición.

## Control de altura

Se propone iniciar con la dinámica vertical o de altura por lo que establecemos el  $i$ -ésimo agente seguidor como en la ecuación (3.4) pero utilizando los parámetros de la identificación del sistema de la siguiente forma

$$\dot{z}_i = Az_i + B(u_{iz} + d_{iz}) \quad (4.11)$$

donde  $z_i = [z_{i1} \ z_{i2}]^\top$  representa los estados del vehículo como es la posición y velocidad respectivamente,  $A = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 0 \\ a_3 \end{bmatrix}$  y  $d_{iz}$  es la incertidumbre o perturbación en la dinámica de altitud. Además, Podemos definir de la misma forma el agente líder de la ecuación (4.11) como:

$$\dot{z}_0 = Az_0 + Br(t) \quad (4.12)$$

donde  $r(t)$  es la referencia a seguir por parte de la altura de los vehículos y de la misma forma podemos definir el error de seguimiento  $\delta_{iz} = z_i - z_0$ , la cual derivando para obtener su dinámica obtenemos.

$$\begin{aligned} \dot{\delta}_{iz} &= Az_i + B(u_{iz} + W_{iz}^{*\top} \Psi_i(z_i) + \epsilon_i) - Az_0 - Br \\ &= A\delta_{iz} + B(u_{iz} + W_{iz}^{*\top} \Psi_i(z_i) + \epsilon_i - r(t)) \end{aligned} \quad (4.13)$$

Para garantizar la estabilidad del error de seguimiento  $\delta_{z_i}$ , diseñamos la siguiente ley de control distribuido con un término de control adaptativo aumentado

$$u_{z_i} = c_z K_z e_{z_i} - \hat{W}_{z_i}^T \Psi_{z_i}(z_i), \quad (4.14)$$

donde  $\hat{W}_{z_i}$  es una estimación de los pesos desconocidos  $W_{z_i}^*$ , el escalar  $c_z$  es una ganancia de

acoplamiento y  $e_{z_i}$  es un error de vecindad representado como

$$e_{z_i} = \sum_{j \in N} a_{ij}(z_i - z_j) + a_{i0}(z_i - z_0), \quad (4.15)$$

Defina  $K_z \in \mathbb{R}^{1 \times 2}$  como  $K_z = -B_z^T M_z$ , donde  $M_z$  es la solución de la desigualdad de Riccati definida como

$$A_z^T M_z + M_z A_z + N_z - M_z B_z B_z^T M_z \leq 0, \quad (4.16)$$

Observe que  $M_z$  y  $N_z \in \mathbb{R}^{2 \times 2}$  son matrices definidas positivas. Si la ley de control de consenso (4.14) se sustituye en (4.3), llegamos a

$$\begin{aligned} \dot{\delta}_{z_i} &= A_z \delta_{z_i} + B_z (c_z K_z e_{z_i} - \hat{W}_{z_i}^T \Psi_{z_i}(z_i) + W_{z_i}^{*T} \Psi_{z_i}(z_i) + \epsilon_{z_i} - r) \\ &= A_z \delta_{z_i} + B_z (c_z K_z e_{z_i} + \tilde{W}_{z_i}^T \Psi_{z_i}(z_i) + \epsilon_{z_i} - r). \end{aligned} \quad (4.17)$$

Cabe señalar de (4.10) que la información sobre el error de estimación de los pesos  $\tilde{W}_{z_i}$  está contenida en  $H_{z_i}$ . Entonces, la ley de adaptación, para los pesos del  $i$ ésimo agente seguidor en la dinámica de altitud se propone como

$$\dot{\hat{W}}_{z_i} = \Gamma_{z_i} (\Psi_{z_i}(z_i) e_{z_i}^T M_z B_z - k_{z_i} H_{z_i}). \quad (4.18)$$

donde  $\Gamma_{z_i}$  y  $k_{z_i}$  son constantes positivas. Luego, establecemos el teorema para el control cooperativo de consenso para la dinámica de altitud del sistema multiagente quadrotor de la siguiente manera. Considere la dinámica de altitud para el agente seguidor  $i$ th dada en la Ecuación (4.11). Si el regresor  $\Psi_{z_i}$  definido en (4.5) cumple la condición PE, se propone la estrategia de control cooperativo  $u_{z_i}$  como en la Ecuación (4.14), la ganancia  $K_z$  se obtiene al resolver la Ecuación de Riccati (4.23), la ganancia de acoplamiento  $c_z$  cumple la condición (4.38), y los pesos neuronales se actualizan por la ley adaptativa dada por la Ecuación (4.18), donde la matriz auxiliar  $H_{z_i}$  está definida en la Ecuación (4.9). Entonces, el error de seguimiento de altitud  $\delta_{z_i}$  y el error de estimación de los pesos neuronales  $\tilde{W}_{z_i}$  para el  $i$ ésimo agente convergen en una región delimitada definida por (4.46).

## Control Horizontal

En este apartado analizaremos el controlador para las dinámicas en el plano  $X - Y$  donde obtendremos las entradas de control tanto para la dinámica en  $X$  como  $u_x$  y para la dinámica en  $Y$  como  $u_y$  ya que ambas se encuentran fuertemente relacionadas con los ángulos deseados para  $\Psi$  y  $\theta$  como se puede ver en las ecuaciones (3.7), por lo que definiremos

$$\dot{x}_i = Ax_i + B(u_{ix} + d_{ix}) \quad (4.19)$$

y

$$\dot{y}_i = Ay_i + B(u_{iy} + d_{iy}) \quad (4.20)$$

donde de la misma forma que en la altitud  $x_i = [x_{i1} \ x_{i2}]^T$   $y_i = [y_{i1} \ y_{i2}]^T$  representan los estados del vehículo como es la posición y velocidad respectivamente en cada dinámica dentro del plano horizontal, aquí la gran diferencia se encuentra en las matrices  $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

y  $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  para ambos casos y tanto  $d_{ix}$  como  $d_{iy}$  representan las perturbaciones en sus respectivas dinámicas. Por lo tanto podemos proponer de la misma forma el controlador nominal distribuido con su respectivo termino adaptable como

$$u_{ix} = cKe_{ix} - \hat{W}_{ix}^T \Psi(x_i) \quad (4.21)$$

y

$$u_{iy} = cKe_{iy} - \hat{W}_{iy}^T \Psi(y_i) \quad (4.22)$$

con sus respectivos errores de sincronización de vecindad  $e_{ix}$  y  $e_{iy}$ , la ganancia de retroalimentación  $K = -B^T P$  se obtiene mediante la desigualdad de Riccati

$$A^T P + PA + Q - PBB^T P \leq 0 \quad (4.23)$$

siendo cuidadosos con las matrices  $A$  y  $B$  correspondientes a este par de dinámicas. Finalmente, podemos definir el error de seguimiento en el plano horizontal como  $\delta_{ix} = x_i - x_0$  y  $\delta_{iy} = y_i - y_0$ , la cual derivando y realizando las substituciones adecuadas obtendremos las dinámicas para

todo el sistema multiagente dentro del plano  $X - Y$

$$\dot{\delta}_x = (I_N \otimes A + cH \otimes BK)\delta_x + (I_N \otimes B)(\tilde{W}_x^T \Psi(x) + \epsilon_x - \underline{r}) \quad (4.24)$$

y

$$\dot{\delta}_y = (I_N \otimes A + cH \otimes BK)\delta_y + (I_N \otimes B)(\tilde{W}_y^T \Psi(y) + \epsilon_y - \underline{r}) \quad (4.25)$$

### 4.3. Análisis de estabilidad y convergencia

*Demostración.* Para realizar la prueba de estabilidad, definimos las siguientes variables,  $\delta_z = [\delta_{z_1}^T, \dots, \delta_{z_N}^T]^T \in \mathbb{R}^{2N}$ ,  $e_z = [e_{z_1}^T, \dots, e_{z_N}^T]^T \in \mathbb{R}^{2N}$ ,  $\epsilon_z = [\epsilon_{z_1}, \dots, \epsilon_{z_N}]^T \in \mathbb{R}^N$ ,  $\Psi_z = [\Psi_{z_1}^T(x_1), \dots, \Psi_{z_N}^T(x_N)]^T \in \mathbb{R}^{Ns}$ ,  $\tilde{W}_z = \text{diag}(\tilde{W}_{z_i}) \in \mathbb{R}^{Ns \times N}$ , y  $\underline{r}_z = [r_z, \dots, r_z]^T \in \mathbb{R}^N$ . A partir de las definiciones anteriores y la dinámica del error de seguimiento para el  $i$ -ésimo agente dada en la Ecuación (4.3), la dinámica del error de seguimiento global  $\delta_z$  se puede escribir como

$$\dot{\delta}_z = \left( I_N \otimes A_z + c_z \mathcal{H} \otimes B_z K_z \right) \delta_z + \left( I_N \otimes B_z \right) \left( \tilde{W}_{z_i}^T \Psi_z(z) + \epsilon_z - \underline{r}_z \right). \quad (4.26)$$

Ahora, considere la función candidata de Lyapunov definida como

$$V_z = \frac{1}{2} \delta_z^T \left( \mathcal{H} \otimes M_z \right) \delta_z + \frac{1}{2} \text{tr} \left\{ \tilde{W}_z^T \Gamma_z^{-1} \tilde{W}_z \right\} \quad (4.27)$$

donde  $\Gamma_z = \text{diag}(\bar{\Gamma}_{z_i}) \in \mathbb{R}^{Ns \times Ns}$ ,  $\bar{\Gamma}_{z_i} = \Gamma_{z_i} I_{x \times s} \in \mathbb{R}^{s \times s}$ , y  $\Gamma_{z_i} > 0$ ,  $i = 1, \dots, N$ . La derivada temporal de  $V_z$  a lo largo de las trayectorias de (4.26) se obtiene como

$$\begin{aligned} \dot{V}_z = & \delta_z^T \left( \mathcal{H} \otimes M_z \right) \left[ \left( I_N \otimes A_z + c_z \mathcal{H} \otimes B_z K_z \right) \delta_z + \left( I_N \otimes B_z \right) \left( \tilde{W}_z^T \Psi_z(z) + \epsilon_z - \underline{r}_z \right) \right] \\ & + \text{tr} \left\{ \tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z \right\} \end{aligned} \quad (4.28)$$

el cual puede ser reescrito como

$$\begin{aligned} \dot{V}_z = & \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes \left( M_z A_z + A_z^T M_z \right) - 2c_z \mathcal{H}^2 \otimes \left( M_z B_z B_z^T M_z \right) \right] \delta_z \\ & + \delta_z^T \left( \mathcal{H} \otimes M_z B_z \right) \left( \tilde{W}_z^T \Psi_z(z) + \epsilon_z - \underline{r}_z \right) + \text{tr} \left\{ \tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z \right\} \end{aligned} \quad (4.29)$$

A partir de las propiedades de la traza, podemos obtener lo siguiente

$$\begin{aligned} \dot{V}_z = & \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z) \right] \delta_z + \delta_z^T \left( \mathcal{H} \otimes M_z B_z \right) \left( \epsilon_z - \underline{r}_z \right) \\ & + \text{tr} \left\{ \tilde{W}_z^T \Psi_z(z) \delta_z^T (\mathcal{H} \otimes M_z B_z) - \tilde{W}_z^T \Gamma_z^{-1} \dot{\hat{W}}_z \right\} \end{aligned} \quad (4.30)$$

Observe que la ley de adaptación para los pesos del  $i$ -ésimo agente presentado en la Ecuación (4.18), se puede reescribir en forma global como

$$\dot{\hat{W}}_z = \Gamma_z \left[ \Psi_z(z) \delta_z^T (\mathcal{H} \otimes M_z B_z) - k_z H_z \right] \quad (4.31)$$

donde  $k_z = \text{diag}(\bar{k}_{z_i}) \in \mathbb{R}^{N_s \times N_s}$ ,  $\bar{k}_{z_i} = k_{z_i} I_{s \times s} \in \mathbb{R}^{s \times s}$ , y  $k_{z_i}$  son constantes positivas y  $H_z = \text{diag}(H_{z_i}) \in \mathbb{R}^{N_s \times N}$  con  $H_{z_i}$  definido en la Ecuación (4.10). Sustituyendo la ley de adaptación global (4.31) en la derivada de la función de Lyapunov (4.32), obtenemos

$$\begin{aligned} \dot{V}_z = & \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z) \right] \delta_z + \delta_z^T \left( \mathcal{H} \otimes M_z B_z \right) \left( \epsilon_z - \underline{r}_z \right) \\ & + \text{tr} \left\{ \tilde{W}_z^T k_z H_z \right\} \end{aligned} \quad (4.32)$$

Definiendo  $P_z = \text{diag}(P_{z_i}) \in \mathbb{R}^{N_s \times N_s}$ ,  $Q_z = \text{diag}(Q_{z_i}) \in \mathbb{R}^{N_s \times N}$  y  $\varrho_z = \text{diag}(\varrho_{z_i}) \in \mathbb{R}^{N_s \times N}$ , podemos expresar  $H_z$  como

$$\begin{aligned} H_z &= P_z \hat{W}_z - Q_z \\ &= -P_z \tilde{W}_z + \varrho_z. \end{aligned} \quad (4.33)$$

Entonces, sustituyendo  $H_z$  en la Ecuación (4.32), obtenemos lo siguiente

$$\begin{aligned} \dot{V}_z = & \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z) \right] \delta_z + \delta_z^T \left( \mathcal{H} \otimes M_z B_z \right) \left( \epsilon_z - \underline{r}_z \right) \\ & - \text{tr} \left\{ \tilde{W}_z^T k_z P_z \tilde{W}_z + \tilde{W}_z^T k_z \varrho_z \right\} \end{aligned} \quad (4.34)$$

De la desigualdad de Young, para el segundo término en (4.34), obtenemos

$$\begin{aligned}
\dot{V}_z &= \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z) \right] \delta_z \\
&\quad + \frac{1}{2} \delta_z^T \left( \mathcal{H} \otimes M_z B_z \right) \left( \mathcal{H}^T \otimes B_z^T M_z \right) \delta_z - \text{tr} \left\{ \tilde{W}_z^T k_z P_z \tilde{W}_z + \tilde{W}_z^T k_z \varrho_z \right\} \\
&\quad + \frac{1}{2} \left( \epsilon_z - \underline{r}_z \right)^T \left( \epsilon_z - \underline{r}_z \right) \\
&= \frac{1}{2} \delta_z^T \left[ \mathcal{H} \otimes (M_z A_z + A_z^T M_z) - (2c_z + 1) \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z) \right] \delta_z \\
&\quad - \text{tr} \left\{ \tilde{W}_z^T k_z P_z \tilde{W}_z + \tilde{W}_z^T k_z \varrho_z \right\} + \frac{1}{2} \left( \epsilon_z - \underline{r}_z \right)^T \left( \epsilon_z - \underline{r}_z \right)
\end{aligned} \tag{4.35}$$

Para simplificar, usamos una matriz unitaria  $U$  tal que  $U^T \mathcal{H} U = \Lambda_z = \text{diag}(\lambda_{z_1}, \dots, \lambda_{z_N})$ , donde  $\lambda_{z_i}$  son los valores propios de la matriz  $\mathcal{H}$ , que es definida positiva. Introduzcamos una transformación de estado dada como  $\xi_z = (U^T \otimes I_n) \delta_z$  con  $\xi_z = [\xi_{z_1}^T, \dots, \xi_{z_N}^T]^T$ . Entonces, introduciendo la transformación de estado en la Ecuación (4.35), obtenemos

$$\begin{aligned}
\dot{V}_z &= \frac{1}{2} \xi_z^T \left[ \Lambda \otimes (M_z A_z + A_z^T M_z) - (2c_z + 1) \Lambda^2 \otimes (M_z B_z B_z^T M_z) \right] \xi_z \\
&\quad - \text{tr} \left\{ \tilde{W}_z^T k_z P_z \tilde{W}_z + \tilde{W}_z^T k_z \varrho_z \right\} + \frac{1}{2} \left( \epsilon_z - \underline{r}_z \right)^T \left( \epsilon_z - \underline{r}_z \right)
\end{aligned} \tag{4.36}$$

que se reescribe en forma desarrollada como

$$\begin{aligned}
\dot{V}_z &= \frac{1}{2} \sum_{i=1}^N \lambda_{z_i} \xi_{z_i}^T \left[ M_z A_z + A_z^T M_z - (2c_z + 1) \lambda_{z_i} M_z B_z B_z^T M_z + N_z \right] \xi_{z_i} - \frac{1}{2} \sum_{i=1}^N \lambda_{z_i} \xi_{z_i}^T N_z \xi_{z_i} \\
&\quad - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T P_{z_i} \tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T \varrho_{z_i} + \frac{1}{2} \left( \epsilon_z - \underline{r}_z \right)^T \left( \epsilon_z - \underline{r}_z \right) \\
&\leq \frac{1}{2} \sum_{i=1}^N \lambda_{z_i} \xi_{z_i}^T \left[ M_z A_z + A_z^T M_z - (2c_z + 1) \min(\lambda_{z_i}) M_z B_z B_z^T M_z + N_z \right] \xi_{z_i} \\
&\quad - \frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \sum_{i=1}^N \xi_{z_i}^T N_z \xi_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T P_{z_i} \tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T \varrho_{z_i} + \epsilon_z^+
\end{aligned} \tag{4.37}$$

donde  $\frac{1}{2} \|\epsilon_z - \underline{r}_z\|^2 \leq \epsilon_z^+$  con  $\epsilon_z^+$  un escalar positivo. Definamos la ganancia de acoplamiento  $c_z$  como

$$c_z \geq \frac{1}{2 \min_{i=1, \dots, N}(\lambda_{z_i})} - \frac{1}{2}, \tag{4.38}$$

tal que  $(2c_z + 1) \min(\lambda_{z_i}) \geq 1$ . Como la parte  $(A_z, B_z)$  es estabilizable, entonces se garantiza la existencia de la matriz solución  $M_z$ , tal que

$$M_z A_z + A_z^T M_z + N_z - (2c_z + 1) \min(\lambda_{z_i}) M_z B_z B_z^T M_z \leq M_z A_z + A_z^T M_z + N_z - M_z B_z B_z^T M_z \leq 0. \quad (4.39)$$

Entonces, de la Ecuación (4.39), podemos reescribir (4.37) como

$$\dot{V}_z \leq -\frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \sum_{i=1}^N \xi_{z_i}^\top N_z \xi_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T P_{z_i} \tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T \varrho_{z_i} + \epsilon_z^+ \quad (4.40)$$

Del Lema 1, las matrices  $P_{z_i}$  son definidas positivas. Entonces, definiendo  $\lambda_{\min}(P_{z_i})$  como el valor propio mínimo de  $P_{z_i}$ ,  $i = 1, \dots, N$ , obtenemos

$$\dot{V}_z \leq -\frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \sum_{i=1}^N \xi_{z_i}^\top N_z \xi_{z_i} - k_z^+ \lambda_{\min}(P_{z_i}) \sum_{i=1}^N \tilde{W}_{z_i}^T \tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i} \tilde{W}_{z_i}^T \varrho_{z_i} + \epsilon_z^+ \quad (4.41)$$

Eliminando las sumatorias y agrupando los términos, obtenemos

$$\dot{V}_z \leq -\frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \xi_z^\top (I_N \otimes N_z) \xi_z - k_z^+ \lambda_{\min}(P_w) \text{tr}\{\tilde{W}_z^\top \tilde{W}_z\} + \left| \text{tr}\{\tilde{W}_z^T k_z \varrho_z\} \right| + \epsilon_z^+, \quad (4.42)$$

De la definición de  $\xi_z$ , obtenemos por la desigualdad de Cauchy–Schwartz y (4.42) lo siguiente

$$\dot{V}_z \leq -\frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \delta_z^\top (I_N \otimes N_z) \delta_z - k_z^+ \lambda_{\min}(P_{z_i}) \|\tilde{W}\|_F^2 + k_{z_F}^+ \varrho_z^+ \|\tilde{W}\|_F + \epsilon_z^+ \quad (4.43)$$

donde  $k_{z_F}^+ = \|k_z\|_F$ ,  $\|\varrho_z\|_F \leq \varrho_{z_F}^+$  con  $\varrho_{z_F}^+$  una constante positiva. Por la desigualdad de Young  $ab \leq \frac{a^2}{2\eta} + \frac{\eta b^2}{2}$  con  $\eta > 0$ , obtenemos

$$\begin{aligned} \dot{V}_z &\leq -\frac{1}{2} \min_{i=1, \dots, N}(\lambda_{z_i}) \lambda_{\min}(N_z) \|\delta_z\|^2 - \left( k_z^+ \lambda_{\min}(P_{z_i}) - \frac{1}{2\eta_z} \right) \|\tilde{W}\|_F^2 + \frac{\eta_z (k_{z_F}^+ \varrho_z)^2}{2} + \epsilon_z^+ \\ &\leq -\alpha_z V_z + \beta_z \end{aligned} \quad (4.44)$$

donde

$$\alpha_z = \min \left( \frac{\lambda_{\min}(\mathcal{H}) \lambda_{\min}(N_z)}{\lambda_{\max}(\mathcal{H}) \lambda_{\max}(M_z)}, \frac{2 \left( k_z^+ \lambda_{\min}(P_{z_i}) - \frac{1}{2\eta_z} \right)}{\sigma_{\max}(\Gamma_z^{-1})} \right), \quad \beta_z = \frac{\eta_z (k_{z_F}^+ \varrho_z)^2}{2} + \epsilon_z^+ \quad (4.45)$$

Teniendo en cuenta la teoría de Lyapunov extendida, es posible concluir que  $\delta_z$  y  $\tilde{W}$  están uniformemente acotados en última instancia y convergen en un pequeño conjunto compacto dado por

$$W_z := \left\{ \delta_z, \tilde{W} \mid \|\delta_z\| \leq \sqrt{\frac{2}{\lambda_{\min}(\mathcal{H})} \frac{\beta_z}{\lambda_{\min}(M_z) \alpha_z}}, \|\tilde{W}\| \leq \sqrt{\frac{2}{\sigma_{\min}(\Gamma_z^{-1})} \frac{\beta_z}{\alpha_z}} \right\} \quad (4.46)$$

Por lo tanto, los errores de seguimiento  $\delta_{z_i}$  para el  $i$ -ésimo agente están últimamente e uniformemente acotados. □

---

## Resultados

---

En este capítulo se describe de forma detallada la aplicación del algoritmo desarrollado dentro del ambiente de simulación *Gazebo*. Este simulador permite probar el comportamiento de los vehículos multirrotores sin la necesidad de utilizar los sistemas físicos que debido a la complejidad de las pruebas y requerir de una cantidad mayor de componentes por el número de vehículos que definamos como agentes seguidores este simulador o (*SITL*) por sus siglas en inglés "Software In The Loop". Nos permite representar el código del autopiloto en lenguaje C++ o Python para crear un ejecutable con el cual podremos hacer experimentos del comportamiento del código sin hardware.

Cuando se ejecuta el SITL, los datos del sensor provienen de un modelo de dinámica de vuelo en un simulador de vuelo, donde en este caso en específico *PX4*, el cual es un controlador o autopiloto de código abierto el cual es capaz de realizar el control de alto nivel de un vehículo multirrotores de forma manual o automática. Este incluye, numerosos modos de vuelo (Acro, Stabilize, Loiter, Guided, Return-to-Launch, Alt Hold, etc) con los que se pueden realizar pruebas tanto de trayectorias como de leyes de control. Además, es un software totalmente flexible y personalizable, de manera que, aunque la gran mayoría de los parámetros y configuraciones están ya preprogramadas, estos se pueden modificar si el usuario o la aplicación lo requiere.

Por otro lado, es fácil añadir nuevos tipos de sensores, objetos y plataformas robóticas simulados a partir de la configuración en *Gazebo* que se trata de un simulador dinámico en 3D que nos permite crear virtualmente sistemas robóticos de manera eficiente y precisa en ambientes complejos en interiores o al exterior. Mediante la definición de uniones (joints), enlaces (links) con colisiones (collisions), etcétera, este simulador ofrece la posibilidad de generar datos tanto en

la cinemática como la dinámica de los robots o sistemas mecánicos y electrónicos, con un alto grado de fidelidad, además de numerosos sensores e interfaces con el usuario. Además su fuerte relación con *ROS* que representa una herramienta flexible para escribir aplicaciones o programas destinados a plataformas robóticas. En concreto, se trata de una colección de herramientas y bibliotecas cuyo principal objetivo es simplificar la tarea de crear un comportamiento robótico complejo y robusto a través de una amplia variedad de sensores y canales de comunicación.

## 5.1. Controlador inteligente usando redes neuronales de base radial para un UAV

En este apartado mostramos los resultados obtenidos para el algoritmo de redes neuronales de base radial utilizando el estimador de parámetros basado en la información del error comparado con un controlador *PD* (proporcional derivativo) y usando la red neuronal con estimación de pesos utilizando el algoritmo adaptable modificación  $-\sigma$ . En las siguientes figuras mostramos el desempeño de estas estrategias de control llegando a una referencia deseada en  $x_d = 0.0$  metros,  $y_d = 0.0$  metros y  $z_d = 6 - 3 \cos\left(\frac{t}{3.5}\right)$  metros con un vector de viento con dirección  $[X \ Y \ Z]^T = [1 \ 1 \ 1]^T$  y magnitud aproximada de  $1.0 \frac{m}{s}$  con una varianza de 0.5 como perturbación  $d_\xi$  para los tres ejes. En la tabla 5.1 podemos observar los parámetros utilizados en el controlador para realizar la simulación Para esta simulación se utilizó una red neuronal de

Tabla 5.1: Parámetros y condiciones iniciales utilizados en los resultados de la simulación.

Parámetro	Valor	Parámetro	Valor
$K_z$	$[-1.72 \ -0.805]$	$l$	13
$K_y$	$[-0.5 \ -0.25]$	$x_{uav}(0)$	$[0 \ 0]^T$
$K_x$	$[-0.5 \ -0.25]$	$y_{uav}(0)$	$[0 \ 0]^T$
$\Gamma$	0.13	$z_{uav}(0)$	$[2.5 \ 0]^T$
$k$	0.3		

base radial. La red neuronal propuesta cuenta con 9 de neuronas, donde el ancho  $b = 1.5$  y los centros de los pesos neuronales se eligieron como

$$c_z = \begin{bmatrix} 0 & 0 & 0 & 3 & 3 & 3 & 6 & 6 & 6 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (5.1)$$

En la Figura 5.1 podemos observar como el algoritmo logra compensar las perturbaciones agrupadas y conforme avanza en el tiempo comienza a reducir el error de seguimiento a cero principalmente podemos ver este efecto en la dinámica del eje  $Z$  en la figura 5.1 ya que esta se encuentra afectada por el termino gravitatorio y el viento. Es importante señalar como con el controlador  $PD$  no logra compensar cuando llega a cada cresta de la función de referencia y en la figura se encuentra un acercamiento comparando los otros dos estrategias de control con un la red neuronal y se ve de una mejor forma como la línea punteada verde converge de una mejor forma con respecto a la línea continua de color magenta.

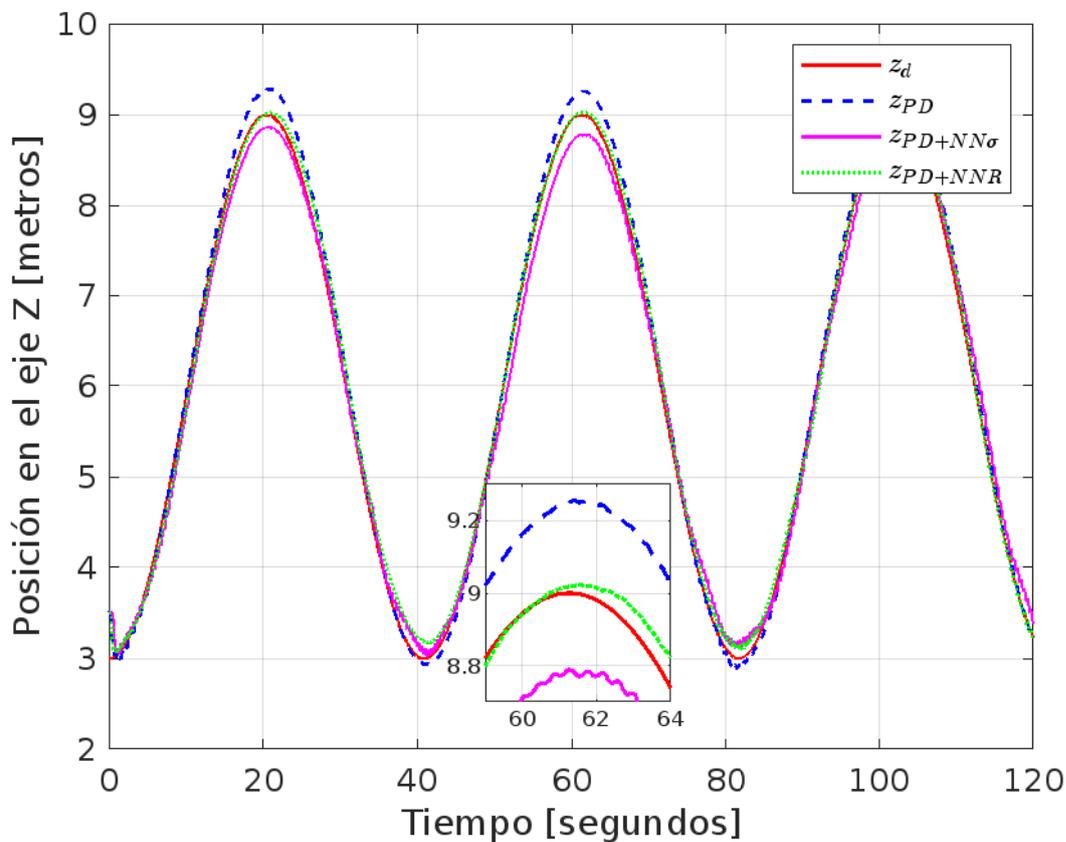


Figura 5.1: Resultado del proceso de seguimiento en el eje  $Z$ , donde  $z_d$  representa la referencia deseada,  $z_{PD}$  es el seguimiento con un controlador Proporcional Derivativo,  $z_{PD+NN\sigma}$  utiliza un controlador PD y el término de la red neuronal con la actualización de pesos tradicional y  $z_{PD+NNR}$  controlador PD y el término de la red neuronal actualizando los pesos mediante el estimador mejorado

Por otro lado en la siguiente Figura 5.2 se muestra el seguimiento para una referencia circular en el plano  $X - Y$  y de la misma forma que en la anterior comparativa podemos ver como compensa de mejor forma la estrategia de control de el  $PD$  mas la red neuronal con el estimador robusto basado en la información del error con la línea punteada de color verde.

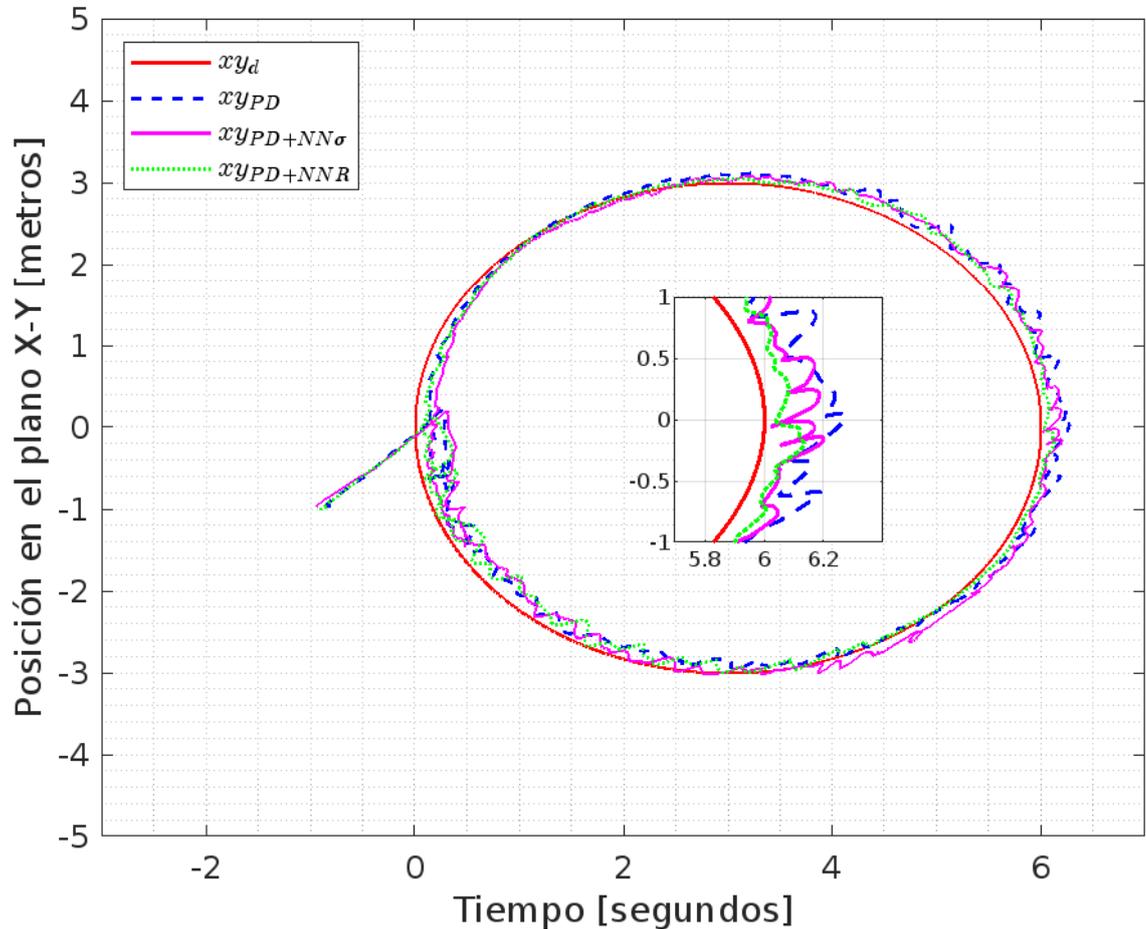


Figura 5.2: Resultado del seguimiento en el plano  $X - Y$ , donde  $xy_d$  representa la referencia deseada,  $xy_{PD}$  es el seguimiento con un controlador Proporcional Derivativo,  $xy_{PD+NN\sigma}$  utiliza un controlador PD y el término de la red neuronal con la actualización de pesos tradicional y  $xy_{PD+NNR}$  controlador PD y el término de la red neuronal actualizando los pesos mediante el estimador mejorado

## 5.2. Controlador inteligente usando redes neuronales recurrentes de alto orden para un UAV

A continuación se presentan los resultados obtenidos al aplicar un controlador basado en redes neuronales dinámicas en un vehículo multi-rotor autónomo en el entorno de simulación

*Gazebo*. El objetivo principal del controlador era realizar el seguimiento de una trayectoria deseada en tiempo real. La evaluación del controlador se realizó mediante la utilización de nodos de ROS y se evaluó su desempeño en términos de convergencia del seguimiento y estabilidad de la trayectoria.

El controlador se compone de dos partes y se propuso utilizando una red neuronal dinámica, que es capaz de modelar la dinámica del vehículo y ajustar tanto la señal de control y la ley de actualización para los pesos de la red neuronal. La ley de actualización ajusta los pesos de la red neuronal en tiempo real, lo que permite que la red neuronal adapte su modelo a la dinámica del vehículo y el entorno de vuelo, permitiendo una adaptabilidad a diferentes circunstancias.

En cuanto a la primera parte de la señal de control esta basada en un controlador "backstepping" por su definición en inglés, es una técnica de control robusto utilizada en el diseño de controladores para sistemas no lineales. El algoritmo utiliza una función de Lyapunov para garantizar la estabilidad del sistema, y se aplica en múltiples pasos, retrocediendo desde la salida del sistema hasta la entrada. En este caso, se utilizó el algoritmo de backstepping para controlar la red neuronal recurrente de alto orden, permitiendo el ajuste en tiempo real del controlador en función de la dinámica del vehículo y las condiciones del ambiente. En los siguientes pruebas se realizó de la misma forma una comparación entre la ley de actualización tradicional y la mejorada. La tabla 5.2 presenta las ganancias y parámetros del controlador y de la ley de actualización para los pesos de la red neuronal. Estos parámetros y ganancias fueron ajustados mediante pruebas y evaluaciones rigurosas, para asegurar que el controlador proporciona un seguimiento estable a la trayectoria a pesar de añadir incertidumbres como ráfagas de viento.

Tabla 5.2: Parámetros y condiciones iniciales utilizados en el control de altitud de un UAV.

Parametro	Valor	Parametro	Valor
$a_{1z}$	1.5	$a_{2z}$	1.5
$g_{1z}$	0.3	$g_{2z}$	0.3
$k_{fz}$	0.3	$k_w$	0.001
$\lambda_z$	1.7	$\Gamma_w$	0.95
$\omega(0)$	115	$\Psi_z$	$\tanh(z, \hat{z}, \tilde{z})$
$z_1(0)$	2.5	$z_2(0)$	0.0

En la figura 5.3 se muestra la gráfica comparativa del seguimiento del UAV en relación a la trayectoria deseada. Como se puede observar, el valor deseado (línea roja sólida) es igual en ca-

da uno de los casos de estudio se introdujo el controlador tradicional  $PD$  (línea azul punteada) para mantener las condiciones del experimento con las redes neuronales estáticas y en cuanto a la utilización de las redes neuronales recurrentes de alto orden (RHONN) tenemos dos casos el primero utilizando la estimación de los pesos con la técnica adaptable de modificación  $-\sigma$  (línea rosa sólida) y con el estimador basado en la información del error el cual denotamos con la letra  $R$  (línea verde sólida) dentro de las gráficas y estos últimos lograron un seguimiento acotado a un error de seguimiento cercano al medio metro en algunos instantes de la trayectoria, manteniendo el vehículo dentro de límites acotados durante todo el vuelo, sin embargo la actualización de los pesos de la red no terminan de converger a una zona ideal y nos deja ver en la figura 5.4 el error de seguimiento, como se mantiene variando durante todo el vuelo, sin embargo es importante señalar que conforme el tiempo pasa los pesos para la estimación tradicional comienzan a diverger.

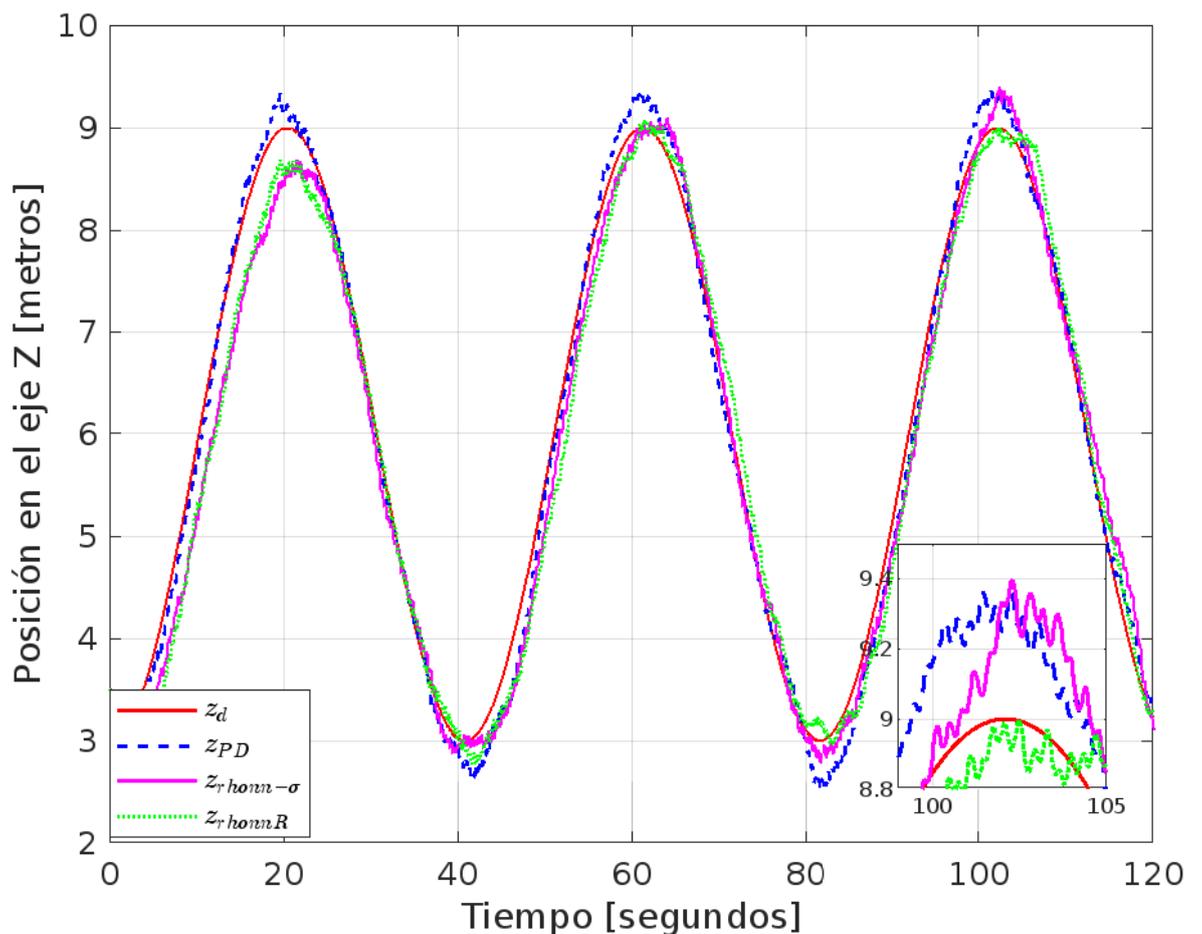


Figura 5.3: Comparativo del UAV en altura utilizando diferentes estrategias de control

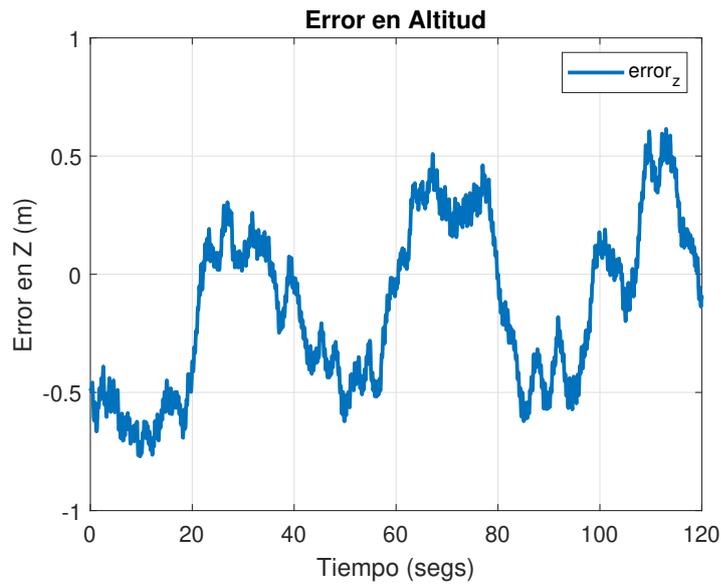


Figura 5.4: Error sobre la dinámica de altura con actualización tradicional

En la Figura 5.4 la gráfica del error de seguimiento, nos permite observar como a través del tiempo diverge el valor de error. Por otro lado, al introducir el error de estimación en la ley de actualización de parámetros el error comienza a converger y esto queda más detallado en la figura 5.5 donde el error de seguimiento, se mantiene acotado entre los valores de  $\pm 0.5$  metros lo que confirma la eficacia del controlador basado en redes neuronales dinámicas.

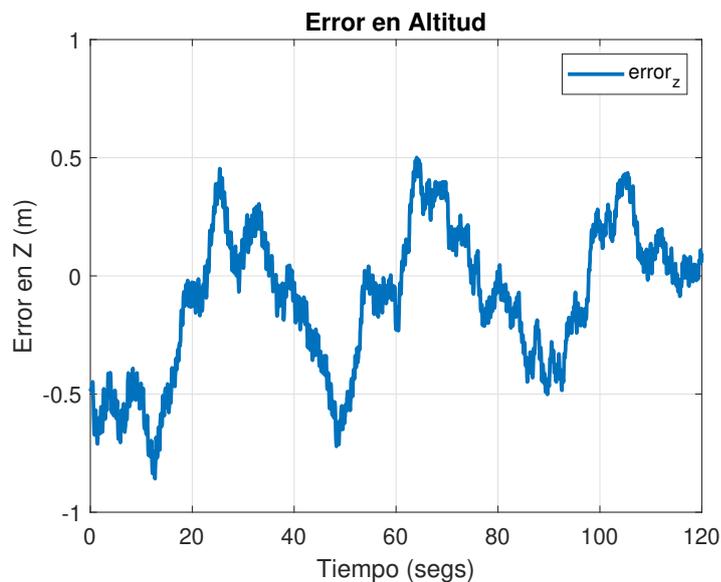


Figura 5.5: Error sobre la dinámica de altura utilizando el estimador mejorado

En resumen, los resultados obtenidos demuestran que el controlador basado en redes neu-

ronales dinámicas es estable para el vehículo y en el seguimiento de trayectorias, con una convergencia acotada. Estos resultados son prometedores y sugieren que el controlador podría ser utilizado en aplicaciones o experimentos reales para el consenso de vehículos aéreos no tripulados en condiciones adversas.

### 5.3. Resultados para multiagentes

En esta sección se presentan los resultados de la simulación para la estrategia robusta de control cooperativo desarrollada. Primero, presentamos el *Entorno de simulación*, donde los pasos necesarios para llevar a cabo las pruebas de simulación en *ROS* y *Gazebo* se resumen en el Algoritmo 1. Después de esto, presentamos dos escenarios de simulación. El primero es para el *control de altitud* de un grupo de cuatro vehículos aéreos multirrotores, donde se realiza un estudio comparativo con un control distribuido tradicional y un control distribuido adaptable utilizando las redes neuronales de base radial. En el segundo, se presentan los resultados de la simulación para el *control cooperativo traslacional* en el plano  $X - Y$ .

#### 5.3.1. Ambiente de simulación

La implementación del algoritmo dentro del simulador de Gazebo, permite probar el comportamiento de vehículos multirrotor sin necesidad de utilizar sistemas físicos debido a la complejidad de las pruebas y requiere de una mayor cantidad de componentes debido a la cantidad de vehículos que definimos como agentes seguidores en este simulador. El entorno se puede constituir a partir de las herramientas y paquetes desarrollados por *PX4* [81] que tienen una serie de subproyectos como entornos de simulación (mundos), algoritmos de control para diferentes tipos de UAV, y la conexión con *ROS* a través de una Unidad de Control de Vuelo o *FCU* (Flight Unit Control por sus siglas en inglés), que soporta la conexión con una computadora embebida todo esto a través de un protocolo estandarizado llamado *Mavlink*. Podemos observar en el algoritmo1, que tenemos que establecer el ambiente ventoso con el que se introduce el *gazebo plugin*, el cual genera las perturbaciones externas, seguido del número de agentes a utilizar, luego los tópicos y servicios se crean para cada vehículo. Desarrollamos un programa en lenguaje *python* para cada vehículo para obtener los estados a través de los sensores del vehículo y también calcular el control de movimiento, esto se puede lograr

publicando a un t3pico llamado *uavN/mavros/setpoint\_raw/attitude*, lo cual es equivalente a proporcionar los 3ngulos y el empuje deseados al  $UAV_N$  tanto del algoritmo propuesto de la red neuronal como de las se3ales de control previamente definidas. Por 3ltimo, ejecutamos el nodo principal donde establecemos las matrices de topolog3a de comunicaci3n, las ganancias para el control distribuido nominal tradicional [82] junto con la ley de actualizaci3n para los t3rminos adaptables, los estados iniciales para los agentes seguidores y el tiempo de ejecuci3n.

---

**Algorithm 1:** Algoritmo para pruebas de un sistema multiagente

---

Paso 1 Descargar e Instalar el paquete del Autopiloto de *PX4* desde su propio repositorio.

Paso 2 Compilar sin lanzar la simulación del modelo *iris* en un mundo "vacío" para *Gazebo*.

Paso 3 Añadir la dirección al código fuente del ambiente a simular.

Paso 4 Lanzar el archivo `multi_uav_mavros_sitl.launch`. ; /\* Modificar el numero de agentes (UAVs) y utilizar el archivo `windy.world` \*/

Paso 5 Definir la matriz de comunicación, los estados iniciales para cada agente seguidor y la referencia para el agente líder virtual.

Paso 6 Definir el tiempo de ejecución del nodo (i.e `exec.t = 90` segundos).

Paso 7 Ejecutar el nodo principal

Paso 8 **if**  $t \leq 10$  **then**

Se colocan los agentes en los estados iniciales ; /\* Se utiliza un controlador PID interno \*/

**if**  $t > 10$  &  $t \leq exec.t$  **then**

Se calcula el error de los agentes vecinos y se envía la información correspondiente al programa individual de cada agente;

Se computa la señal de control y la función de aproximación obtenida mediante la red neuronal;

Se publica la estrategia de control inteligente a los respectivos tópicos de cada vehiculo;

Se guardan los estados y señales de control de cada UAV en un archivo separado por comas .csv para su futuro procesamiento;

**end**

**else**

Se envía el comando de Auto aterrizaje para todos los agentes involucrados

**end**

---

### 5.3.2. Escenarios de simulación

En estos resultados de simulación consideramos la configuración del sistema multiagente con  $N = 4$  vehículos multirroto. Cada avión tiene una masa  $m = 1.5 \text{ kg}$  y los gráficos de comunicación utilizados en las simulaciones se representan en la Fig. 5.6.

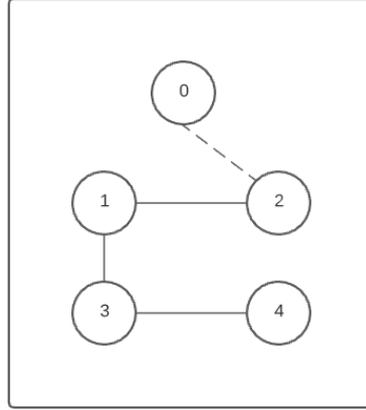


Figura 5.6: Grafo de comunicación propuesto para los sistemas multiagente.

Para el gráfico de comunicación de la Fig. 5.6, las matrices de adyacencia y de grado se definen como

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

Asimismo, las matrices de adyacencia laplaciana y líder se definen como

$$\mathcal{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{L} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad (5.3)$$

Para mostrar el desempeño de nuestro enfoque propuesto, presentamos un análisis comparativo con otras dos estrategias de control de consenso presentadas en la literatura.

- *Control distribuido tradicional.* En este enfoque, se utiliza una estrategia de control estáti-

co, que viene dada por

$$u_{z_i} = c_z K_z e_{z_i}, \quad (5.4)$$

que usa solo la ganancia de acoplamiento  $c_z$ , la ganancia de retroalimentación  $K_z$  y el error de vecindad  $e_{z_i}$  [83].

- *Control de consenso con red neuronal.* En esta estrategia, se emplea una red neuronal de base radial para compensar las perturbaciones agrupadas. La estrategia cooperativa de consenso y la ley de actualización de los pesos neuronales se presentan a continuación, donde los pesos se actualizan utilizando el método convencional [84]

$$\begin{aligned} u_{z_i} &= c_z K_z e_{z_i} - \hat{W}_{z_i}^T \Psi_{z_i}(z_i) \\ \dot{\hat{W}}_{z_i} &= \Gamma_{z_i} (\Psi_{z_i}(z_i) e_{z_i}^T M_z B_z - k_{z_i} \hat{W}_{z_i}) \end{aligned} \quad (5.5)$$

En comparación con nuestro enfoque, observe que para los pesos neurales la ley de adaptación (5.5) incluye la estimación de los pesos dada por el término  $\hat{W}_{z_i}$ .

### 5.3.2.1. Control de altitud

El modelo matemático para la dinámica de altitud del  $i$ -ésimo agente se puede escribir de la siguiente manera

$$\dot{z}_i = \begin{bmatrix} 0 & 1 \\ -0.1093 & -0.1862 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ 1.2737 \end{bmatrix} (u_{z_i} + d_{z_i}) \quad (5.6)$$

Para el agente líder, seleccionamos una entrada acotada dada como  $r(t) = 6 - 3 \cos\left(\frac{t}{3.5}\right)$ .

La Tabla 5.3 presenta los parámetros utilizados para este escenario de simulación, incluyendo las ganancias de control, las ganancias del estimador robusto y las ganancias por actualización de los pesos neuronales. Se utilizó una red neuronal de base radial, donde la salida de esta red neuronal se da en la ecuación (2.10). La red neuronal propuesta tiene 25 de neuronas, donde los centros de los pesos neuronales se eligieron como

Tabla 5.3: Parámetros y condiciones iniciales utilizados en los resultados de la simulación multiagente.

Parámetro	Valor	Parámetro	Valor
$c_z$	2.7	$k_{wz}$	0.17
$K_z$	$[-0.75 \quad -0.35]$	$z_1(0)$	$[3.0 \quad 0.0]^\top$
$l$	13	$z_2(0)$	$[2.7 \quad 0.0]^\top$
$\Gamma$	0.13	$z_3(0)$	$[2.5 \quad 0.0]^\top$
$k_{fz}$	0.3	$z_4(0)$	$[3.7 \quad 0.0]^\top$

$$c_z = \left[ \begin{array}{cccccccccccccccccccccccccccc} -2 & -2 & 2 & 2 & 2 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 5 & 5 & 5 & 5 & 5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 9 & 9 & 9 & 9 & 9 \\ -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 \end{array} \right] \quad (5.7)$$

con  $b = 0.7$  como ancho de la neurona oculta. Además, los pesos iniciales de la red se seleccionan aleatoriamente.

Como podemos ver en la figura 5.7 se utiliza un controlador clásico el cual es estacionario por lo que el error de seguimiento que realizan los agentes es constante en los segundos 20 y 60, también en el rectángulo ampliado podemos ver como los vehículos oscilan más en comparación con el agente virtual de referencia, en el segundo escenario, la Figura 5.8 podemos ver los efectos del término adaptativo en el cual el error de seguimiento comienza a disminuir sin embargo la ley de actualización para estimar los parámetros genera oscilaciones en específico en la parte inferior en los segundos 40 y 80, se introduce el estimador robusto en el tercer escenario y podemos observar en la Figura 5.9 como converge el error de seguimiento y se reducen las oscilaciones. En este trabajo, para comprobar la eficacia de la estrategia propuesta, utilizamos dos índices de desempeño basados en el error de seguimiento entre cada uno de los agentes seguidores y el agente líder. Los índices utilizados son el error absoluto de tiempo integral (*ITAE* Integral Time Absolut Error por sus siglas en ingles) y el error cuadrático de tiempo integral (*ITSE* Integral Time Square Error por sus siglas en ingles). Los resultados de estos desempeños se muestran en la Tabla 5.4 donde, como se puede ver, los valores en cada índice disminuyen y mejoran en nuestro enfoque en comparación con los otros métodos.

Tabla 5.4: Tabla de comparación de los índices de desempeño integral de error

Índice de desempeño			
Controlador	UAV	ITAE	ITSE
Controlador distribuido tradicional	Agente 1	0.568	0.141
	Agente 2	0.567	0.139
	Agente 3	0.575	0.143
	Agente 4	0.598	0.157
Controlador distribuido tradicional con termino adaptable	Agente 1	0.465	0.083
	Agente 2	0.463	0.081
	Agente 3	0.467	0.085
	Agente 4	0.494	0.091
Algoritmo propuesto	Agente 1	0.259	0.025
	Agente 2	0.251	0.023
	Agente 3	0.263	0.027
	Agente 4	0.267	0.029

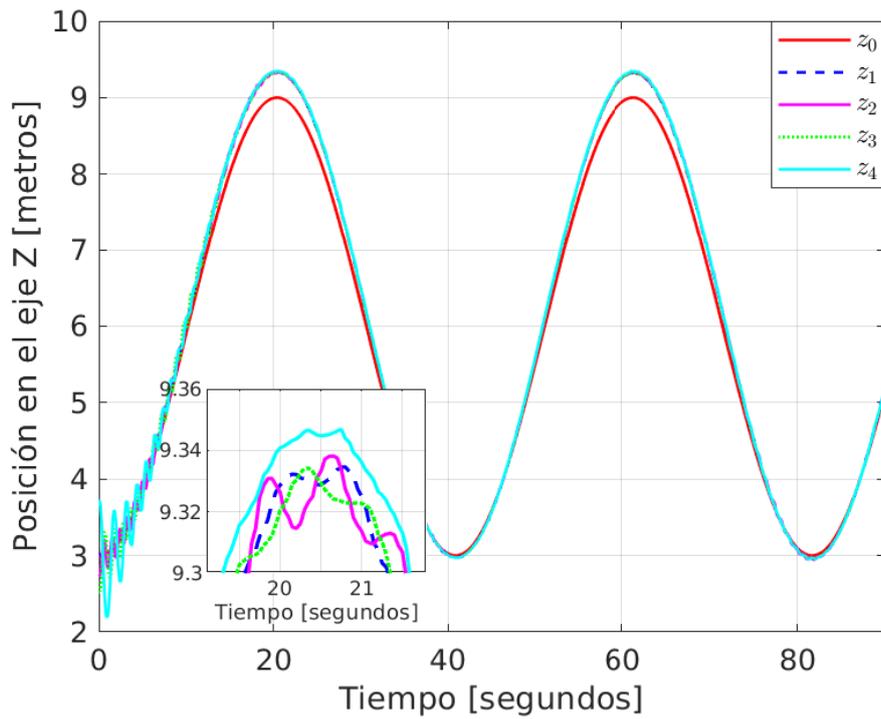


Figura 5.7: Consenso con controlador distribuido tradicional

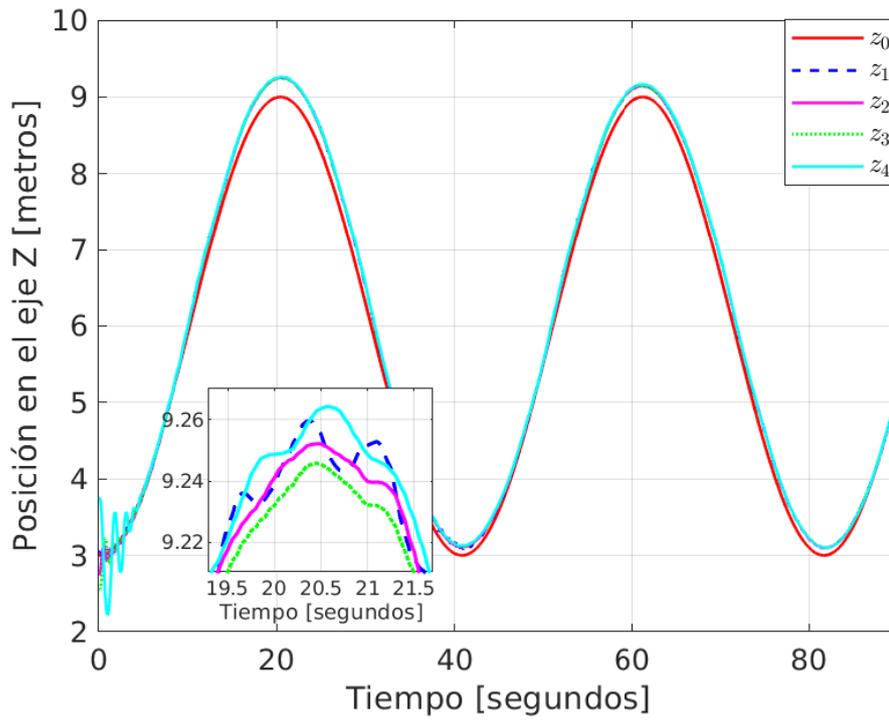


Figura 5.8: Consenso con término adaptable

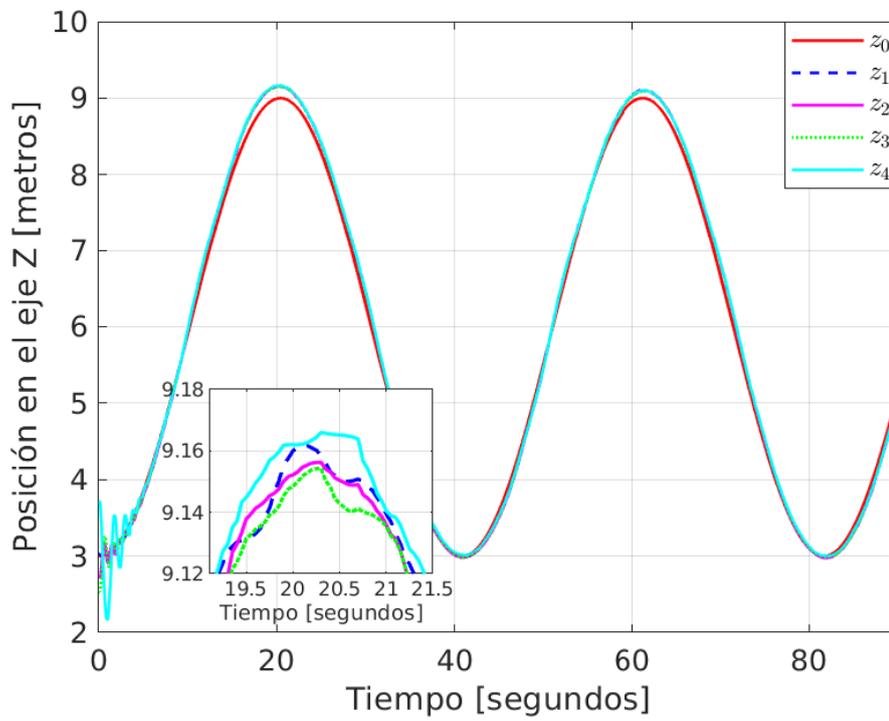


Figura 5.9: Consenso con nuestro algoritmo propuesto

### 5.3.2.2. Control cooperativo traslacional

El modelo dinámico en plano horizontal formado por los ejes  $X$  y  $Y$  se puede escribir de la siguiente forma:

$$\begin{aligned}\dot{x}_i &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u_{x_i} + d_{x_i}) \\ \dot{y}_i &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} y_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u_{y_i} + d_{y_i})\end{aligned}\tag{5.8}$$

Cabe mencionar que a partir de las señales de control de consenso  $u_{x_i}$  y  $u_{y_i}$ , se obtienen los ángulos de referencia deseados  $\phi_{d_i}$  y  $\theta_{d_i}$  y se utilizan como control. entradas para los vehículos multirrotores. Para seguir una trayectoria circular, la referencia para el agente líder en los ejes  $X$  y  $Y$  se da como

$$\begin{aligned}r_x(t) &= (-3 \cos\left(\frac{t\pi}{20}\right) + 3) - 3 \\ r_y(t) &= -3 \sin\left(\frac{\pi}{20}t\right)\end{aligned}\tag{5.9}$$

El número de neuronas es similar a la red neuronal en altura dinámica, el ancho  $b = 0,7$  y los centros de los pesos neuronales

$$c_x = \begin{bmatrix} -6 & -6 & -6 & -6 & -6 & -3 & -3 & -3 & -3 & -3 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 & 6 & 6 & 6 & 6 & 6 \\ -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 \end{bmatrix}\tag{5.10}$$

Los parámetros y las condiciones iniciales cambiarán en efecto de analizar el desempeño del algoritmo propuesto como podemos observar en la Tabla 5.5. Tenga en cuenta que los centroides y los parámetros descritos en la tabla 5.5 usados para la dinámica del eje  $X$  son los mismos para la dinámica del eje  $Y$  del sistema.

A continuación presentamos los resultados de las simulaciones con las condiciones mencionadas anteriormente, la referencia del agente líder virtual es una trayectoria circular en el plano  $X$ - $Y$  con centro en  $[0, 0]$  y las referencias circulares para cada vehículo mantienen un desplazamiento de sus centros de  $\pm 6$  metros. Por último, en las Figuras 5.10 y 5.11 podemos observar el comportamiento de la posición de los UAVs en el eje  $X$  e  $Y$  respectivamente donde podemos

Tabla 5.5: Parámetros y condiciones iniciales utilizados en el ejemplo horizontal.

Parametro	Valor	Parametro	Valor
$c_x$	30	$\Gamma_x$	0.3
$K_x$	$[-0.2 \quad -0.01]$	$\lambda_x$	13
$k_{f(x)}$	0.3	$k_{w(x)}$	2.9
$x_1(0)$	$[0.0 \quad 0.0]^\top$	$x_3(0)$	$[0.0 \quad 0.0]^\top$
$x_2(0)$	$[-9.0 \quad 0.0]^\top$	$x_4(0)$	$[-9.0 \quad 0.0]^\top$
$y_1(0)$	$[-3.0 \quad 0.0]^\top$	$y_3(0)$	$[3.0 \quad 0.0]^\top$
$y_2(0)$	$[-3.0 \quad 0.0]^\top$	$y_4(0)$	$[3.0 \quad 0.0]^\top$

ver dos vehículos siguiendo la trayectoria circular en sus respectivos compensa a pesar de tener una perturbación de viento  $d_{x,y} = 1 \frac{m}{s}$  con dirección positiva en el plano horizontal  $X - Y$  y en la figura 5.12 mostramos un punto de vista con la trayectoria de los agentes seguidores donde podemos observar cómo converge el error de seguimiento y se puede compartir un video que muestra las simulaciones en el entorno *Gazebo* en <https://youtu.be/i6Qp5MMB2Ws>.

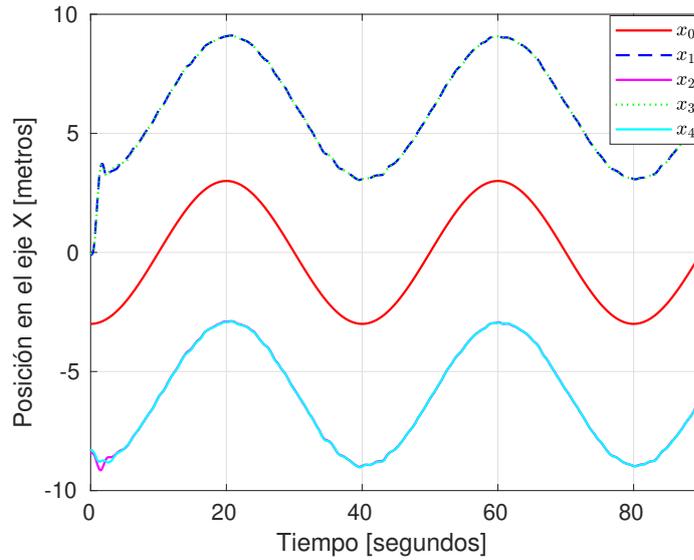


Figura 5.10: Desempeño del consenso en el eje  $X$

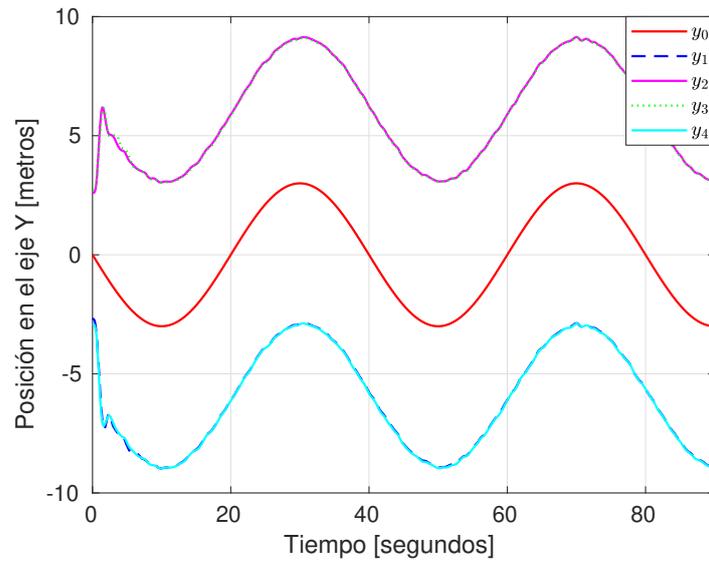


Figura 5.11: Desempeño del consenso en el eje  $Y$

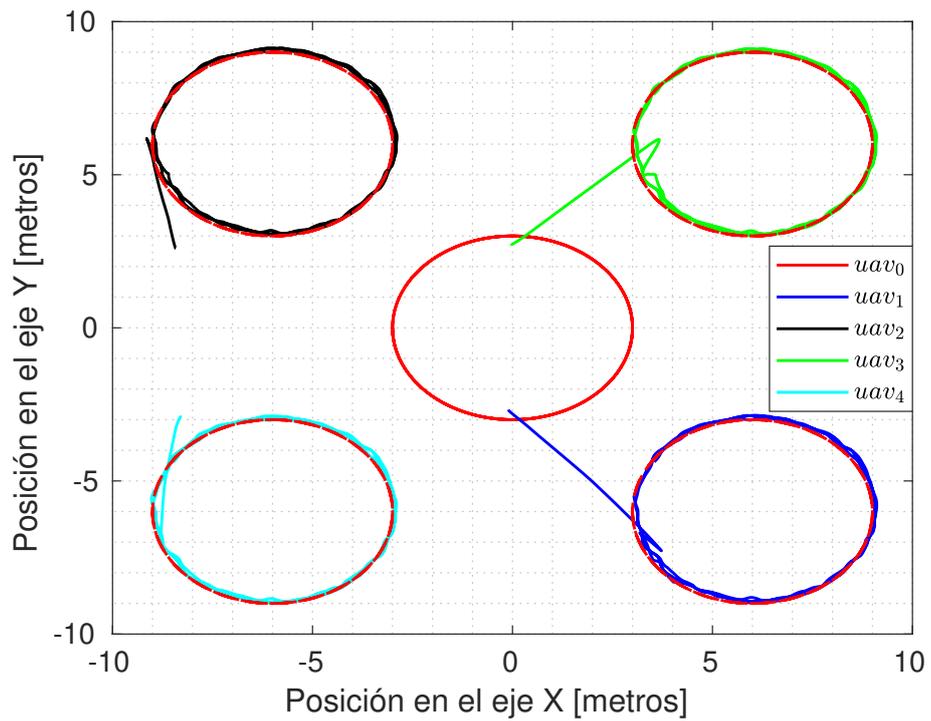


Figura 5.12: Desempeño del consenso en el plano horizontal  $X - Y$



---

## Conclusiones

---

Durante el desarrollo de este trabajo de tesis se logro obtener lo siguiente:

- Se desarrollo una estrategia de control para un vehículo UAV basado en redes neuronales de base radial, donde los pesos de la red se actualizan considerando la información del error de estimación. Los resultados de la simulación en Gazebo mostraron que la estrategia propuesta logró una buena precisión en la respuesta del sistema. Además, la técnica de adaptación de pesos permitió que el controlador mantuviera un seguimiento de trayectoria estable e incluso cuando se presentaron perturbaciones agrupadas.
- Se diseño una estrategia de control para un vehículo UAV basado en redes neuronales recurrentes de alto orden, donde los pesos se actualizan con la estimación mejorada. Los resultados de la simulación en Gazebo demostraron que la estrategia propuesta logró una buena estabilidad, sin embargo la respuesta presenta desafíos para mejorar el desempeño de este algoritmo. Sin embargo, es una buena herramienta el uso de este tipo de redes neuronales si se desconoce el modelo matemático del sistema o la perturbaciones externas.
- Se propuso una estrategia de control cooperativo líder-seguidor basada en consenso y redes neuronales de base radial para un conjunto de vehículos multirrotores sujeto a perturbaciones externas y dinámicas no modeladas, donde los pesos de la red neuronal se adaptan tomando en cuenta el error de estimación de los pesos. Los resultados de la simulación en Gazebo demostraron que la estrategia propuesta logró una buena precisión y mejora en el consenso y seguimiento de los agentes con el agente líder. Por otro lado, la

presencia de un grafo de comunicación también genero un reto importante ya que parece generar una especie de retraso entre el sistema multiagente.

- Se realizó la prueba de estabilidad de la estrategia propuesta para garantizar las condiciones de estabilidad sobre el sistema multi-agente en lazo cerrado. Los resultados de la prueba de estabilidad mostraron que las estrategias de control propuestas son estables y pueden garantizar el seguimiento de una referencia con el agente líder y la convergencia a los pesos ideales de la red neuronal del sistema en un entorno cooperativo.

En resumen, la tesis de posgrado logró desarrollar estrategias de control cooperativo del tipo líder-seguidor basadas en el algoritmo de consenso utilizando redes neuronales estáticas, y logró cumplir con los objetivos particulares propuestos, demostrando su eficacia en el control de sistemas de vehículos aéreos multirrotores en un entorno cooperativo. Los resultados obtenidos sugieren que estas estrategias de control pueden ser útiles para aplicaciones en la industria aeroespacial y la robótica, y pueden ser objeto de futuras investigaciones en el campo del control de sistemas multi-agente.

## 6.1. Trabajo Futuro

- Implementar la estrategia cooperativa desarrollada con múltiples vehículos para observar el desempeño en tiempo real y bajo perturbaciones externas reales.
- Realizar el desarrollo matemático y verificación con pruebas experimentales utilizando las redes neuronales recurrentes de alto orden para sistemas multiagentes.
- Ampliar esta investigación a la dinámica completa del vehículo multirroto al considerar las ecuaciones en el subsistema de orientación mediante alguna técnica de *backstepping*.
- La estrategia propuesta se puede ampliar para desarrollar tareas cooperativas con múltiples vehículos como el transporte de carga o la vigilancia y monitoreo de alguna zona en específica.
- Extender estos resultados en sistemas heterogéneos multiagentes.

# Bibliografía

---

- [1] F. Messina, A. Vasilakos, and P. De Meo, “Introduction to the special section on recent trends in flocking control and communication for unmanned vehicles,” *Computers & Electrical Engineering*, vol. 80, p. 106495, 2019.
- [2] Y. Qiu and L. Xiang, “Distributed adaptive coordinated tracking for coupled non-holonomic mobile robots,” *IET Control Theory & Applications*, vol. 8, no. 18, pp. 2336–2345, 2014.
- [3] H. Li, Q. Liu, G. Feng, and X. Zhang, “Leader–follower consensus of nonlinear time-delay multiagent systems: A time-varying gain approach,” *Automatica*, vol. 126, p. 109444, 2021.
- [4] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle routing problems for drone delivery,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.
- [5] R. Bähnemann, D. Schindler, M. Kamel, R. Siegwart, and J. Nieto, “A decentralized multi-agent unmanned aerial system to search, pick up, and relocate objects,” in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 123–128, 2017.
- [6] W.-N. Chen and J. Zhang, “Ant colony optimization for software project scheduling and staffing with an event-based scheduler,” *IEEE Transactions on Software Engineering*, vol. 39, no. 1, pp. 1–17, 2013.

- [7] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang, and B. Ning, "A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems," *Neurocomputing*, vol. 275, pp. 1684–1701, 2018.
- [8] B. Zhou, W. Wang, and H. Ye, "Cooperative control for consensus of multi-agent systems with actuator faults," *Computers & Electrical Engineering*, vol. 40, no. 7, pp. 2154–2166, 2014.
- [9] C. Hua, A. Jiang, and K. Li, "Adaptive neural network finite-time tracking quantized control for uncertain nonlinear systems with full-state constraints and applications to quavs," *Neurocomputing*, vol. 440, pp. 264–274, 2021.
- [10] M. Zhang and H. H. T. Liu, "Cooperative tracking a moving target using multiple fixed-wing uavs," *Journal of Intelligent & Robotic Systems*, vol. 81, pp. 505–529, Mar 2016.
- [11] K. K. Dhiman, A. Abhishek, and M. Kothari, *Cooperative Load Control and Transportation*.
- [12] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 73–82, 2017.
- [13] D. Wang, Q. Zong, B. Tian, F. Wang, and L. Dou, "Finite-time fully distributed formation reconfiguration control for uav helicopters," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 18, pp. 5943–5961, 2018.
- [14] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *2014 American Control Conference*, pp. 2077–2082, 2014.
- [15] O. Saif, *Reactive navigation of a fleet of drones in interaction*. Theses, Université de Technologie de Compiègne, Mar. 2016.
- [16] T. Chen, Q. Gao, and M. Guo, "An improved multiple uavs cooperative flight algorithm based on leader follower strategy," in *2018 Chinese Control And Decision Conference (CCDC)*, pp. 165–169, 2018.

- [17] K. Klausen, T. Fossen, and T. Johansen, “Nonlinear control with swing damping of a multirotor uav with suspended load,” *Journal of Intelligent Robotic Systems*, vol. 88, pp. 379–394, 2017.
- [18] S. Safavi and U. A. Khan, “Leader-follower consensus in mobile sensor networks,” *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2249–2253, 2015.
- [19] J. Zhang, J. Yan, P. Zhang, and X. Kong, “Collision avoidance in fixed-wing uav formation flight based on a consensus control algorithm,” *IEEE Access*, vol. 6, pp. 43672–43682, 2018.
- [20] S. T. Yuanyuan Xu, Tieshan Li, “Event-triggered adaptive fuzzy bipartite consensus control of multiple autonomous underwater vehicles,” *IET Control Theory Applications*, vol. 14, pp. 3632–3642(10), December 2020.
- [21] X. Cai, C. Wang, G. Wang, L. Xu, J. Liu, and Z. Zhang, “Leader-following consensus control of position-constrained multiple euler-lagrange systems with unknown control directions,” *Neurocomputing*, vol. 409, pp. 208–216, 2020.
- [22] W. Cao, J. Zhang, and W. Ren, “Leader–follower consensus of linear multi-agent systems with unknown external disturbances,” *Systems Control Letters*, vol. 82, pp. 64–70, 2015.
- [23] J. Bai, G. Wen, and A. Rahmani, “Leaderless consensus for the fractional-order nonlinear multi-agent systems under directed interaction topology,” *International Journal of Systems Science*, vol. 49, no. 5, pp. 954–963, 2018.
- [24] R. Gao, J. Huang, and L. Wang, “Leaderless consensus control of uncertain multi-agents systems with sensor and actuator attacks,” *Information Sciences*, vol. 505, pp. 144–156, 2019.
- [25] R. Olfati-Saber and R. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [26] A. Abdessameud and A. Tayebi, “On consensus algorithms design for double integrator dynamics,” *Automatica*, vol. 49, no. 1, pp. 253–260, 2013.

- [27] B. Tian, Z. Zuo, and H. Wang, “Leader–follower fixed-time consensus of multi-agent systems with high-order integrator dynamics,” *International Journal of Control*, vol. 90, no. 7, pp. 1420–1427, 2017.
- [28] G.-X. Wen, “Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems,” *IET Control Theory Applications*, vol. 9, pp. 1927–1934(7), August 2015.
- [29] A. A. Neto, L. A. Mozelli, and F. O. Souza, “Control of air-ground convoy subject to communication time delay,” *Computers & Electrical Engineering*, vol. 76, pp. 213–224, 2019.
- [30] L. Ma, Z. Wang, Y. Liu, and F. E. Alsaadi, “A note on guaranteed cost control for nonlinear stochastic systems with input saturation and mixed time-delays,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4443–4456.
- [31] R. Sakthivel, R. Sakthivel, B. Kaviarasan, and F. Alzahrani, “Leader-following exponential consensus of input saturated stochastic multi-agent systems with markov jump parameters,” *Neurocomputing*, vol. 287, pp. 84–92, 2018.
- [32] J. Wang, Y. Xu, Y. Xu, and D. Yang, “Time-varying formation for high-order multi-agent systems with external disturbances by event-triggered integral sliding mode control,” *Applied Mathematics and Computation*, vol. 359, pp. 333–343, 2019.
- [33] S. Yu and X. Long, “Finite-time consensus for second-order multi-agent systems with disturbances by integral sliding mode,” *Automatica*, vol. 54, pp. 158–165, 2015.
- [34] C. Liu, S. Sun, C. Tao, Y. Shou, and B. Xu, “Sliding mode control of multi-agent system with application to uav air combat,” *Computers Electrical Engineering*, vol. 96, p. 107491, 2021.
- [35] Y. Yin, X. Bu, P. Zhu, and W. Qian, “Point-to-point consensus tracking control for unknown nonlinear multi-agent systems using data-driven iterative learning,” *Neurocomputing*, vol. 488, pp. 78–87, 2022.

- [36] L. Zhang, J. Li, Y. Zhu, H. Shi, and K.-S. Hwang, "Multi-agent reinforcement learning by the actor-critic model with an attention interface," *Neurocomputing*, vol. 471, pp. 275–284, 2022.
- [37] H. Wu, J. Zhang, Z. Wang, Y. Lin, and H. Li, "Sub-avg: Overestimation reduction for cooperative multi-agent reinforcement learning," *Neurocomputing*, vol. 474, pp. 94–106, 2022.
- [38] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2017.
- [39] Y. Li and C. Tan, "A survey of the consensus for multi-agent systems," *Systems Science & Control Engineering*, vol. 7, no. 1, pp. 468–482, 2019.
- [40] J. E. Sierra and M. Santos, "Wind and payload disturbance rejection control based on adaptive neural estimators: Application on quadrotors," *Complexity*, vol. 2019, p. 6460156, Jan 2019.
- [41] Z. Peng, D. Wang, H. Zhang, and Y. Lin, "Cooperative output feedback adaptive control of uncertain nonlinear multi-agent systems with a dynamic leader," *Neurocomputing*, vol. 149, pp. 132–141, 2015.
- [42] M. Shahvali and K. Shojaei, "Distributed adaptive neural control of nonlinear multi-agent systems with unknown control directions," *Nonlinear Dynamics*, vol. 83, pp. 2213–2228, 2016.
- [43] C. Dong, Q. Ye, and S.-L. Dai, "Neural-network-based adaptive output-feedback formation tracking control of usvs under collision avoidance and connectivity maintenance constraints," *Neurocomputing*, vol. 401, pp. 101–112, 2020.
- [44] Z. Peng, D. Wang, and J. Wang, "Predictor-based neural dynamic surface control for uncertain nonlinear systems in strict-feedback form," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2156–2167, 2017.

- [45] M. Hashemi and G. Shahgholian, “Distributed robust adaptive control of high order nonlinear multi agent systems,” *ISA Transactions*, vol. 74, pp. 14–27, 2018.
- [46] P. Ioannou and P. Kokotovic, “Robust redesign of adaptive control,” *IEEE Transactions on Automatic Control*, vol. 29, no. 3, pp. 202–211, 1984.
- [47] K. Narendra and A. Annaswamy, “A new adaptive law for robust adaptation without persistent excitation,” *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, 1987.
- [48] V. Adetola, M. Guay, and D. Lehrer, “Adaptive estimation for a class of nonlinearly parameterized dynamical systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2818–2824, 2014.
- [49] V. Adetola and M. Guay, “Finite-time parameter estimation in adaptive control of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 807–811, 2008.
- [50] J. Na, G. Herrmann, X. Ren, M. N. Mahyuddin, and P. Barber, “Robust adaptive finite-time parameter estimation and control of nonlinear systems,” in *2011 IEEE International Symposium on Intelligent Control*, pp. 1014–1019, 2011.
- [51] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, “Robust adaptive finite-time parameter estimation and control for robotic systems,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.
- [52] J. Na, M. N. Mahyuddin, G. Herrmann, R. Xuemei, and P. Barber, “Robust adaptive finite-time parameter estimation and control for robotic systems,” *International Journal of Robust and Nonlinear Control*, vol. 25, 09 2014.
- [53] M. Mahyuddin, S. Khan, and G. Herrmann, “A novel robust adaptive control algorithm with finite-time online parameter estimation of a humanoid robot arm,” *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 294–305, 2014. Advances in Autonomous Robotics — Selected extended papers of the joint 2012 TAROS Conference and the FIRA Robo-World Congress, Bristol, UK.

- [54] B. Jing, J. Na, G. Gao, and G. Sun, "Robust adaptive control for robotic systems with guaranteed parameter estimation," in *Proceedings of the 2015 Chinese Intelligent Systems Conference* (Y. Jia, J. Du, H. Li, and W. Zhang, eds.), (Berlin, Heidelberg), pp. 341–352, Springer Berlin Heidelberg, 2016.
- [55] J. Yang, J. Na, Y. Guo, and X. Wu, "Adaptive estimation of road gradient and vehicle parameters for vehicular systems," *IET Control Theory and Applications*, vol. 9, no. 6, pp. 935–943, 2015.
- [56] M. N. Mahyuddin, J. Na, G. Herrmann, X. Ren, and P. Barber, "Adaptive observer-based parameter estimation with application to road gradient and vehicle mass estimation," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2851–2863, 2014.
- [57] J. Na, Y. Huang, X. Wu, G. Gao, G. Herrmann, and J. Z. Jiang, "Active adaptive estimation and control for vehicle suspensions with prescribed performance," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2063–2077, 2018.
- [58] J. Zhao, X. Wang, G. Gao, J. Na, H. Liu, and F. Luan, "Online adaptive parameter estimation for quadrotors," *Algorithms*, vol. 11, no. 11, 2018.
- [59] Q. Zhao and G. Duan, "Adaptive finite-time tracking control of 6dof spacecraft motion with inertia parameter identification," *IET Control Theory and Applications*, vol. 13, pp. 2075–2085, September 2019.
- [60] C. Yang, T. Teng, B. Xu, Z. Li, J. Na, and C. Su, "Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence," *International Journal of Control, Automation and Systems*, vol. 15, pp. 1916–1924, 2017.
- [61] F. Luan, J. Na, Y. Huang, and G. Gao, "Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence," *Neurocomputing*, vol. 337, pp. 153–164, 2019.
- [62] J. Na, S. Wang, Y.-J. Liu, Y. Huang, and X. Ren, "Finite-time convergence adaptive neural network control for nonlinear servo systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2568–2579, 2020.

- [63] Y. Jiang, Y. Wang, Z. Miao, J. Na, Z. Zhao, and C. Yang, “Composite-learning-based adaptive neural control for dual-arm robots with relative motion,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1010–1021, 2022.
- [64] C. Tabasso, N. Mimmo, V. Cichella, and L. Marconi, “Optimal motion planning for localization of avalanche victims by multiple uavs,” *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2054–2059, 2021.
- [65] Z. Xu, D. Deng, and K. Shimada, “Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, 2021.
- [66] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” p. 155, 2007.
- [67] N. T. Nguyen, I. Prodan, F. Stoican, and L. Lefèvre, “Reliable nonlinear control for quadcopter trajectory tracking through differential flatness,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6971–6976, 2017. 20th IFAC World Congress.
- [68] X. Zhang, L. Liu, and G. Feng, “Leader–follower consensus of time-varying nonlinear multi-agent systems,” *Automatica*, vol. 52, pp. 8–14, 2015.
- [69] J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*, pp. 339–362. 01 2013.
- [70] J. Park and I. W. Sandberg, “Universal Approximation Using Radial-Basis-Function Networks,” *Neural Computation*, vol. 3, pp. 246–257, 06 1991.
- [71] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, Dec 1989.
- [72] E. Kosmatopoulos, M. Polycarpou, M. Christodoulou, and P. Ioannou, “High-order neural network structures for identification of dynamical systems,” *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 422–431, 1995.
- [73] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. USA: Prentice-Hall, Inc., 1989.

- [74] S. Sastry, M. Bodson, and J. F. Bartram, “Adaptive control: Stability, convergence, and robustness,” *The Journal of the Acoustical Society of America*, vol. 88, no. 1, pp. 588–589, 1990.
- [75] H. Xu and P. A. Ioannou, “Robust adaptive control of linearizable nonlinear single input systems,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 391–396, 2002. 15th IFAC World Congress.
- [76] J. Na, G. Herrmann, X. Ren, M. N. Mahyuddin, and P. Barber, “Robust adaptive finite-time parameter estimation and control of nonlinear systems,” in *2011 IEEE International Symposium on Intelligent Control*, pp. 1014–1019, 2011.
- [77] J. Na, M. N. Mahyuddin, G. Herrmann, and X. Ren, “Robust adaptive finite-time parameter estimation for linearly parameterized nonlinear systems,” in *Proceedings of the 32nd Chinese Control Conference*, pp. 1735–1741, 2013.
- [78] B. E. Jackson, T. A. Howell, K. Shah, M. Schwager, and Z. Manchester, “Scalable cooperative transport of cable-suspended loads with uavs using distributed trajectory optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3368–3374, 2020.
- [79] B. Shirani, M. Najafi, and I. Izadi, “Cooperative load transportation using multiple uavs,” *Aerospace Science and Technology*, vol. 84, pp. 158–169, 2019.
- [80] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen, “Cooperative control for multirotors transporting an unknown suspended load under environmental disturbances,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 653–660, 2020.
- [81] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.
- [82] H. Zhang, F. L. Lewis, and A. Das, “Optimal design for synchronization of cooperative systems: State feedback, observer and output feedback,” *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1948–1952, 2011.

- [83] F. C. Souza, S. R. B. D. Santos, A. M. d. Oliveira, and S. N. Givigi, "Influence of network topology on uavs formation control based on distributed consensus," in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–8, 2022.
- [84] Z. Peng, D. Wang, H. Zhang, and G. Sun, "Distributed neural network control for adaptive synchronization of uncertain dynamical multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1508–1519, 2014.
- [85] S.-H. Yu and A. M. Annaswamy, "Adaptive control of nonlinear dynamic systems using -adaptive neural networks," *Automatica*, vol. 33, no. 11, pp. 1975–1995, 1997.
- [86] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [87] M. N. O. Manipon and <https://orcid.org/0000-0003-0104-5465>, neorico27.mnm@gmail.com, Saint Mary's University, Philippines, "Effectiveness of ChemiCooking as a gamified intervention in nomenclature of compounds: Learning experiences of grade 11 students in a public school," *International Multidisciplinary Research Journal*, vol. 5, Feb. 2023.
- [88] S.-L. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1717–1727, 2016.
- [89] V. Adetola and M. Guay, "Finite-time parameter estimation in adaptive control of nonlinear systems," in *2007 American Control Conference*, pp. 1920–1924, 2007.
- [90] B. Pasik-Duncan, "Adaptive control [second edition, by karl j. astrom and bjorn wittenmark, addison wesley (1995)]," *IEEE Control Systems Magazine*, vol. 16, no. 2, pp. 87–, 1996.
- [91] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control: Algorithms, Analysis and Applications - 1rst edition*. Communications and Control Engineering, springer, 2011.