



UNIVERSIDAD MICHOACANA DE  
SAN NICOLÁS DE HIDALGO

FACULTAD DE INGENIERÍA ELÉCTRICA

CONTROL PID NEURONAL E IDENTIFICACIÓN  
DE SISTEMAS POR MEDIO DE REDES  
NEURONALES Y FILTRO DE KALMAN  
EXTENDIDO APLICADO A UN ROBOT MÓVIL  
DIFERENCIAL

TESIS

Que para obtener el título de:

**Maestro en Ciencias en Ingeniería  
Eléctrica**

Presenta:

**Manuel Elias Pech Dzul**

Directos de Tesis:

**Dr. Juan Anzures Marín**

Co-Director de Tesis:

**Dra. Alma Yolanda Alanís García**



Morelia, Michoacán a Mayo de 2025



*Dedico esta tesis a todas las personas que me apoyaron a lo largo de este proceso. Su respaldo y confianza en mí me dieron la fortaleza necesaria para superar las dificultades y alcanzar este logro. A mi esposa, mi familia, mis amigos y mis profesores.*

*A todos, muchas gracias.*



# Lista de Publicaciones

”PID Control of a PMDC Motor Identified with RHONN and EKF for Performance Improvement”.

Autores: Luis Rodrigo Ramírez Cisneros, Manuel Elías Pech Dzul, Juan Anzuarez Marin, Roberto Tapia Sánchez, Jorge Daniel Ríos Arranaga, Alma Yolanda Alanís García.

Publicado en 2024 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC 2024), Ixtapa, Mexico.



# Resumen

La tarea de controlar sistemas complejos y no lineales continúa siendo un desafío en lo que respecta a la aplicación de un control robusto; se busca que sea resistente a las perturbaciones y que pueda gestionar las no linealidades. Para lograr este objetivo, una de las técnicas a las que se ha recurrido es la combinación de técnicas de control convencional y metaheurísticas, como las presentadas en este trabajo, en el que se emplean diversas redes neuronales recurrentes de alto orden (RHONN por sus siglas en inglés) para modelar y controlar un sistema no lineal como lo es un robot móvil diferencial. Una red neuronal identificadora la cual proporciona un modelo equivalente, filtrando ruido y otras dos que generan un control adaptivo usando un PID que modificará las ganancias en tiempo real.

Uno de los puntos clave de este trabajo es abordar la problemática que frecuentemente se encuentra en la aplicación de un control experimental al necesitar parámetros del sistema para un correcto modelado, un filtrado de datos cuando existan perturbaciones externas, o en su defecto los actuadores o sensores pueden contar con retardos al ejecutar una determinada acción.

Este trabajo apunta a resolver los problemas que a menudo se generan al aplicar controles experimentales, como la dificultad de modelar el sistema cuando se desconocen los parámetros, el procesamiento de datos cuando hay ruidos, o los retrasos de los actuadores o sensores al implementar una acción. El objetivo es obtener un sistema equivalente, utilizando redes neuronales artificiales, que facilite el uso de los datos obtenidos del sistema y que, a la vez, proporcione un identificador robusto, capaz de resistir perturbaciones.

Este tipo de identificadores tienen una mayor importancia cuando se llevan a cabo pruebas experimentales, ya que son capaces de compensar variaciones del sistema, desgastes o perturbaciones al obtener los datos del sistema. Un punto clave de la propuesta del uso de redes neuronales para identificar parámetros desconocidos de un sistema, es el entrenamiento de esta. En este trabajo se usará el Filtro Extendido de Kalman (EKF por sus siglas en inglés) como entrenamiento para las diferentes redes neuronales que se usarán, permitiendo a dichas redes neuronales manejar de mejor manera las no linealidades y posibilita un entrenamiento en línea, que, si bien requiere un mayor poder computacional, este entrenamiento recompensa teniendo un rápido, preciso y robusto entrenamiento.

Posteriormente, se utiliza un control PID adaptativo, que emplea una RHONN

de una sola capa para modificar las ganancias de este controlador. Esto supone un gran beneficio sobre un controlador convencional, ya que se ajustará en línea y también tendrá la robustez que una RHONN adquiere cuando es entrenada por el EKF como se indica anteriormente. De esta manera se unen un identificador que nos permite obtener un sistema equivalente, de un sistema no lineal con parámetros desconocidos, siendo capaz de tolerar perturbaciones y también un controlador que se adaptará constantemente considerando su sistema no lineal y complejo. Las técnicas descritas se llevan a prueba con robots previamente instrumentados, demostrando el potencial de esta metodología y cómo puede simplificar aplicaciones prácticas que con un control convencional podrían requerir una técnica más compleja.

**Palabras Clave:** Redes Neuronales, Filtro Extendido de Kalman, Identificador, Control PID Adaptivo, Robot Móvil Diferencial



# Abstract

The task of controlling complex and nonlinear systems remains a challenge in the implementation of robust control; it is sought to be resilient to disturbances and capable of managing nonlinearities. To achieve this goal, one of the techniques that has been resorted to is the combination of conventional control techniques and metaheuristics, as presented in this work, in which Recurrent High-Order Neural Networks (RHONN) are employed to model a nonlinear system such as that of a differential mobile robot.

One of the key points of this work is to address the problem often encountered in the application of experimental control when system parameters are needed for accurate modeling, data filtering when necessary disturbances exist, or alternatively, actuators or sensors may have delays in executing a certain action.

This work aims to solve the problems that often arise when applying experimental control, such as the difficulty of modeling the system when parameters are unknown, data processing when there is noise, or delays of actuators or sensors when implementing an action. The goal is to obtain an equivalent system, using artificial neural networks, that facilitates the use of data obtained from the system and at the same time provides a robust identifier, capable of withstanding disturbances.

Such identifiers are of greater importance when carrying out experimental tests, as they are able to compensate for system variations, wear, or disturbances when obtaining system data. A key point of the proposal to use neural networks to identify unknown parameters of a system is the training of these networks. In this work, the Extended Kalman Filter (EKF) will be used as training for the different neural networks that will be used, allowing these networks to better handle nonlinearities and enabling online training, which, while requiring greater computational power, rewards this training by being fast, accurate, and robust.

Subsequently, an adaptive PID control is used, which employs a single-layer RHONN to modify the gains of this controller. This represents a significant benefit over a conventional controller, as it will adjust online and also have the robustness that a RHONN acquires when trained by the EKF as indicated above. In this way, an identifier is combined that allows us to obtain an equivalent system, from a nonlinear system with unknown parameters, capable of tolerating disturbances, and also a controller that will constantly adapt

considering its nonlinear and complex system. The previously described techniques are tested with previously instrumented robots, demonstrating the potential of this methodology and how it can simplify practical applications that with conventional control might require a more complex technique.

# Contenido

|                                                                           |      |
|---------------------------------------------------------------------------|------|
| Dedicatoria . . . . .                                                     | III  |
| Lista de Publicaciones . . . . .                                          | V    |
| Resumen . . . . .                                                         | VII  |
| Abstract . . . . .                                                        | IX   |
| Contenido . . . . .                                                       | XI   |
| Lista de Figuras . . . . .                                                | XIII |
| Lista de Tablas . . . . .                                                 | XV   |
| Lista de Símbolos . . . . .                                               | XVII |
| <br>                                                                      |      |
| 1. Introducción . . . . .                                                 | 1    |
| 1.1. Planteamiento del Problema . . . . .                                 | 1    |
| 1.2. Estado del Arte . . . . .                                            | 2    |
| 1.3. Objetivos de la Tesis . . . . .                                      | 7    |
| 1.3.1. Objetivo general . . . . .                                         | 7    |
| 1.3.2. Objetivos particulares . . . . .                                   | 7    |
| 1.4. Justificación . . . . .                                              | 8    |
| 1.5. Hipótesis . . . . .                                                  | 9    |
| 1.6. Metodología . . . . .                                                | 9    |
| 1.7. Descripción de Capítulos . . . . .                                   | 10   |
| <br>                                                                      |      |
| 2. Marco Teórico . . . . .                                                | 13   |
| 2.1. Robots Móviles Diferenciales . . . . .                               | 13   |
| 2.2. Redes Neuronales Artificiales . . . . .                              | 15   |
| 2.2.1. Función de Activación . . . . .                                    | 17   |
| 2.2.2. Clasificación de las Redes Neuronales . . . . .                    | 20   |
| 2.3. Entrenamiento de Redes Neuronales . . . . .                          | 23   |
| 2.3.1. Filtro de Kalman . . . . .                                         | 25   |
| 2.3.2. Filtro Extendido de Kalman . . . . .                               | 29   |
| 2.4. Redes Neuronales Recurrentes de Alto Orden . . . . .                 | 31   |
| 2.4.1. Red neuronal recurrente de alto orden en tiempo discreto . . . . . | 34   |
| 2.5. Entrenamiento de una RHONN por medio del EKF . . . . .               | 36   |
| 2.6. Conclusión de Capítulo . . . . .                                     | 39   |

---

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 3. Modelado e Identificación del Sistema                             | 41 |
| 3.1. Modelo Cinemático . . . . .                                     | 41 |
| 3.2. Diseño de Identificador del Sistema . . . . .                   | 46 |
| 3.2.1. Identificación del Sistema . . . . .                          | 46 |
| 3.2.2. Identificación Neuronal . . . . .                             | 48 |
| 3.2.3. Estructura General . . . . .                                  | 49 |
| 3.2.4. Diseño Implementado . . . . .                                 | 50 |
| 3.3. Conclusión de Capítulo . . . . .                                | 54 |
| 4. Diseño y Aplicación del Control PID Adaptivo                      | 55 |
| 4.1. Diseño de PID Adaptivo del Sistema . . . . .                    | 56 |
| 4.2. Simulación: Identificador Neuronal y Control Neuronal . . . . . | 59 |
| 4.3. Pruebas Experimentales . . . . .                                | 66 |
| 4.3.1. Caso 1: Mayor distancia . . . . .                             | 68 |
| 4.3.2. Caso 2: Distancia intermedia . . . . .                        | 74 |
| 4.3.3. Caso 3: Menor distancia, más precisión . . . . .              | 78 |
| 4.4. Conclusión de Capítulo . . . . .                                | 83 |
| 5. Conclusiones                                                      | 85 |
| 5.1. Conclusiones Generales . . . . .                                | 85 |
| 5.2. Trabajos Futuros . . . . .                                      | 87 |
| Referencias                                                          | 89 |

# Lista de Figuras

|                                                                                                                                                   |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. Modelo Neuronal Artificial . . . . .                                                                                                         | 15 |
| 2.2. Modelo Neuronal Artificial Polarizado . . . . .                                                                                              | 17 |
| 2.3. Función Umbral . . . . .                                                                                                                     | 18 |
| 2.4. Función Sigmoide . . . . .                                                                                                                   | 19 |
| 2.5. Función Tangente Hiperbólica . . . . .                                                                                                       | 20 |
| 2.6. Red Neuronal de una Sola Capa . . . . .                                                                                                      | 21 |
| 2.7. Red Neuronal Multicapa . . . . .                                                                                                             | 22 |
| 2.8. Red Neuronal Recurrente Multicapa . . . . .                                                                                                  | 22 |
| 2.9. Sistema dinámico lineal en tiempo discreto . . . . .                                                                                         | 26 |
| 2.10. RHONN en tiempo discreto . . . . .                                                                                                          | 33 |
| 2.11. RHONN en tiempo continuo . . . . .                                                                                                          | 34 |
| 3.1. Diagrama de cuerpo libre del Robot Móvil Diferencial . . . . .                                                                               | 42 |
| 3.2. Diagrama de cuerpo libre del Robot Móvil Diferencial punto original (c) en el eje de las llantas . . . . .                                   | 43 |
| 3.3. Diagrama de cuerpo libre del Robot Móvil Diferencial punto proyectado (h) . . . . .                                                          | 44 |
| 3.4. Diagrama de cuerpo libre del Robot Móvil Diferencial punto original (c) y punto extendido (h) . . . . .                                      | 45 |
| 3.5. Diagrama del proceso general de la identificación. . . . .                                                                                   | 47 |
| 3.6. Estructura del Identificador Neuronal en tiempo discreto. . . . .                                                                            | 51 |
| 3.8. Gráfica de las posiciones cartesianas comparando el sistema equivalente dado por el identificador contra los datos reales del robot. . . . . | 52 |
| 3.7. Proceso de la Identificación Neuronal. . . . .                                                                                               | 53 |
| 3.9. Gráfica del ángulo comparando el sistema equivalente dado por el identificador contra los datos reales del robot. . . . .                    | 53 |
| 4.1. Esquema del PID adaptivo de una sola neurona para cada una de las variables a controlar. . . . .                                             | 58 |
| 4.2. Esquema del EKF-SNPID o de una sola neurona para cada una de las variables a controlar. . . . .                                              | 60 |
| 4.3. Diagrama de bloques del sistema completo con todas las redes neuronales usadas. . . . .                                                      | 61 |
| 4.4. Identificador y PID adaptivo simulado en la coordenada $x = 5$ , $y = 2$ . . . . .                                                           | 62 |

|                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------|----|
| 4.5. Seguimiento de trayectoria circular donde el robot inicia al centro del círculo en las coordenadas $X$ e $Y$ . . . . . | 62 |
| 4.6. Valores de $X$ e $Y$ a lo largo del tiempo comparados con la trayectoria deseada. . . . .                              | 63 |
| 4.7. Valores de $X$ e $Y$ a lo largo del tiempo comparados con la trayectoria deseada, acercamiento de la recta. . . . .    | 64 |
| 4.8. Error del proceso de identificación neuronal. . . . .                                                                  | 65 |
| 4.9. Error de trayectoria del controlador neuronal. . . . .                                                                 | 65 |
| 4.10. Turtlebot3 <sup>®</sup> Waffle Pi Kit . . . . .                                                                       | 66 |
| 4.11. Model Kit utilizado de un robot móvil diferencial junto a los periféricos conectados. . . . .                         | 68 |
| 4.12. PID adaptivo experimental en la coordenada $x = 3, y = 3$ . . . . .                                                   | 69 |
| 4.13. Identificador y PID adaptivo experimental en la coordenada $x = 3, y = 3$ . . . . .                                   | 70 |
| 4.14. Variables solo PID adaptivo en la coordenada $x = 3, y = 3$ . . . . .                                                 | 70 |
| 4.15. Variables con identificador y PID adaptivo experimental en la coordenada $x = 3, y = 3$ . . . . .                     | 71 |
| 4.16. Ley de control solo PID adaptivo en la coordenada $x = 3, y = 3$ . . . . .                                            | 72 |
| 4.17. Ley de control con identificador y PID adaptivo experimental en la coordenada $x = 3, y = 3$ . . . . .                | 72 |
| 4.18. Error de variables solo PID adaptivo en la coordenada $x = 3, y = 3$ . . . . .                                        | 73 |
| 4.19. Error de variables con identificador y PID adaptivo experimental en la coordenada $x = 3, y = 3$ . . . . .            | 73 |
| 4.20. PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .                                                 | 74 |
| 4.21. Identificador y PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .                                 | 75 |
| 4.22. Ley de control con PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .                              | 75 |
| 4.23. Ley de control con identificador y PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .              | 76 |
| 4.24. Error de variables solo PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .                         | 77 |
| 4.25. Error de variables con identificador y PID adaptivo experimental en la coordenada $x = 2.5, y = 1$ . . . . .          | 77 |
| 4.26. PID adaptivo experimental en la coordenada $x = 1.8, y = 2.1$ . . . . .                                               | 78 |
| 4.27. Identificador y PID adaptivo experimental en la coordenada $x = 1.8, y = 2.1$ . . . . .                               | 79 |
| 4.28. Ley de control solo con PID adaptivo en la coordenada $x = 1.8, y = 2.1$ . . . . .                                    | 79 |
| 4.29. Ley de control con identificador y PID adaptivo experimental en la coordenada $x = 1.8, y = 2.1$ . . . . .            | 80 |
| 4.30. Variable $\theta$ solo con PID adaptivo en la coordenada $x = 1.8, y = 2.1$ . . . . .                                 | 80 |
| 4.31. Variable $\theta$ con identificador y PID adaptivo experimental en la coordenada $x = 1.8, y = 2.1$ . . . . .         | 81 |
| 4.32. Error de variables solo PID adaptivo en la coordenada $x = 1.8, y = 2.1$ . . . . .                                    | 82 |
| 4.33. Error de variables con identificador y PID adaptivo experimental en la coordenada $x = 1.8, y = 2.1$ . . . . .        | 82 |

# Lista de Tablas

|                                                                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. Clasificación de Robots Móviles Terrestres según el Método de Locomoción                                                                                              | 14 |
| 2.2. Clasificación de Robots Móviles Terrestres según la Holonomía . . . . .                                                                                               | 14 |
| 2.3. Métodos de aprendizaje en redes neuronales . . . . .                                                                                                                  | 24 |
| 2.4. Tipos de aprendizaje en redes neuronales . . . . .                                                                                                                    | 24 |
| 2.5. Paradigmas de aprendizaje en redes neuronales . . . . .                                                                                                               | 25 |
| 2.6. Características de las redes neuronales . . . . .                                                                                                                     | 33 |
| 3.1. Comparación de métodos para modelar sistemas. . . . .                                                                                                                 | 46 |
| 4.1. RMS y últimos valores de las variables del error del identificador neuronal y el controlador neuronal. . . . .                                                        | 66 |
| 4.2. Comparación del RMS del error en las variables $X$ y $Y$ entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto $X = 3, Y = 3$ .             | 71 |
| 4.3. Comparación del RMS del error en las variables $X$ e $Y$ entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto $X = 2.5, Y = 1$ .           | 76 |
| 4.4. Comparación del RMS del error en las variables $X$ e $Y$ entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto $X = 1.8, Y = 2.1$ . . . . . | 81 |





# Lista de Símbolos

|                           |                                                                              |
|---------------------------|------------------------------------------------------------------------------|
| $x$                       | Vector de estados del sistema.                                               |
| $\dot{x}(t)$              | Derivada total de una función $x$ dependiente del tiempo.                    |
| $\mathbb{R}^n$            | Espacio vectorial de dimensión $n$ .                                         |
| $\mathbb{R}^m$            | Espacio vectorial de dimensión $m$ .                                         |
| $\mathbb{R}^{3 \times 1}$ | Vector columna de dimensión 3.                                               |
| $\omega$                  | Vector de pesos del sistema.                                                 |
| $e(k)$                    | Error de seguimiento en el instante $k$ .                                    |
| $\eta$                    | Tasa de aprendizaje del Filtro de Kalman Extendido (EKF).                    |
| $u(k)$                    | Señal de control en el instante $k$ .                                        |
| $v(k)$                    | Combinación lineal ponderada de entradas en el instante $k$ .                |
| $x_1(k), x_2(k), x_3(k)$  | Variables de entrada del sistema en el instante $k$ .                        |
| $\alpha$                  | Escalar de saturación en la función de activación.                           |
| $P$                       | Matriz de covarianza del error de estimación.                                |
| $Q$                       | Matriz de covarianza del ruido de proceso.                                   |
| $R$                       | Matriz de covarianza del ruido de medición.                                  |
| $K_p$                     | Ganancia proporcional del PID adaptivo.                                      |
| $K_i$                     | Ganancia integral del PID adaptivo.                                          |
| $K_d$                     | Ganancia derivativa del PID adaptivo.                                        |
| $\eta$                    | Tasa de aprendizaje del Filtro de Kalman Extendido (EKF).                    |
| $u(k)$                    | Señal de control en el instante $k$ .                                        |
| $v(k)$                    | Combinación lineal ponderada de entradas en el instante $k$ .                |
| $x_1(k), x_2(k), x_3(k)$  | Variables de entrada del sistema en el instante $k$ .                        |
| $L$                       | Distancia entre el centro de masas de las llantas (m).                       |
| $h$                       | Distancia del punto $c$ perpendicular al eje de giro de las llantas (m).     |
| $V$                       | Velocidad lineal del robot móvil diferencial (m/s).                          |
| $w$                       | Velocidad angular del robot móvil diferencial (rad/s).                       |
| $\theta$                  | Ángulo de inclinación del robot (rad).                                       |
| $x_c, y_c$                | Posición del punto $c$ en $X$ y $Y$ (m).                                     |
| $x_h, y_h$                | Posición del punto $h$ en $X$ y $Y$ (m).                                     |
| $F_{k+1,k}$               | Matriz de transición de estados de la iteración $k$ a la iteración $k + 1$ . |
| $H(k)$                    | Matriz de medición.                                                          |
| $y(k)$                    | Vector de salida del proceso (medible en la iteración $k$ ).                 |
| $v(k)$                    | Vector del ruido de medición (gaussiano con media cero).                     |

|                  |                                                                         |
|------------------|-------------------------------------------------------------------------|
| $K(k)$           | Matriz de ganancia de Kalman.                                           |
| $\hat{w}(k)$     | Estado estimado en la iteración $k$ .                                   |
| $P(k)$           | Matriz de covarianza del error en la iteración $k$ .                    |
| $z_i$            | Vector de conexiones de alto orden en la red neuronal.                  |
| $\epsilon_{z_i}$ | Aproximación acotada del error en la red neuronal.                      |
| $w_i^*$          | Vector de pesos ideal en la red neuronal.                               |
| $\tilde{w}_i(k)$ | Error de estimación de pesos en la red neuronal.                        |
| $e_i(k)$         | Error de identificación en la red neuronal.                             |
| $\psi(v)$        | Función de activación de la neurona.                                    |
| $\varphi(\cdot)$ | Función de activación no lineal.                                        |
| $b_n$            | Polarización (bias) de la neurona.                                      |
| $v_n$            | Potencial de activación de la neurona.                                  |
| $y_n$            | Señal de salida de la neurona.                                          |
| $w_{nj}$         | Peso sináptico de la neurona $n$ con la neurona $j$ .                   |
| $x_j$            | Señal de entrada a la neurona $j$ .                                     |
| $u_n$            | Combinación lineal de las entradas ponderadas por los pesos sinápticos. |

# Lista de Acrónimos

|       |                                                                                      |
|-------|--------------------------------------------------------------------------------------|
| ANN   | Artificial Neural Network (Red Neuronal Artificial).                                 |
| ARMA  | Autoregressive Moving Average (Modelo Autorregresivo de Media Móvil).                |
| EKF   | Extended Kalman Filter (Filtro Extendido de Kalman).                                 |
| LDS   | Laser Distance Sensor (Sensor de Distancia Láser).                                   |
| LMS   | Least Mean Squares (Mínimos Cuadrados).                                              |
| LQR   | Linear Quadratic Regulator (Regulador Cuadrático Lineal).                            |
| MIMO  | Multiple Input Multiple Output (Múltiples Entradas y Múltiples Salidas).             |
| NARX  | Nonlinear Autoregressive Exogenous Model (Modelo No Lineal Autorregresivo).          |
| PID   | Proportional-Integral-Derivative (Proporcional-Integral-Derivativo).                 |
| SNPID | Single Neuron Proportional-Integral-Derivative (PID de una sola neurona).            |
| RHONN | Recurrent High-Order Neural Network (Red Neuronal Recurrente de Alto Orden).         |
| RLS   | Recursive Least Squares (Mínimos Cuadrados Recursivos).                              |
| RMS   | Root Mean Square (Raíz Cuadrada de la Media).                                        |
| ROS   | Robot Operating System (Sistema Operativo Robótico).                                 |
| SLAM  | Simultaneous Localization and Mapping (Localización y Mapeo Simultáneo).             |
| SGUUB | Semiglobal Uniformly Ultimately Bounded (Acotado Uniformemente en Última Instancia). |



# Capítulo 1

## Introducción

### 1.1. Planteamiento del Problema

En el contexto moderno de sistemas complejos, como los robots de tracción diferencial y cinemática, existen múltiples variables e incertidumbres paramétricas. En estos casos, los modelos clásicos requieren un conocimiento preciso del sistema, además de ser altamente sensibles a errores y perturbaciones externas. Debido a esta sensibilidad, obtener un modelo exacto resulta extremadamente difícil, lo que limita considerablemente la aplicación de técnicas modernas de control y dificulta el diseño de estrategias de control robustas y adaptables [Siwek et al., 2023].

Es necesario diseñar controladores considerando las perturbaciones, ruido en la medición de sensores y errores de modelado. Un modelo matemático del sistema es esencial para el diseño de un controlador adecuado [Rossomando, 2014]. Además, este modelo tiene diversas aplicaciones, incluyendo la identificación de perturbaciones, la simulación y la regulación del propio sistema [Gómez Ávila et al., 2020]. La identificación del sistema es una alternativa útil para obtener un modelo equivalente, en el que las técnicas de identificación neuronal sobresalen.

Por lo general, es necesario tener un conocimiento completo de todas las variables de estado para poder llevar a cabo tareas de control. Sin embargo, no siempre es posible conocer todas las variables de estado; el uso de sensores y observadores es una opción

para obtener estimaciones de esas variables de estado no accesibles. Una de las ventajas de las redes neuronales es su capacidad para modelar sistemas no lineales y complejos, lo que facilita el diseño de un control para dichos sistemas [Akgun et al., 2019]. La implementación de una red neuronal identificadora facilita el empleo de controles clásicos, y el entrenamiento en línea de la red potencia significativamente su capacidad de aprendizaje.

Es necesario mencionar que los controles tradicionales presentan limitaciones al gestionar sistemas complejos y no lineales [Castro-Romero et al., 2024]. En muchos casos, es necesaria una linealización para su manejo efectivo, lo cual implica sacrificios y establece restricciones significativas. Ya que hay fenómenos no lineales que no podrían obtenerse por medio de modelos lineales, por ejemplo, tiempo de escape finito, ciclos límite, múltiples equilibrios aislados, caos, etc.

Este trabajo plantea como solución diseñar una RHONN que pueda generar un modelo equivalente del sistema, basándose en los datos obtenidos de los sensores correspondientes, y a su vez, aplicar un control robusto y adaptable. Asimismo, para el control buscamos aplicar un PID adaptativo usando una RHONN [Hernández-Barragán et al., 2021] de una sola capa, logrando así tener unas ganancias que estarán constantemente adaptándose al comportamiento de la primera red neuronal identificadora. De esta manera, al controlar un sistema con parámetros inciertos y aplicando el Filtro de Kalman Extendido (EKF), se aporta robustez y versatilidad al proceso.

## 1.2. Estado del Arte

En la actualidad, el uso de robots móviles en la industria en tareas que implican periodicidad, desplazamiento, exploración o reconocimiento se ha vuelto una tendencia relevante de investigación. Ya sea para la construcción de robots con capacidad de dar respuestas rápidas en catástrofes o desarrollos en áreas de investigación [Dadvar and Habibian, 2022, Alverhed et al., 2024], como el control, mecatrónica o electrónica. Las técnicas modernas de control han permitido que puedan diversificarse en diferentes tipos y complejidades. Esto ha generado la necesidad de aumentar la robustez [Reis et al., 2022, Zhang et al., 2018] y tolerancias de los controles aplicados en estos, siendo las redes neuronales la alternativa que

se busca desarrollar en este trabajo.

El uso de redes neuronales aplicadas al control y a la identificación de modelos complejos no lineales de parámetros desconocidos es un campo en constante crecimiento con una enorme versatilidad y variedad. Su enfoque dinámico y robusto ha permitido ser una alternativa consistente [Lin et al., 2022], la cual está fundamentada en trabajos de diversas áreas de interés, tanto para la identificación como el control. Como se muestra en [Hernández Barragán et al., 2020] representando las diversas estructuras del PID adaptivo. En [Ríos et al., 2022] se controlan múltiples sistemas por medio de una red compleja. Mostrando el potencial del EKF como un entrenamiento en tiempo real como se muestra en [Hernández-Barragán et al., 2021].

## Modelado y control de sistemas

El modelado de un sistema complejo es un proceso nada trivial. Por ello, existen técnicas de control modernas capaces de sobrellevar perturbaciones, ruido y fallas inesperadas; sin embargo, es innegable que muchas técnicas de control no son capaces de manejar todas las dinámicas reales que un sistema complejo, como los robots móviles, puede tener. Sumado a esto, los medios en sí mismos son condiciones variables difíciles de modelar, lo cual puede generar aún más perturbaciones no consideradas al momento de modelar, en especial cuando siguen trayectorias complejas, como menciona en [Lin et al., 2022]. En [Rios, 2022] se hace mención del uso de un identificador neuronal que utiliza datos previamente adquiridos para la obtención de un modelo que estima el comportamiento de la glucosa e insulina en un paciente diabético por medio de una RHONN, la cual está siendo entrenada con el EKF. A la par, usa otra RHONN para predecir el comportamiento de esta usando los resultados del estimador neuronal, aplicando un control óptimo inverso neurodifuso.

La ventaja clara que demuestran las redes neuronales para la estimación de un modelo basado en datos, tomando en cuenta dinámicas y perturbaciones que un modelado convencional no podría manejar, es evidente en [Gómez Ávila et al., 2020], donde se utiliza el Filtro de Kalman Extendido para entrenar un perceptrón multicapa, logrando que un vehículo no tripulado tenga mayor robustez ante perturbaciones desconocidas. Además, el entrenamiento por EKF ha demostrado una velocidad más alta de aprendi-

zaje y tiempo de convergencia comparada con la retropropagación, como se muestra en [Rios et al., 2020]. Asimismo, en [Akgun et al., 2019] se comparan varios algoritmos de identificación de parámetros contra el Filtro de Kalman Extendido. Aunque este requiere más poder de cómputo, su velocidad, precisión, fidelidad y tasa de convergencia en conjunto superan a otros algoritmos de estimación de parámetros, como los Modelos No Lineales Autorregresivos Exógenos, Mínimos Cuadrados, Mínimos Cuadrados Recursivos y Redes Neuronales Multicapa. Los modelos adquiridos de cada técnica fueron probados a través de un control PI y posteriormente comparados para definir sus características más importantes.

Como antecedente, es pertinente comentar que el Filtro de Kalman Extendido es un algoritmo capaz de estimar en sí mismo, como se muestra en [Deng et al., 2021]. Sin embargo, se necesita un algoritmo que permita optimizar su velocidad y capacidad de convergencia. En este caso, se utilizan los modelos autoregresivos de media móvil, que permiten seleccionar una matriz de covarianza apropiada. Como se mencionó previamente, se usarán redes neuronales para la estimación de parámetros, recalcando el papel fundamental del EKF como método de entrenamiento.

En [Rios et al., 2020], se presenta un identificador neuronal de un tanque robot cuyo modelo matemático es desconocido, lo cual lo convierte en un claro ejemplo del potencial de un estimador de parámetros neuronal. Este se compara con un observador neural que predice su comportamiento, con el propósito de analizar el identificador como un observador con una salida diferente a la de uno convencional. Esto permite comparar los errores entre ellos y concluir que estos pueden disminuir considerablemente, reduciendo también el tiempo de muestreo.

Modelos estimados por redes neuronales para robots móviles diferenciales o cinemáticos, en general, se han convertido en un área de investigación con muchas oportunidades debido a los complejos sistemas que manejan y su vulnerabilidad a perturbaciones. Por ello, controles como el PD pueden combinarse con las RHONN, como se menciona en [Hernández-Barragán et al., 2021]. Si bien las redes neuronales han demostrado su alta capacidad para controlar sistemas no lineales, el enfoque de [Lu et al., 2024] busca compensar las desventajas de un controlador PID convencional.

En [Zhu et al., 2019], se observa una red neuronal recurrente difusa (RFNN por



sus siglas en inglés) entrenada por el Filtro de Kalman Extendido. Al combinar lógica difusa con un modelo de red neuronal, se aumenta el rendimiento para estimar estados del sistema de un robot terrestre autónomo. Es importante recalcar que el EKF es capaz de entrenar a la red neuronal recursiva difusa en línea, optimizando el control sobre este sistema y convirtiéndolo en un sistema robusto y tolerante a fallas. Este entrenamiento en línea permite identificar errores y corregir trayectorias.

Previamente se ha mencionado la capacidad de las redes neuronales para los sistemas no lineales y la obtención de su modelo. De la misma forma, estas son capaces de crear un control tolerante a fallas, como en [Conchas et al., 2023], donde una RHONN se combina con modos deslizantes para crear un controlador robusto de modos deslizantes neuronal. El uso de redes neuronales permite desarrollar un modelo no lineal que optimiza la utilización de modos deslizantes, minimizando así la trayectoria del error.

## Control Neuronal para Robots Móviles

Respecto a las técnicas de control adaptivo, con un enfoque neuronal o haciendo uso de técnicas metaheurísticas, como se menciona en [Lin et al., 2022], estas muestran una tendencia a investigar debido a su versatilidad y su enfoque en generar controles híbridos usando técnicas clásicas y enfoques basados en datos de sistemas complejos. Como se muestra en [Khan et al., 2021], se comparan varios controles convencionales en los cuales es necesario linealizar sistemas con no linealidades, buscando obtener un sistema equivalente linealizado. Las limitantes de una linealización son claras, ya que esta puede implicar sacrificios significativos. Se compararon los controles Proporcional (P), Proporcional-Integral (PI), Proporcional-Integral-Derivativo (PID) y Regulador Cuadrático Lineal (LQR). Los valores de estos controles fueron ajustados experimentalmente en los casos donde fue necesario ajustar ganancias o condiciones iniciales. Los resultados de esa comparativa demostraron que el LQR, para el control de motores de un robot móvil, puede ayudar significativamente con la convergencia y los sobreimpulsos. El mismo trabajo menciona que se recomienda el uso de controles más complejos, como los difusos o híbridos.

Las redes neuronales han tenido diversas aplicaciones cuando se trata del control inteligente aplicado a robots móviles o conjuntos de estos, como se muestra en [Ríos et al., 2022],

donde se utilizan redes complejas mediante el control de anclaje de redes complejas (pinning control por su nombre en inglés), buscando controlar las coordenadas de traslación de diversos robots móviles. Como menciona [Ríos et al., 2022], el control de anclaje implica controlar solo algunos nodos en la red, lo que estabiliza la red completa usando sus ecuaciones cinemáticas para diseñar las ganancias de sus velocidades lineales y angulares.

Como se menciona en [Yousfi Allagui et al., 2021], el PID tiene limitaciones claras cuando se usa en sistemas no lineales. Una de las principales alternativas para compensarlo es hacer las ganancias PID adaptables, como lo hacen en [Yousfi Allagui et al., 2021], utilizando lógica difusa para ajustar los parámetros del controlador y mejorar así la respuesta en velocidad, robustez y adaptabilidad. Este enfoque ofrece un diseño simple pero muy eficiente.

Una de las cualidades de usar redes neuronales para ajustar los valores del control PID aplicado a sistemas no lineales es la capacidad de disminuir el efecto “windup”, como se muestra en [Hernández Barragán et al., 2020], cuando los actuadores se saturan. Este PID adaptativo es entrenado por el EKF, permitiendo un entrenamiento en línea con rápida convergencia comparado con el método de retropropagación. En [Hernández Barragán et al., 2020], se demuestra cómo se puede diseñar un esquema de neurona única para las tres ganancias adaptables, demostrando que un PID adaptativo de una sola neurona puede tener una mejor respuesta que controles PID convencionales o de otra estructura neuronal.

## 1.3. Objetivos de la Tesis

### 1.3.1. Objetivo general

Diseñar e implementar un sistema de control adaptativo, robusto y de rápido aprendizaje para robots móviles diferenciales, mediante el uso de una RHONN combinada con un identificador de sistemas. Este enfoque permitirá obtener un sistema equivalente tolerante a perturbaciones y adaptable en línea, mejorando el desempeño y la autonomía de los robots en entornos dinámicos.

### 1.3.2. Objetivos particulares

- Diseñar una red neuronal identificadora de tipo recurrente de alto orden para robots móviles, así obteniendo un sistema equivalente usando los datos del sistema, facilitando el diseño de un control para dicho modelo.
- Aplicar un control PID adaptativo por medio de una red neuronal RHONN logrando así un control robusto y eficiente.
- Adquisición y gestión de conjuntos de datos procedentes del sistema sometido a diversos puntos de operación.
- Aplicación de identificador neuronal más el controlador adaptivo en un prototipo experimental, previamente instrumentando, utilizando el Sistema Operativo Robótico (ROS por sus siglas en inglés) para una correcta comunicación y control.
- Analizar el comportamiento del identificador para tolerar perturbaciones en conjunto al control PID adaptativo y por separado al identificador.

## Contribuciones de la tesis

1. La tesis aporta al desarrollo de un control robusto y versátil a robots móviles diferenciales. Este control se basa en redes neuronales entrenadas por el EKF, lo que permite una mayor resistencia y adaptabilidad

2. El uso en conjunto de un identificador neuronal junto a un control adaptivo logrando así, usar control en sistemas de parámetros desconocidos y con un enfoque en darle robustez y tolerancia a perturbaciones desconocidas en sistemas no lineales, mediante un entrenamiento en línea.

## 1.4. Justificación

El uso de inteligencia artificial por medio de redes neuronales para la obtención de sistemas robustos y adaptables para el control de robots móviles diferenciales es un campo en continuos desafíos y actualizaciones, en áreas de control, seguimiento de trayectoria, seguridad o dinámicas desconocidas, como menciona [Gómez Ávila et al., 2020]. Un control adaptivo eficiente es crucial para garantizar la seguridad y fiabilidad de los sistemas autónomos. Al aplicar redes neuronales para este propósito, se amplían las áreas de interés, como el envío de paquetes, la agricultura y el monitoreo en general.

Al complementar la red neuronal que ajusta las ganancias del control PID adaptivo previamente mencionada con otra RHONN enfocada en la identificación del modelo desconocido, creando un sistema equivalente basado en parámetros desconocidos, se logra una mayor versatilidad, la cual tiene cualidades muy útiles, especialmente para la robótica móvil, debido a la complejidad de sus condiciones operativas o a las dinámicas desconocidas para un modelado clásico.

Un aspecto vital al diseñar un control robusto y versátil por medio de redes neuronales es el entrenamiento de estas. Por ello, el Filtro Extendido de Kalman, adaptado para el entrenamiento de las redes neuronales, es capaz de considerar las no linealidades del sistema, con la capacidad de ajustarse de manera dinámica a los cambios del sistema, siendo robusto frente a ruidos y perturbaciones. El EKF también puede incorporar información a priori o restricciones sobre los parámetros, lo que mejora el rendimiento y la generalización de la red. Al aplicar el EKF a las redes neuronales de alto orden, tanto para el control como identificación, se obtiene un sistema de control adaptable, que diagnostica y compensa perturbaciones en tiempo real, así como identifica el modelo desconocido del sistema autónomo.

Las pruebas experimentales pueden generar diferentes perturbaciones o retrasos inesperados debido al comportamiento de los actuadores, sensores o por el diseño del sistema en sí mismo. Debido a esto, los controles convencionales requieren consideraciones extra al momento de aplicaciones reales. Una de las ventajas de unir al identificador neuronal con el PID adaptativo es que los ajustes se harán en tiempo real, permitiendo una rápida estabilización del sistema y una gran adaptabilidad.

## 1.5. Hipótesis

La implementación de redes neuronales entrenadas por el Filtro Extendido de Kalman en robots móviles diferenciales y cinemáticos, aumentará significativamente su robustez y tolerancia perturbaciones. Además, la implementación de un control adaptivo ajustando parámetros del control PID nos permite compensar las deficiencias del control PID convencional.

Se espera que el control PID adaptivo, en conjunto con la RHONN identificadora, ofrezca un rendimiento óptimo, proporcionando mejoras significativas en la fiabilidad y seguridad de estos sistemas. Finalmente, se anticipa que la recopilación y análisis de datos obtenidos en pruebas experimentales demuestren que esta combinación de control e identificador neuronal represente una solución altamente eficaz para aplicaciones en sistemas complejos y/o no lineales.

## 1.6. Metodología

El desarrollo de esta tesis consistió en la recolección de datos del sistema en tiempo real, identificando simultáneamente sus variables y parámetros. Estos datos fueron utilizados como entradas para el entrenamiento de la RHONN identificadora. Para garantizar la precisión del modelo, se empleó una red neuronal previamente diseñada en simulaciones, lo que permitió evaluar la viabilidad del sistema equivalente obtenido.

Se implementaron dos tipos de redes neuronales. La primera, una RHONN identificadora, fue entrenada para generar un modelo equivalente del comportamiento del sistema.

La segunda red neuronal se encargó del ajuste en línea de los parámetros del control PID. El sistema equivalente generado por la RHONN identificadora permitió mejorar el desempeño del control, ya que utilizó datos obtenidos directamente del sistema real.

Ambas redes neuronales fueron entrenadas mediante el EKF, lo que facilitó el entrenamiento en línea. Este aspecto resultó fundamental, ya que el ajuste de los parámetros de ambas redes requirió múltiples pruebas debido a la complejidad y no linealidad del sistema, lo que hizo que su configuración no fuera trivial.

Las pruebas se realizaron en modelos previamente instrumentados, los cuales incluían sensores diseñados para recopilar información del sistema, sin necesidad de utilizar parámetros predeterminados. No obstante, en los casos en que un parámetro era conocido, se incorporó en el diseño de la red neuronal para estructurar de manera lógica la relación entre las variables previamente identificadas.

## 1.7. Descripción de Capítulos

Esta tesis está compuesta por seis capítulos. El **Capítulo 1, Introducción**, presenta el problema central del trabajo, justificando la necesidad de desarrollar controladores robustos y adaptativos mediante redes neuronales. En el **Capítulo 2, Marco Teórico**, se expone el marco conceptual relacionado con robots móviles diferenciales y redes neuronales artificiales. Se explican las RHONN, el EKF y su aplicación como método de entrenamiento. También se detallan conceptos fundamentales sobre el modelado y el control de sistemas no lineales.

El **Capítulo 3, Modelado e Identificación del Sistema**, describe el diseño y la implementación de un identificador neuronal basado en una RHONN. Se incluye el modelado cinemático del robot móvil diferencial y el diseño del identificador para obtener un modelo equivalente del sistema en tiempo discreto. En el **Capítulo 4, Diseño y Aplicación del Control PID Adaptivo**, se presenta el diseño del controlador PID adaptativo, empleando una RHONN para ajustar de forma dinámica las ganancias. Se muestran simulaciones y pruebas experimentales para validar el rendimiento del sistema en distintos escenarios, comparando el desempeño del PID adaptivo solo contra el Identificador neural

en conjunto al PID adaptivo, destacando su capacidad de adaptarse a perturbaciones y errores.

El **Capítulo 5, Conclusiones y Trabajos Futuros**, resume las principales contribuciones del trabajo, destacando la efectividad del uso de redes neuronales para el control adaptativo en conjunto con una identificación neuronal como una alternativa robusta y adaptable. Además, se sugieren posibles trabajos futuros para desarrollar el enfoque planteado.





## Capítulo 2

# Marco Teórico

Este capítulo aborda temas específicos, mencionados previamente, incluyendo así una sección sobre la Robótica Móvil Terrestre, la estructura de una Red Neuronal Recurrente de Alto Orden, el Filtro Extendido de Kalman como herramienta de entrenamiento, el diseño de un identificador neuronal y el diseño de una red neuronal capaz de convertir un control PID clásico en un sistema adaptivo.

### 2.1. Robots Móviles Diferenciales

La robótica móvil, como rama de la robótica, ha generado tendencia como área de investigación. En particular, los Robots Móviles Terrestres (RMT) han demostrado adquirir mucha relevancia [[Alverhed et al., 2024](#), [Dadvar and Habibian, 2022](#)], ya que se busca que estos ejecuten tareas más complejas de forma autónoma. También es importante comprender el comportamiento mecánico de estos para diseñar robots móviles apropiados para sus tareas, así como para entender cómo diseñar algoritmos de control adecuados a las demandas de dichos sistemas [[Siegwart et al., 2011](#)].

En este tipo de robots, se pueden utilizar diferentes métodos de locomoción para su navegación, adaptados al entorno y a las características del medio [[Vázquez et al., 2021](#)]. Como se muestra en la Tabla 2.1, se busca clasificar a este tipo de robots por sus categorías, por su método de locomoción y su tipo de holonomía, ayudando así a conocer las capacidades de los robots móviles terrestres, así como sus limitaciones, mediante su método de

desplazamiento y su capacidad de maniobra.

Tabla 2.1: Clasificación de Robots Móviles Terrestres según el Método de Locomoción

| Robots con Ruedas             | Descripción                                                                                                    |
|-------------------------------|----------------------------------------------------------------------------------------------------------------|
| Ackerman                      | Similar al sistema de dirección de un automóvil, con una rueda motriz y otra direccional.                      |
| Triciclo                      | Una rueda direccional y dos ruedas traseras motrices.                                                          |
| Omnidireccional               | Capaces de moverse en cualquier dirección utilizando ruedas especiales.                                        |
| Diferencial                   | Dos ruedas motrices independientes y una o dos ruedas de apoyo, que permiten giros sobre su propio eje.        |
| Robots con Patas              | Simulan la locomoción de animales y pueden moverse en terrenos irregulares.                                    |
| Robots con Orugas (Serpiente) | Utilizan cintas continuas que les permiten moverse sobre terrenos difíciles, como robots de rastreo o tanques. |

Tabla 2.2: Clasificación de Robots Móviles Terrestres según la Holonomía

| Tipo de Robot  | Descripción                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Holonómicos    | Robots que pueden moverse en todas las direcciones en el plano al instante, sin restricciones en su movilidad.                                                     |
| No Holonómicos | Robots cuya movilidad está limitada a ciertas direcciones o trayectorias, lo que implica que no pueden moverse instantáneamente en todas las direcciones posibles. |

Como se menciona previamente en la Tabla 2.2, un robot móvil diferencial es del tipo no holonómico. Posee dos ruedas motrices individuales (un motor por cada una de las ruedas) y una o dos de apoyo, como menciona [Villalobos Álvarez, 2022].

El RMD es comúnmente empleado en ámbitos académicos y de investigación porque se puede controlar a través de su modelo cinemático [Siegwart et al., 2011], es estable y requiere menos componentes para su construcción. Su tracción diferencial le permite girar

sobre su propio eje con menos maniobras en comparación con otras configuraciones que necesitan más piezas. Como menciona [Vázquez et al., 2021], los principales problemas de control en estos son la regulación y el seguimiento de trayectoria en la robótica móvil.

## 2.2. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN, por sus siglas en inglés, Artificial Neural Network) son modelos matemáticos simplificados de las redes neuronales biológicas. Generalmente, se implementan utilizando componentes electrónicos o se simulan con software [Sánchez and Alanis, 2006]. Su objetivo es replicar el comportamiento de una red neuronal biológica, en especial el comportamiento del cerebro humano. Las redes neuronales son sistemas no lineales altamente interconectados que poseen propiedades importantes, lo que las convierte en herramientas muy útiles para resolver problemas complejos en la ingeniería [Rios et al., 2020].

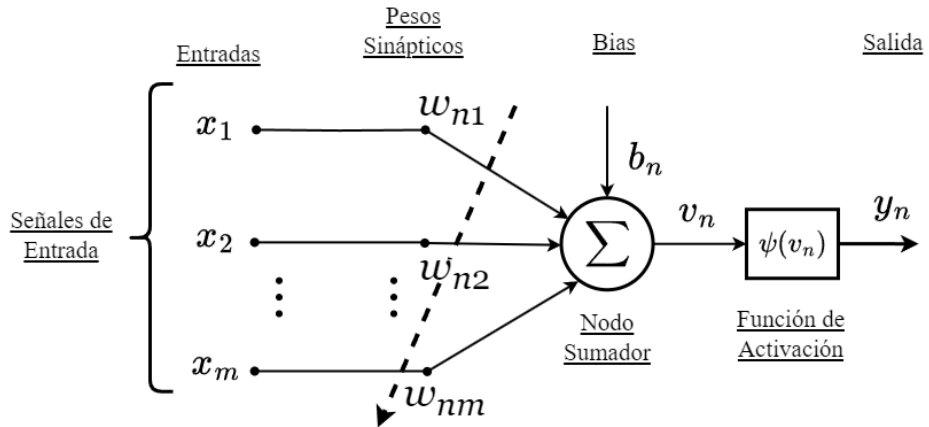


Figura 2.1: Modelo Neuronal Artificial

El diagrama de bloques presentado en la Figura 2.1 muestra las partes que componen una neurona, las cuales son la base de las redes aplicadas en este trabajo. Podemos desglosar los elementos de la siguiente manera [Haykin, 2011, Sánchez and Alanis, 2006]:

- **Enlaces de conexión:** Parametrizados por los pesos sinápticos  $w_{nj}$ . Es importante notar que el primer subíndice corresponde a la neurona receptora, mientras que el

segundo subíndice corresponde a la neurona emisora. Si  $w_{nj} > 0$ , entonces la conexión es excitadora; de igual forma, si  $w_{nj} < 0$ , la conexión es inhibidora.

- **Sumador ( $\psi$ ):** Suma los componentes de las señales de entrada multiplicadas por  $w_{nj}$ .
- **Función de activación ( $\varphi$ ):** Realiza una transformación no lineal.

$$u_n = \sum_{j=1}^m w_{nj}x_j \quad (2.1)$$

y

$$y_n = \psi(u_n + b_n) \quad (2.2)$$

donde  $x_1, x_2, \dots, x_m$  son las señales de entrada;  $w_{n1}, w_{n2}, \dots, w_{nm}$  son los pesos sinápticos de la neurona  $n$ ;  $u_n$  es la combinación lineal de las entradas ponderadas por los pesos sinápticos;  $b_n$  es la polarización;  $\varphi(\bullet)$  es la función de activación; y finalmente,  $y_n$  es la señal de salida de la neurona [Sánchez and Alanis, 2006, Rios et al., 2020].

La polarización es un parámetro externo de la neurona  $n$ , pero es posible considerarla como parte de las señales de entrada. Los modelos de la Figura 2.1 y la Figura 2.2 son matemáticamente equivalentes [Haykin, 2011]. Si se combinan las ecuaciones 2.1 y 2.2, se obtiene:

$$v_n = \sum_{j=0}^m w_{nj}x_j \quad (2.3)$$

y

$$y_n = \psi(v_n) \quad (2.4)$$

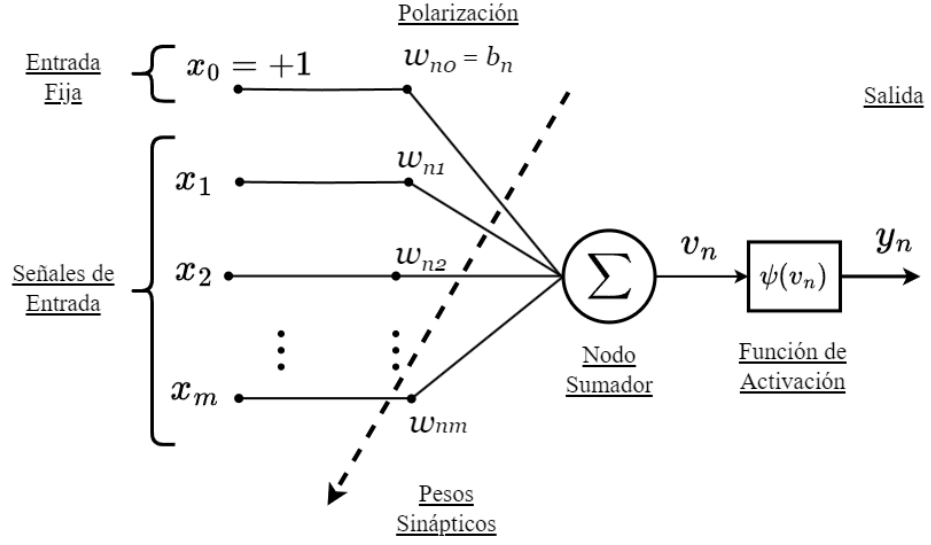


Figura 2.2: Modelo Neuronal Artificial Polarizado

A  $v_n$  se le denomina potencial de activación. En la ecuación 2.3, se ha agregado una nueva sinapsis como se observa en la Figura 2.2. Su entrada es:

$$x_0 = +1 \quad (2.5)$$

y el peso correspondiente es:

$$w_{n0} = b_n \quad (2.6)$$

### 2.2.1. Función de Activación

La función de activación, denotada por  $\psi(v)$ , define la salida de una neurona en función del campo local inducido  $v$  [Haykin, 2011]. A continuación, se identifican dos tipos básicos de funciones de activación:

**1. Función Umbral.** Para este tipo de función de activación, como se muestra en la Figura 2.3, se tiene:

$$\psi(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ 0 & \text{si } v < 0 \end{cases} \quad (2.7)$$

En ingeniería, este tipo de función umbral es conocida como *función de Heaviside*.

La salida de la neurona  $k$  que utiliza esta función se expresa como:

$$y_k = \begin{cases} 1 & \text{si } v_k \geq 0 \\ 0 & \text{si } v_k < 0 \end{cases} \quad (2.8)$$

donde  $v_k$  es el campo local inducido de la neurona, es decir:

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (2.9)$$

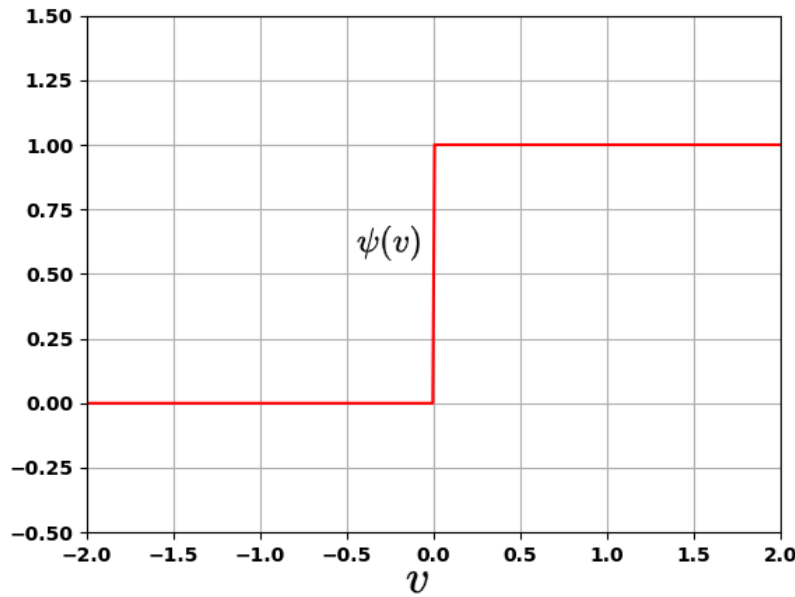


Figura 2.3: Función Umbral

**2. Función Sigmoide.** La función sigmoide, como se muestra en la Figura 2.4, es la forma más común de función de activación utilizada en la construcción de redes neuronales [Haykin, 2011]. Es estrictamente creciente, con un comportamiento asintótico. Un ejemplo es la *función logística*:

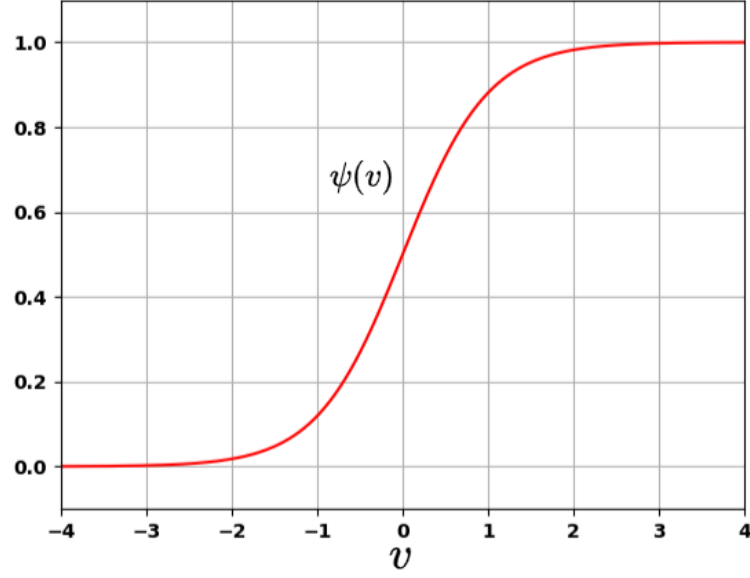


Figura 2.4: Función Sigmoide

$$\psi(v) = \frac{1}{1 + \exp(-av)} \quad (2.10)$$

Las funciones de activación descritas hasta aquí toman valores en el intervalo cerrado  $[0, 1]$ . Sin embargo, también se puede permitir que estas funciones tomen valores en el intervalo cerrado  $[-1, 1]$  [Sánchez and Alanis, 2006, Hernández Barragán et al., 2020].

Para la forma correspondiente de la función sigmoide, se puede usar la función tangente hiperbólica, como se muestra en la Figura 2.5, definida como:

$$\psi(v) = \tanh(v) \quad (2.11)$$

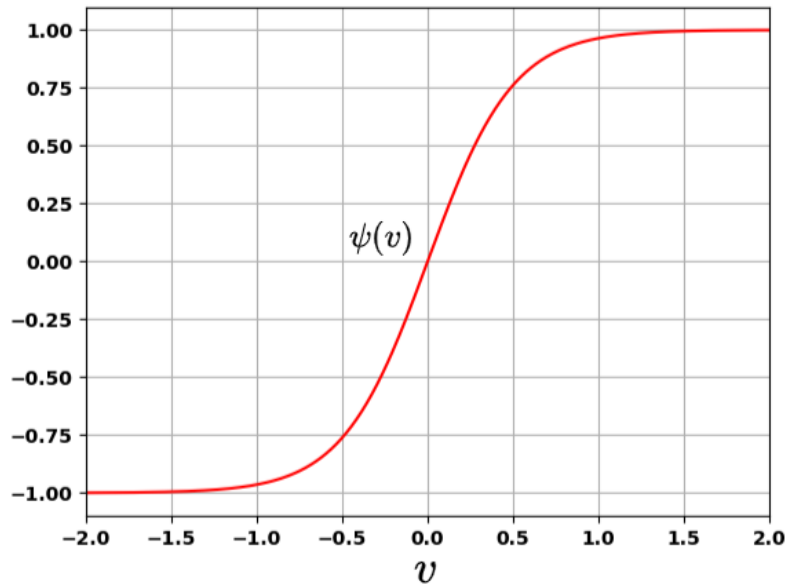


Figura 2.5: Función Tangente Hiperbólica

### 2.2.2. Clasificación de las Redes Neuronales

La manera en que las redes neuronales son entrenadas está íntimamente relacionada con su estructura. Si bien hay diversas estructuras para conectar redes neuronales, podemos identificar tres estructuras fundamentales [Haykin, 2011, Rios et al., 2020]:

- **Redes neuronales unicapa**

En el caso más simple, una red neuronal unicapa tiene la capa de entrada conectada directamente a la capa de neuronas de salida por medio de las sinapsis. Se le da la designación de unicapa porque solo tiene una capa con nodos computacionales; para esta designación no se toma en cuenta la capa que contiene los nodos de entrada, como se muestra en la Figura 2.6.

- **Redes neuronales multicapa**

Las redes neuronales multicapa se distinguen por tener una o más capas de neuronas



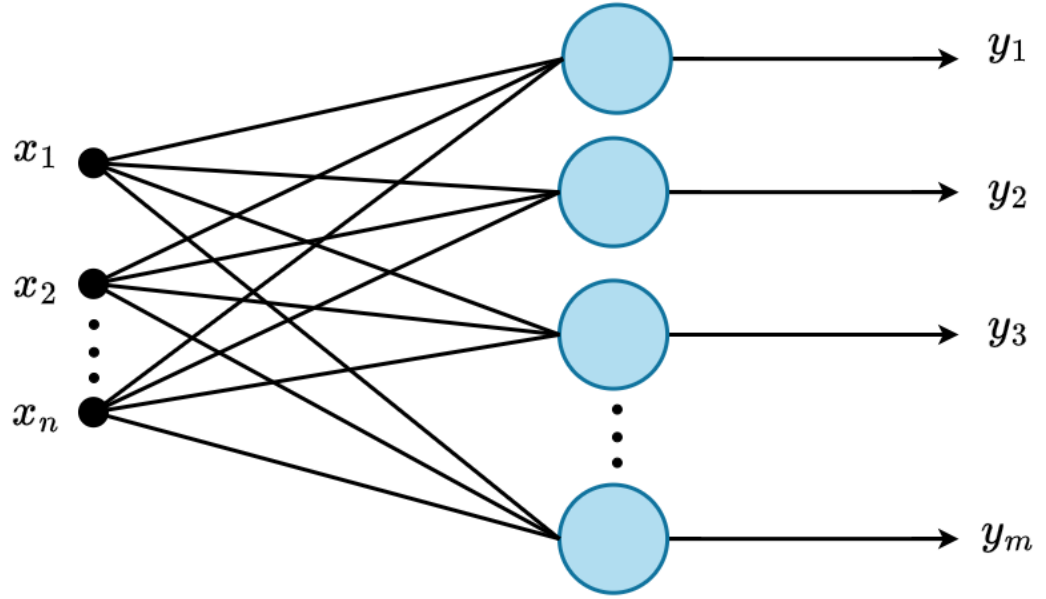


Figura 2.6: Red Neuronal de una Sola Capa

ocultas, cuyos nodos computacionales son llamados neuronas ocultas. La red neuronal mostrada en la Figura 2.7 se dice totalmente conectada, en el sentido de que todos los nodos en cada capa de la red están conectados a todos los otros nodos en la siguiente capa. En el caso de que falte alguna de las sinapsis de la red, entonces se dice que la red es parcialmente conectada [Sánchez and Alanis, 2006].

- **Redes neuronales recurrentes**

Las redes neuronales recurrentes se caracterizan porque al menos tienen un lazo de retroalimentación. La presencia de una estructura recurrente tiene un impacto significativo en la capacidad de aprendizaje y representación de la red neuronal. El uso de estos bucles de retroalimentación implica la utilización de unidades de retardo (denotadas como  $z^{-1}$  en el caso discreto) y unidades de integración (denotadas como  $1/s$  en el caso continuo) [Rios et al., 2020].

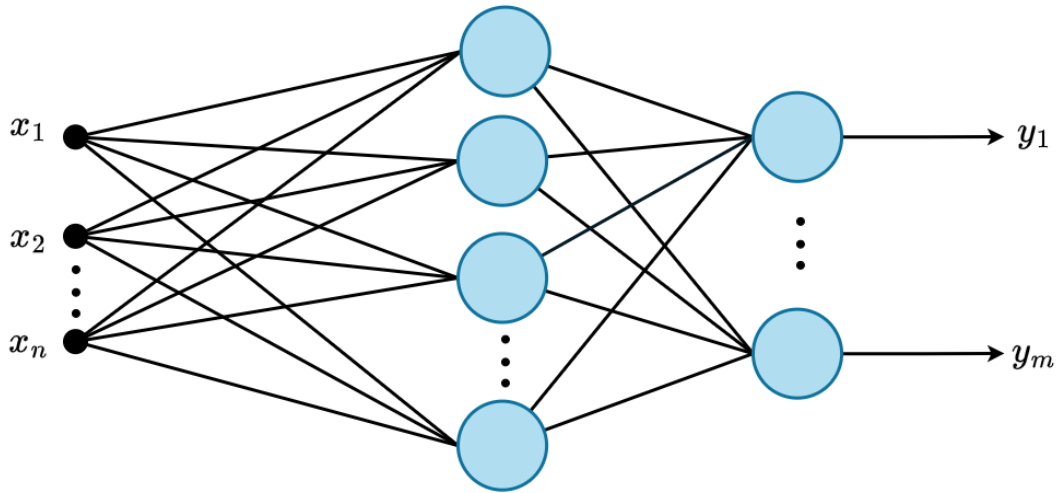


Figura 2.7: Red Neuronal Multicapa

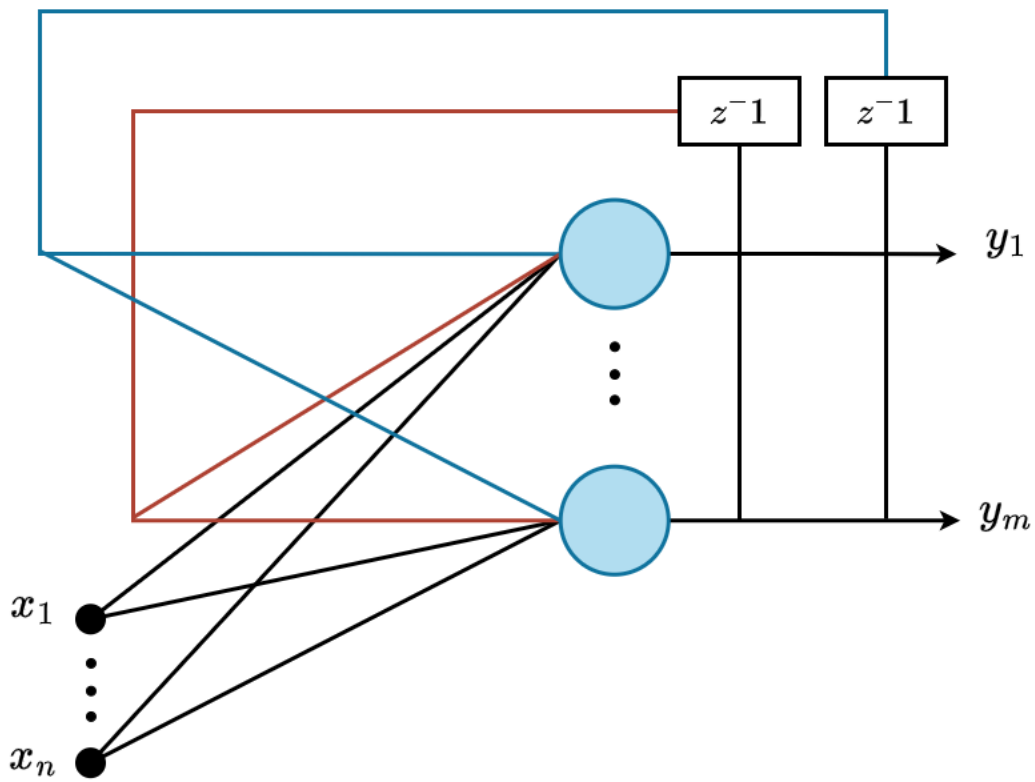


Figura 2.8: Red Neuronal Recurrente Multicapa

## 2.3. Entrenamiento de Redes Neuronales

Uno de los principales paradigmas del aprendizaje de las redes neuronales es el aprendizaje con el medio ambiente y el mejoramiento de su desempeño por medio de este aprendizaje [Haykin, 2011]. Dicho aprendizaje se desarrolla variando los parámetros de la red neuronal, sus pesos sinápticos y el umbral de polarización.

El tipo de aprendizaje se determina por el tipo de adaptación que la red tenga respecto a los datos que obtiene de su medio. Las redes neuronales artificiales son sistemas de aprendizaje basado en datos (ejemplos). La capacidad de una red para resolver un problema está relacionada con los datos adquiridos en el proceso de aprendizaje. El objetivo principal del aprendizaje adecuado para una red neuronal artificial implica determinar los valores de los pesos y bias de todas sus conexiones, lo que permite resolver un problema de manera eficiente. El criterio de convergencia depende del tipo de red utilizada o del tipo de problema a resolver. La finalización del período de aprendizaje se puede determinar [Sánchez and Alanis, 2006]:

- Mediante un número fijo de ciclos.
- Cuando el error descienda por debajo de una cantidad preestablecida.
- Cuando la modificación de los pesos sea irrelevante.
- Por combinaciones de los anteriores.

Los pesos de la red neuronal se ajustan por el mecanismo mostrado en la ecuación 2.12. Los algoritmos de aprendizaje difieren en cómo determinar  $\Delta w_{ij}(k)$ .

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \quad (2.12)$$

Los principales mecanismos de aprendizaje son:

Tabla 2.3: Métodos de aprendizaje en redes neuronales

| Métodos de Aprendizaje      |
|-----------------------------|
| Corrección del error        |
| Boltzmann                   |
| Ley de efectos de Thorndike |
| Regla de Hebb (Delta)       |
| Competitivo                 |

Cabe destacar que estos mecanismos de aprendizaje son conceptos fundamentales para el entrenamiento de redes neuronales y están desarrollados en [Sánchez and Alanis, 2006]. En este trabajo se hará uso de otro tipo de entrenamiento debido a la complejidad del sistema y a sus diversas necesidades, que se abordarán en capítulos posteriores.

El proceso de aprendizaje puede dividirse de la siguiente forma [Rios et al., 2020, Haykin, 2011]:

Tabla 2.4: Tipos de aprendizaje en redes neuronales

| Tipo de Aprendizaje | Descripción                                                                                                                                                                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Offline</b>      | La red neuronal actualiza los pesos en cada paso hasta que se alcanza un estado estable. Una vez que el entrenamiento termina, los pesos permanecen fijos. En este tipo de entrenamiento, la red aprende en un proceso independiente de su aplicación. |
| <b>Online</b>       | El proceso de aprendizaje solo finaliza cuando se detiene la aplicación de la red neuronal. Los pesos en este tipo de aprendizaje varían para adaptarse al problema continuamente.                                                                     |

Tabla 2.5: Paradigmas de aprendizaje en redes neuronales

| Tipo de Aprendizaje   | Descripción                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Supervisado</b>    | El maestro proporciona a la red las respuestas deseadas para las entradas dadas; luego, la red ajusta sus pesos para emular al maestro. De esta manera, el conocimiento del maestro se transfiere a la red a través del entrenamiento y se almacena en los pesos sinápticos.                                                                          |
| <b>No supervisado</b> | La red neuronal puede formar representaciones de las características de las entradas; en consecuencia, crea nuevas clases automáticamente. En este tipo de aprendizaje, se define una medida independiente para evaluar la calidad de la representación que el sistema de aprendizaje adquiere. Los parámetros se ajustan para optimizar esta medida. |
| <b>Reforzamiento</b>  | Se define una función de costo como el costo acumulado de las acciones realizadas en una secuencia de pesos en lugar de un costo inmediato. El sistema opera bajo un reforzamiento positivo o negativo proporcionado por un observador crítico para generar una señal heurística de reforzamiento.                                                    |

La forma en que la red interactúa con su medio se considera un paradigma. Los principales paradigmas se presentan en la Tabla 2.5 [Rios et al., 2020, Haykin, 2011].

### 2.3.1. Filtro de Kalman

El filtro de Kalman básico es un sistema lineal, en tiempo discreto y de dimensión finita, que posee una estructura recursiva adecuada para su implementación en una computadora digital. La solución es recursiva, lo que significa que cada actualización del estado estimado se calcula a partir del estimado anterior y del nuevo dato de entrada. Por lo tanto, solo es necesario almacenar el estimado anterior y no todos los datos históricos. Una propiedad clave del filtro de Kalman es que actúa como el estimador de mínima va-

rianza en el sentido de mínimos cuadrados para el estado de un sistema dinámico lineal [Sánchez and Alanis, 2006].

Se considera el sistema dinámico lineal en tiempo discreto, mostrado en la Figura 2.9, como se ilustra en [Sánchez and Alanis, 2006]. El concepto de *estado* es fundamental en esta descripción. El *vector de estado*, o simplemente *estado*, denotado por  $w(k)$ , se define como el conjunto mínimo de datos necesarios para describir el comportamiento dinámico del sistema. El subíndice  $k$  indica el instante de muestreo.

En otras palabras, el estado representa la cantidad mínima de información del comportamiento pasado del sistema necesaria para *predecir* su comportamiento futuro [Haykin, 2011, Sánchez and Alanis, 2006]. Típicamente, el vector de estados  $w(k)$  es *desconocido*. Para estimarlo, se utiliza un conjunto de datos *medidos*, denotados por  $y(k)$ .

El diagrama de bloques presentado en la Figura 2.9 ilustra este sistema dinámico y puede describirse matemáticamente mediante el siguiente par de ecuaciones:

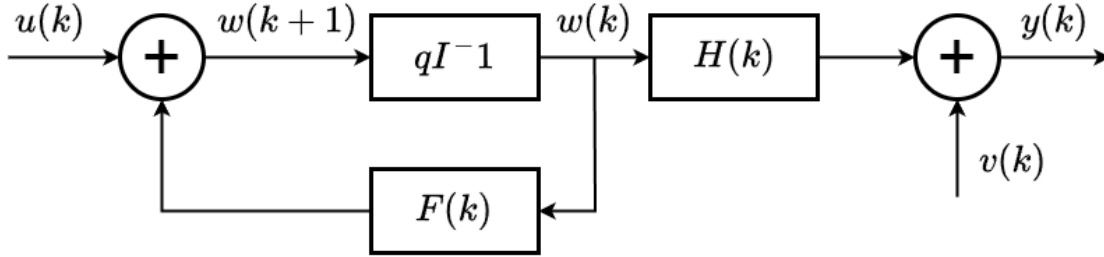


Figura 2.9: Sistema dinámico lineal en tiempo discreto

### Ecuación del proceso

$$w(k+1) = F_{k+1,k}w(k) + u(k) \quad (2.13)$$

donde:  $F_{k+1,k} \in \mathbb{R}^{L \times L}$  es la matriz de transición de estados de la iteración  $k$  a la iteración  $k+1$ ;  $w(k) \in \mathbb{R}^L$  es el vector de estados en la iteración  $k$ ;  $u(k) \in \mathbb{R}^L$  representa el vector del ruido del proceso (**gaussiano con media cero**).

La matriz de covarianza del ruido de proceso se define como:

$$E\{u(n)u^T(k)\} = \begin{cases} Q(k) & \text{para } n = k \\ 0 & \text{para } n \neq k \end{cases} \quad (2.14)$$

El operador  $E\{\cdot\}$  representa el **valor esperado**. La variable  $u(n)$  representa el ruido del proceso en el instante  $n$ . Por su parte,  $u^T(k)$  es la transpuesta de  $u(k)$ , utilizada para formar la matriz de covarianza. Finalmente,  $Q(k)$  es la **matriz de covarianza del ruido del proceso**, que modela la incertidumbre del sistema en el tiempo de muestreo  $k$ .

### Ecuación de la salida (medición)

$$y(k) = H(k)w(k) + v(k) \quad (2.15)$$

donde:  $H \in \mathbb{R}^{m \times L}$  es la matriz de medición;  $y \in \mathbb{R}^m$  es el vector de salida del proceso (medible en la iteración  $k$ );  $v \in \mathbb{R}^m$  representa el vector del ruido de medición (**gaussiano con media cero**).

La matriz de covarianza del ruido de medición se define como [Haykin, 2004]:

$$E\{v(n)v^T(k)\} = \begin{cases} R(k) & \text{para } n = k \\ 0 & \text{para } n \neq k \end{cases} \quad (2.16)$$

El operador  $E\{\cdot\}$  representa el **valor esperado**. La variable  $v(n)$  es un **vector aleatorio** que representa el ruido de medición en el instante  $n$ . Por su parte,  $v^T(k)$  es la transpuesta de  $v(k)$ , utilizada para formar la matriz de covarianza. Finalmente,  $R(k)$  es la **matriz de covarianza del ruido de medición**, que modela la incertidumbre asociada a las observaciones del sistema en el tiempo de muestreo  $k$ . El ruido de medición  $v(k)$  está descorrelacionado del ruido del proceso  $u(k)$ .

Entonces, el vector de estados  $w(k)$  del sistema lineal descrito por 2.13 y 2.15 se puede estimar con el filtro de Kalman, el cual queda definido por las siguientes ecuaciones:

*Propagación del estado estimado*

$$\hat{w}^-(k) = F_{k+1,k}\hat{w}^-(k-1) \quad (2.17)$$

*Propagación de la covarianza del error*

$$P^-(k) = F_{k+1,k}P(k-1)F^T(k, k-1) + Q(k-1) \quad (2.18)$$

*Matriz de ganancia de Kalman*

$$K(k) = P^-(k)H^T(k) [H(k)P^-(k)H^T(k) + R(k)]^{-1} \quad (2.19)$$

*Actualización del estado estimado*

$$\hat{w}(k) = \hat{w}^-(k) + K(k) (y(k) - H(k)\hat{w}^-(k)) \quad (2.20)$$

*Actualización de la covarianza del error*

$$P(k) = (I - K(k)H(k)) P^-(k) \quad (2.21)$$

*Inicialización para  $k = 0$ :*

$$\hat{w}(0) = E\{w(0)\} \quad (2.22)$$

$$P(0) = E\left\{(w(0) - E\{w(0)\})(w(0) - E\{w(0)\})^T\right\} \quad (2.23)$$

El filtro de Kalman es el mejor estimador lineal si  $w(0)$ ,  $u(k)$ ,  $v(k)$  tienen características estadísticas arbitrarias. Si  $w(0)$ ,  $u(k)$ ,  $v(k)$  son variables aleatorias normales, entonces el filtro de Kalman es el estimador óptimo de  $w(k)$  [Sánchez and Alanis, 2006].

La aplicación del filtro de Kalman para entrenar redes neuronales distribuye los pesos sinápticos en los estados que el filtro de Kalman estima, y la salida de la red es la medición usada por el filtro de Kalman. Del planteamiento previamente mencionado surge la alternativa de visualizar el entrenamiento de redes neuronales como un problema de filtrado óptimo. Las aplicaciones de la teoría del filtro de Kalman pueden extenderse a sistemas dinámicos no lineales [Haykin, 2011].

- La ecuación del proceso que define la evolución del estado en el tiempo.
- La ecuación de salida (medida) que define las variables observables en función del estado.



*El modelo es estocástico debido a la presencia aditiva de ruido de proceso y ruido de medición, los cuales se asumen como gaussianos con media cero y matrices de covarianza conocidas [Haykin, 2004].*

### 2.3.2. Filtro Extendido de Kalman

El problema de filtrado de Kalman discutido se ha centrado en la estimación de un vector de estado en un modelo lineal de un sistema dinámico. No obstante, si el modelo es no lineal, es posible adaptar el enfoque del filtrado de Kalman aplicando un procedimiento de linealización. Este tipo de filtro adaptado se denomina filtro de Kalman extendido (EKF) [Haykin, 2004].

Considérese un sistema dinámico no lineal descrito por el siguiente modelo en espacio de estado.

$$w(k+1) = f(k, w(k)) + u(k) \quad (2.24)$$

$$y(k) = h(k, w(k)) + v(k) \quad (2.25)$$

Como antes,  $u(k)$  y  $v(k)$  son ruidos independientes, gaussianos con matrices de covarianza  $R(k)$  y  $Q(k)$ , respectivamente.  $f(k, w(k))$  denota la función matricial no lineal de transición, posiblemente variante en el tiempo, y  $h(k, w(k))$  denota la función matricial de medición no lineal que también puede ser variante en el tiempo [Sánchez and Alanis, 2006].

El principio fundamental del Filtro de Kalman Extendido consiste en linearizar el modelo en espacio de estado de las ecuaciones 2.24 y 2.25 en cada instante de tiempo, tomando como referencia el estado estimado más reciente. Una vez que se obtiene el modelo lineal, se aplican las ecuaciones del Filtro de Kalman [Alanis et al., 2020, Haykin, 2004].

Más explícitamente, la aproximación procede en dos pasos:

**Paso 1.** Se construyen las siguientes matrices:

$$F_{k+1,k} = \left. \frac{\partial f(k, w(k))}{\partial w} \right|_{w=\hat{w}(k)} \quad (2.26)$$

$$H(k) = \left. \frac{\partial h(k, w(k))}{\partial w} \right|_{w=\hat{w}^-(k)} \quad (2.27)$$

**Paso 2.** Una vez que las matrices  $F(\bullet)$  y  $H(\bullet)$  son evaluadas, se emplean en una aproximación en series de Taylor de primer orden para las funciones no lineales  $f(k, w(k))$  y  $h(k, w(k))$  alrededor de  $\hat{w}(k)$  y  $\hat{w}^-(k)$ , respectivamente. Específicamente, se aproximan como sigue [Sánchez and Alanis, 2006, Rios et al., 2020]:

$$f(k, w(k)) \approx F(w, \hat{w}(k)) + F_{k+1,k} w(k, \hat{w}(k)) \quad (2.28)$$

$$h(k, w(k)) \approx H(k, \hat{w}^-(k)) + H_{k+1,k} w(k, \hat{w}^-(k)) \quad (2.29)$$

En base a las ecuaciones 2.28 y 2.29, se aproximan las ecuaciones de estado no lineales 2.24 y 2.25 como se muestra:

$$w(k+1) \approx F_{k+1,k} w(k) + u(k) + d(k) \quad (2.30)$$

$$\bar{y}(k) \approx H(k) w(k) + v(k) \quad (2.31)$$

Ahora introduciremos dos nuevas variables:

$$\bar{y}(k) = y(k) - \{h(w, \hat{w}^-(k)) - H(k) \hat{w}^-(k)\} \quad (2.32)$$

$$d(k) = f(w, \hat{w}(k)) - F_{k+1,k} \hat{w}(k) \quad (2.33)$$

Dadas las ecuaciones del modelo en espacio de estado linealizadas, se procede a realizar el mismo procedimiento que en el caso del filtro de Kalman [Haykin, 2004].

El vector de estados  $w(k)$  del sistema no lineal, descrito por las ecuaciones 2.24 y 2.25, puede estimarse utilizando el filtro de Kalman. Este proceso considera  $u(k)$  y  $v(k)$  como ruidos independientes y gaussianos, con matrices de covarianza  $Q(k)$  y  $R(k)$ , respectivamente, y se define mediante las siguientes ecuaciones:

*Propagación del estado estimado*

$$\hat{w}^-(k) = f(k, \hat{w}(k-1)) \quad (2.34)$$

*Propagación de la covarianza del error*

$$P^-(k) = F_{k,k-1} P(k-1) F_{k,k-1}^T + Q(k-1) \quad (2.35)$$

*Matriz de la ganancia de Kalman*

$$K(k) = P^-(k)H^T(k) [H(k)P^-(k)H^T(k) + R(k)]^{-1} \quad (8.22)$$

*Actualización del estado estimado*

$$\hat{w}(k) = \hat{w}^-(k) + K(k) [y(k) - h(k, \hat{w}^-(k))] \quad (2.36)$$

*Actualización de la covarianza del error*

$$P(k) = [I - K(k)H(k)] P^-(k) \quad (2.37)$$

donde:

$$F_{k+1,k} = \left. \frac{\partial f(k, w(k))}{\partial w} \right|_{w=\hat{w}(k)} \quad (2.38)$$

$$H(k) = \left. \frac{\partial h(k, w(k))}{\partial w} \right|_{w=\hat{w}^-(k)} \quad (2.39)$$

inicializando con:

$$\hat{w}(0) = E\{w(0)\} \quad (2.40)$$

$$P(0) = E \left\{ (w(0) - E\{w(0)\}) (w(0) - E\{w(0)\})^T \right\} \quad (2.41)$$

La matriz de covarianza  $P(0)$  representa la incertidumbre en la estimación del estado inicial. Aquí,  $w(0) - E\{w(0)\}$  es el error entre el estado real y su estimación, y  $E\{\cdot\}$  denota el valor esperado. La matriz de covarianza  $P(0)$  es una medida de cuánto se desvía el estado real de su estimación inicial.

## 2.4. Redes Neuronales Recurrentes de Alto Orden

Considere el siguiente sistema no lineal en tiempo discreto con retardos desconocidos de múltiples entradas y múltiples salidas:

$$\begin{aligned} x(k+1) &= F(x(k-l), u(k)) + d(k), \\ y(k) &= Cx(k), \end{aligned} \quad (2.42)$$

donde  $x \in \mathbb{R}^n$  es el vector de estado del sistema,  $u \in \mathbb{R}^m$  es el vector de entrada,  $y \in \mathbb{R}^p$  es el vector de salida,  $C \in \mathbb{R}^{p \times n}$  es la matriz de salida,  $d \in \mathbb{R}^n$  es el vector de perturbaciones,  $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  es una función no lineal, y  $l = 1, 2, \dots$  es el retardo desconocido [Arana-Daniel et al., 2020].

Las redes neuronales recurrentes de alto orden (RHONN) son redes neuronales en tiempo discreto que se han utilizado ampliamente para modelar e identificar sistemas dinámicos en problemas de control [Alanis et al., 2020]. Estas redes son capaces de aproximar el comportamiento de algunos sistemas dinámicos, especialmente si cuentan con suficientes conexiones de alto orden, permitiendo la existencia de un vector de pesos tal que el comportamiento de la RHONN aproxima cualquier sistema dinámico cuya trayectoria de estado permanezca en un conjunto compacto [Arana-Daniel et al., 2020].

Además, las RHONN destacan por su capacidad para capturar la respuesta dinámica de sistemas no lineales complejos [Rovithakis and Christodoulou, 2012] y de sistemas con retardo temporal. Esto se debe a sus características, como un modelo flexible que facilita la incorporación de información a priori sobre el sistema a identificar, así como sus capacidades de aproximación, robustez frente al ruido, entrenamiento en línea y comportamiento dinámico, producto de sus conexiones recurrentes. Estas propiedades mejoran sus capacidades de aprendizaje y rendimiento en comparación con redes neuronales convencionales [Arana-Daniel et al., 2020, Rovithakis and Christodoulou, 2012].

Entre las principales características de las RHONN se encuentran [Rios et al., 2020]:

Las RHONN en tiempo discreto Figura 2.10 presentan las mismas características que las continuas Figura 2.11. Además, son ideales para el modelado, identificación y control de sistemas dinámicos no lineales complejos discretos [Alanis et al., 2020]. Presentadas a continuación en la Tabla 2.6

Tabla 2.6: Características de las redes neuronales

| N° | Característica                                                                              |
|----|---------------------------------------------------------------------------------------------|
| 1  | Permiten el modelado eficiente de sistemas dinámicos complejos.                             |
| 2  | Son buenos candidatos para la identificación de sistemas, observadores de estado y control. |
| 3  | Son robustas contra el ruido.                                                               |
| 4  | Son fáciles de implementar.                                                                 |
| 5  | Su estructura es relativamente simple.                                                      |
| 6  | Se puede incorporar información a priori del sistema en el modelo de la red.                |
| 7  | Los parámetros pueden ajustarse con entrenamiento en línea o fuera de línea.                |

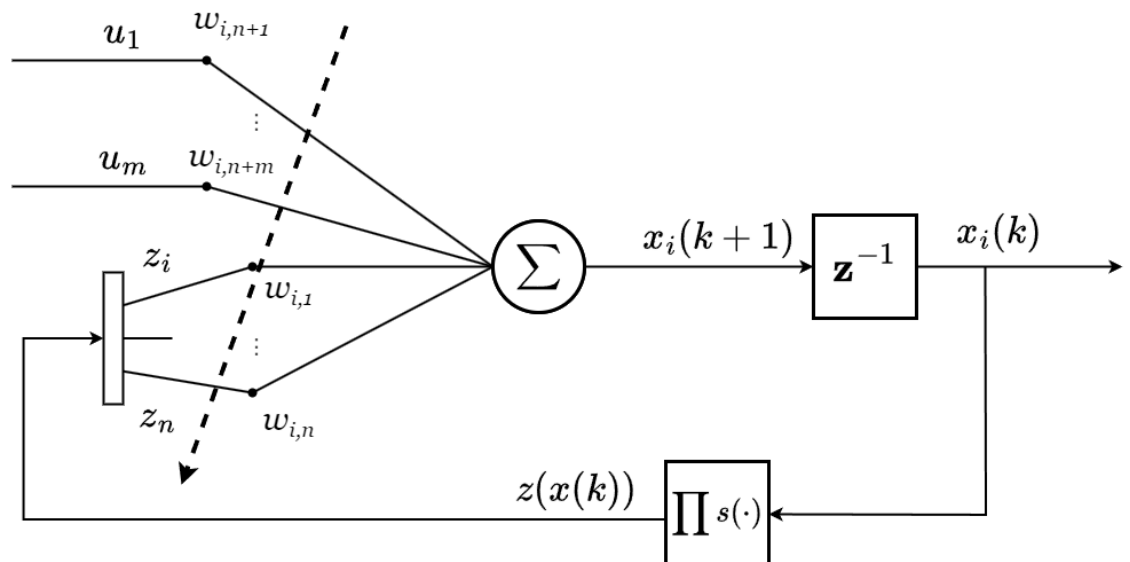


Figura 2.10: RHONN en tiempo discreto

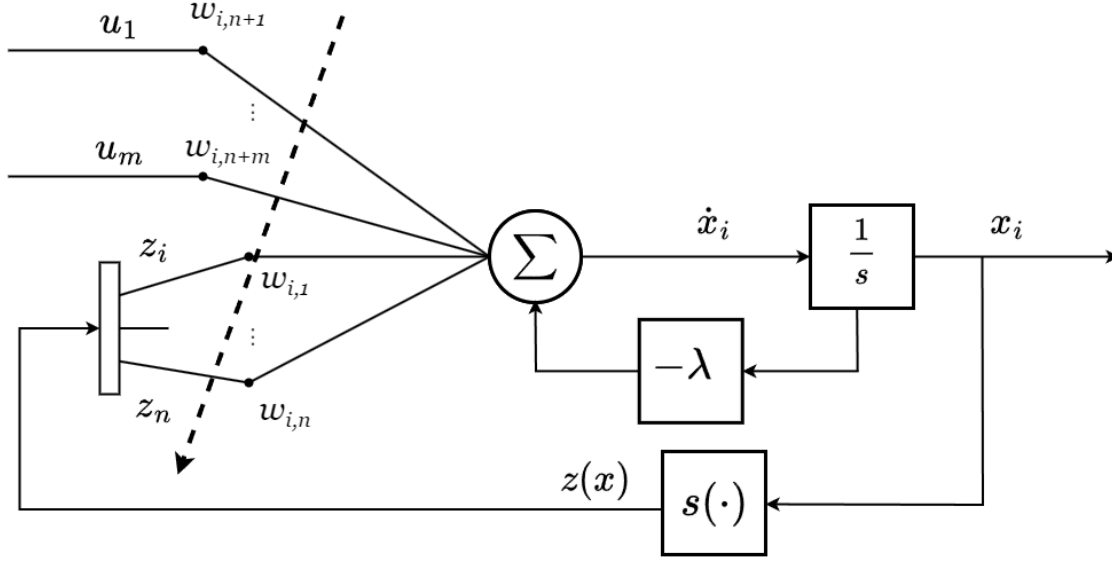


Figura 2.11: RHONN en tiempo continuo

### 2.4.1. Red neuronal recurrente de alto orden en tiempo discreto

Existen dos modelos de RHONNs [Arana-Daniel et al., 2020, Rios et al., 2020]:

- Modelo paralelo,

$$\hat{x}_i(k+1) = w_i^T z_i(\hat{x}(k), u(k)). \quad (2.43)$$

- Modelo serie-paralelo,

$$\hat{x}_i(k+1) = w_i^T z_i(x(k), u(k)), \quad (2.44)$$

donde  $n$  es la dimensión del estado,  $i = 1, \dots, n$ ,  $\hat{x}$  es el vector de estado de la red neuronal,  $w$  es el vector de pesos,  $x$  es el vector de estado de la planta, y  $u = [u_1, u_2, \dots, u_m]^T$  es el vector de entrada de la red neuronal [Rovithakis and Christodoulou, 2012].

Para el modelo serie-paralelo,  $z_i(\cdot)$  se define de la siguiente manera:

$$z_i(x(k), u(k)) = \begin{bmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{iL_i} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} \xi_{ij}^{d_{ij}(1)} \\ \prod_{j \in I_2} \xi_{ij}^{d_{ij}(2)} \\ \vdots \\ \prod_{j \in I_{L_i}} \xi_{ij}^{d_{ij}(L_i)} \end{bmatrix}, \quad (2.45)$$

Este vector representa la salida de la  $i$ -ésima neurona en la red. Cada componente  $z_{il}$  (donde  $l = 1, 2, \dots, L_i$ ) es un producto de términos elevados a potencias.

$$z_{il} = \prod_{j \in I_l} \xi_{ij}^{d_{ij}(l)} \quad (2.46)$$

Cada componente  $z_{il}$  es un producto de términos  $\xi_{ij}$  elevados a potencias  $d_{ij}(l)$ . Aquí,  $\xi_{ij}$  representa las entradas a la neurona, que pueden ser estados  $x(k)$  o entradas de control  $u(k)$ .

$$\{I_1, I_2, \dots, I_{L_i}\} \quad (2.47)$$

Estos son subconjuntos no ordenados de índices que indican qué entradas están involucradas en cada producto. Los índices pueden referirse a estados o entradas de control.

$$d_{ij}(l) \quad (2.48)$$

Estos son enteros no negativos que representan las potencias a las que se elevan las entradas  $\xi_{ij}$  en cada producto.

$$L_i \quad (2.49)$$

Este es el número de términos de producto (conexiones de alto orden) para la  $i$ -ésima neurona.

Donde  $L_i$  es el respectivo número de conexiones de alto orden,  $\{I_1, I_2, \dots, I_{L_i}\}$  es una colección de subconjuntos no ordenados de  $\{1, 2, \dots, n + m\}$ ,  $d_{ij}(k)$  son enteros no negativos, y  $\xi_i$  se define como:

$$\xi_i = \begin{bmatrix} \xi_{i1} \\ \xi_{i2} \\ \vdots \\ \xi_{i(n+m)} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad (2.50)$$

y  $S(\cdot)$  se define como cualquier función sigmoide

$$S(\varsigma) = \frac{1}{1 + \exp(-\beta\varsigma)}, \quad \beta > 0, \quad (2.51)$$

donde  $\varsigma$  es una variable con cualquier valor real.

Para el modelo paralelo,  $z_i(\cdot)$  es 2.45 con la modificación de cambiar  $x$  a  $\hat{x}$ , es decir [Rios et al., 2020],

$$\xi_i = \begin{bmatrix} \xi_{i1} \\ \vdots \\ \xi_{in} \\ \xi_{in+1} \\ \vdots \\ \xi_{i(n+m)} \end{bmatrix} = \begin{bmatrix} S(\hat{x}_1) \\ \vdots \\ S(\hat{x}_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix}. \quad (2.52)$$

## 2.5. Entrenamiento de una RHONN por medio del EKF

El Filtro de Kalman estima el estado de un sistema lineal con ruido blanco aditivo, tanto en el estado como en la salida, utilizando una solución recursiva en la cual el estado estimado se calcula a partir del estado anterior y la entrada actual [Rios et al., 2020]. El EKF es un método efectivo y eficiente para el entrenamiento de una RHONN [Alanis et al., 2020]. Este enfoque no sólo ofrece una rápida convergencia, sino que también supera al filtro de Kalman tradicional en su capacidad para manejar sistemas no lineales.

En este proceso, el EKF estima los pesos de la red neuronal, que representan el estado del sistema, y considera el error entre la salida medida de la planta y la salida de la red neuronal como ruido blanco aditivo, mejorando así la precisión y robustez del modelo [Arana-Daniel et al., 2020].

El EKF es la base de un método de entrenamiento de redes neuronales de segundo orden que ofrece una alternativa práctica y eficiente a los métodos de segundo orden por lotes. Este enfoque recursivo del EKF permite que, durante el proceso de entrenamiento, los pesos de la red se actualicen de manera secuencial en lugar de por lotes. Además, se mantiene y ajusta una matriz de covarianza de error aproximada, que proporciona información de segundo orden relevante para el problema de entrenamiento [Haykin, 2004].



El objetivo del entrenamiento es encontrar el vector de pesos óptimo que minimice el error de predicción. El EKF se utiliza porque la relación de mapeo de la red neuronal es no lineal [Alanis et al., 2020, Haykin, 2004].

El algoritmo de entrenamiento para una RHONN basado en el EKF utilizado en este trabajo se basa en el modelo serie-paralelo 2.44 y se define de la siguiente manera:

$$\begin{aligned}\omega_i(k+1) &= \omega_i(k) + \eta_i K_i(k) e_i(k), \\ K_i(k) &= P_i(k) H_i(k) M_i(k), \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^T(k) P_i(k) + Q_i(k),\end{aligned}\tag{2.53}$$

con

$$M_i(k) = [R_i(k) + H_i^T(k) P_i(k) H_i(k)]^{-1},\tag{2.54}$$

$$e_i(k) = x_i(k) - \hat{x}_i(k),\tag{2.55}$$

$$H_{ij} = \left[ \frac{\partial \hat{x}_i(k)}{\partial \omega_{ij}(k)} \right]^T,\tag{2.56}$$

donde  $e_i \in \mathbb{R}$  es el error de identificación,  $\eta \in \mathbb{R}$  es el factor de aprendizaje del filtro de Kalman. El factor de aprendizaje  $\eta_t$  determina qué tan rápido o lento se ajusta el estado estimado  $\omega_i(k)$  en respuesta al error  $e_i(k)$ .

Si  $\eta$  es grande, el filtro realizará ajustes más grandes en cada iteración, lo que puede acelerar la convergencia pero también puede introducir inestabilidad.

Si  $\eta$  es pequeño, el filtro realizará ajustes más pequeños, lo que puede ralentizar la convergencia pero proporcionará una estimación más estable.

$P_i \in \mathbb{R}^{L_i \times L_i}$  es la matriz de covarianza de la predicción del error,  $\omega_i \in \mathbb{R}^{L_i}$  es el vector de pesos adaptado en línea,  $\hat{x}_i$  es la  $i$ -ésima variable de estado de la red neuronal,  $K_i \in \mathbb{R}^{L_i}$  es el vector de ganancia de Kalman,  $Q_i \in \mathbb{R}^{L_i \times L_i}$  es la matriz de covarianza del ruido de estimación,  $R_i \in \mathbb{R}$  es la matriz de covarianza del ruido del error de salida, y  $H_i \in \mathbb{R}^{L_i}$  es un vector en el cual cada entrada se define como  $H_{ij}$ , que es la derivada del

estado de la red neuronal ( $\hat{x}_i$ ) respecto a un peso de la red neuronal ( $\omega_{ij}$ ) y se define en 2.56, donde  $i = 1, \dots, n$  y  $j = 1, \dots, L_i$  [Rios et al., 2020].

En el caso del modelo paralelo 2.43, el entrenamiento es una modificación de 2.53 con las siguientes modificaciones:  $e_i \in \mathbb{R}^p$ ,  $R_i \in \mathbb{R}^{p \times p}$ , y  $H_i \in \mathbb{R}^{L_i \times p}$ , donde  $e$  y  $H_{ij}$  se definen como [Rios et al., 2020]:

$$e_i(k) = y_i(k) - \hat{y}_i(k), \quad (2.57)$$

$$H_{ij} = \left[ \frac{\partial \hat{y}_i(k)}{\partial \omega_{ij}(k)} \right]^T. \quad (2.58)$$

Por la regla de la cadena obtenemos

$$\frac{\partial \hat{y}(k)}{\partial \omega(k)} = \frac{\partial \hat{y}(k)}{\partial x(k)} \frac{\partial \hat{x}(k)}{\partial \omega(k)}. \quad (2.59)$$

Reescribiendo 2.43 como

$$\hat{x}(k+1) = F(x(k), u(k), w(k)), \quad (2.60)$$

donde  $F(\cdot, \cdot, \cdot)$  es una función no lineal que determina la transición del estado de la neurona:

$$\begin{aligned} \frac{\partial \hat{x}(k+1)}{\partial \omega(k)} &= \frac{\partial F(x(k), u(k), w(k))}{\partial x(k)} \frac{\partial x(k)}{\partial \omega(k)} \\ &+ \frac{\partial F(x(k), u(k), w(k))}{\partial u(k)} \frac{\partial u(k)}{\partial \omega(k)} \\ &+ \frac{\partial F(x(k), u(k), w(k))}{\partial w(k)} \frac{\partial w(k)}{\partial \omega(k)}. \end{aligned} \quad (2.61)$$

como

$$\frac{\partial u(k)}{\partial \omega(k)} = 0, \quad (2.62)$$

$$\frac{\partial w(k)}{\partial \omega(k)} = 1, \quad (2.63)$$

$$\begin{aligned} \frac{\partial \hat{x}(k+1)}{\partial \omega(k)} &= \frac{\partial F(x(k), u(k), w(k))}{\partial x(k)} \frac{\partial x(k)}{\partial \omega(k)} \\ &+ \frac{\partial F(x(k), u(k), w(k))}{\partial \omega(k)} \frac{\partial \omega(k)}{\partial \omega(k)}. \end{aligned} \quad (2.64)$$

Suponiendo que el estado inicial de la red neuronal no depende funcionalmente de los pesos [Rios et al., 2020], se establece que  $H_{ij}(0) = 0$ ; además,  $P_i$  y  $Q_i$  se inicializan como matrices diagonales con valores  $P_i(0)$  y  $Q_i(0)$ , respectivamente. Cabe destacar que  $H_i(k)$ ,  $K_i(k)$  y  $P_i(k)$  para el EKF están delimitados [Arana-Daniel et al., 2020].

## 2.6. Conclusión de Capítulo

Los conceptos preliminares mencionados en este capítulo constituyen los fundamentos matemáticos y de programación sobre los cuales se desarrolla este trabajo. La estructura de las RHONN se utilizará para la construcción del conjunto identificador del sistema y el control adaptativo. Dichas redes serán retroalimentadas con un aprendizaje en tiempo real, como el que aporta el EKF, tal como se muestra en los Capítulos 3 y 4. A su vez, en el Capítulo 4 se demuestra cómo estas redes pueden trabajar en conjunto como un sistema robusto de control.



## Capítulo 3

# Modelado e Identificación del Sistema

### 3.1. Modelo Cinemático

Un robot móvil diferencial se puede definir como un dispositivo capaz de desplazarse de manera que cada llanta de cada lado del vehículo sea capaz de controlar su velocidad de forma independiente. Un diseño simple general de estos es aquel en el que cada llanta trasera tiene su control de velocidad de manera independiente, y otra sección, delante o atrás, provee estabilidad, como se menciona en [\[Corke, 2023\]](#).

El modelo cinemático asociado al RMD describe su posición y velocidad (angular y lineal) en el plano. Antes de determinar el estado (posiciones y velocidades), es necesario seleccionar un punto específico en el robot móvil que servirá de base para obtener su modelo cinemático. Se considera el modelo del robot móvil diferencial descrito como el visto en la Figura 3.1. En general, se pueden diferenciar dos escenarios para el modelo cinemático del RMD: el punto de operación 'c' y el punto 'h', donde el punto 'c' está sobre la línea perpendicular a la velocidad de las llantas y sobre el centro de giro del robot móvil, y 'h' siendo un punto del plano escogido para propósitos del control aplicado [\[Villalobos Álvarez, 2022\]](#).

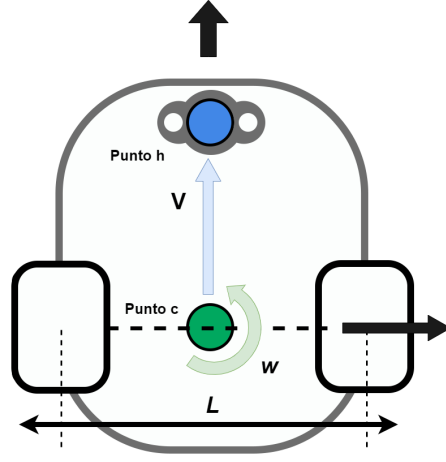


Figura 3.1: Diagrama de cuerpo libre del Robot Móvil Diferencial

Se presenta la nomenclatura del modelo cinemático que se utilizará a lo largo del capítulo [Arana-Daniel et al., 2020, Villalobos Álvarez, 2022]:

- $L$  Distancia entre el centro de masas de las llantas ( $m$ ).
- $h$  Distancia del punto  $c$  perpendicular al eje de giro de las llantas ( $m$ ).
- $V$  Velocidad lineal del RMD ( $m/s$ ).
- $w$  Velocidad angular del RMD ( $rad/s$ ).
- $\theta$  Ángulo de inclinación del robot ( $rad$ ).
- $x_c$  Posición del punto  $c$  en  $X$  ( $m$ ).
- $y_c$  Posición del punto  $c$  en  $Y$  ( $m$ ).

El modelo matemático asociado a un robot móvil diferencial describe la cinemática de su posición y velocidad en un plano [Siegwart et al., 2011]. Como se muestra en la Figura 3.2, es importante diferenciar puntos de operación, donde el punto  $[x_c, y_c]$  [Ríos et al., 2022] corresponde al punto medio del eje de las ruedas del robot y  $\theta$  a su ángulo de orientación.

Es necesario mencionar que el punto que usamos para el control PID adaptativo es el punto extendido  $[x_h, y_h]$  [Guerrero-Castellanos and González-Romeo, 2021]. Sin embargo, la RHONN identificadora usará el modelo de origen para el sistema equivalente.

$$\begin{aligned}\dot{x} &= v \cos \theta, \\ \dot{y} &= v \sin \theta, \\ \dot{\theta} &= \omega\end{aligned}\tag{3.1}$$

Donde  $\theta$  se mide en relación con el eje horizontal X. Esto da lugar a un sistema MIMO no lineal compuesto por tres ecuaciones con dos entradas (velocidad lineal  $v$  y velocidad angular  $w$ ) y dos salidas (posiciones  $x$  y  $y$ ). Despejando la velocidad lineal  $v$  entre las ecuaciones 3.1 e igualando, se obtiene:

$$\dot{x}_c \sin(\theta) - \dot{y}_c \cos(\theta) = 0\tag{3.2}$$

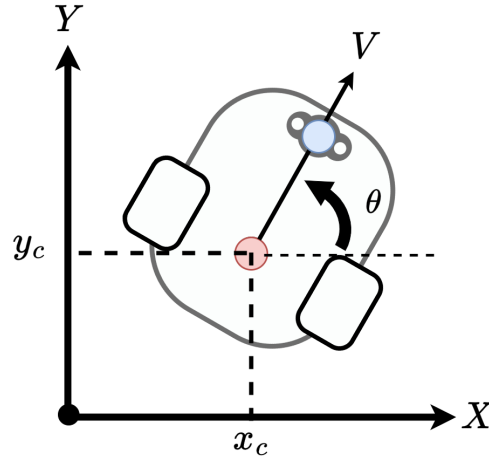


Figura 3.2: Diagrama de cuerpo libre del Robot Móvil Diferencial punto original (c) en el eje de las llantas

Sea el diagrama de cuerpo libre mostrado en la Figura 3.3. Considerando que el punto  $h$  (punto en azul), se observa que el punto  $c$  puede considerarse como un pivote de  $h$ . Por lo tanto, la posición de  $h$  se describe como [Rodríguez-Cortes and Aranda-Bricaire, 2007]:

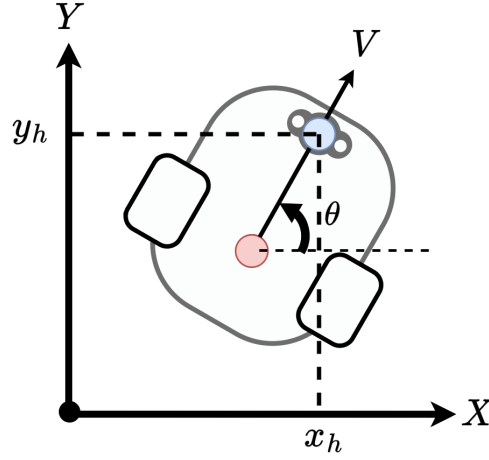


Figura 3.3: Diagrama de cuerpo libre del Robot Móvil Diferencial punto proyectado (h)

$$\begin{aligned} x_h &= x_c + h \cos(\theta), \\ y_h &= y_c + h \sin(\theta) \end{aligned} \quad (3.3)$$

Derivando la posición para obtener las velocidades cartesianas:

$$\begin{aligned} \dot{x}_h &= \dot{x}_c - h \sin(\theta) w, \\ \dot{y}_h &= \dot{y}_c + h \cos(\theta) w, \\ \dot{\theta} &= w \end{aligned} \quad (3.4)$$

Obteniendo las ecuaciones del plano extendido:

$$\begin{aligned} \dot{x}_h &= V \cos(\theta) - h \sin(\theta) w, \\ \dot{y}_h &= V \sin(\theta) + h \cos(\theta) w, \\ \dot{\theta} &= w \end{aligned} \quad (3.5)$$

Esto da como resultado el modelo cinemático del RMD en el punto  $h$ . Finalmente, se presenta en su forma matricial mostrado en 3.6, donde se observa que la matriz jacobiana de velocidades ( $\mathbf{J}$ ) tiene un determinante diferente de cero. Esta característica permite la realización de una cinemática inversa en el diseño de controladores, lo que no es posible en el modelo en el punto  $c$  [Villalobos Álvarez, 2022].



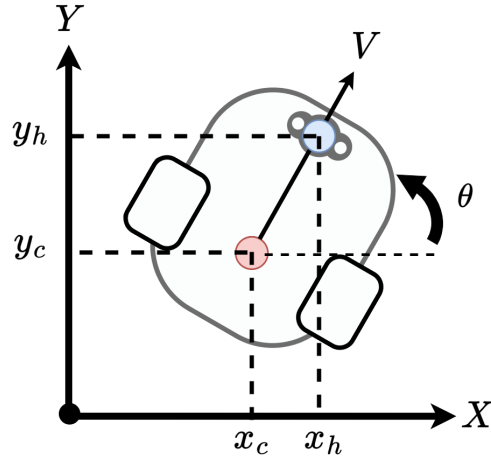


Figura 3.4: Diagrama de cuerpo libre del Robot Móvil Diferencial punto original (c) y punto extendido (h)

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -h \sin(\theta) \\ \sin(\theta) & h \cos(\theta) \end{bmatrix}}_{\mathbf{J}(\theta)} \begin{bmatrix} v \\ w \end{bmatrix} \quad (3.6)$$

Donde  $\mathbf{J}(\theta)$  es invertible siempre y cuando  $h \neq 0$  [Hernández Barragán et al., 2023]. La ley de control ( $u_x$  y  $u_y$ ) es diseñada como  $\dot{x}_h = u_x$  y  $\dot{y}_h = u_y$ . Se utiliza la siguiente transformación de entrada en la ecuación 3.6. Esta particularidad permite realizar una cinemática inversa en el diseño de controladores, lo cual no es posible en el modelo de la Figura 3.2 [Villalobos Álvarez, 2022]. Donde los estados son la velocidad lineal  $v$  y la velocidad angular  $\omega$  como se muestra en la Ecuación 3.7 usando el jacobiano junto a la ley de control previamente mencionada.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}(\theta)^{-1} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta)/h & \cos(\theta)/h \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \quad (3.7)$$

## 3.2. Diseño de Identificador del Sistema

### 3.2.1. Identificación del Sistema

Los sistemas de control modernos tienden a requerir un conocimiento estructurado del sistema a controlar; por ello, dicho modelo debe representarse mediante ecuaciones diferenciales o en diferencias. Existen diversas técnicas para la obtención de dicho modelo, como se muestra en la Tabla 3.1 [Rios et al., 2020], las cuales pueden agruparse en dos categorías: deducción, aplicando las leyes de la física, o inferencia, a partir de datos obtenidos durante un experimento práctico [Alanis et al., 2020].

Tabla 3.1: Comparación de métodos para modelar sistemas.

| Deductivo                                                                                                                                                                         | Identificación                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| El modelo del sistema se obtiene utilizando las leyes de la física. La mayoría de los casos se requiere una gran cantidad de tiempo.                                              | El modelo se infiere a partir de un conjunto de datos obtenidos de un experimento práctico realizado con el sistema dentro de su rango operativo.                               |
| Es poco probable o incluso imposible obtener un modelo exacto del sistema debido a la dificultad de modelar perturbaciones, retardos, saturaciones y dinámicas de los actuadores. | Este enfoque puede ser práctico como un atajo para derivar modelos matemáticos, pero no siempre proporciona un modelo exacto, sino uno satisfactorio con un esfuerzo razonable. |

El proceso de identificación, de manera general, se puede dividir de la siguiente manera [Rios et al., 2020]:

#### 1. Experimentación.

Consiste en recopilar datos que describan cómo se comporta el sistema bajo ciertas condiciones.

**2. Selección de la estructura del modelo.**

Implica elegir la configuración adecuada para representar el sistema de manera precisa.

**3. Estimación del modelo.**

Se enfoca en calcular los parámetros que definen el modelo basado en los datos recopilados.

**4. Validación.**

Es el proceso de verificar si el modelo obtenido cumple con los criterios y requisitos establecidos.

**5. Regresar a cualquier paso del proceso.**

En caso de que ocurra un error en alguna etapa, se permite retroceder para realizar ajustes y correcciones necesarias.

De forma general, se puede estructurar el proceso de identificación de modelo del sistema como lo muestra el diagrama de la Figura 3.5.

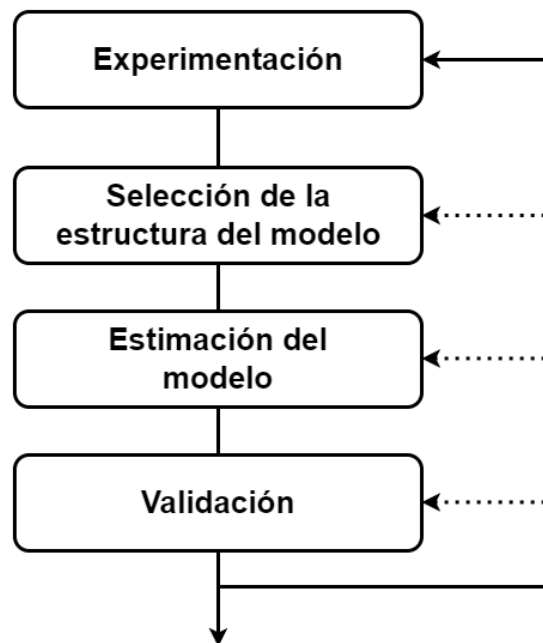


Figura 3.5: Diagrama del proceso general de la identificación.

### 3.2.2. Identificación Neuronal

El uso de redes neuronales simplifica el proceso de modelado y el diseño de control. Las redes neuronales pueden emplearse en controladores altamente especializados o en controladores de propósito general. Su principal ventaja radica en su capacidad para modelar una amplia gama de sistemas, lo que no solo reduce los tiempos de desarrollo, sino que también mejora el desempeño en comparación con las técnicas de control tradicionales [Rios et al., 2020].

Existen otras razones para utilizar redes neuronales en la identificación de sistemas, tales como:

- Debido a que el modelo aproximado a nuestro sistema real no da un resultado óptimo debido a una condición física.
- Se necesita un modelo no lineal porque el modelo existente no describe adecuadamente el comportamiento del sistema.
- El sistema es demasiado complejo para ser modelado con un método tradicional.
- No se tiene acceso a la medición o descripción de todas las variables involucradas en el sistema.

La identificación neuronal puede definirse como [Rios et al., 2020]:

**Una técnica para obtener un modelo de un sistema que consiste en seleccionar un modelo adecuado de red neuronal y ajustar sus pesos de acuerdo con una ley de adaptación para que la red neuronal aproxime satisfactoriamente la respuesta del sistema real al mismo estímulo.**

Los pasos descritos en la Figura 3.7 para llevar a cabo el proceso de identificación neuronal en este trabajo se implementan de la siguiente forma [Rios et al., 2020]:

- **Paso 1:** Recopilar la información necesaria del sistema para el proceso de identificación neuronal.
- **Paso 2:** Seleccionar un modelo de red neuronal adecuado; en este caso, se utiliza una RHONN con configuración serie-paralelo.

- **Paso 3:** Ejecutar el entrenamiento del modelo empleando el algoritmo basado en el Filtro Extendido de Kalman, el cual es en línea.
- **Paso 4:** Verificar los resultados obtenidos, asegurándose de que el error entre las salidas del sistema y las del identificador sea mínimo para las mismas entradas.
- **Paso 5:** Aplicar un entrenamiento en línea que concluirá automáticamente cuando la aplicación detenga su funcionamiento.

### 3.2.3. Estructura General

Considérese el problema de estimar el sistema discreto no lineal con retardos desconocidos (2.42) utilizando la siguiente RHONN en un modelo serie-paralelo:

$$\hat{x}_i(k+1) = w_i^T z_i(x(k-l), u(k)) + \epsilon_{z_i}, \quad i = 1, \dots, n, \quad (3.8)$$

$z_i(x(k-l), u(k))$  es la función de activación que depende de los valores actuales y pasados de las entradas del sistema.  $x(k-l)$  son las entradas pasadas (en este caso, el valor de  $x$  en el paso de tiempo  $k-l$ ), lo que indica la naturaleza recurrente del modelo. Esta entrada pasada tiene un retardo de  $l$  pasos.

$u(k)$  son las entradas actuales al sistema (pueden ser señales de control o cualquier otra entrada relevante para el modelo en el paso de tiempo  $k$ ).

donde  $\epsilon_{z_i}$  es una aproximación acotada del error que puede ser reducida o ampliada incrementando el número de pesos ajustables [Rovithakis and Christodoulou, 2012]. Supongamos que existe un vector de pesos ideal  $w_i^*$  tal que  $\|\epsilon_{z_i}\|$  puede minimizarse en un conjunto compacto  $\Omega_{z_i} \subset \mathbb{R}^{L_i}$ . El vector de pesos ideal  $w_i^*$  es una cantidad artificial necesaria para fines analíticos [Rios et al., 2020]. En general, se asume que este vector existe y es constante pero desconocido. Definamos su estimación como  $w_i$  y el error de estimación como:

$$\tilde{w}_i(k) = w_i^* - w_i(k). \quad (3.9)$$

Entonces, considerando (2.53), la dinámica de (3.9) puede definirse como

$$\tilde{w}_i(k+1) = \tilde{w}_i(k) - \eta_i K_i(k) e(k). \quad (3.10)$$

El error de identificación se define como

$$e_i(k) = x_i(k) - \hat{x}_i(k). \quad (3.11)$$

La dinámica de (3.11) puede expresarse como

$$e_i(k+1) = \tilde{w}_i(k) z_i(x(k-l), u(k)) + \epsilon_{z_i}. \quad (3.12)$$

**Teorema 3.1.** La RHONN (3.8) entrenada con el algoritmo modificado basado en EKF (2.53) para identificar la planta no lineal (2.42) garantiza que el error de identificación (3.11) está acotado en última instancia de manera semiglobal y uniformemente (*SGUUB*); esto implica que existe una cota uniforme para el error de identificación. es decir, el error no crece indefinidamente y permanece dentro de un límite predecible. Ser semiglobal indica que la cota del error depende de un conjunto inicial de condiciones y parámetros del sistema, cuales no garantizan una estabilidad para cada condición inicial. Además, los pesos de la RHONN permanecen acotados. Este teorema tiene su demostración en las fuentes: [Rios et al., 2020, Alanis et al., 2020].

### 3.2.4. Diseño Implementado

El diseño implementado para el caso del robot móvil diferencial fue estructurado utilizando 3 neuronas, una para cada variable de estado del sistema, entrenadas por medio del EKF [Rios et al., 2020]. Dado que el modelo original usado cuenta con 3 ecuaciones representando su comportamiento, se propuso el siguiente sistema:

$$x(k+1) = f(x(k)) + g(x(k))u(k), \quad (3.13)$$

$$\begin{aligned} f_1 &= W_1^T Z_1, \\ f_2 &= W_2^T Z_2, \\ f_3 &= W_3^T Z_3. \end{aligned} \quad (3.14)$$

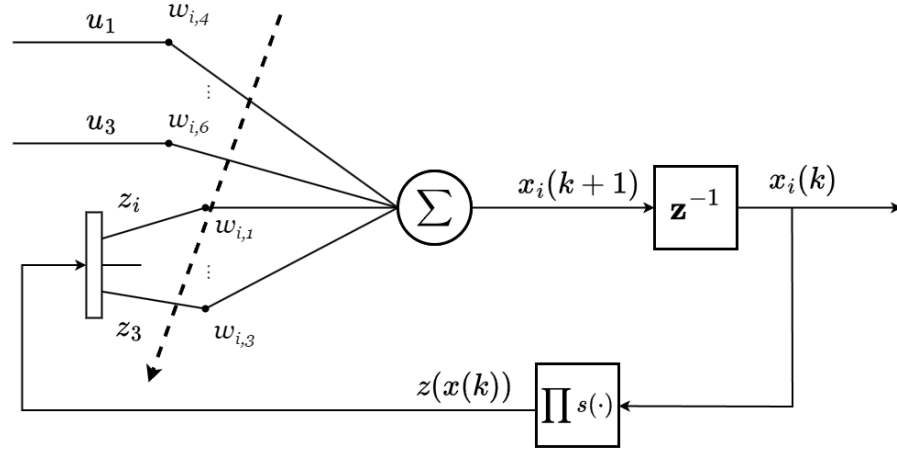


Figura 3.6: Estructura del Identificador Neuronal en tiempo discreto.

y

$$g = \begin{pmatrix} 0.001 & 0.001 \\ 0.001 & 0.001 \\ 0.001 & 0.001 \end{pmatrix}. \quad (3.15)$$

Donde  $X_{\text{medida}}$ ,  $Y_{\text{medida}}$  y  $\theta_{\text{medida}}$  representan los valores obtenidos al excitar el sistema dentro de un rango de operación estándar. Con base en la ecuación 3.6, esta se utiliza como referencia para diseñar, de manera heurística, la red neuronal, considerando modelar las dinámicas de estado asociadas a cada neurona, como se muestra en la Figura 3.6.

A partir de lo mencionado previamente, se diseñan las funciones sigmoides, tal como se muestra:

$$s_1 = \begin{bmatrix} x_{\text{medida}} \\ y_{\text{medida}} \\ \theta_{\text{medida}} \end{bmatrix}, \quad s_2 = \begin{bmatrix} x_{\text{medida}} \\ y_{\text{medida}} \\ \theta_{\text{medida}} \end{bmatrix}, \quad s_3 = \begin{bmatrix} x_{\text{medida}} \\ y_{\text{medida}} \\ \theta_{\text{medida}} \end{bmatrix}. \quad (3.16)$$

donde:

$$Z_1 = \frac{1}{1 + \exp(-s_1)}, \quad Z_2 = \frac{1}{1 + \exp(-s_2)}, \quad Z_3 = \frac{1}{1 + \exp(-s_3)}. \quad (3.17)$$

Cada una de las  $Z_i$  será entrenada por el Filtro Extendido de Kalman:

$$\begin{aligned}
 H_i &= Z_i, \\
 K_i &= \frac{P_i H_i}{R_i + H_i P_i H_i}, \\
 W_i &+= g_i K_i e_i, \\
 P_i &= P_i - K_i H_i P_i + q_i.
 \end{aligned} \tag{3.18}$$

El proceso de identificación desarrollado anteriormente, de forma general, se puede ilustrar en el diagrama de la Figura 3.7. Los datos obtenidos de los sensores del robot móvil diferencial fueron usados como entrada para esta red neuronal, cuyos resultados se muestran en las Figuras 3.8 y 3.9. Estos resultados demuestran con claridad la velocidad con la que el identificador se adapta al sistema de parámetros desconocidos con eficacia, donde  $x_1$  es la posición en  $X$ ,  $x_2$  es la posición en  $Y$  y  $x_3$  es el ángulo  $\theta$  del sistema.

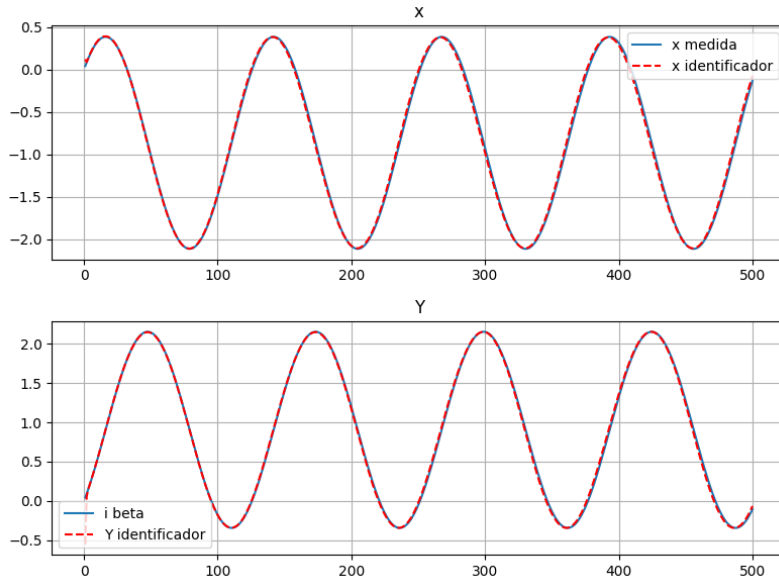


Figura 3.8: Gráfica de las posiciones cartesianas comparando el sistema equivalente dado por el identificador contra los datos reales del robot.



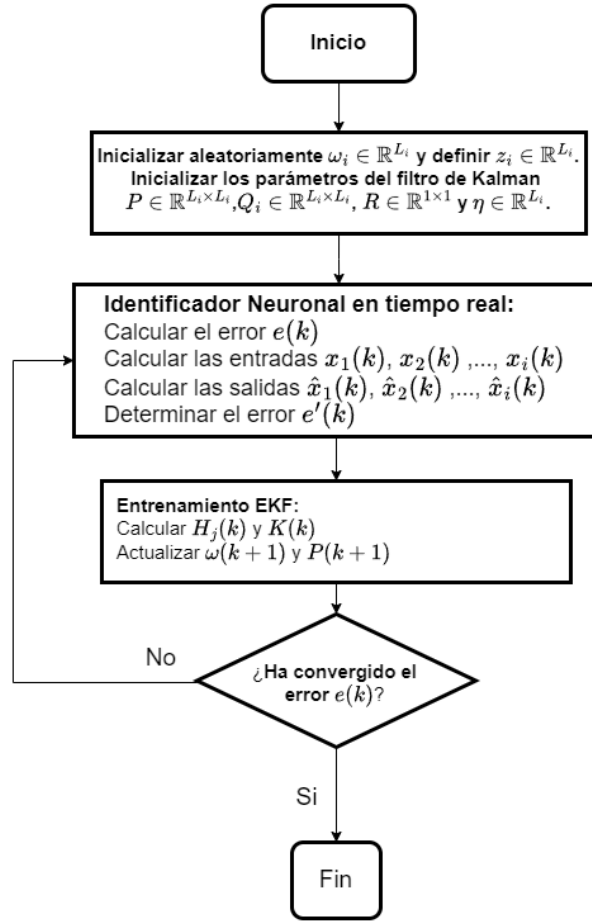


Figura 3.7: Proceso de la Identificación Neuronal.

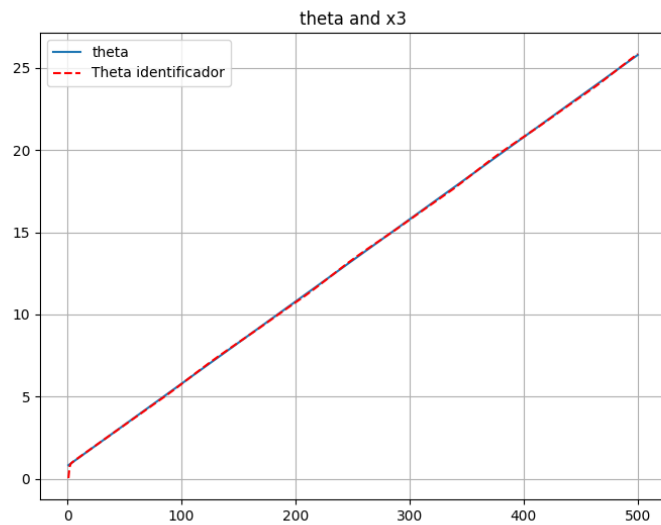


Figura 3.9: Gráfica del ángulo comparando el sistema equivalente dado por el identificador contra los datos reales del robot.

### 3.3. Conclusión de Capítulo

En esta sección, se diseña un modelo neuronal equivalente al modelo cinemático de un robot móvil diferencial. Al proyectar el sistema de esta manera, se facilita la comprensión del funcionamiento del modelo clásico de estos robots, tomando como entradas la velocidad lineal y angular. Con el objetivo de captar la relación entre las variables del sistema, se propone un modelo neuronal diseñado heurísticamente, en el cual un identificador neuronal juega un papel clave. Este identificador contribuye a mejorar la robustez y versatilidad del sistema. Finalmente, el modelo neuronal desarrollado será empleado en el controlador neuronal presentado en el siguiente capítulo.

## Capítulo 4

# Diseño y Aplicación del Control PID Adaptivo

### Diseño del Control PID Adaptivo

El controlador Proporcional-Integral-Derivativo (PID) sigue siendo uno de los controles más populares en la industria, en especial cuando se habla de aplicaciones en la industria [Hernández Barragán et al., 2020]. Si bien este control, en condiciones reguladas y principalmente para sistemas lineales, puede funcionar adecuadamente, los sistemas no lineales que se encuentran en un constante cambio necesitan que el controlador y sus ganancias estén adaptándose. Por ello, el desempeño del PID puede ser deficiente al encontrarse con:

1. **Desempeño ineficaz ante cambios:** El PID no responde bien ante variaciones en las condiciones de operación del sistema, lo que puede llevar a respuestas oscilatorias o lentas si la ganancia del proceso se desvía de su valor nominal [Tian et al., 1999].
2. **Problemas con incertidumbres paramétricas:** Los controladores PID degradan su rendimiento significativamente cuando hay incertidumbre en las condiciones operativas o en los parámetros ambientales del sistema [Khalil, 2002, Tian et al., 1999].
3. **Sensibilidad a perturbaciones:** Los controladores PID pueden tener un desem-

peño pobre presente en la señal de entrada, en especial en perturbaciones externas constantes [Khalil, 2002].

4. **Problemas con errores grandes:** Un error de control grande puede llevar a una respuesta integral agresiva, resultando en el indeseado fenómeno de reset windup” (por su nombre en inglés), donde el integrador se satura y afecta la estabilidad del sistema, pudiendo llegar al punto de no convergencia [Tian et al., 1999].
5. **Dependencia de modelos:** Aunque los PID son independientes del modelo, no tienen la capacidad de adaptarse de manera dinámica a sistemas con variaciones significativas en sus parámetros o comportamiento no lineal, requiriendo ajustes manuales constantes [Tian et al., 1999].
6. **Complejidad en el ajuste de parámetros:** El ajuste de los parámetros del PID ( $K_p$ ,  $K_i$ ,  $K_d$ ) en sistemas no lineales suele ser complejo y puede no garantizar un desempeño óptimo. Técnicas avanzadas, como la optimización robusta, son necesarias para superar esta limitación [Dorf and Bishop, 2011].

La principal desventaja de los controladores PID convencionales es que no son adecuados para sistemas no lineales y variantes en el tiempo. Los controladores neuronales adaptativos son una alternativa para superar este problema.

## 4.1. Diseño de PID Adaptivo del Sistema

$$e(k) = y_r(k) - y(k) \quad (4.1)$$

$$x_1(k) = e(k) \quad (4.2)$$

$$x_2(k) = e(k) - e(k-1) \quad (4.3)$$

$$x_3(k) = \sum_{j=1}^k e'(k) \quad (4.4)$$

$$v(k) = \sum_{i=1}^3 \omega_i(k) x_i(k) \quad (4.5)$$

$$u(k) = \alpha \tanh(v(k)) \quad (4.6)$$

Con el sistema equivalente previamente definido en las ecuaciones (3.14) y (3.15), se usará para aplicar el control adaptivo PID. Como se menciona en [Hernández Barragán et al., 2020], se usará un esquema de una sola neurona donde los pesos  $w_1$ ,  $w_2$  y  $w_3$  son entrenados en línea por medio del Filtro Extendido de Kalman, como se ve en la figura, las cuales representan las ganancias  $k_P$ ,  $k_I$  y  $k_D$ .

Donde  $x_1$ ,  $x_2$  y  $x_3$  son las entradas de la red neuronal, cuyo valor son los errores proporcional, integral y diferencial. Donde  $v$  es la suma de los pesos por la entrada y  $u$  [Hernández-Barragán et al., 2021] es la salida de la red después de la función de activación. El valor  $v$  se calcula como la suma ponderada de las entradas de la neurona en 4.5. Finalmente, la salida de la neurona  $\hat{y}$  se calcula con 4.6, donde se selecciona la función de activación  $\tanh(\cdot)$ .

La función de activación reacciona en el rango  $(-1, 1)$ . Sin embargo, se puede seleccionar el parámetro  $\alpha$  para ajustar la acción de control, ya que la salida de la neurona es directamente la señal de control  $u(k) = \hat{y}(k)$ .

Cuyo entrenamiento EKF está descrito de la siguiente manera, cuyo conjunto se llama EKF-SNPID, por la combinación de la estructura de la red y el entrenamiento previamente mencionado en [Hernández Barragán et al., 2020].

$$K(k) = P(k)H(k) \left[ R(k) + H^\top(k)P(k)H(k) \right]^{-1} \quad (4.7)$$

$$\omega(k+1) = \omega(k) + \eta K(k)e(k) \quad (4.8)$$

$$P(k+1) = P(k) - K(k)H^\top(k)P(k) + Q(k) \quad (4.9)$$

$$H_j(k) = \left[ \frac{\partial u(k)}{\partial \omega_j(k)} \right] = \left[ \frac{\partial u(k)}{\partial v(k)} \quad \frac{\partial v(k)}{\partial \omega_j(k)} \right] = \alpha \operatorname{sech}^2(v(k))x_j(k) \quad (4.10)$$

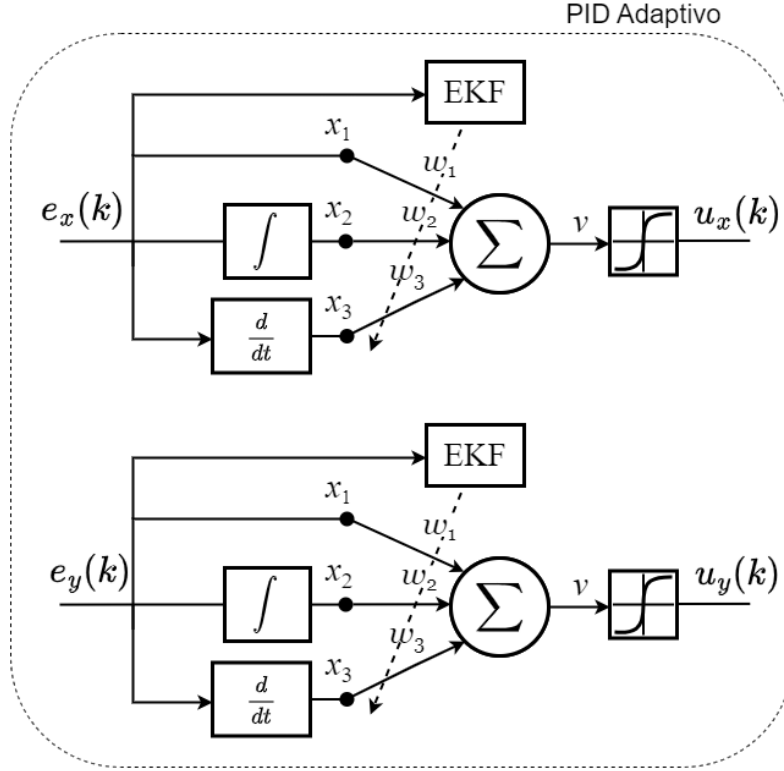


Figura 4.1: Esquema del PID adaptivo de una sola neurona para cada una de las variables a controlar.

$$\begin{aligned}
 P &= 10 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} && \text{(matriz diagonal de tamaño 3 general),} \\
 Q &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} && \text{(matriz diagonal de tamaño 3 general),} \\
 R &= 0.1, \\
 \alpha &= 0.05 && \text{(valor predeterminado),} \\
 \eta_1 &= 0.01 && \text{(tasa de aprendizaje primera neurona),} \\
 \eta_2 &= 0.03 && \text{(tasa de aprendizaje segunda neurona).}
 \end{aligned} \tag{4.11}$$

Donde  $\omega \in \mathbb{R}^3$  es el vector de pesos,  $K \in \mathbb{R}^{3 \times 1}$  es el vector de ganancia de Kalman,  $P \in \mathbb{R}^{3 \times 3}$ ,  $Q \in \mathbb{R}^{3 \times 3}$  y  $R \in \mathbb{R}^{1 \times 1}$  son matrices de covarianza del error de estimación de pesos, ruido de estimación y ruido de error, respectivamente.

Los parámetros  $\alpha$ ,  $\eta_1$ ,  $\eta_2$  fueron conseguidos de forma heurística al probar el modelo de forma experimental.  $\eta \in \mathbb{R}$  es la tasa de aprendizaje del filtro de Kalman, y  $H \in \mathbb{R}^{3 \times 1}$  es una matriz cuyas entradas  $H_j$  son la derivada de la salida de la red neuronal con respecto a cada peso representado en la ecuación 4.10.  $u \in \mathbb{R}$  es la salida de la neurona, y el error  $e \in \mathbb{R}$  se define como la diferencia entre la salida deseada y la salida de la neurona mostrada en la ecuación 4.1 [Hernández Barragán et al., 2020].

El objetivo del proceso de entrenamiento es determinar un conjunto de pesos  $\omega$  que permita a la señal  $u$  minimizar el error entre la referencia deseada y la salida del sistema. En la Figura 4.2 se muestra un diagrama de flujo que, de forma general, demuestra el proceso que sigue el EKF-SNPID [Hernández Barragán et al., 2020]. En el algoritmo EKF [Gómez Ávila et al., 2020], la tasa de aprendizaje  $\eta$  se considera un escalar. No obstante, se puede plantear la utilización de  $\eta \in \mathbb{R}^{3 \times 1}$  para asignar una tasa de aprendizaje individualizada a cada peso  $\omega_i$ , ajustándolos de forma independiente mediante  $\eta_i$  [Hernández-Barragán et al., 2021].

## 4.2. Simulación: Identificador Neuronal y Control Neuronal

Con los diseños de ambas redes, se llevaron a cabo pruebas simuladas que incluyen el seguimiento de trayectorias y el seguimiento a puntos específicos con una trayectoria libre. Dichas simulaciones permiten confirmar que el control aplicado al identificador opera de manera óptima en simulación.

Dado que una simulación no considera ruidos y perturbaciones que las pruebas experimentales sí incluyen, ajustar las ganancias y las condiciones iniciales de las redes neuronales utilizadas no es necesariamente equivalente al ajuste realizado para pruebas experimentales.

Es importante recalcar que ambas redes neuronales se entrenan en línea utilizando el Filtro Extendido de Kalman. En conjunto, el diagrama presentado en la Figura 4.3

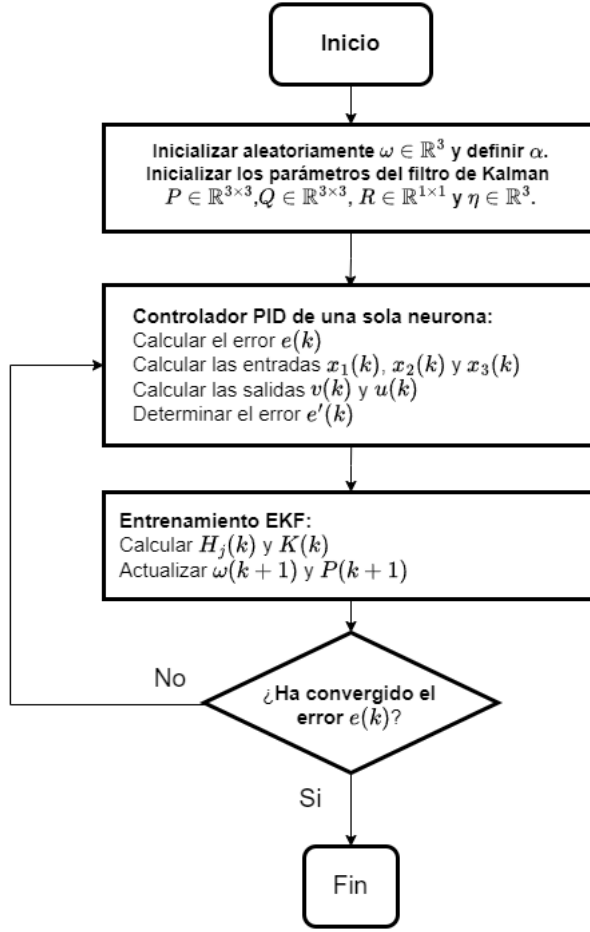


Figura 4.2: Esquema del EKF-SNPID o de una sola neurona para cada una de las variables a controlar.

ilustra el comportamiento general del sistema, considerando la construcción del Identificador Neuronal, mostrada en la Figura 3.6, junto con el control neuronal EKF-SNPID presentado en la Figura 4.1. Este diagrama sintetiza de manera clara y completa el sistema simulado y el prototipo experimental utilizado.

Debido a que el ajuste de variables se realiza de manera heurística o, en su defecto, utilizando un algoritmo metaheurístico que ajuste automáticamente dichas variables, las pruebas físicas necesitarán sus propias condiciones iniciales. Sin embargo, tener las simulaciones como punto de partida nos permite encontrar más rápido las condiciones iniciales óptimas.



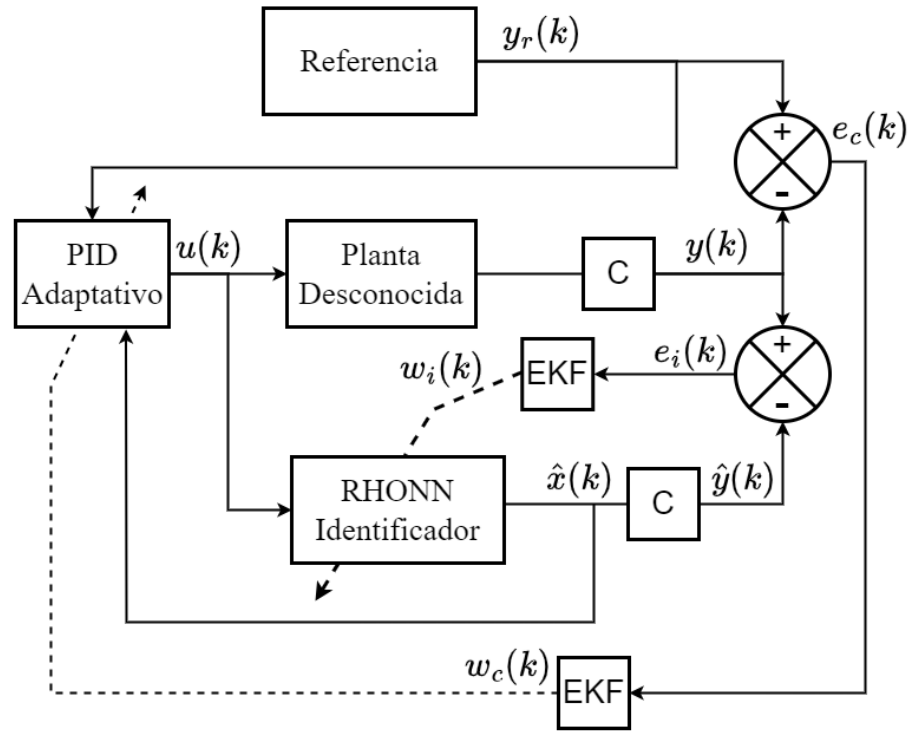


Figura 4.3: Diagrama de bloques del sistema completo con todas las redes neuronales usadas.

En la Figura 4.4 se observa una simulación en tiempo discreto de un robot móvil diferencial iniciando en la posición  $(0,0)$ . Se le permite llegar, con trayectoria libre, al punto  $(5,2)$  utilizando el identificador y el PID adaptativo, logrando llegar al punto indicado de manera rápida tanto por el identificador como por la planta desconocida.

A continuación, se plantea una trayectoria circular con un radio de 3 unidades, cuyo centro está ubicado en el origen  $(0,0)$ . Esta trayectoria define el punto de partida para el robot móvil diferencial, tal como se muestra en la Figura 4.5. Al ejecutar la simulación, se observa una disminución progresiva del error a medida que el robot comienza a alinearse con la trayectoria establecida.

Se puede analizar un período de adaptación que se ve claramente representado en la Figura 4.5, completando el seguimiento de trayectoria con un error reducido casi a cero. Como se muestra en la Figura 4.6, se representa el comportamiento oscilante de la trayectoria deseada en verde, comparada con el identificador neuronal y la planta desconocida, las cuales

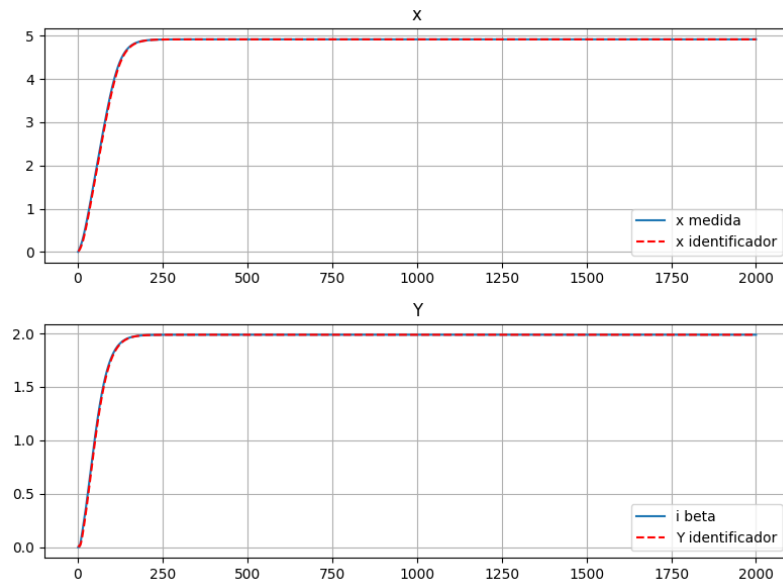


Figura 4.4: Identificador y PID adaptivo simulado en la coordenada  $x = 5$ ,  $y = 2$ .

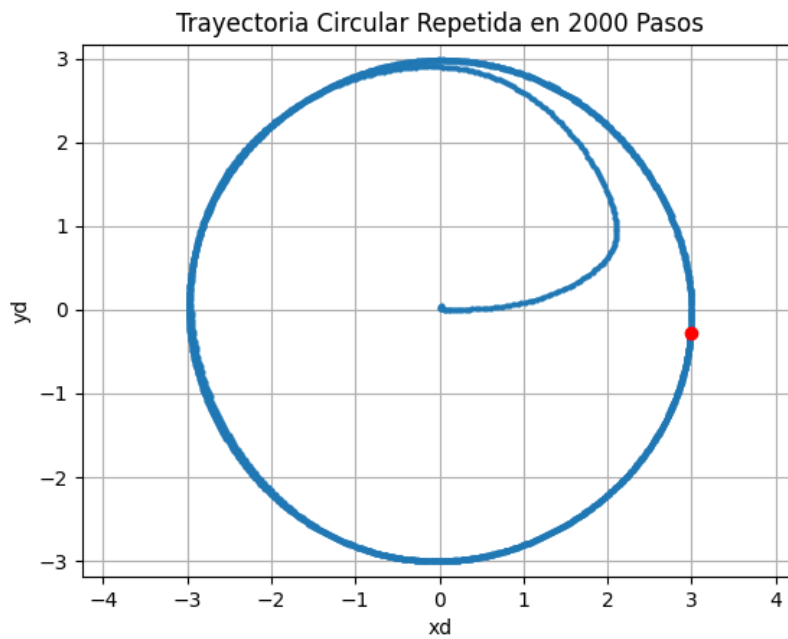


Figura 4.5: Seguimiento de trayectoria circular donde el robot inicia al centro del círculo en las coordenadas  $X$  e  $Y$ .

se encuentran solapadas.

Se resalta el comportamiento de cada red neuronal en la Figura 4.7, donde se compara la planta desconocida con el modelo identificado de ambas variables de posición. El error resultante se ilustra en la Figura 4.8, donde la diferencia entre la planta desconocida y el identificador muestra un error menor a 0.05, y el valor RMS es inferior a 0.045, como se detalla en la Tabla 4.1. Además, el error de trayectoria en el PID adaptivo, presentado en la Figura 4.9, disminuye rápidamente durante los primeros ciclos, ya que el punto de inicio está lejos de la trayectoria definida. Cuando el movimiento se estabiliza en la variable  $Y$ , el error disminuye más lentamente.

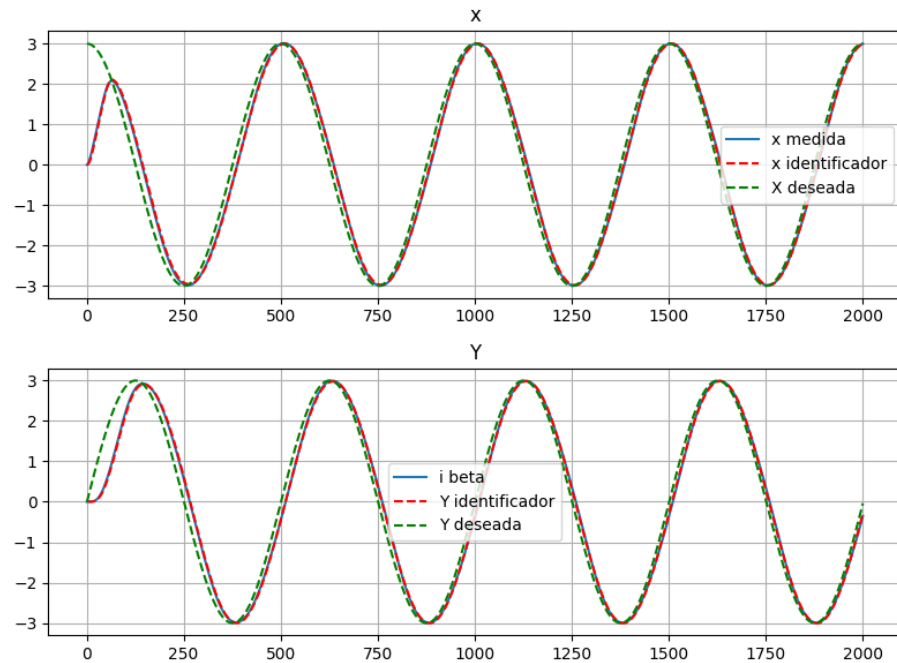


Figura 4.6: Valores de  $X$  e  $Y$  a lo largo del tiempo comparados con la trayectoria deseada.

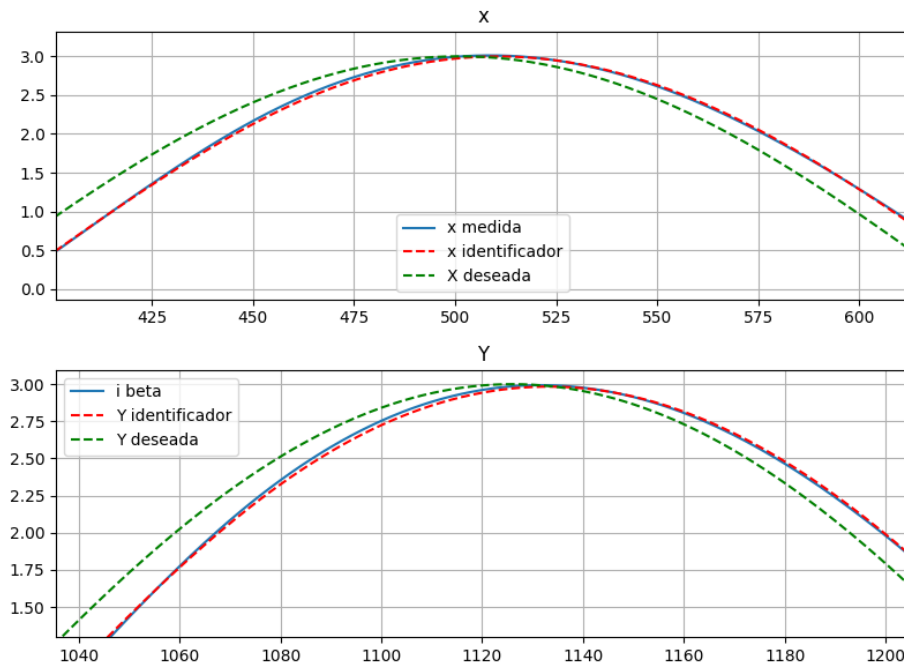


Figura 4.7: Valores de  $X$  e  $Y$  a lo largo del tiempo comparados con la trayectoria deseada, acercamiento de la recta.

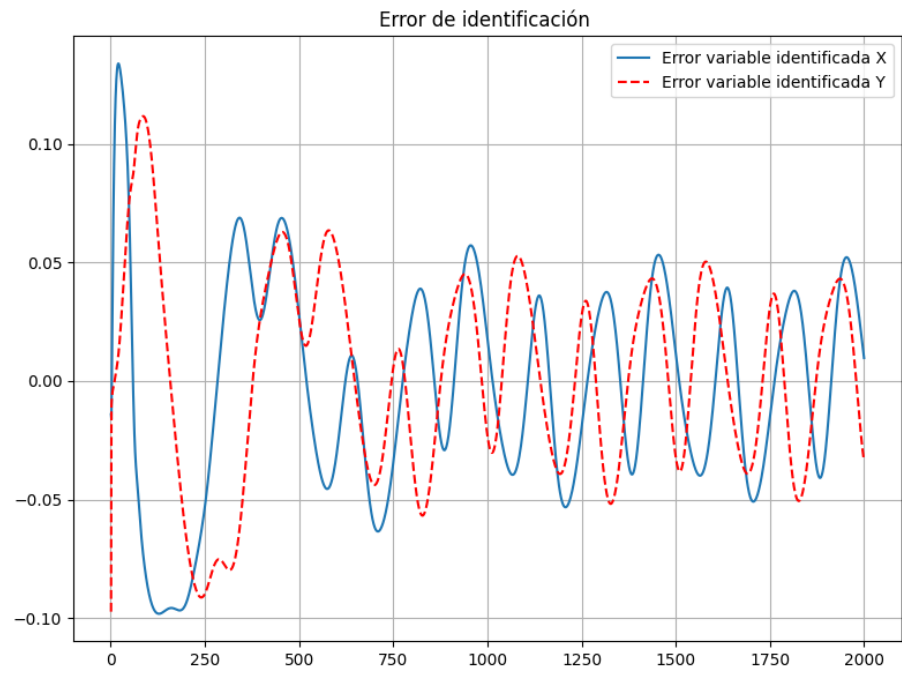


Figura 4.8: Error del proceso de identificación neuronal.

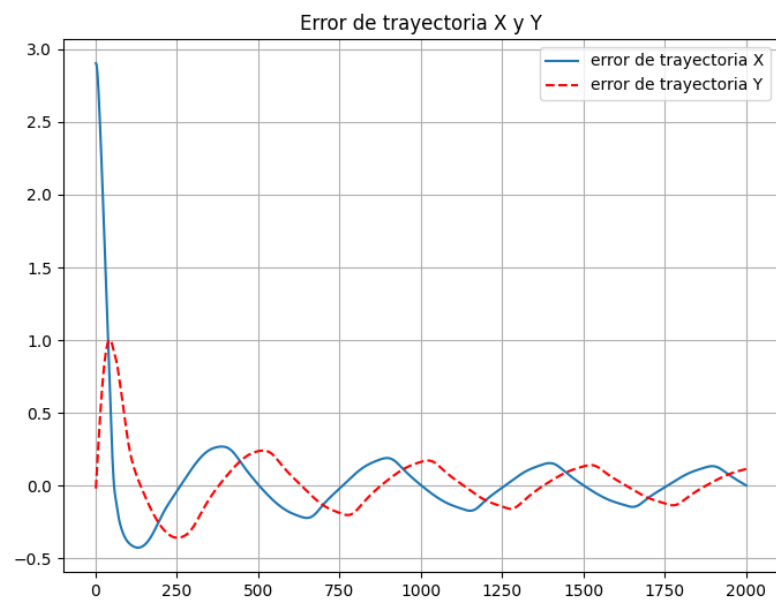


Figura 4.9: Error de trayectoria del controlador neuronal.

Tabla 4.1: RMS y últimos valores de las variables del error del identificador neuronal y el controlador neuronal.

| Variable                 | RMS    | Último Valor |
|--------------------------|--------|--------------|
| Error Identificador $X$  | 0.0453 | 0.0097       |
| Error Identificador $Y$  | 0.0431 | -0.0337      |
| Error de Trayectoria $X$ | 0.3415 | 0.0020       |
| Error de Trayectoria $Y$ | 0.2134 | 0.1154       |

### 4.3. Pruebas Experimentales



Figura 4.10: Turtlebot3<sup>®</sup> Waffle Pi Kit

Para la fase experimental, se utiliza el robot Turtlebot3<sup>®</sup> Waffle Pi, ilustrado en la Figura 4.10. Entre los elementos de hardware de un robot Turtlebot3<sup>®</sup> Waffle Pi se incluyen una Raspberry Pi 3, un sensor LiDAR de 360 grados, un sensor de distancia LDS-01, y una IMU con giroscopio de 3 ejes y acelerómetro de 3 ejes [Ríos et al., 2022]. La pose del robot se calcula mediante un mensaje de odometría utilizando ROS, lo que proporciona su posición y orientación. Además, su posición se reinicia para que siempre parta de manera constante

desde un punto de origen.

A continuación, se describen los componentes utilizados junto al modelo kit para su correcto funcionamiento y configuración [ROBOTIS, 2024a, ROBOTIS, 2024b]:

1. **Turtlebot3 Waffle Pi:** El TurtleBot3 Waffle utiliza una placa Raspberry Pi, un sistema LiDAR de 360° de ROBOTIS, una unidad de control OpenCR, una cámara Raspberry Pi y dos motores servo Dynamixel para la locomoción.
2. **Sensor Láser LDS-02:** Este escáner láser 2D proporciona datos del entorno en un rango completo de 360 grados, esenciales para aplicaciones de SLAM (Localización y Mapeo Simultáneos) y navegación. Opera con una tensión de 5V DC, consume 240 mA y ofrece una tasa de muestreo fija de 2.3 kHz.
3. **Robot Operating System (ROS):** El sistema ROS es un entorno de código abierto para el desarrollo de sistemas autónomos. ROS permite el uso de lenguajes de programación como Python, utilizado en este trabajo para ejecutar el código principal de identificación y control neuronal, permitiendo su implementación en equipos de cómputo de capacidades moderadas.
4. **Raspberry Pi 4 Model B:** Está equipada con un procesador Broadcom BCM2711 quad-core Cortex-A72 de 64 bits a 1.5 GHz, donde se carga el modelo del Turtlebot Waffle Pi junto a los paquetes adecuados para aplicar el control en tiempo real de forma inalámbrica.



Figura 4.11: Model Kit utilizado de un robot móvil diferencial junto a los periféricos conectados.

Esta prueba experimental nos permite evaluar la robustez de las redes utilizadas, dado que la exactitud de las mediciones depende tanto del entorno como del desgaste del robot. Por esta razón, se emplea un identificador neuronal; sin embargo, también es importante conocer los parámetros del robot para comparar el impacto en condiciones ambientales similares.

Los siguientes resultados se obtuvieron definiendo un punto final para el robot con una trayectoria libre en diversos puntos, comparando el PID adaptativo solo contra el identificador más el PID adaptativo, con el objetivo de analizar sus comportamientos, similitudes, ventajas y desventajas.

#### 4.3.1. Caso 1: Mayor distancia

Como se muestra en las Figuras 4.12 y 4.13, se define el punto final como (3,3), las variables controladas  $X$  e  $Y$ , la señal de control de la velocidad angular y lineal, los vectores



de error respecto al punto deseado previamente mencionado, y  $\theta$  como una variable que se deja sin referencia.

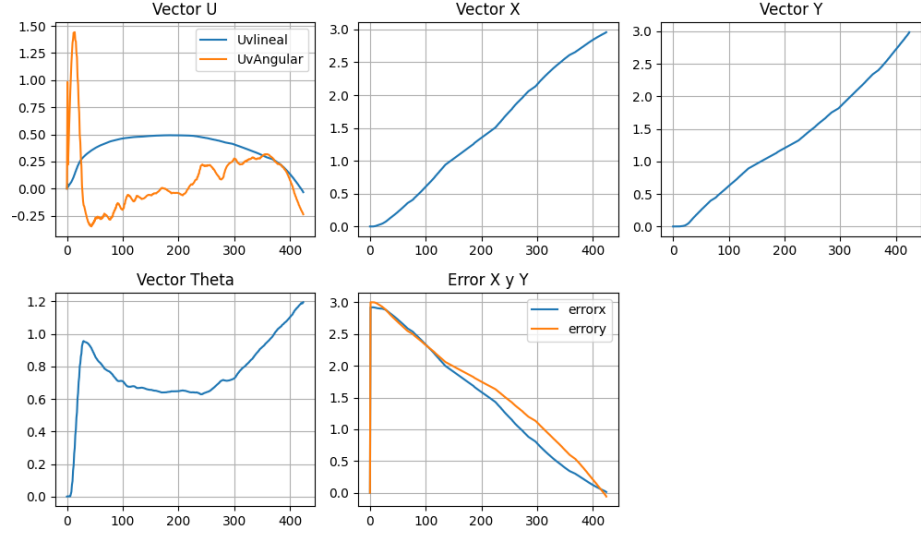


Figura 4.12: PID adaptivo experimental en la coordenada  $x = 3$ ,  $y = 3$ .

Se ilustra el conjunto del identificador junto con el control PID adaptativo, el cual demuestra una estabilización más rápida, con una transición más gradual y suave, y con menos fluctuaciones e irregularidades. No obstante, el sistema que utiliza únicamente el control PID adaptativo también converge de forma rápida, y en ambos casos se logra disminuir el error.

Cabe destacar que, en el punto inicial, el identificador, cuyas condiciones iniciales son aleatorias, presenta un breve periodo de acoplamiento a la planta desconocida. Esto permite mantener una correcta identificación y un control más suave, lo cual es un aspecto relevante a considerar en pruebas experimentales.

Un aspecto a recalcar del comportamiento generado al combinar el identificador neuronal con el PID adaptativo se evidencia en la ley de control, como se muestra en la Figura 4.17. Al compararla con la Figura 4.16, se observa un aumento en los valores de la velocidad lineal y angular. En particular, la velocidad lineal desciende más rápidamente desde 0.5, presentando una mayor suavidad en su comportamiento. En cuanto a la velocidad angular, ilustrada también en la Figura 4.17, los valores se mantienen dentro del rango de

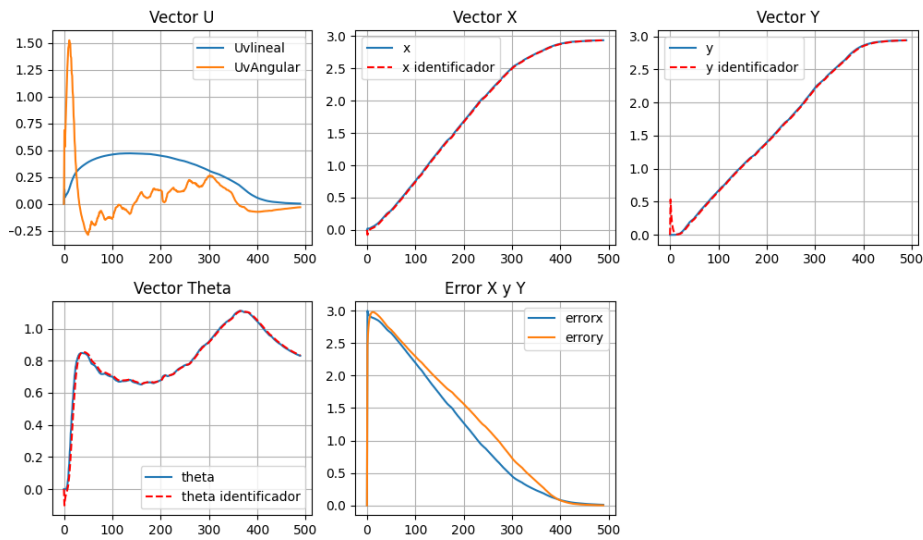


Figura 4.13: Identificador y PID adaptivo experimental en la coordenada  $x = 3$ ,  $y = 3$ .

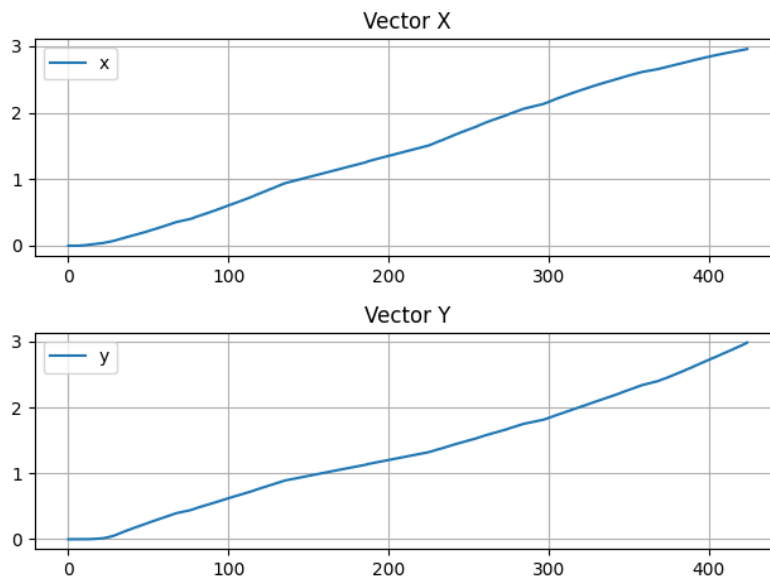


Figura 4.14: Variables solo PID adaptivo en la coordenada  $x = 3$ ,  $y = 3$ .

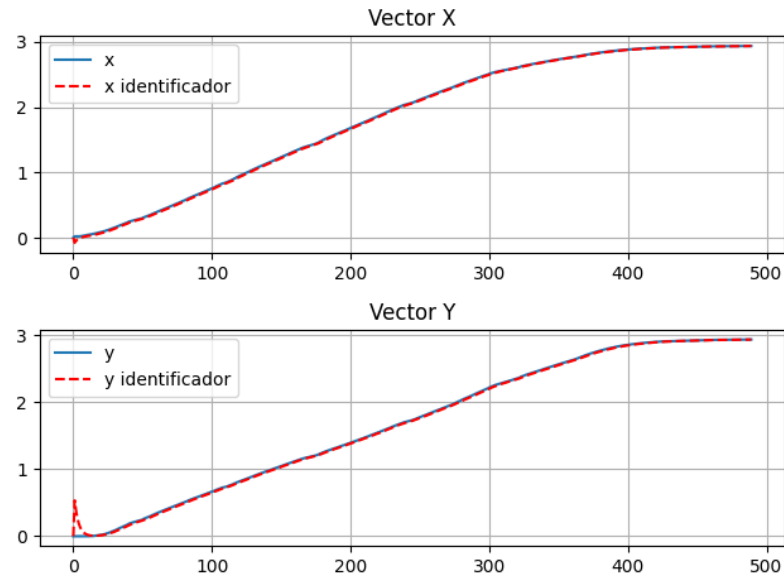


Figura 4.15: Variables con identificador y PID adaptivo experimental en la coordenada  $x = 3$ ,  $y = 3$ .

-0.25 a 0.25, con un cambio más gradual y alcanzando el valor 0 en menor tiempo. Esto demuestra las ventajas de utilizar la combinación del identificador neuronal con el PID adaptativo.

La comparativa del error en las variables  $X$  e  $Y$  respecto al punto deseado se presenta en las Figuras 4.18 y 4.19, donde se hace énfasis en la eficiencia de la convergencia del error cuando se usa el identificador neuronal, como se ilustra en la Figura 4.19, demostrando una transición más gradual y suave.

Tabla 4.2: Comparación del RMS del error en las variables  $X$  y  $Y$  entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto  $X = 3$ ,  $Y = 3$ .

| Método   | RMS del error en X | RMS del error en Y |
|----------|--------------------|--------------------|
| ID+SNPID | 1.4838             | 1.5966             |
| SNPID    | 1.7298             | 1.8124             |

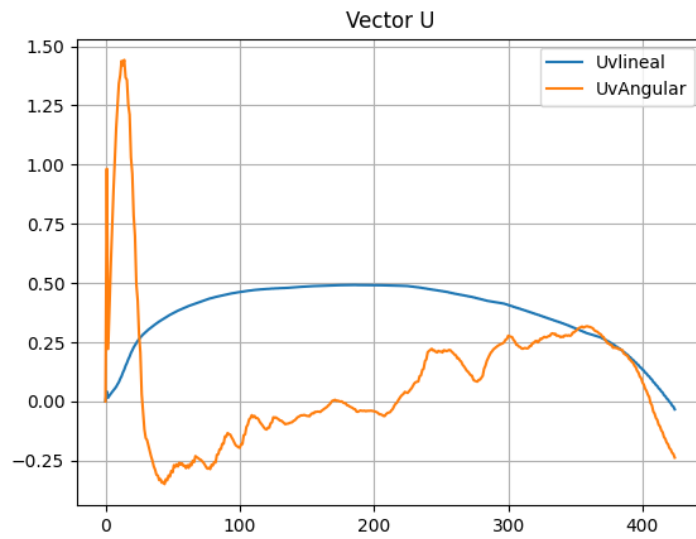


Figura 4.16: Ley de control solo PID adaptivo en la coordenada  $x = 3$ ,  $y = 3$ .

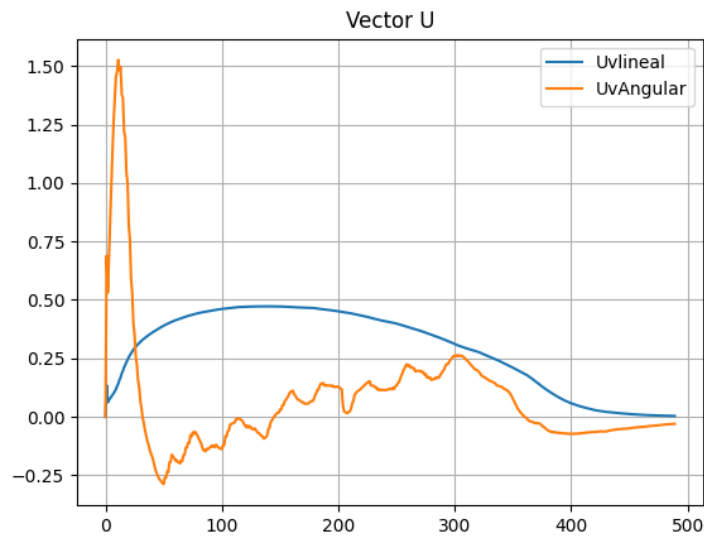


Figura 4.17: Ley de control con identificador y PID adaptivo experimental en la coordenada  $x = 3$ ,  $y = 3$ .

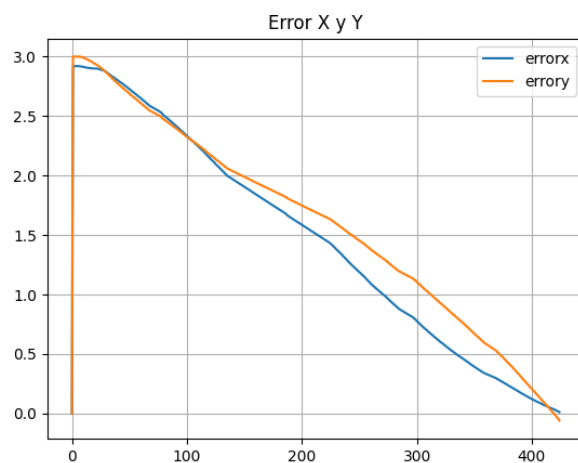


Figura 4.18: Error de variables solo PID adaptivo en la coordenada  $x = 3$ ,  $y = 3$ .

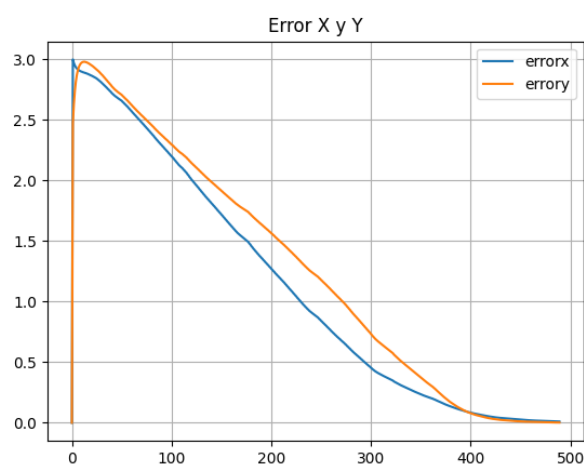


Figura 4.19: Error de variables con identificador y PID adaptivo experimental en la coordenada  $x = 3$ ,  $y = 3$ .

### 4.3.2. Caso 2: Distancia intermedia

Como se menciona en el Caso 1, se comparan los resultados obtenidos en ambas pruebas: el PID adaptativo sin identificador (Figura 4.20) y el PID adaptativo con el identificador neuronal (Figura 4.21), siguiendo una trayectoria libre hasta el punto  $X = 2.5, Y = 1$ . El objetivo es comparar los resultados y destacar las ventajas de combinar ambas técnicas o, en su defecto, determinar si el PID adaptativo por sí solo es igual de efectivo en diversas pruebas experimentales.

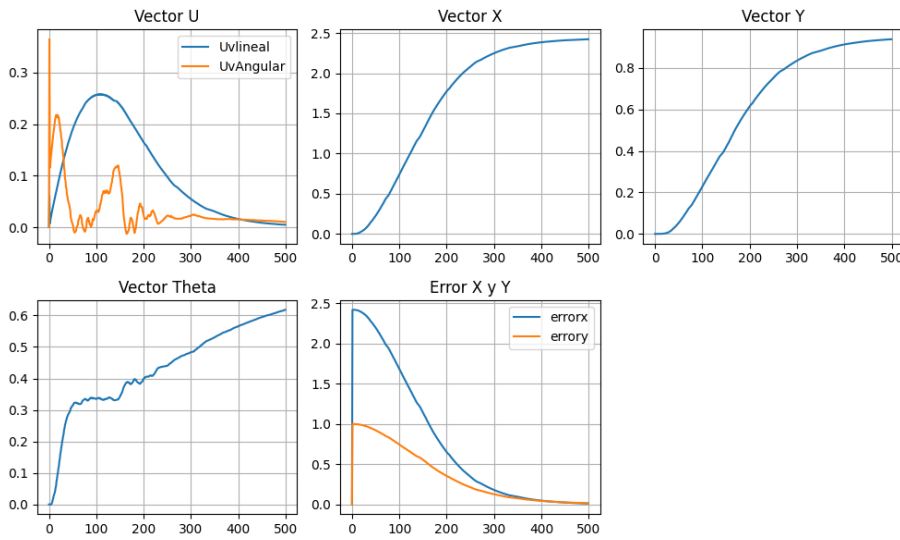


Figura 4.20: PID adaptivo experimental en la coordenada  $x = 2.5, y = 1$ .

Si bien se presentaron casos como los mostrados en las Figuras 4.22 y 4.23, en los cuales los sistemas se comportaron adecuadamente, en la Figura 4.23 se observa que el sobreimpulso de la señal de control de la velocidad angular es disminuido. En ambos casos, se mantiene un desplazamiento gradual y suave, lo que demuestra que la combinación del identificador con el control PID adaptativo es capaz de mantener la robustez del sistema y facilitar la gestión de perturbaciones e incógnitas paramétricas de manera eficiente.

Al comparar los valores enviados para controlar las entradas del sistema de planta desconocida, ilustrados en las Figuras 4.22 y 4.23, se evidencia el sobreimpulso inicial previamente mencionado que disminuye al utilizar el identificador. Sin embargo, en ambos casos el sistema logra estabilizarse en un punto cercano a los 500 ciclos.

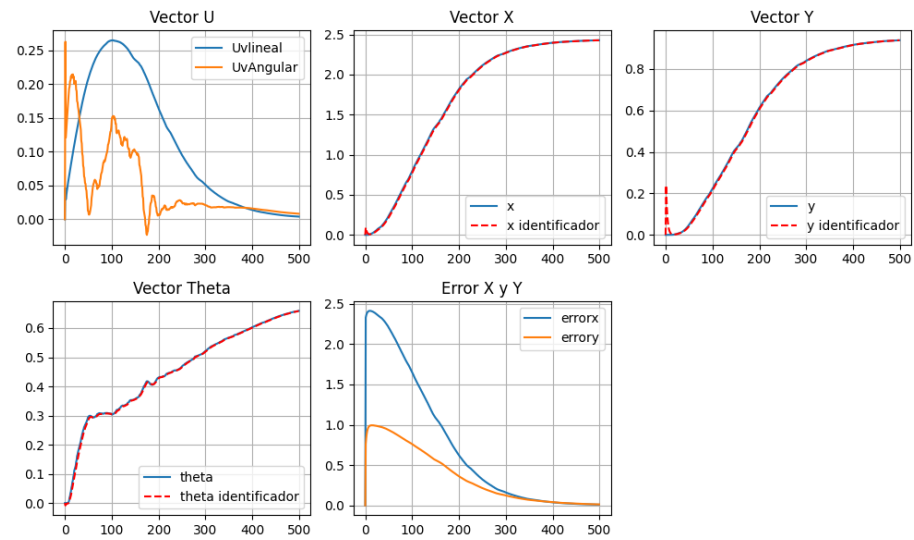


Figura 4.21: Identificador y PID adaptivo experimental en la coordenada  $x = 2.5$ ,  $y = 1$ .

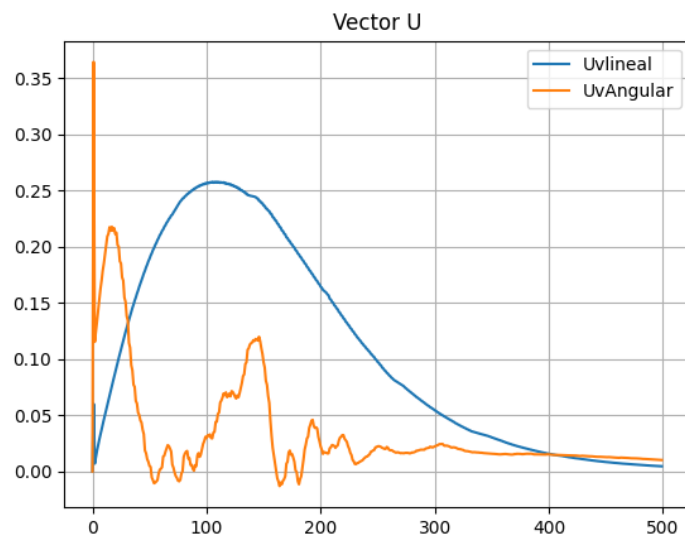


Figura 4.22: Ley de control con PID adaptivo experimental en la coordenada  $x = 2.5$ ,  $y = 1$ .

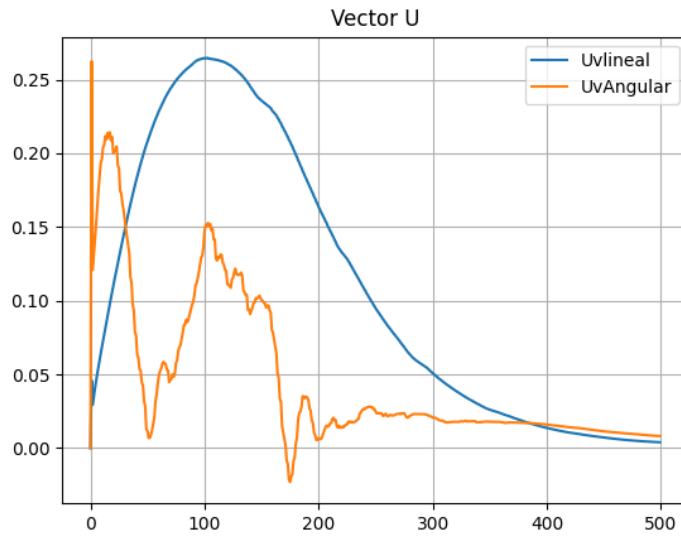


Figura 4.23: Ley de control con identificador y PID adaptivo experimental en la coordenada  $x = 2.5$ ,  $y = 1$ .

El error en las variables  $X$  e  $Y$ , mostrado en las Figuras 4.24 y 4.25, evidencia que la diferencia entre ambas es mínima, dado que el comportamiento de las variables es similar. El objetivo de este caso es demostrar que el identificador y el PID adaptativo pueden trabajar de manera conjunta de forma efectiva, incluso en casos donde los resultados no difieren significativamente del PID adaptativo por sí solo, como se muestra en la Tabla 4.3. Esto suele ocurrir en situaciones experimentales con baja exigencia, ya sea debido a una trayectoria más simple o al desempeño óptimo de ambos sistemas.

Tabla 4.3: Comparación del RMS del error en las variables  $X$  e  $Y$  entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto  $X = 2.5$ ,  $Y = 1$ .

| Método   | RMS del error en X | RMS del error en Y |
|----------|--------------------|--------------------|
| ID+SNPID | 1.1017             | 0.4932             |
| SNPID    | 1.1146             | 0.4894             |



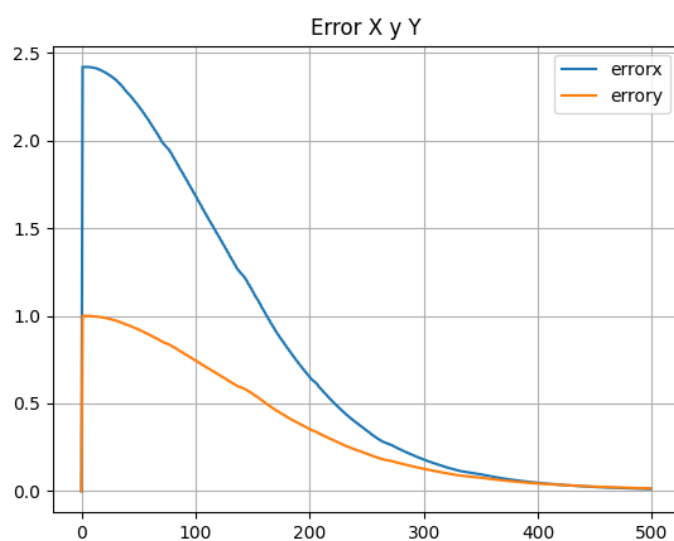


Figura 4.24: Error de variables solo PID adaptivo experimental en la coordenada  $x = 2.5$ ,  $y = 1$ .

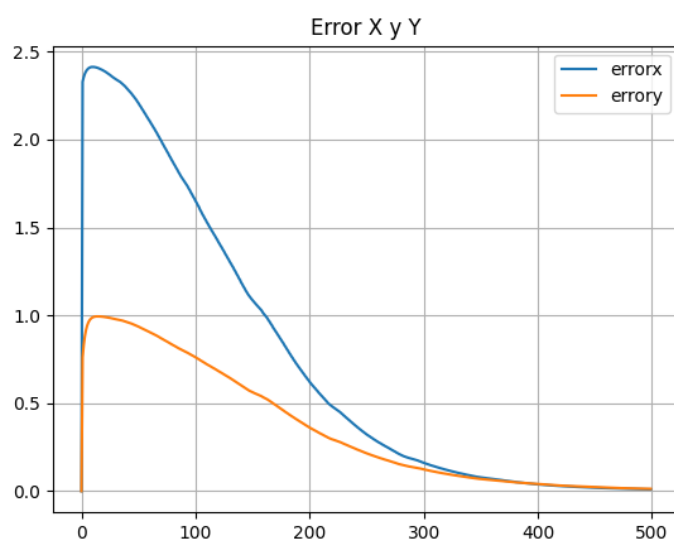


Figura 4.25: Error de variables con identificador y PID adaptivo experimental en la coordenada  $x = 2.5$ ,  $y = 1$ .

### 4.3.3. Caso 3: Menor distancia, más precisión

El caso presentado en las Figuras 4.26 y 4.27 destaca las cualidades del identificador neuronal combinado con el PID adaptivo. Este enfoque demuestra su capacidad para atenuar el sobreimpulso de la señal de control durante los primeros ciclos y reducir el error. Además, se observa que la variable  $\theta$ , la cual se deja con una referencia libre, contribuye a que el error disminuya de manera más suave y controlada, como se muestra más adelante.

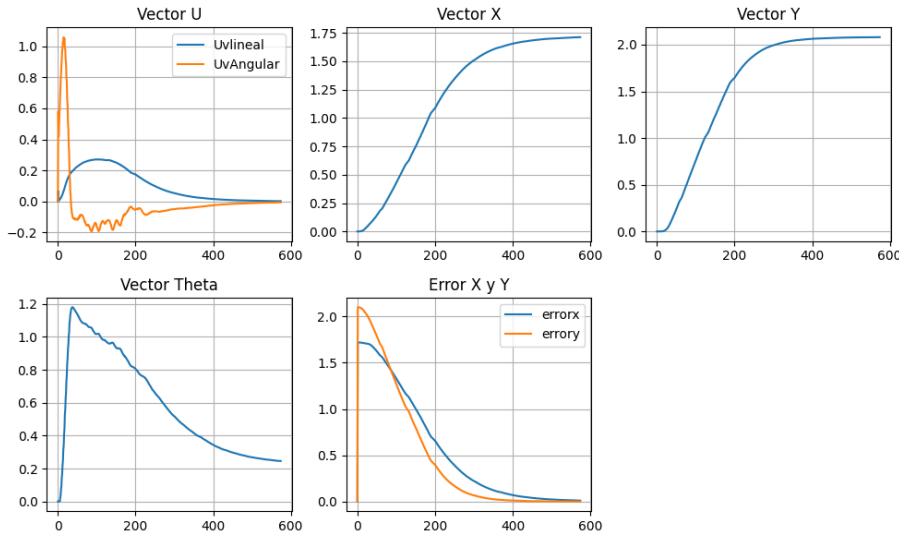


Figura 4.26: PID adaptivo experimental en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

El comportamiento de la ley de control, mostrado en las Figuras 4.28 y 4.29, reafirma la capacidad del identificador neuronal combinado con el PID adaptivo para disminuir el sobreimpulso de la velocidad angular. Experimentalmente, esto se traduce en un giro más suave y controlado, lo que permite alcanzar un error inferior a 0.1 con 25 ciclos menos.

Aunque  $\theta$  es una variable sin referencia fija, permitiendo una trayectoria libre, al comparar las Figuras 4.30 y 4.31, se observa que la Figura 4.31, donde se utiliza el identificador neuronal, muestra un cambio de ángulo con mayor suavidad y control. Además, necesita girar menos en comparación con el uso exclusivo del PID adaptivo. Este comportamiento es una constante en casa caso debido a que esta variable tiene mayor variabilidad cuando el PID adaptivo está solo, demostrando una diferencia clara entre ambas técnicas.

Respecto al error en este caso, se proporciona una referencia con mayor precisión

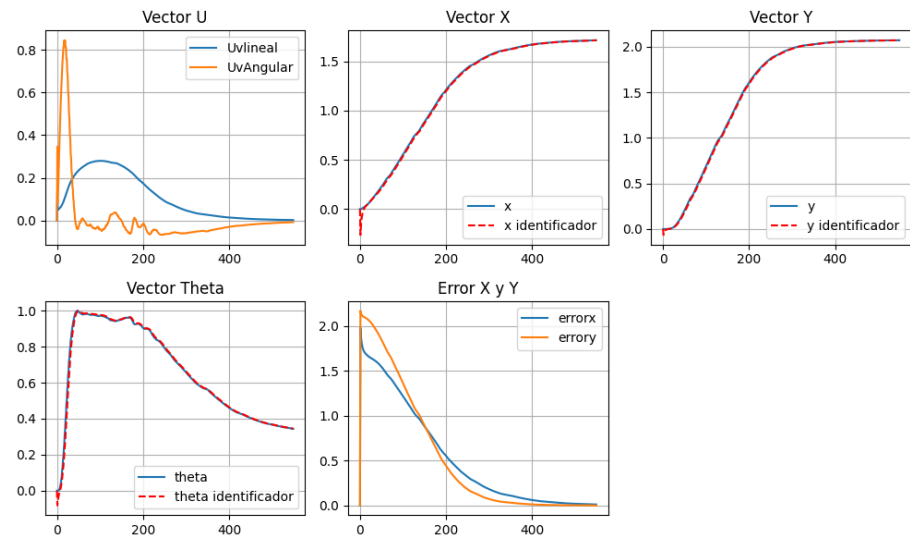


Figura 4.27: Identificador y PID adaptivo experimental en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

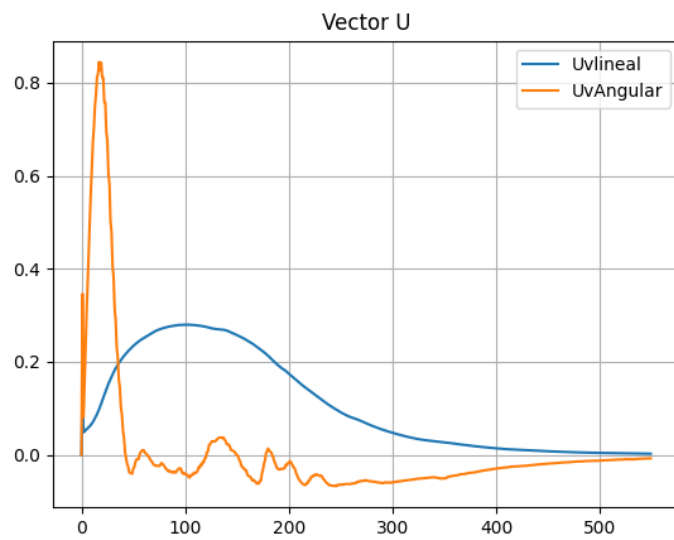


Figura 4.28: Ley de control solo con PID adaptivo en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

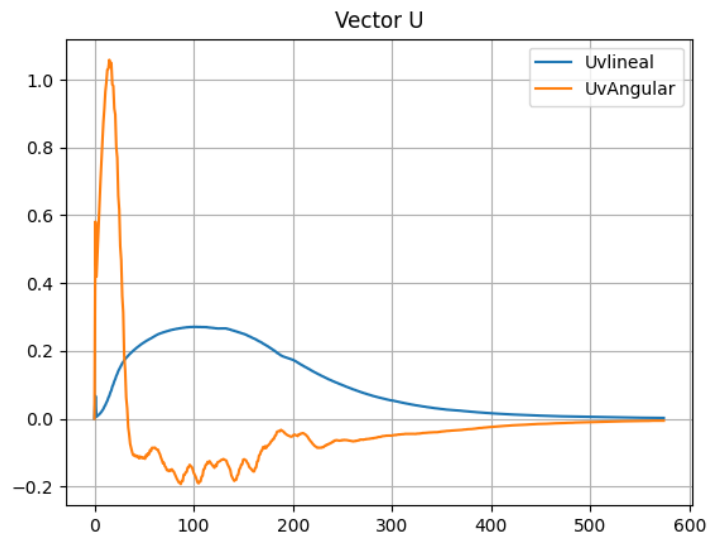


Figura 4.29: Ley de control con identificador y PID adaptivo experimental en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

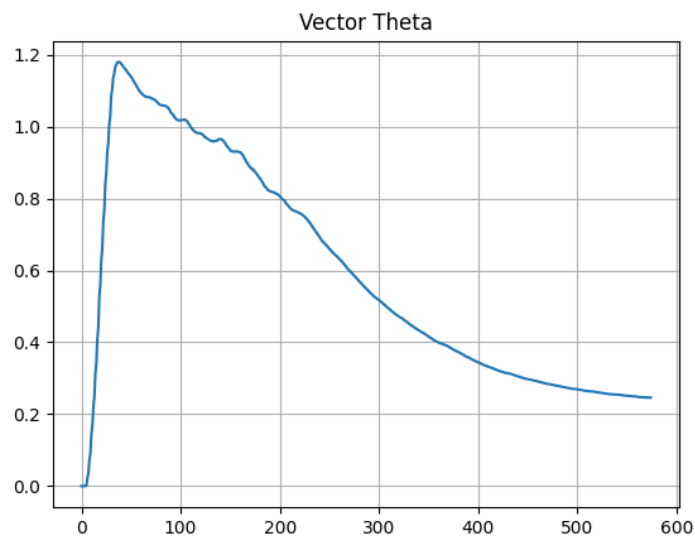


Figura 4.30: Variable  $\theta$  solo con PID adaptivo en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

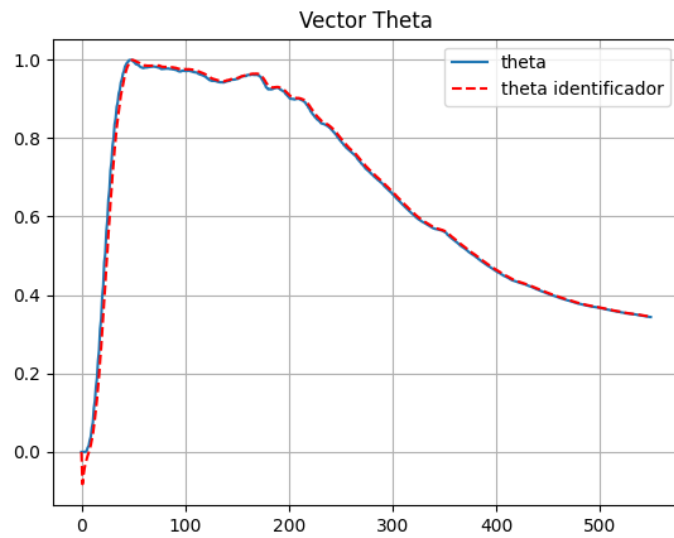


Figura 4.31: Variable  $\theta$  con identificador y PID adaptivo experimental en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

que en los dos casos anteriores, como se ilustra en las Figuras 4.32 y 4.33. En ambos casos, el error alcanza un valor inferior a 0.1. Sin embargo, al analizar la Tabla 4.4, se observa que el error RMS del PID adaptativo en  $X$  es superior en 0.03, mientras que en  $Y$ , el PID adaptativo presenta un error de 0.05. Esto se detalla más claramente mediante la variable  $\theta$ , mostrada en la Figura 4.31, donde se evidencia un cambio más suave en su comportamiento.

Tabla 4.4: Comparación del RMS del error en las variables  $X$  e  $Y$  entre los métodos: Identificador con PID adaptivo y PID adaptivo, para el punto  $X = 1.8$ ,  $Y = 2.1$ .

| Método   | RMS del error en X | RMS del error en Y |
|----------|--------------------|--------------------|
| ID+SNPID | 0.7731             | 0.8790             |
| SNPID    | 0.8088             | 0.8250             |

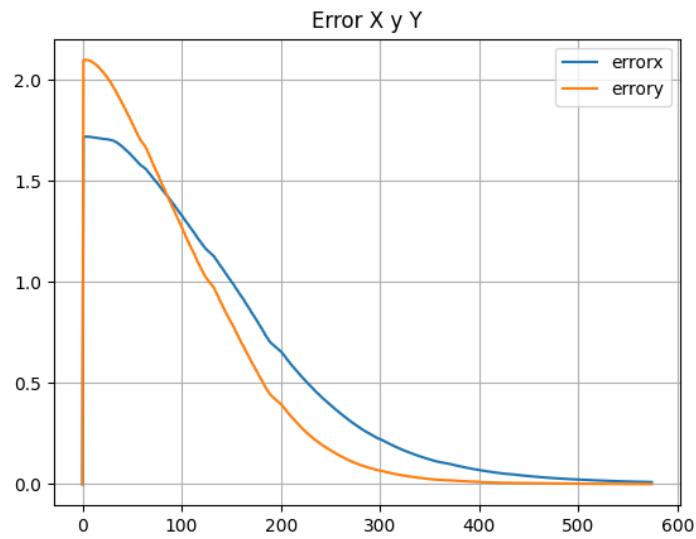


Figura 4.32: Error de variables solo PID adaptivo en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

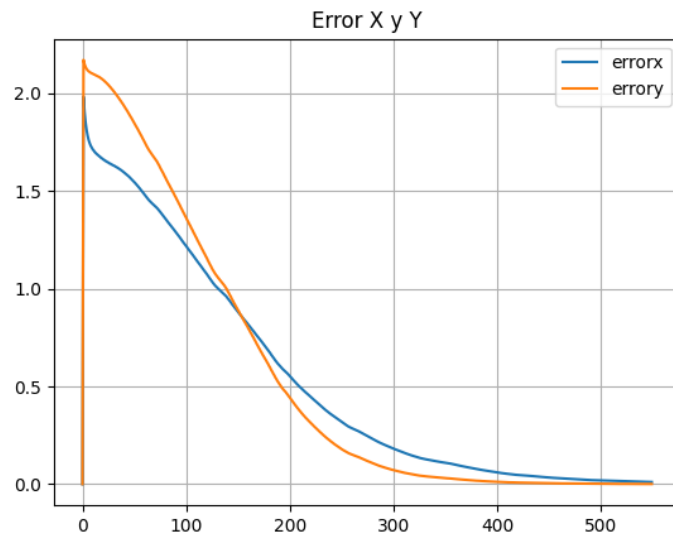


Figura 4.33: Error de variables con identificador y PID adaptivo experimental en la coordenada  $x = 1.8$ ,  $y = 2.1$ .

#### 4.4. Conclusión de Capítulo

El correcto diseño del PID adaptivo para las variable de estado a controlar es indispensable, junto a su entrenamiento por medio del Filtro Extendido de Kalman, ya que esto compensa las carencias del PID clásico al poder adaptarse en tiempo real. En este capítulo, se combinan el identificador neuronal con el control PID adaptivo, el cual es una RHONN de una sola capa, ambos entrenados en tiempo real, por medio del EKF, durante todo el proceso de operación. Esto se demuestra en simulación, evidenciando su adaptabilidad y comparando el funcionamiento del PID adaptivo por sí solo con el identificador neuronal combinado con el PID adaptivo.

Posteriormente, se realizan pruebas experimentales en un robot móvil diferencial "Turtlebot 3 Waffle Pi", permitiendo comparar de nuevo el funcionamiento del conjunto. Los resultados obtenidos resaltan la capacidad del sistema para suavizar el comportamiento del robot, dándole una trayectoria más libre, disminuyendo el sobreimpulso inicial cuando se presenta un error significativo y evitando que el robot tenga un comportamiento de cambio demasiado rápido que afecte su desempeño en pruebas reales.





## Capítulo 5

# Conclusiones

### 5.1. Conclusiones Generales

Este trabajo de tesis aborda la problemática asociada a las dificultades de realizar pruebas experimentales, destacando cómo las incertidumbres en los parámetros, las perturbaciones y el ruido pueden representar un desafío significativo al intentar implementar un control convencional. En este contexto, la alternativa propuesta de utilizar un identificador basado en una RHONN entrenada mediante el Filtro Extendido de Kalman, capaz de realizar un entrenamiento en tiempo real y en línea, demostró ser una herramienta eficaz para mejorar el comportamiento del sistema. Esto se debe a que el sistema equivalente obtenido es capaz de tolerar ruido y perturbaciones en comparación con un modelo clásico.

Cabe recalcar la importancia de usar Python como alternativa para el código fuente, ya que nos permite obtener algoritmos de mayor complejidad sin requerir una gran capacidad de cómputo. Además, facilita el uso de ROS como medio de interacción entre prototipos experimentales, aprovechando medios de comunicación más modernos y robustos frente a fallas en la comunicación.

Adicionalmente, el Control PID adaptivo, también entrenado mediante el EKF, permite superar las limitaciones de un control PID convencional, especialmente en lo que respecta a la sensibilidad frente a perturbaciones, la acumulación de error y la complejidad en el manejo de no linealidades. Sin embargo, un inconveniente a considerar es que, aun-

que las redes neuronales permiten ajustar parámetros o ganancias, estas mismas requieren métodos heurísticos para optimizar su desempeño, garantizar la convergencia y mejorar su precisión. Por ello, se recomienda el uso de algoritmos metaheurísticos que ajusten automáticamente los parámetros de las redes neuronales. A pesar de esta limitación, resulta más eficaz ajustar los valores de una red neuronal mediante algoritmos que adaptar controles clásicos para sistemas altamente complejos, ya sean de respuesta rápida, lenta o no lineales.

Es importante destacar la adaptabilidad del sistema gracias a la selección del EKF como método de entrenamiento. Al trabajar en conjunto el identificador y el controlador adaptativo, se añadieron tres neuronas en línea junto con sus respectivos entrenamientos. En aplicaciones prácticas, las diferencias en el comportamiento del sistema en conjunto (identificador-controlador) respecto al controlador por si solo fueron mínimas, manteniendo una alta precisión en ambos casos. Sin embargo, el uso del conjunto de redes permitió reducir la incertidumbre en los valores finales al alcanzar un punto deseado, además de disminuir la trayectoria recorrida.

Esto permite aproximar el comportamiento del sistema mediante una mayor cantidad de redes más simples, pero eficaces en tareas específicas. Además, permite un enfoque útil para la adaptación a sistemas reales, donde estas redes funcionan como filtros y compensan incertidumbres paramétricas o variaciones entre el sistema real y el modelo clásico.

Una consideración importante al escoger este método es la necesidad de ajustar con precisión las condiciones iniciales, ya que el uso de múltiples redes incrementa la complejidad del proceso. Asimismo, la adaptación de un sistema no lineal mediante una red que genera un modelo equivalente, para luego aplicar un control adaptativo, da lugar a un algoritmo significativamente más grande y complejo.

Finalmente, se confirma el cumplimiento de los objetivos generales y específicos, demostrando que esta técnica conjunta representa una alternativa prometedora para implementaciones experimentales. Este esquema de control permite obtener aproximaciones simples a sistemas complejos, mejorando su tolerancia a perturbaciones y consolidándose como una opción viable en aplicaciones prácticas.

## 5.2. Trabajos Futuros

### Trabajos Futuros

1. Comprobación matemática de estabilidad del uso conjunto del Identificador neuronal y el PID adaptivo.
2. Diseño de una red neuronal recurrente de alto orden (RHONN) clasificadora que nos permita la detección de diferentes tipos de fallas en el sistema.
3. Diseño de un control tolerante a fallas haciendo el uso del identificador, control adaptivo y red neuronal clasificadora.
4. Implementando medidas de ciberseguridad para los robots móviles autónomos así logrando contra medidas robustas.
5. Diseño de un observador neuronal, así logrando el uso de un compensador neuronal que proporcionará a los posibles trabajos previamente mencionados, con su versatilidad y precisión.
6. Diseño de Redes Complejas para el manejo de múltiples robots móviles, terrestres, aéreos o híbridos.



# Referencias

- [Akgun et al., 2019] Akgun, G., Hasan Khan, H. u., Hebaish, M., and Gohringer, D. (2019). System identification using lms, rls, ekf and neural network. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE.
- [Alanis et al., 2020] Alanis, A. Y., Rios, J. D., Gomez-Avila, J., Zuniga, P., and Jurado, F. (2020). Discrete-time neural control of quantized nonlinear systems with delays: Applied to a three-phase linear induction motor. *Electronics*, 9(8):1274.
- [Alverhed et al., 2024] Alverhed, E., Hellgren, S., Isaksson, H., Olsson, L., Palmqvist, H., and Flodén, J. (2024). Autonomous last-mile delivery robots: a literature review. *European Transport Research Review*, 16(1).
- [Arana-Daniel et al., 2020] Arana-Daniel, N., Alanis, A. Y., and Lopez-Franco, C. (2020). *Neural Networks for Robotics: An Engineering Perspective*. Automation and Control Engineering Series. CRC Press.
- [Castro-Romero et al., 2024] Castro-Romero, L. A., Garcia-Lozano, H. N., Santiaguillo-Salinas, J., and Pérez-Castro, N. (2024). Control-observador para seguimiento en un robot diferencial de segundo orden. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 12(Especial2):1–7.
- [Conchas et al., 2023] Conchas, R. F., Sanchez, E. N., Ricalde, L. J., Alvarez, J. G., and Alanis, A. Y. (2023). Sensor fault-tolerant control for a doubly fed induction generator in a smart grid. *Engineering Applications of Artificial Intelligence*, 117:105527.

- [Corke, 2023] Corke, P. (2023). *Robotics, Vision and Control: Fundamental Algorithms in Python*. Springer International Publishing.
- [Dadvar and Habibian, 2022] Dadvar, M. and Habibian, S. (2022). Contemporary research trends in response robotics. *ROBOMECH Journal*, 9(1).
- [Deng et al., 2021] Deng, F., Levi, C., Yin, H., and Duan, M. (2021). Identification of an autonomous underwater vehicle hydrodynamic model using three kalman filters. *Ocean Engineering*, 229:108962.
- [Dorf and Bishop, 2011] Dorf, R. C. and Bishop, R. H. (2011). *Modern Control Systems*. Pearson.
- [Guerrero-Castellanos and González-Romeo, 2021] Guerrero-Castellanos, J. F. and González-Romeo, L. L. (2021). Position control system via active disturbance rejection for laser optical systems. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 19(1):61–73.
- [Gómez Ávila et al., 2020] Gómez Ávila, J., Villaseñor, C., Hernández Barragán, J., Arana-Daniel, N., Alanís, A. Y., and Lopez-Franco, C. (2020). Neural pd controller for an unmanned aerial vehicle trained with extended kalman filter. *Algorithms*, 13(2):40.
- [Haykin, 2004] Haykin, S. (2004). *Kalman filtering and neural networks*. John Wiley & Sons.
- [Haykin, 2011] Haykin, S. (2011). *Neural Networks and Learning Machines*. Pearson Education.
- [Hernández-Barragán et al., 2021] Hernández-Barragán, J., D. Ríos, J., Gómez-ávila, J., Arana-Daniel, N., Lopez-Franco, C., and Alanís, A. Y. (2021). Adaptive neural pd controllers for mobile manipulator trajectory tracking. *PeerJ Computer Science*, 7:e393.
- [Hernández Barragán et al., 2023] Hernández Barragán, J., Hernández, T., Ríos, J. D., Perez-Cisneros, M., and Alanís, A. Y. (2023). Edge-weighted consensus-based formation control with collision avoidance for mobile robots based on multi-strategy mutation differential evolution. *Mathematics*, 11(17):3633.

- [Hernández Barragán et al., 2020] Hernández Barragán, J., Ríos, J. D., Alanís, A. Y., Lopez-Franco, C., Gomez-Avila, J., and Arana-Daniel, N. (2020). Adaptive single neuron anti-windup pid controller based on the extended kalman filter algorithm. *Electronics*, 9(4):636.
- [Khalil, 2002] Khalil, H. K. (2002). *Nonlinear Systems*. Prentice Hall.
- [Khan et al., 2021] Khan, H., Khatoon, S., and Gaur, P. (2021). Comparison of various controller design for the speed control of dc motors used in two wheeled mobile robots. *International Journal of Information Technology*, 13(2):713–720.
- [Lin et al., 2022] Lin, S., Liu, A., Wang, J., and Kong, X. (2022). A review of path-planning approaches for multiple mobile robots. *Machines*, 10(9):773.
- [Lu et al., 2024] Lu, Y., Huang, J., Jiang, Z., Tang, T., Tang, H., and Shi, L. (2024). Pid adaptive feedback motor system based on neural network. *IEEE Access*, 12:60149–60154.
- [Reis et al., 2022] Reis, W. P. N. d., Couto, G. E., and Junior, O. M. (2022). Automated guided vehicles position control: a systematic literature review. *Journal of Intelligent Manufacturing*, 34(4):1483–1545.
- [Rios, 2022] Rios, Y., G.-R. J. S. E. A. A. R.-V. E. . P.-G. A. (2022). On board neuro fuzzy inverse optimal control for type 1 diabetes mellitus treatment: In-silico testing. *Advances in Electrical and Computer Engineering*, 22(3):3–14.
- [Rios et al., 2020] Rios, J. D., Alanis, A. Y., Arana-Daniel, N., and Lopez-Franco, C. (2020). *Neural Networks Modeling and Control: Applications for Unknown Nonlinear Delayed Systems in Discrete Time*. Academic Press.
- [ROBOTIS, 2024a] ROBOTIS (2024a). Turtlebot3 e-manual: Features.
- [ROBOTIS, 2024b] ROBOTIS (2024b). Turtlebot3 e-manual: Lds-02 appendix.
- [Rodriguez-Cortes and Aranda-Bricaire, 2007] Rodriguez-Cortes, H. and Aranda-Bricaire, E. (2007). Observer based trajectory tracking for a wheeled mobile robot. In *2007 American Control Conference*. IEEE.

- [Rossomando, 2014] Rossomando, F. G. (2014). Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural pid. *Neural Computing and Applications*, 26(5):1179–1191.
- [Rovithakis and Christodoulou, 2012] Rovithakis, G. A. and Christodoulou, M. A. (2012). *Adaptive control with recurrent high-order neural networks: theory and industrial applications*. Springer Science & Business Media.
- [Ríos et al., 2022] Ríos, J. D., Ríos-Rivera, D., Hernández Barragán, J., Pérez-Cisneros, M., and Alanís, A. Y. (2022). Formation control of mobile robots based on pin control of complex networks. *Machines*, 10(10):898.
- [Sánchez and Alanis, 2006] Sánchez, E. and Alanis, A. (2006). *Redes neuronales: conceptos fundamentales y aplicaciones a control automático*. Automática y Robótica. Pearson Educación.
- [Siegwart et al., 2011] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. Intelligent Robotics and Autonomous Agents series. MIT Press, second edition.
- [Siwek et al., 2023] Siwek, M., Panasiuk, J., Baranowski, L., Kaczmarek, W., Prusaczyk, P., and Borys, S. (2023). Identification of differential drive robot dynamic model parameters. *Materials*, 16(2):683.
- [Tian et al., 1999] Tian, Y.-C., Tade, M. O., and Tang, J. (1999). A nonlinear pid controller with applications. *IFAC Proceedings Volumes*, 32(2):2657–2661.
- [Villalobos Álvarez, 2022] Villalobos Álvarez, E. F. (2022). Control de múltiples robots no holonómicos bajo un esquema líder seguidor. Master’s thesis, Universidad Tecnológica de la Mixteca, H. Cd. de Huajuapán de León, Oaxaca, México.
- [Vázquez et al., 2021] Vázquez, U., González-Sierra, J., Fernández-Anaya, G., and Hernández-Martínez, E. G. (2021). Análisis del desempeño de un control pid de orden —onal en un robot móvil diferencial. *Revista Iberoamericana de Automática e Informática industrial*, 19(1):74–83.



- 
- [Yousfi Allagui et al., 2021] Yousfi Allagui, N., Salem, F. A., and Aljuaid, A. M. (2021). Artificial fuzzy-pid gain scheduling algorithm design for motion control in differential drive mobile robotic platforms. *Computational Intelligence and Neuroscience*, 2021:1–13.
- [Zhang et al., 2018] Zhang, H.-y., Lin, W.-m., and Chen, A.-x. (2018). Path planning for the mobile robot: A review. *Symmetry*, 10(10):450.
- [Zhu et al., 2019] Zhu, Q., Han, Y., Liu, P., Xiao, Y., Lu, P., and Cai, C. (2019). Motion planning of autonomous mobile robot using recurrent fuzzy neural network trained by extended kalman filter. *Computational Intelligence and Neuroscience*, 2019:1–16.

# Manuel Elias Pech Dzul

## Control PID neuronal e identificación de sistemas por medio de redes neuronales y filtro de Kalman e

 Universidad Michoacana de San Nicolás de Hidalgo

### Detalles del documento

Identificador de la entrega

trn:oid:::3117:458225961

Fecha de entrega

12 may 2025, 10:11 a.m. GMT-6

Fecha de descarga

12 may 2025, 10:58 a.m. GMT-6

Nombre de archivo

Control PID neuronal e identificación de sistemas por medio de redes neuronales y filtro de Kalm....pdf

Tamaño de archivo

14.0 MB

113 Páginas

20.411 Palabras

104.494 Caracteres

# Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado  
Universidad Michoacana de San Nicolás de Hidalgo



A quien corresponda,

Por este medio, quien abajo firma, bajo protesta de decir verdad, declara lo siguiente:

- Que presenta para revisión de originalidad el manuscrito cuyos detalles se especifican abajo.
- Que todas las fuentes consultadas para la elaboración del manuscrito están debidamente identificadas dentro del cuerpo del texto, e incluidas en la lista de referencias.
- Que, en caso de haber usado un sistema de inteligencia artificial, en cualquier etapa del desarrollo de su trabajo, lo ha especificado en la tabla que se encuentra en este documento.
- Que conoce la normativa de la Universidad Michoacana de San Nicolás de Hidalgo, en particular los Incisos IX y XII del artículo 85, y los artículos 88 y 101 del Estatuto Universitario de la UMSNH, además del transitorio tercero del Reglamento General para los Estudios de Posgrado de la UMSNH.

| Datos del manuscrito que se presenta a revisión |                                                                                                                                                    |                               |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Programa educativo                              | Maestro en Ciencias en Ingeniería Eléctrica                                                                                                        |                               |
| Título del trabajo                              | CONTROL PID NEURONAL E IDENTIFICACIÓN DE SISTEMAS POR MEDIO DE REDES NEURONALES Y FILTRO DE KALMAN EXTENDIDO APLICADO A UN ROBOT MÓVIL DIFERENCIAL |                               |
|                                                 | Nombre                                                                                                                                             | Correo electrónico            |
| Autor/es                                        | Manuel Elias Pech Dzul                                                                                                                             | manuel.elias.pech@umich.mx    |
| Director                                        | Juan Anzures Marín                                                                                                                                 | juan.anzures@umich.mx         |
| Codirector                                      | Alma Yolanda Alanís García                                                                                                                         | alma.alanis@academicos.udg.mx |
| Coordinador del programa                        | Norberto Garcia Barriga                                                                                                                            | norberto.garcia@umich.mx      |

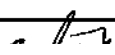
| Uso de Inteligencia Artificial |             |             |
|--------------------------------|-------------|-------------|
| Rubro                          | Uso (sí/no) | Descripción |
| Asistencia en la redacción     | no          |             |

# Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado  
Universidad Michoacana de San Nicolás de Hidalgo



| Uso de Inteligencia Artificial             |             |                                                                                      |
|--------------------------------------------|-------------|--------------------------------------------------------------------------------------|
| Rubro                                      | Uso (sí/no) | Descripción                                                                          |
| Traducción al español                      | no          |                                                                                      |
| Traducción a otra lengua                   | no          |                                                                                      |
| Revisión y corrección de estilo            | no          |                                                                                      |
| Análisis de datos                          | no          |                                                                                      |
| Búsqueda y organización de información     | no          |                                                                                      |
| Formateo de las referencias bibliográficas | si          | Estructurar las bibliografías en el formato correspondiente y verificar las propias. |
| Generación de contenido multimedia         | no          |                                                                                      |
| Otro                                       |             |                                                                                      |

| Datos del solicitante |                                                                                                                                   |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Nombre y firma        | <br>Manuel Elias Pech Dzul                   |
| Lugar y fecha         | <br>Morelia, Michoacán a 9 de mayo del 2025. |