

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

INGENIERÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN

**DISEÑO, PROGRAMACIÓN Y SIMULACIÓN DE
CONTROLADORES INTELIGENTES APLICADOS A UN ROBOT
MÓVIL DE TRACCIÓN DIFERENCIAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y AUTOMATIZACIÓN**

KEVIN ALEXANDER PUENTES BENALCÁZAR

kevin.puentes@epn.edu.ec

DIRECTOR:

DR. LUIS ALBERTO MORALES ESCOBAR

luis.morales@epn.edu.ec

Quito, abril 2023

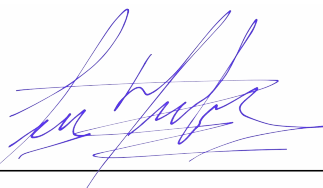
CERTIFICACIONES

Yo, Kevin Alexander Puentes Benalcázar declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Kevin Alexander Puentes Benalcázar

Certifico que el presente trabajo de integración curricular fue desarrollado por Kevin Alexander Puentes Benalcázar, bajo mi supervisión.



Luis Alberto Morales Escobar
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



Kevin Alexander Puentes Benalcázar



Luis Alberto Morales Escobar

DEDICATORIA

Dedico este trabajo a mis padres Nancy y Belisario, y a mi tía Irma, por motivarme e inspirarme a culminar esta etapa académica y seguir avanzando hacia un mejor futuro.

AGRADECIMIENTO

A mi familia, profesores y amigos que han aportado a la elaboración de este trabajo, a todos ellos gracias por siempre estar presentes.

CONTENIDO

DECLARACIÓN DE AUTORÍA	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
CONTENIDO	v
LISTA DE FIGURAS	vi
LISTA DE TABLAS	x
RESUMEN.....	xi
ABSTRACT	xii
1. INTRODUCCIÓN	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcances.....	2
1.4 Marco teórico.....	3
1.4.1 Sistema de control en lazo cerrado	3
1.4.2 Métodos de aproximación a un modelo FOPTD.....	4
1.4.3 Robótica móvil	5
1.4.4 Robot móvil de tracción diferencial.....	6
1.4.5 Tipos de robots de tracción diferencial en V-REP	6
1.4.6 Redes neuronales.....	7
1.4.6.1 Elementos básicos que componen una red neuronal	8
1.4.6.2 Mecanismos de aprendizaje.....	10
1.4.6.3 Control Neuronal	11
1.4.7 Índices de desempeño	12
2. METODOLOGÍA	13
2.1 Redes Neuronales Aprendizaje	13
2.1.1 Algoritmo de Retropropagación.....	13
2.2 Robot Pioneer 3DX.....	16
2.2.1 Modelo cinemático	18

2.2.2	Modelo dinámico	19
2.2.3	Generación de trayectorias	20
2.2.4	Diseño de controladores cinemático y dinámico.....	21
2.2.4.1	Controlador Cinemático.....	21
2.2.4.2	Controladores Dinámicos	22
2.3	Interfaz Gráfica	30
3.	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	31
3.1	Sintonización de controladores.....	31
3.2	Prueba de seguimiento de trayectorias.....	32
3.2.1	Trayectoria Cuadrada	32
3.2.2	Trayectoria Circular.....	36
3.2.3	Trayectoria Lemniscata.....	38
3.2.4	Trayectoria Abierta.....	41
3.3	Pruebas ante perturbaciones.....	43
3.3.1	Trayectoria Cuadrada con Plano Inclinado.....	43
3.3.2	Trayectoria Circular con Plano Inclinado	46
3.3.3	Trayectoria Lemniscata con Plano Inclinado	49
3.3.4	Trayectoria Abierta con Plano Inclinado	51
3.4	CONCLUSIONES.....	54
3.5	RECOMENDACIONES.....	56
4.	BIBLIOGRAFÍA	57
5.	ANEXOS	62

LISTA DE FIGURAS

Figura 1.1.	Esquema de un sistema de control en lazo cerrado [10]	3
Figura 1.2.	(a) Señal paso, (b) Curva de reacción del sistema [3], [10]	4
Figura 1.3.	Robots móviles de tracción diferencial	6
Figura 1.6.	Ejemplo de una red neuronal multicapa [12], [14].....	8
Figura 1.7.	Esquema de aprendizaje supervisado [12].....	10

Figura 1.8. Esquema de aprendizaje no supervisado [12].....	11
Figura 1.9. Esquema de control inverso con aprendizaje no supervisado [25]	11
Figura 1.10. Esquema de control PID+NN [25]	12
Figura 2.1. (a) Robot móvil Pioneer 3DX. (b) Dimensiones en milímetros del robot Pioneer 3DX	17
Figura 2.2. Parámetros del robot móvil de tracción diferencial [34]	18
Figura 2.3. Respuesta paso de velocidad (a) lineal y, (b) angular de robot Pioneer 3DX	19
Figura 2.4. Red neuronal en controlador PID Neuronal.....	23
Figura 2.5. Esquema de control PID Neuronal para seguimiento de trayectoria de un robot móvil usando redes neuronales	25
Figura 2.6. Red neuronal PID+NN	27
Figura 2.7. Esquema de control PID+NN para seguimiento de trayectoria de un robot móvil usando redes neuronales	29
Figura 2.8. Enlace entre el entorno Matlab y V-REP [34]	30
Figura 2.9. Interfaz Gráfica.....	30
Figura 3.1. Creación de trayectoria mediante el interfaz de gráfica	32
Figura 3.2. Seguimiento de trayectoria cuadrada de 3 m de lado.....	33
Figura 3.3. Error cuadrático instantáneo de posición del seguimiento de trayectoria cuadrada de 3 m de lado a una velocidad lineal de 0,2 m/s.....	33
Figura 3.4. Velocidad lineal de 0,2 m/s para el seguimiento de trayectoria cuadrada de 3 m de lado	34
Figura 3.5. Velocidad angular para el seguimiento de trayectoria cuadrada de 3 m de lado para una velocidad lineal de 0,2 m/s.....	34
Figura 3.6. Comparación cuantitativa para figura cuadrada	35
Figura 3.7. (a) Seguimiento de trayectoria circular de 1,5 m de radio, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular	36
Figura 3.8. (a) Seguimiento de trayectoria circular de 2,5 m de radio, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,2 m/s	36
Figura 3.9. Seguimiento de trayectoria circular de 2,5 m de radio, velocidad (a) lineal de 0,2 m/s y (b) angular.....	37

Figura 3.10. Comparación cuantitativa para figura circular (a) ISE, (b) IAE	37
Figura 3.11. Comparación cuantitativa para figura circular (a) ISCO U, (b) ISCO W ..	37
Figura 3.12. (a) Seguimiento de trayectoria lemniscata de 1,5 m de radio, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,3 m/s	38
Figura 3.13. Seguimiento de trayectoria lemniscata de 1,5 m de radio velocidad (a) lineal de 0,3 m/s y (b) angular.....	39
Figura 3.14. (a) Seguimiento de trayectoria lemniscata de 2,5 m de radio, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular	39
Figura 3.15. Comparación cuantitativa en figura lemniscata (a) ISE, (b) IAE	40
Figura 3.16. Comparación cuantitativa en figura lemniscata (a) ISCO U, (b) ISCO W	40
Figura 3.17. (a) Seguimiento de trayectoria abierta, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular.....	41
Figura 3.18. (a) Seguimiento de trayectoria abierta, (b) error cuadrático instantáneo de posición a una velocidad lineal de 0,2 m/s.....	41
Figura 3.19. Seguimiento de trayectoria abierta a velocidad (a) lineal de 0,2 m/s y (b) angular	42
Figura 3.20. Comparación cuantitativa (a) ISE, (b) IAE, en la trayectoria abierta	42
Figura 3.21. Comparación cuantitativa (a) ISCO U, (b) ISCO W, en la trayectoria abierta	42
Figura 3.22. Plano inclinado.....	43
Figura 3.23. (a) Seguimiento de trayectoria cuadrada de 3 m de lado en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular	44
Figura 3.24. (a) Seguimiento de trayectoria cuadrada de 5 m de lado en plano inclinado, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,2 m/s..	44
Figura 3.25. Seguimiento de trayectoria cuadrada de 5 m de lado en plano inclinado, velocidad (a) lineal de 0,2 m/s y (b) angular	45
Figura 3.26. Comparación cuantitativa (a) ISE, (b) IAE, en trayectorias cuadradas en plano inclinado.....	45
Figura 3.27. Comparación cuantitativa (a) ISCO U, (b) ISCO W, en trayectorias cuadradas en plano inclinado.	45

Figura 3.28. (a) Seguimiento de trayectoria circular de 1,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, a velocidad lineal de 0,3 m/s.	46
Figura 3.29. Seguimiento de trayectoria circular de 1,5 m de radio en plano inclinado, velocidad (c) lineal de 0,3 m/s y (d) angular.....	47
Figura 3.30. (a) Seguimiento de trayectoria circular de 2,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular	47
Figura 3.31. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria circular con plano inclinado	48
Figura 3.32. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria circular con plano inclinado	48
Figura 3.33. (a) Seguimiento de trayectoria lemniscata de 1,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular	49
Figura 3.34. (a) Seguimiento de trayectoria lemniscata de 2,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular	50
Figura 3.35. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria lemniscata con plano inclinado.....	50
Figura 3.36. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria lemniscata con plano inclinado	51
Figura 3.38. (a) Seguimiento de trayectoria abierta en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular	52
Figura 3.39. (a) Seguimiento de trayectoria abierta en plano inclinado, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,2 m/s	52
Figura 3.40. Seguimiento de trayectoria abierta en plano inclinado, velocidad (a) lineal de 0,2 m/s y (b) angular.....	53
Figura 3.41. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria abierta con plano inclinado	53
Figura 3.42. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria abierta con plano inclinado.....	53
Figura I.1. Pantalla de inicio	62
Figura I.2. Interfaz de Usuario	63

Figura I.3. Selección de trayectoria	64
Figura I.4. Tamaño y tiempo de trayectoria	64
Figura I.5. Selección del controlador	65
Figura I.6. Selección del Plano.....	65
Figura I.7. Simulación V-REP.....	66
Figura I.8. Simulación Interfaz de Usuario.....	66
Figura I.9. Simulación detenida V-REP	67
Figura I.10. Gráfica de resultados	67
Figura I.11. Navegación de resultados.....	68
Figura I.12. Habilitación de comparación de controladores	69
Figura I.13. Comparación de resultados de controladores	69
Figura I.14. Generación de nueva trayectoria	70

LISTA DE TABLAS

Tabla 1.1. Robots móviles de tracción diferencial en VREP	7
Tabla 2.1. Especificaciones técnicas del Robot Pioneer 3DX.....	17
Tabla 2.2. Parámetros del FOPDT de la velocidad lineal y angular del robot móvil Pioneer 3DX.....	20
Tabla 2.3. Ecuaciones consideradas para la generación de trayectorias	20
Tabla 3.1. Parámetros del controlador PID	31

RESUMEN

En este documento se presenta un análisis para la implementación de controladores inteligentes basados en redes neuronales para el seguimiento de trayectorias de un robot móvil.

El sistema es simulado mediante el software Virtual Robot Experimentation Platform (V-REP), el cual contiene un entorno de desarrollo integrado basado en una arquitectura de control distribuido con objetos que pueden ser controlados a través de un script, creado dentro de la plataforma mismo o comandado por un cliente remoto Application Programming Interface (API).

Se hará uso del robot móvil diferencial Pioneer 3XD y mediante la plataforma de cliente remoto, se comandará el robot desde Matlab, ya que este software posee un entorno de escritorio perfeccionado para el análisis y cálculo numérico de complejos algoritmos como lo son los controladores inteligentes (por ejemplo, basados en redes neuronales) a desarrollar en este proyecto.

El esquema propuesto consiste en el diseño de un controlador cinemático basado en el modelo inverso para control de posición que permita el seguimiento de trayectorias y un controlador dinámico basado en redes neuronales que, ante cambios del entorno del robot, permita controlar su velocidad lineal y angular.

PALABRAS CLAVE: Control Cinemático, Control dinámico, Control Inteligente, Redes Neuronales, V-REP.

ABSTRACT

This document presents an analysis for the implementation of intelligent controllers based on neural networks for trajectory tracking of a mobile robot.

The system is simulated using the Virtual Robot Experimentation Platform (V-REP) software, which contains an integrated development environment based on a distributed control architecture with objects that can be controlled through a script, created within the platform itself. or commanded by a remote client Application Programming Interface (API).

The Pioneer 3XD differential mobile robot will be used and through the remote client platform, the robot will be commanded from Matlab, since this software has a perfected desktop environment for the analysis and numerical calculation of complex algorithms such as the intelligent controllers (for example, based on neural networks) to be developed in this project.

The proposed scheme consists of the design of a kinematic controller based on the inverse model for position control that allows trajectory tracking and a dynamic controller based on neural networks that, when faced with changes in the robot's environment, allows its linear and angular speed to be controlled.

KEYWORDS: Dynamic Control, Intelligent Control, Kinematic Control, Neural Networks, V-REP.

1. INTRODUCCIÓN

La robótica móvil es una herramienta que permite el desarrollo de tecnologías de navegación autónoma, debido a su gran utilidad ha sido empleada en áreas industriales como: exploración y mapeo en ambientes peligrosos, sanidad, tácticas militares y extinción de incendios, también en ambientes educativos, así como en el sector del entretenimiento. Por lo tanto, la navegación de robots móviles debe tener un alto grado de precisión al seguir una trayectoria. El algoritmo para el control de los robots móviles se clasifica en [1]–[5]:

- Control aplicado al modelo cinemático y,
- Control aplicado al modelo dinámico.

La navegación de los robots móviles se basa en el modelo cinemático, donde se estudia el movimiento con respecto a una trayectoria generada, por lo que, el control cinemático tiene como fin la disminución del error al seguir dicha trayectoria.

Por otro lado, el modelo dinámico describe directamente la relación entre las fuerzas y pares generados por cada uno de los motores, donde la velocidad línea y angular son considerados entradas del sistema del robot.

Se han publicado varios estudios sobre el diseño de controladores aplicados a robots móviles [2], [5]–[9]. Por lo que, la motivación de este trabajo se ha enfocado en hacer uso de controladores inteligentes para mejorar el desempeño en seguimiento de trayectorias considerando que la dinámica del robot puede verse afectada por factores externos, y es indispensable que el controlador presente características inteligentes, es decir, tenga la capacidad de aprender y auto-ajustarse para mantener un comportamiento adecuado.

En base a lo antes descrito, se propone el diseño de un controlador basado en redes neuronales aplicado a la dinámica del robot, y un control cinemático para el control de posición. Estos dos controladores serán aplicados al sistema en configuración cascada (controlador neuronal interno, y controlador cinemático externo) para realizar el seguimiento de trayectorias. Adicionalmente, se cambiará el entorno de desplazamiento del robot de modo que la dinámica del robot se vea afectada con la finalidad de evaluar el desempeño del controlador neuronal ante perturbaciones. Finalmente, se realizarán pruebas que validen el funcionamiento del sistema ante diferentes trayectorias y se realizará un análisis comparativo con técnicas de control convencionales.

1.1 Objetivo general

Diseñar, programar y simular un controlador basado en redes neuronales aplicado a un robot móvil de tracción diferencial para seguimiento de trayectorias.

1.2 Objetivos específicos

- Realizar una revisión bibliográfica de los diferentes tipos de robots móviles de tracción diferencial existentes en el software Virtual Robot Experimentation Platform y una revisión bibliográfica de controladores inteligentes basados en redes neuronales para control de sistemas robóticos.
- Realizar el diseño de un controlador basado en redes neuronales aplicado al modelo dinámico de un robot móvil de tracción diferencial en tareas de seguimiento de trayectorias y su respectivo controlador cinemático.
- Programar los controladores propuestos en el software Matlab e implementar una interfaz de usuario para operar el sistema a través de una estructura Software in Loop.
- Simular los controladores programados en Matlab utilizando el modelo virtual del robot de tracción diferencial dispuesto en el software Virtual Robot Experimentation Platform.
- Realizar pruebas de funcionamiento de los controladores cinemático y dinámico para validarlos en seguimiento de diferentes trayectorias, y evaluar su comportamiento mediante índices de desempeño.

1.3 Alcances

- Se realizará una revisión bibliográfica de los diferentes tipos de robots móviles de tracción diferencial existentes en el software Virtual Robot Experimentation Platform y una revisión bibliográfica de controladores inteligentes basados en redes neuronales para control de sistemas robóticos.
 - Se realizará el diseño de dos controladores basados en redes neuronales aplicados al modelo dinámico de un robot móvil de tracción diferencial en tareas de seguimiento de trayectorias y su respectivo controlador cinemático.
 - Se programará los controladores propuestos en el software Matlab e implementar una interfaz de usuario para operar el sistema a través de una estructura Software in Loop.
 - Se simulará los controladores programados en Matlab utilizando el modelo virtual del robot de tracción diferencial dispuesto en el software Virtual Robot Experimentation Platform.

- Se realizará pruebas de funcionamiento de los controladores cinemático y dinámicos para validarlos en seguimiento de diferentes trayectorias, y evaluar su comportamiento mediante índices de desempeño.

1.4 Marco teórico

En esta sección se hará una descripción bibliográfica de los sistemas de control para el seguimiento de trayectoria de un robot móvil, el tipo de tracción para el robot y una introducción a las redes neuronales para la implantación de controladores basados en redes neuronal, que se desarrollará en el Capítulo 2.

1.4.1 Sistema de control en lazo cerrado

Un sistema de control en lazo cerrado se caracteriza por retroalimentar la variable de salida y compararla con la referencia (calculando el error) para controlar una variable del proceso [10].

Los controladores por retroalimentación son aquellos que para mantener el punto de operación basan su decisión en el cálculo del error.

La compensación de la perturbación, por medio de la variable controlada es desviada de su punto de control (debido a una perturbación), el controlador realiza la acción de cambiar su salida con el objetivo de lograr el regreso de la variable a su punto de control. En la Figura 1.1. se observa un ejemplo de un sistema de control realimentado.

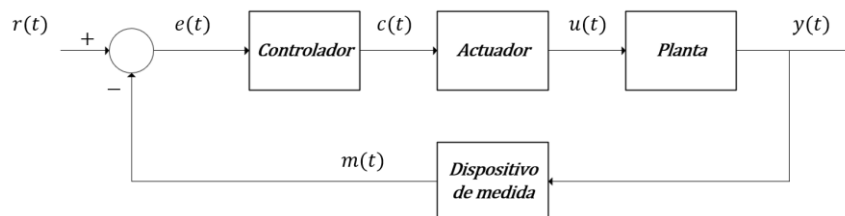


Figura 1.1. Esquema de un sistema de control en lazo cerrado [10]

Donde [11]:

- Señal de referencia $r(t)$: La entrada (o referencia) es el valor deseado a ser obtenido mediante el control de la planta. Esta referencia puede ser estática o variable como una trayectoria, velocidad lineal o angular.
- Error actuante $e(t)$: Es la diferencia entre la señal de referencia y la señal medida como se observa en la Ecuación (1.1). Los controladores diseñados en este documento tienen el objetivo de minimizar el error del seguimiento de trayectoria, velocidad lineal y velocidad angular.

$$e(t) = r(t) - m(t) \quad (1.1)$$

- Señal de control $c(t)$: La variable de control es la señal enviada hacia el actuador del sistema físico para controlar la variable de salida.
- Variable manipulada $u(t)$: La variable manipulada, corresponde a la transformación de la variable de control a una variable que afecta la dinámica del sistema, para el caso es la velocidad angular de las ruedas del sistema.
- Salida $y(t)$: La variable de salida del sistema, es la variable física que el sistema de control desea modificar, la posición del robot móvil.
- Variable medida $m(t)$: Esta señal se obtiene luego de cambiar la señal física a una señal eléctrica, mediante un elemento de sensado, el mismo que debido al error de medición por la calibración del instrumento y errores intrínsecos del sensor, es posible que dicho valor sea diferente al valor actual, siendo la velocidad lineal y angular real.

1.4.2 Métodos de aproximación a un modelo FOPTD

El método de aproximación de primer orden con retardo FOPTD (First Order Plus Dead Time, por sus siglas en inglés) es usado para obtener modelos de sistemas que tienen una respuesta no lineal, ya que, debido a la naturaleza del proceso (dinámica del robot móvil) estudiado en el documento, la ecuación que gobierna el sistema tiende hacer de orden superior, por lo que se justifica el uso de métodos de aproximación como el uso de FOPTD [10], [12]. Mediante el uso de la curva de reacción del proceso vistas en la Figura 1.2. se realiza la identificación de los parámetros; la ganancia, los tiempos de reacción de la curva y el tiempo de retardo, de este modo se crea la función de transferencia de primer orden con retardo como se describe en la Ecuación (1.2).

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K}{(\tau s + 1)} e^{-t_0 s} \quad (1.2)$$

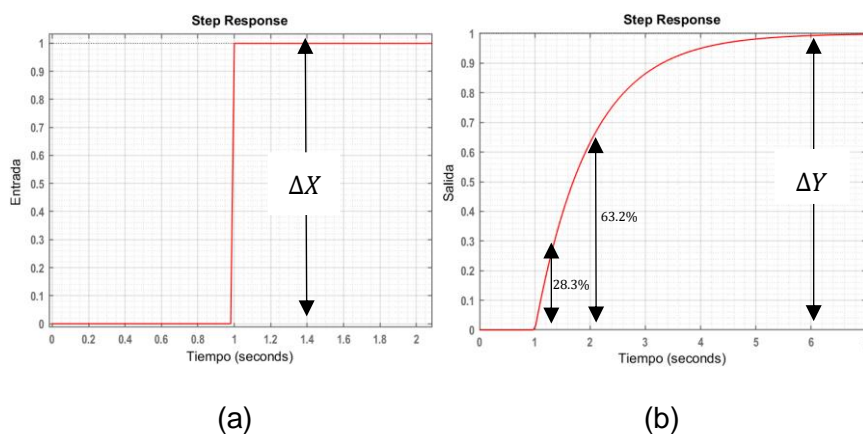


Figura 1.2. (a) Señal paso, (b) Curva de reacción del sistema [3], [10]

Los parámetros del modelo de primer orden están definidos de la siguiente manera:

- Ganancia (K): Es la variación de la salida respecto al cambio de la entrada como lo indica en la Ecuación (1.3), la ganancia indica estado estable del proceso que se caracteriza por indicar la rapidez en que aumenta la respuesta del sistema.

$$K = \frac{\Delta Y}{\Delta x} = \frac{\text{Variación de señal salida}}{\text{Variación de señal referencia}} \quad (1.3)$$

- Constante de tiempo del proceso (τ): Es el indicador de reacción del proceso ante una perturbación, es decir, es el tiempo de respuesta hasta que llegue al 63.2% de su cambio total, influyendo en la dinámica del sistema.
- Tiempo muerto del proceso (t_0): Es el tiempo en que tarda en responder el proceso ante un cambio de la referencia o perturbación.

Para realizar la identificación de los parámetros de FOPDT se hace uso el método de Smith, que permite obtener un sistema de primer orden basado en dos puntos con un alto grado de similitud con el sistema real [10], siguiendo los siguientes pasos:

1. Enviar una señal paso con un valor inicial hacia el sistema y luego se realizar una variación del 10% al valor final en un lapso como se muestra en la Figura 1.2a, encontrar la ganancia con (1.3).
2. Tomar los tiempos al $0.632\Delta Y$ y $0.283\Delta Y$ de la señal de salida como se indica en la Figura 1.2b.
3. Obtener el valor de constante de tiempo τ y el tiempo de retardo t_0 definidos por las Ecuaciones (1.4) y (1.5) respectivamente.

$$\tau = 1.5(t_{0.632\Delta Y} - t_{0.283\Delta Y}) \quad (1.4)$$

$$t_0 = t_{0.632\Delta Y} - \tau \quad (1.5)$$

1.4.3 Robótica móvil

El estudio de la robótica impulsó al crecimiento de las diferentes áreas científica (física, geología, óptica, vulcanología, etc [6], [13]) y la creación de nuevos dispositivos aplicables a distintos fines (construcción, milicia, aeronáutica, transporte, etc [14]–[16]), que posteriormente se orientaron hacia sistema electromecánico reprogramable, que permite realizar diferentes tareas repetitivas que requieren un grado elevado de precisión. Existe toda una gama de formas y aplicaciones para los robots, dependiendo de la tarea que deban realizar; de esta forma se pueden considerar diferentes criterios para su clasificación, como son: su arquitectura, su generación, el nivel de inteligencia, el nivel de control o el nivel de su lenguaje de programación [17]–[19].

La robótica móvil es un sistema mecatrónico capaz de navegar por distintos entornos, además, cuenta con distintas capacidades que permite ejecutar tareas complejas, ya sea de forma autónoma capaz de desplazarse de manera autónoma sin estar sujeto físicamente a un solo punto o controlado por un operador [20]–[22]. De este modo, los robots móviles son dispositivos autónomos dotados de sensores, actuadores, sistemas de cómputo embebido, sistemas de comunicación y sistemas de alimentación, que permiten movilizarse en determinado ambiente de trabajo, que permiten monitorear a cada momento su posición relativa a su punto de origen y a su punto de destino, por medio de un control en lazo cerrado.

1.4.4 Robot móvil de tracción diferencial

Los robots móviles basados en ruedas son sistemas sencillos, de fácil construcción y control. Sin embargo, en un terreno irregular, las ruedas pueden no resultar tan eficiente como los sistemas de patas articuladas u oruga, ya que, no tiene suficiente agarre en esta clase de superficies [6].

Para desplazarse los robots móviles con tracción diferencial están compuestos de dos ruedas traseras que son usadas para dirección y tracción, y una rueda delantera que permite la estabilidad del robot [6], [14], como se observa en la Figura 1.3.

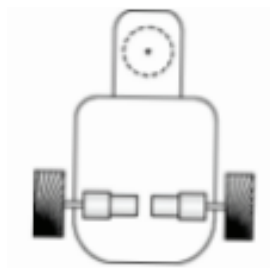


Figura 1.3. Robots móviles de tracción diferencial [6]

1.4.5 Tipos de robots de tracción diferencial en V-REP

Los robots móviles con tracción diferencial que se encuentran en la librería del software V-REP se encuentran descritos en la Tabla 1.1., se ha tomado las características de cada robot, directamente del fabricante, donde se puede corroborar que la tracción del robot es en modo diferencial y es aplicable al presente proyecto de titulación de integración curricular.

Tabla 1.1. Robots móviles de tracción diferencial en VREP [3], [6], [23]

	<p>E-Puck. – Robot pequeño con tracción diferencial diseñado por Michael Bonani y Francesco Mondada en el EPFL con fines educativos y de investigación. Contiene 2 motores, donde cada rueda pueden producir una velocidad de 0 a 18 cm/s de forma independientemente. Contiene un dispositivo de rango y demora para comunicación local entre robots, debido a la gran cantidad de sensores (proximidad, distancia y unidad de medición inercial) y por esta razón son especialmente usados para equipos de tipo enjambre.</p>
	<p>Pioneer 3 DX. – Es un robot con tracción diferencial usado en laboratorios en interiores o aulas, cuenta con gran cantidad de sensores (sonar frontal, giroscopio, GPs, y se puede adicionar otros módulos) usado por su durabilidad y preferido para sistemas inteligentes avanzados. Contiene 2 motores independientes que puede llegar a una velocidad máxima de 1.2 m/s.</p>

1.4.6 Redes neuronales

En los últimos años, se ha realizado investigaciones, tanto para mejorar los métodos tradicionales y su sintonización, así como la posibilidad de encontrar nuevas estrategias de control, utilizando técnicas emergentes [7], [14], [24]–[26].

Las redes neuronales (en inglés conocido como Neural Networks “NN”) tienen como objetivo simular el comportamiento del cerebro humano, haciendo uso de neuronas artificiales y entrenando a la red con un conjunto de datos significativos y diversos,

mediante el proceso de aprendizaje, que consiste en la representación precisa de los pesos en todas las interconexiones de capa y hallar una solución eficiente para problema complejos; modificando los pesos y comprobando si se cumple la convergencia y de no ser el caso, se repetirá el proceso [26]–[28].

Los controladores neuronales, han sido propuestos para resolver sistemas caóticos, aproximar funciones no lineales, sistemas desconocidos con error de sincronización, como por ejemplo; control infinito sincronizado de redes neuronales para sistemas caóticos con retardo de tiempo, control adaptativo de vuelo con arreglos de microelectrodos, aplicaciones estadísticas con redes neuronales para diagnóstico de fallo de rotores en la máquina de inducción y algoritmos de control de seguimiento de trayectoria de bucle cerrado doble de un robot móvil con ruedas basado en la estabilidad global [13], [29]–[31].

1.4.6.1 Elementos básicos que componen una red neuronal

La red está constituida por neuronas interconectadas y arregladas en capas, como se observa en la Figura 1.6. Los datos ingresan por medio de la capa de entrada pasan a través de la capa oculta y salen por la capa de salida. La capa oculta puede estar constituida por varias capas, dependiendo del caso estas capas pueden llegar a ser simplificadas [14].

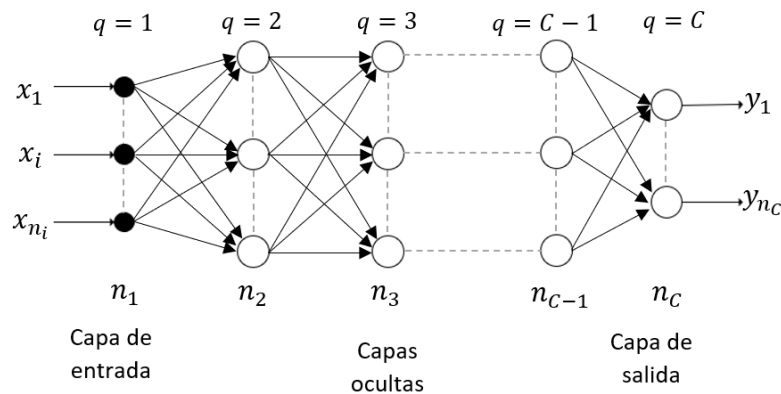


Figura 1.6. Ejemplo de una red neuronal multicapa [12], [14]

1.4.6.1.1 Función de entrada

La capa de entrada contiene un número ínfimo de neuronal y no se encuentran interconectadas entre sí, por lo que, se debe establecer una entrada global a_i^q (1.6). La manera para lograrlo es a través de la función de entrada, la cual se calcula a partir de un vector de entrada. La función de entrada puede describirse como [14], [25]:

$$a_i^q = f\left(\sum_{j=1}^{n_{q-1}} w_{ij}^{q-1} a_j^{q-1} + u_i^q\right) \quad (1.6)$$

Donde, a representa el vector de entrada, i la neurona, q la capa, u el umbral y ω_{ij}^q es el peso de la neurona i a la neurona j .

1.4.6.1.2 Función de activación

La función de activación calcula el estado de una neurona; transformando la función de entrada global menos el umbral en un valor de activación, cuyo rango normalmente va de [0 a 1] o de [-1 a 1]. Esto es así, porque una neurona puede estar totalmente inactiva [0 o -1] o activa [1].

Las funciones de activación más habitualmente utilizadas en controladores son [14], [25]:

1) *Función lineal:*

$$f(x) = \begin{cases} -1 & x \leq -\frac{1}{a} \\ kx & -\frac{1}{k} < x < \frac{1}{k} \\ 1 & x \geq \frac{1}{a} \end{cases}, \text{ con } x = a_i^q - u_i, \text{ y } a > 0 \quad (1.7)$$

Los valores de salida obtenidos por medio de esta función de activación serán: $k(a_i^q - u_i)$ comprendidos dentro del rango $\left[-\frac{1}{k}, \frac{1}{k}\right]$ [14].

2) *Función sigmoidea:*

$$f(x) = \frac{1}{1 + e^{-x}}, \text{ con } x = a_i^q - u_i \quad (1.8)$$

Los valores de salida que entrega esta función están comprendidos dentro de un rango que va de 0 a 1 [14].

3) *Función tangente hiperbólica:*

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \text{ con } x = a_i^q - u_i \quad (1.9)$$

Los valores de salida de la función tangente hiperbólica están comprendidos dentro de un rango que va de -1 a 1 [14].

1.4.6.1.3 Función de salida

La función de salida "y" determina el valor que se transfiere a las neuronas vinculadas. Si la función de activación y' está por debajo de un umbral u_i , ninguna salida pasa a la

neurona subsiguiente, es decir, no existe una activación de la neurona, denotada en la Ecuación 1.10. Normalmente, cualquier valor no es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos entre $[0, 1]$ o $[-1, 1]$, también se puede tener valores reales que se aproximen a estos valores [14].

$$y = \begin{cases} 1 & \text{si } y' > u_i \\ -1 & \text{en caso contrario} \end{cases} \quad (1.10)$$

1.4.6.2 Mecanismos de aprendizaje

Dentro de los mecanismos de aprendizaje para controladores basados en redes neuronales se tienen el aprendizaje supervisado y no supervisado.

1.4.6.2.1 Aprendizaje supervisado

El aprendizaje supervisado se caracteriza porque el proceso de ajuste de parámetros se realiza mediante un entrenamiento, donde, dando los patrones de entrada y salida para realizar el ajuste de la red neuronal se obtiene la salida. En el caso de que ésta no coincida con la salida deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada, como se observa en la Figura 1.7. [12], [32].

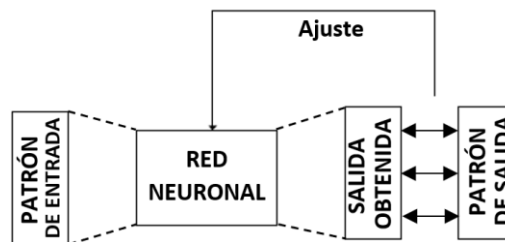


Figura 1.7. Esquema de aprendizaje supervisado [12]

1.4.6.2.2 Aprendizaje no supervisado

Las redes con aprendizaje no supervisado no requieren un patrón para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta, como se observa en la Figura 1.8.

En algunos casos, el aprendizaje no supervisado es llamado clustering que permite ajustar los pesos de la red neuronal para encontrar a la salida y a qué clase pertenece la información presentada a la entrada, donde se tienen las correlaciones entre las informaciones presentadas [12].

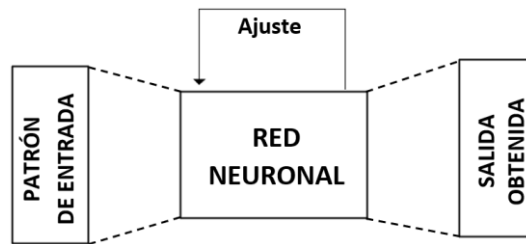


Figura 1.8. Esquema de aprendizaje no supervisado [12]

1.4.6.3 Control Neuronal

Los controles neuronales basados en el aprendizaje no supervisado obtienen la salida de modelos inciertos a partir de una entrada mediante el ajuste de los pesos de la NN.

1.4.6.3.1 PID Neuronal

El controlador PID neuronal tiene un sistema de aprendizaje y aplicación continuo, por lo que, la red neuronal está continuamente ajustando los pesos de las ganancias proporcional, integral y derivativa, empleando el método de gradiente descendente, durante cada iteración del cálculo de la acción de control para decrementar el error [17].

A diferencia del método general, que debe ser entrenado off-line, el procedimiento no supervisado se entrena de manera on-line, es decir, mientras el sistema está realizando su trabajo, sus parámetros se van modificando. El procedimiento no supervisado puede presentar problemas de estabilidad durante el entrenamiento on-line, si la velocidad de aprendizaje no se adecua a las constantes de tiempo del sistema [25].

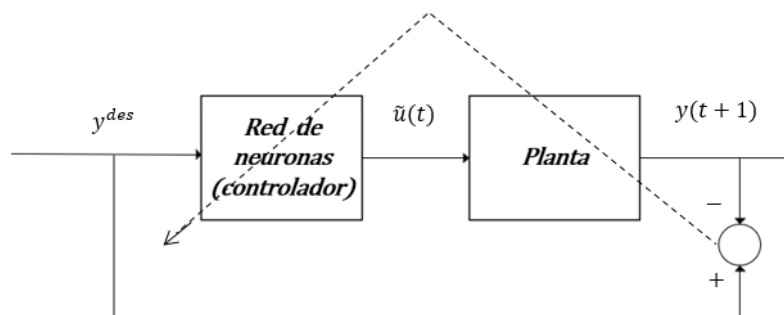


Figura 1.9. Esquema de control inverso con aprendizaje no supervisado [25]

1.4.6.3.2 PID + NN

El controlador PID + NN tiene dos fases; una de aprendizaje y una etapa de aplicación. En la Figura 1.10., se observa el bloque de identificación neuronal que es usado para obtener el modelo del sistema, mientras que en la etapa de aplicación la red neuronal es utilizado como controlador, realizando un aprendizaje continuo (no supervisado) [24].

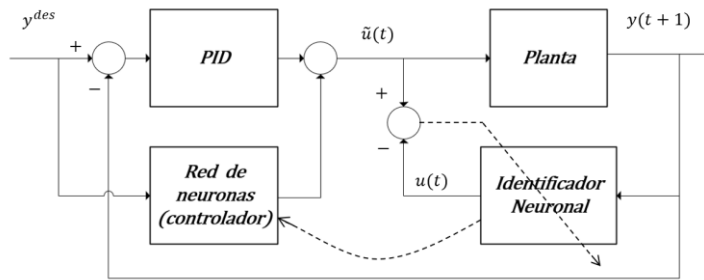


Figura 1.10. Esquema de control PID+NN [25]

1.4.7 Índices de desempeño

Los índices de desempeño utilizados para analizar los resultados del trabajo de integración curricular son: el error cuadrático integral (ISE) permite penalizar errores grandes más que los pequeños, por lo tanto, se obtiene el valor mínimo que se desempeña mejor al inicio de la respuesta o en la parte del transitorio (1.11), el error absoluto integral (IAE) calcula el error en el tiempo para sistemas que posean un amortiguamiento razonable y una respuesta transitoria satisfactoria (1.12), mientras que el integral de la salida del controlador cuadrática (ISCO) permite penalizar las acciones de control grande respecto a las pequeñas, debido a que el cuadrado de un gran error será mucho más importante (1.13) [33], [34].

$$ISE = \int_0^t e^2(t)dt \quad (1.11)$$

$$EAI = \int_0^t e(t)dt \quad (1.12)$$

$$ISCO = \int_0^t u(t)dt \quad (1.13)$$

2. METODOLOGÍA

El presente trabajo es de carácter exploratorio – experimental a nivel de simulación donde se validará la propuesta correspondiente a los algoritmos de control propuestos. En base a la metodología de investigación, este trabajo dispone de cuatro fases. La fase teórica, que ha sido presentada en el Capítulo 1 donde se hace una revisión bibliográfica de los tipos de robots móviles dentro de la plataforma V REP, y de las redes neuronales donde se ha identificado diferentes esquemas de control factibles de implementación en este trabajo. La fase de diseño desarrollada en el Capítulo 2, presenta una revisión de las características físicas del robot Pioneer 3DX y el proceso de comunicación entre Matlab y VREP [3], [11], [35]. A continuación, se presenta el detalle de los algoritmos de aprendizaje de las redes neuronales a ser aplicadas en los controladores, y el diseño de los esquemas de control propuestos para el modelo cinemático y dinámico del robot, así como los criterios considerados para la generación de diferentes trayectorias. La fase de implementación detallada en el Capítulo 2, presenta la configuración para la comunicación entre Matlab y V REP mediante una interfaz gráfica, y el detalle de la programación de los algoritmos de control combinando el controlador cinemático y dinámico en cascada. Finalmente, en la fase de pruebas detallada en el Capítulo 3, se presenta las pruebas de funcionamiento de seguimiento de trayectorias para validar los esquemas de control propuestos.

2.1 Redes Neuronales Aprendizaje

La etapa de aprendizaje en una red neuronal de su importancia, ya que permite conocer la mejor forma de control de un proceso, a continuación, se explica el algoritmo de aprendizaje utilizado en los controladores neuronales propuestos.

2.1.1 Algoritmo de Retropropagación

Para el ajuste de los pesos de una red neuronal, el error debe ser propagado hacia atrás desde la capa de salida [25]. Esto permite que los pesos sobre las conexiones de las neuronas ubicadas en las capas ocultas cambien durante el entrenamiento. La función del error E tiene la siguiente forma:

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (2.1)$$

Donde N es el número de patrones o muestras, $e(n)$ es el error cometido por la red (2.2) [36], $S_i(n)$ es el vector de salidas deseadas y $Y_i(n)$ el vector de salidas de la red para el patrón n de cada neurona.

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (S_i(n) - Y_i(n))^2 \quad (2.2)$$

El cambio de los pesos en las conexiones de las neuronas además de influir sobre la entrada global influye en la activación y por consiguiente en la salida de una neurona. Por lo tanto, es de gran utilidad considerar las variaciones de la función de activación al modificarse el valor de los pesos $w(n)$. Esto se llama sensibilidad de la función de activación con la tasa de aprendizaje α , de acuerdo con el cambio en los pesos, como en la Ecuación (2.3) [14], [25].

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w} \quad (2.3)$$

Las expresiones de la propagación del error en la capa $C-1$ son:

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{ji}^{C-1}} \quad (2.4)$$

Derivando el error (2.2) respecto al peso de la capa anterior se tiene;

$$\begin{aligned} \frac{\partial e(n)}{\partial w_{ji}^{C-1}} &= -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \\ \frac{\partial e(n)}{\partial w_{ji}^{C-1}} &= \frac{\partial e(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \end{aligned} \quad (2.5)$$

Siendo la salida $y_i(n)$ definido en la Ecuación (2.6);

$$y_i = a_i^C = f \left(\sum_{j=1}^{n_{C-1}} w_{ij}^{C-1} a_j^{C-1} + u_i^C \right) \text{ para } i = 1, 2, \dots, n_c \quad (2.6)$$

Derivando la salida (2.6) se obtiene la Ecuación (2.7):

$$\frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} = f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) a_j^{C-1}(n) \quad (2.7)$$

Estableciendo un nuevo término $\delta_i^C(n)$ en la Ecuación (2.8),

$$\begin{aligned} \delta_i^C(n) &= -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \\ \delta_i^C(n) &= \frac{\partial e(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \frac{1}{a_j^{C-1}(n)} \end{aligned} \quad (2.8)$$

Despejando $\frac{\partial e(n)}{\partial w_{ji}^{C-1}}$ se tiene la Ecuación (2.9),

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = \delta_i^c(n) a_j^{c-1}(n) \quad (2.9)$$

Reemplazando la Ecuación (2.4) en (2.9) se tiene la actualización del peso visto en la Ecuación (2.10).

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) - \alpha \delta_i^c(n) a_j^{c-1}(n) \quad (2.10)$$

para $j = 1, 2, \dots, n_{c-1}$ $i = 1, 2, \dots, n_c$

Y para la actualización del umbral se tiene en la Ecuación (2.11)

$$u_i^c(n) = u_i^c(n-1) - \alpha \delta_i^c(n) \quad (2.11)$$

para $i = 1, 2, \dots, n_c$

Para la Capa Oculta $C - 2$, donde los pesos y umbrales para la capa $q + 1$; $q = 1, \dots, C - 2$ y considerando el peso de la capa $C - 2$ a la capa $C - 1$; con la Ecuación (2.12)

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{kj}^{c-2}} \quad (2.12)$$

para $k = 1, 2, \dots, n_{c-1}$ $j = 1, 2, \dots, n_c$

Sea el peso de la capa $C - 2$ afecta a todas las salidas en la Ecuación (2.13)

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \quad (2.13)$$

Derivando la salida respecto al peso;

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} = \frac{\partial a_j^c}{\partial a_j^{c-1}} \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (2.14)$$

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) w_{ji}^{c-1} \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}}$$

Estableciendo un nuevo parámetro $\delta_i^c(n)$ donde:

$$\delta_i^c(n) = -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \quad (2.15)$$

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \rightarrow \frac{\partial e(n)}{\partial w_{kj}^{c-2}} = \sum_{i=1}^{n_c} \delta_i^c(n) w_{ji}^{c-1} \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}}$$

Reemplazando $\delta_i^{c-1}(n)$ o en cualquier capa es:

$$\delta_j^{c-1}(n) = f' \left(\sum_{j=1}^{n_{c-2}} w_{kj}^{c-2} a_k^{c-2} + u_j^{c-1} \right) \sum_{i=1}^{n_c} \delta_i^c(n) w_{ji}^{c-1} \quad (2.16)$$

Remplazando la derivada del error se obtiene,

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = \delta_j^{c-1}(n) a_k^{c-2}(n) \quad (2.17)$$

Ajustando los pesos en la Capa Oculta $C - 2$ es,

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) - \alpha \delta_j^{c-1}(n) a_k^{c-2}(n) \quad (2.18)$$

para $k = 1, 2, \dots, n_{c-2}$ y $j = 1, 2, \dots, n_{c-1}$

Generalizando para las expresiones para $\delta_j^{c+1}(n)$, $w_{kj}^{c+1}(n)$ y $u_j^{c+1}(n)$,

$$\delta_j^{c+1}(n) = f' \left(\sum_{k=1}^{n_c} w_{kj}^c a_k^c + u_j^c \right) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c \quad (2.19)$$

$$w_{kj}^c(n) = w_{kj}^c(n-1) - \alpha \delta_j^{c+1}(n) a_k^c(n) \quad (2.20)$$

para $k = 1, 2, \dots, n_c$, $j = 1, 2, \dots, n_{c+1}$ y $c = 1, 2, \dots, C - 2$

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) - \alpha \delta_j^{c+1}(n) \quad (2.21)$$

para $j = 2, \dots, n_{c+1}$ y $c = 2, \dots, C - 2$

2.2 Robot Pioneer 3DX

El robot Pioneer 3DX es un robot de tracción diferencial de dos ruedas con dos motores que se encuentra contenido en las librerías de la plataforma VREP y ha sido seleccionado para realizar el trabajo de Trabajo de Integración Curricular.

El seguimiento de trayectorias en robótica móvil es una de las aplicaciones más significativas en el desarrollo de sistemas de control, por lo que, durante años ha sido estudiado para mejorar el desempeño de estos mecanismos. El Pioneer 3DX es un robot usado para investigación, el cual incluye un controlador de movimiento con realimentación de encoder, y tiene la posibilidad de ser monitoreado usando mapeo/localización o controlado mediante teleoperación [31], [37], [38]. La plataforma VREP permite controlar dicho robot (simulado) individualmente a través de un script que puede ser desarrollado en Matlab. Para mayor facilidad se ha realizado una interfaz gráfica que permite observar los resultados de los controladores propuestos (véase el Anexo I para manejar el Interfaz de Usuario). Debido a la versatilidad y prestaciones que presenta este software se lo ha seleccionado en este trabajo. En la Figura 2.1a se

observa la estructura física del Robot Móvil y en la Figura 2.1b se presenta sus dimensiones.

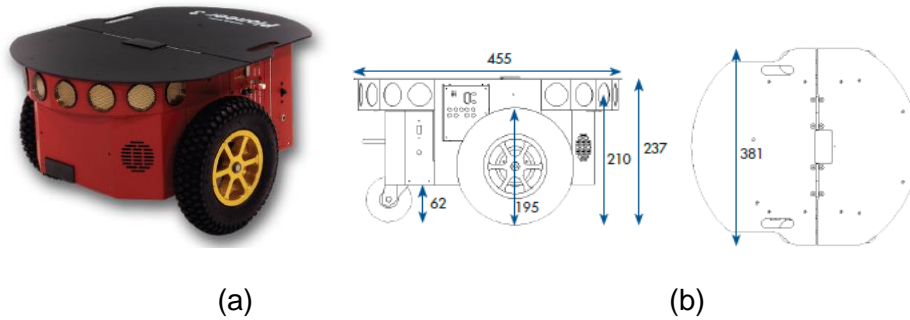


Figura 2.1. (a) Robot móvil Pioneer 3DX. (b) Dimensiones en milímetros del robot Pioneer 3DX [23]

Para obtener el modelo, tanto cinemático como dinámico del robot móvil, es necesario conocer las características técnicas del robot, las cuales se presentan en la Tabla 2.1.

Tabla 2.1. Especificaciones técnicas del Robot Pioneer 3DX [23]

CARACTERÍSTICA	ESPECIFICACIONES
Construcción	Cuerpo: 1,6 mm de aluminio. Neumáticos: Caucho de espuma.
Operación	Peso del robot: 9 kg.
Movimiento	Velocidad de avance/retroceso máximo: 1,2 m/s. Velocidad de rotación máxima: 300°/s. Periodo de muestreo: 100ms.
Alimentación	Voltaje: 12 V. Capacidad: 7200 mAh (por cada batería de ácido de plomo). Tiempo de carga: 12h o 2,4h con un cargado de alta capacidad.

En base a las características físicas del robot móvil algunas de las cuales se presentan en la Figura 2.2., es posible obtener su modelo cinemático.

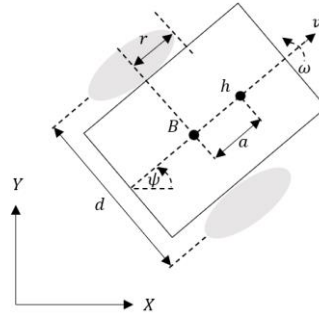


Figura 2.2. Parámetros del robot móvil de tracción diferencial [34]

Donde:

d , es la distancia entre las ruedas del robot.

B , es el punto medio entre los ejes de las ruedas.

h , es el punto de referencia de la posición a obtener.

a , es la distancia entre el punto medio de los ejes de las ruedas y el punto de referencia de la posición a obtener, del robot.

r , es el radio de las ruedas.

ω , es la velocidad angular del robot.

v , es la velocidad lineal del robot.

ψ , es el ángulo de orientación del robot.

2.2.1 Modelo cinemático

El modelo cinemático permite el estudio del movimiento de los sistemas mecánicos sin considerar las fuerzas que afectan dicho movimiento, modelando la velocidad lineal y angular del robot en función a las velocidades de las ruedas conjuntamente a los parámetros geométricos del robot [36].

A partir de la velocidad lineal y angular del robot móvil se obtienen las velocidades lineales de cada rueda aplicando la Ecuación (2.22) [1], [3], [11], [20], [34]–[37].

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} 1 & \frac{d}{2} \\ 1 & -\frac{d}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.22)$$

Para el caso de las velocidades angulares de cada motor se divide la velocidad lineal para el radio " r " de la rueda (*ver* 2.23)

$$\begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & 0 \\ 0 & \frac{1}{r} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix} \quad (2.23)$$

La representación matemática del modelo cinemático discretizado del robot móvil de tracción diferencial se observa en la Ecuación (2.24), considerando que T_s es el tiempo de muestreo, donde [34]:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \psi(k+1) \end{bmatrix} = T_s \begin{bmatrix} \cos \psi(k) & -a \sin \psi(k) \\ \sin \psi(k) & a \cos \psi(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \end{bmatrix} \quad (2.24)$$

2.2.2 Modelo dinámico

El modelo dinámico permite calcular el esfuerzo y la carga sobre la estructura o sistemas, tomando en cuenta el movimiento acelerado de un cuerpo, es decir, las distintas fuerzas actuantes sobre este, donde el modelo se compone de las partes estructurales del robot que pueden influir en el movimiento.

Debido a la gran cantidad de parámetros involucrados en el modelo dinámico como variables físicas, la identificación del modelo es obtenida mediante el modelo aproximado, a través de la curva de reacción vista en la Sección 1.4.2. en la que se aplica una señal tipo escalón al sistema, para evaluar el comportamiento de la salida de la velocidad lineal y angular [10], [12].

La Figura 2.3. muestra la respuesta del sistema, cuando una señal paso es aplicada, lo que permite determinar los modelos aproximados.

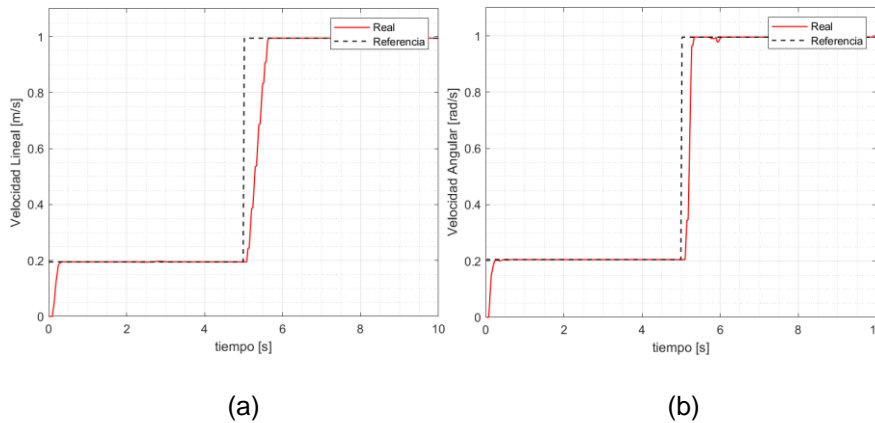


Figura 2.3. Respuesta paso de velocidad (a) lineal y, (b) angular de robot Pioneer 3DX

Aplicando las ecuaciones de la Sección 1.4.2 se obtiene los parámetros del FOPDT para la velocidad lineal y angular, los cuales se pueden observar en la Tabla 2.2.

Tabla 2.2. Parámetros del FOPDT de la velocidad lineal y angular del robot móvil Pioneer 3DX

Parámetros	Velocidad lineal	Velocidad angular
Ganancia (K)	1	1
Constante de tiempo (τ)	0,33435	0,1527
Tiempo muerto (t_0)	0,12975	0,0833

Una vez determinados los modelos aproximados, se propone el diseño de los controladores neuronales de este trabajo, en la Sección 2.2.4.1.

2.2.3 Generación de trayectorias

La generación de trayectorias se realizará creando un vector de muestras, las cuales el robot móvil deberá seguir en un tiempo establecido.

Para el cálculo de las muestras se debe utilizar el tiempo de duración de la trayectoria y el tiempo de muestreo (2.25) [3].

$$m = \frac{t}{T_s} \quad (2.25)$$

m , número de elementos del vector del tiempo de la trayectoria,

t , tiempo de duración de la trayectoria,

T_s , tiempo de muestreo,

Para el cálculo de las componentes X y Y correspondientes a la trayectoria se presenta la Tabla 2.3., donde los elementos de los vectores son las componentes $x_{ref}(k)$ y $y_{ref}(k)$, que inician en la coordenada $(x, y) = (0,0)$. J es el radio del círculo y radio de la lemniscata, y L es el lado del cuadrado.

Tabla 2.3. Ecuaciones consideradas para la generación de trayectorias

Trayectoria	Componente X	Componente Y
Cuadrada	$\frac{L}{2} \forall km \in [0, km];$	$\left(-\frac{L}{2} + 4kmL\right) \forall km \in [0, km];$

	$\left(\frac{L}{2} - 4kmL\right) \forall km \in [km, 2km];$ $-\frac{L}{2} \forall km \in [2km, 3km];$ $\left(-\frac{L}{2} + 4kmL\right) \forall km \in [3km, 4km];$	$\frac{L}{2} \forall k \in [km, 2km];$ $\left(\frac{L}{2} - 4kmL\right) \forall km \in [2km, 3km];$ $-\frac{L}{2} \forall km \in [3km, 4km];$
Circular	$J \cos(2\pi km)$	$J \sin(2\pi km)$
Lemniscata	$0.6J \sin(4\pi km)$	$J \sin(2\pi km)$
Abierta	$2.5 \forall km \in [0, 2km];$ $2.5 - km \forall km \in [2km, 3km];$ $0 \forall km \in [3km, 5km];$ $0 - km \forall km \in [5km, 6km];$ $-2.5 \forall km \in [6km, 8km];$	$-2.5 + km \forall km \in [0, 2km];$ $2.5 \forall km \in [2km, 3km];$ $2.5 - km \forall km \in [3km, 5km];$ $-2.5 \forall km \in [5km, 6km];$ $-2.5 + km \forall km \in [6km, 8km];$

2.2.4 Diseño de controladores cinemático y dinámico

El esquema de control propuesto calcula la salida necesaria para llevar la referencia al sistema. La estructura propuesta presenta un control en cascada cuyo lazo exterior está constituido del controlador cinemático y el lazo interior se dispone del controlador dinámico basado en redes neuronales.

2.2.4.1 Controlador Cinemático

El control cinemático (2.27) está basado en el modelo cinemático, considerando las coordenadas de los puntos de interés $[x, y]^T$. Donde la ley de control es [36], [39]:

$$\begin{aligned}
 \begin{bmatrix} v_{ref}^c(k) \\ \omega_{ref}^c(k) \end{bmatrix} &= \begin{bmatrix} \frac{\cos \psi(k)}{T_s} & \frac{\sin \psi(k)}{T_s} \\ -\frac{1 \sin \psi(k)}{a T_s} & \frac{1 \cos \psi(k)}{a T_s} \end{bmatrix} \\
 &\times \begin{bmatrix} x_{ref}(k+1) + l_x \tanh\left(\frac{k_x}{l_x} e_x(k)\right) - x_{ref}(k) \\ y_{ref}(k+1) + l_y \tanh\left(\frac{k_y}{l_y} e_y(k)\right) - y_{ref}(k) \end{bmatrix} \quad (2.27)
 \end{aligned}$$

Donde $[v_{ref}^c(k) \ \omega_{ref}^c(k)]^T$, es la salida del controlador cinemático, $e_x(k) = x_{ref}(k) - x(k)$, y $e_y(k) = y_{ref}(k) - y(k)$ son los errores de posición para los ejes X y Y respectivamente, k_x, k_y son las ganancias del controlador, $l_x, l_y \in \mathbb{R}$ son constantes de saturación y T_s es el tiempo de muestreo usado en el caso de estudio.

La función $\tanh(\cdot)$ es agregada para saturar las acciones de control en el caso de que el error de posición sea demasiado grande.

En el análisis de estabilidad, la tracción de velocidad es considerada, $v_{ref}^c(k) = v(k)$ y $\omega_{ref}^c(k) = \omega(k)$. Reemplazando (2.26) en (2.24), la ecuación del lazo cerrado es:

$$\begin{bmatrix} e_x(k+1) \\ e_y(k+1) \end{bmatrix} + \begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} e_x(k)\right) \\ l_y \tanh\left(\frac{k_y}{l_y} e_y(k)\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.28)$$

Definido el vector de errores de salida como $\tilde{h}(k) = [e_x(k) \ e_y(k)]^T$

$$\tilde{h}(k+1) = - \begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} e_x(k)\right) \\ l_y \tanh\left(\frac{k_y}{l_y} e_y(k)\right) \end{bmatrix} \quad (2.29)$$

En [36] ha sido seleccionada la candidata de la función de Lyapunov para la ley de control cinemática como $V(k) = \frac{1}{2} \tilde{h}^T(k) \tilde{h}(k)$. En el trabajo citado se demuestra la estabilidad del control cinemático para el seguimiento de trayectorias si los parámetros son configurados como $k_x > 0, k_y > 0, l_x > 0$ y $l_y > 0$, entonces $\tilde{h}(k) \rightarrow 0$ para $k \rightarrow \infty$.

2.2.4.2 Controladores Dinámicos

En este trabajo se presenta dos propuestas de controladores neuronales diferentes. El controlador dinámico contiene los controladores basados en redes neuronales que se encargarán de alcanzar la referencia de velocidad lineal y angular del robot móvil. Las dos propuestas son:

- Control PID Neuronal
- Control PID + NN

2.2.4.2.1 Control PID Neuronal

El modelo dinámico mostrado en la Sección 2.2.2. permite diseñar el controlador PID neuronal, de manera que, aplicando el esquema de control inverso con aprendizaje no supervisado, se pretende llegar al valor de referencia de la velocidad lineal y angular, ya que debido a que el caso de estudio establece que la dinámica del robot cambie, el robot debe ajustar sus parámetros de peso y realizar el control para mantener la trayectoria ante cualquier perturbación.

En la Sección 1.4.4.2.2. se explica el funcionamiento del aprendizaje no especializado de manera general, donde, se establece que el sistema puede llegar a ser inestable si la velocidad de aprendizaje no se adecua a las constantes de tiempo del sistema. Para evitar la inestabilidad del sistema se ha procedido a combinar las estrategias de control PID clásico y aprendizaje especializado colocando como pesos iniciales (K_p, K_I, K_D) las constantes del PID, para empezar el entrenamiento de la red de forma on-line.

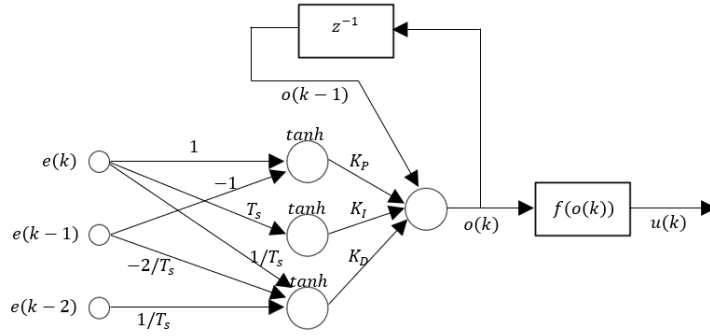


Figura 2.4. Red neuronal en controlador PID Neuronal [25]

Para deducir el controlador PID neuronal se parte de un controlador PID clásico, donde la referencia y la salida del modelo dinámico son usados como entradas, descrito como $e(k) = x_{ref}(k) - x(k)$, donde x representa la velocidad lineal v o angular ω y la salida del controlador $o_i(k)$, se describe como (2.30) :

$$o_i(k) = K_p e(k) + K_I \sum_{j=1}^k e(j) T_s + \frac{K_D}{T_s} [e(k) - e(k-1)] \quad (2.30)$$

Y la salida en el instante $k - 1$ es:

$$o_i(k-1) = K_p e(k-1) + K_I \sum_{j=1}^{k-1} e(j) T_s + \frac{K_D}{T_s} [e(k-1) - e(k-2)] \quad (2.31)$$

Obteniendo la diferencia $o_i(k) - o_i(k-1)$ se obtiene:

$$o_i(k) = o_i(k-1) + K_p [e(k) - e(k-1)] + K_I e(k) T_s + \frac{K_D}{T_s} [e(k) - 2e(k-1) + e(k-2)] \quad (2.32)$$

En la Figura 2.4., se puede observar una red neuronal haciendo uso de los parámetros de un controlador PID (2.32), donde, el vector de entradas son los errores $(e(k), e(k-1)$ y $e(k-2))$, el umbral es el estado anterior de la salida del controlador $o(k-1)$, los pesos son las constantes K_p, K_I y K_D que se ajustaran dentro del controlador. Y cada una de las neuronas tiene una función de activación tangente

hiperbólica, ya que, debido a la naturaleza de la planta, el robot móvil puede ir en sentido horario como en sentido antihorario $[1, -1]$.

La salida $o(k)$ también es sometida a una función de activación como se observa en la ecuación (2.33), donde δ permite tener valores mayores a $[1, -1]$.

$$f(o(k)) = \delta \tanh\left(\frac{o(k)}{\delta}\right) = u(k) \quad (2.33)$$

En la ecuación (2.34) se tiene la salida del controlador neuronal, sin la función de activación.

$$o(k) = o(k-1) + K_p f[e(k) - e(k-1)] + K_i f(e(k)T_s) + K_D f\left(\frac{[e(k) - 2e(k-1) + e(k-2)]}{T_s}\right) \quad (2.34)$$

Para el aprendizaje y ajuste de los pesos, se propaga el error hacia atrás, haciendo uso la regla delta [25], para ello se plantea la derivada del error a minimizar (2.35) [36] respecto a los pesos que permitirá converger el error a cero, ajustando el valor de las constantes mediante una derivación en cadena (2.36).

$$E(k) = \frac{1}{2} (x_{ref}(k) - x(k))^2 \quad (2.35)$$

$$\begin{aligned} \frac{\partial E(k)}{\partial K_p} &= \frac{\partial E(k)}{\partial x(k)} \frac{\partial x(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o(k)} \frac{\partial o(k)}{\partial K_p} \\ \frac{\partial E(k)}{\partial K_i} &= \frac{\partial E(k)}{\partial x(k)} \frac{\partial x(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o(k)} \frac{\partial o(k)}{\partial K_i} \\ \frac{\partial E(k)}{\partial K_D} &= \frac{\partial E(k)}{\partial x(k)} \frac{\partial x(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o(k)} \frac{\partial o(k)}{\partial K_D} \end{aligned} \quad (2.36)$$

Realizando la derivación:

- Del error respecto a la salida de la planta se tiene:

$$\frac{\partial E(k)}{\partial x(k)} = -e(k) \quad (2.37)$$

- De la salida respecto a la variable de control, donde h es un cambio en la salida.

$$\frac{\partial x(k)}{\partial u(k)} = \frac{x(k+h) - x(k)}{h} \quad (2.38)$$

- De la función de activación (2.33), donde:

$$\frac{\partial u(k)}{\partial o(k)} = [1 - f^2(o(k))] \quad (2.39)$$

Donde $f(x) = \tanh(x)$ y $f'(x) = 1 - \tanh^2(x)$.

- De la salida del controlador respecto a al peso K_p :

$$\frac{\partial o(k)}{\partial K_p} = f(e(k) - e(k - 1)) \quad (2.40)$$

- De la salida del controlador respecto a al peso K_I :

$$\frac{\partial o(k)}{\partial K_I} = f(e(k)T_s) \quad (2.41)$$

- De la salida del controlador respecto a al peso K_D :

$$\frac{\partial o(k)}{\partial K_D} = f\left(\frac{[e(k) - 2e(k - 1) + e(k - 2)]}{T_s}\right) \quad (2.42)$$

Obteniendo las constantes del PID en el instante $k + 1$, donde η es la tasa de aprendizaje y θ , son K_p, K_I y K_D .

$$K_\theta(k + 1) = K_\theta(k) - \eta \frac{\partial E(k)}{\partial K_\theta} \quad (2.43)$$

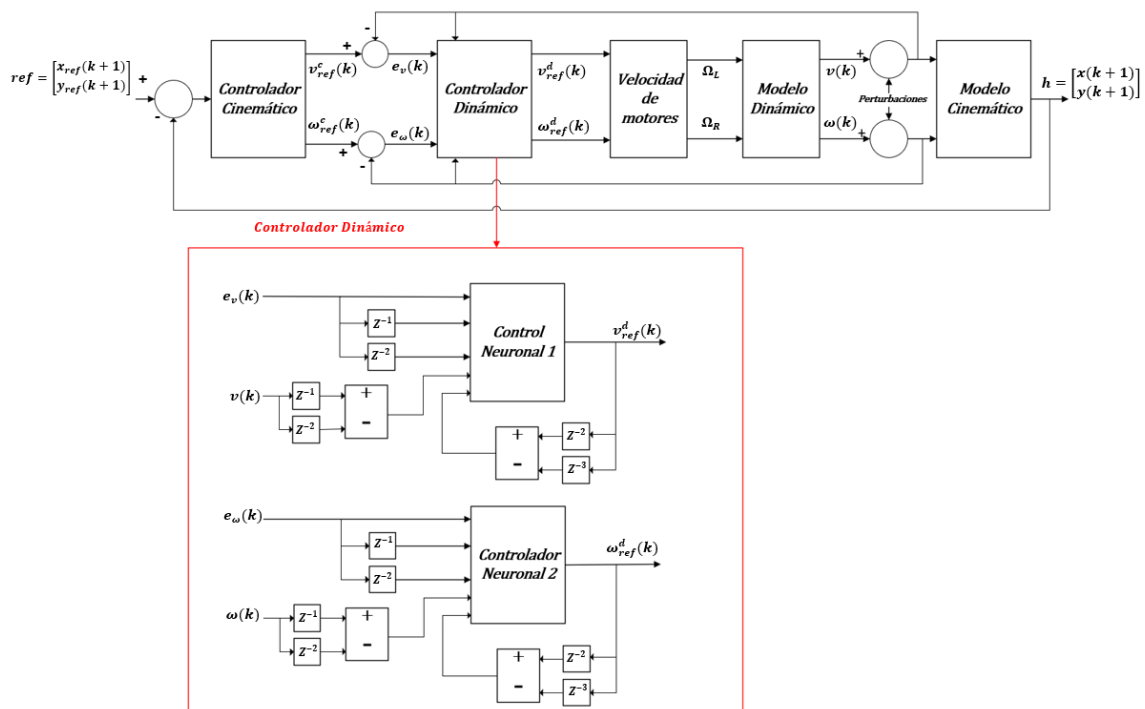


Figura 2.5. Esquema de control PID Neuronal para seguimiento de trayectoria de un robot móvil usando redes neuronales

En el esquema de la Figura 2.5., el bloque de la red neuronal se coloca entre la acción de control de la velocidad lineal calculada por el control dinámico $v_{ref}^d(k)$ y la variable medida v , para aprender la dinámica inversa del sistema usando información pasada.

La velocidad lineal v y su estado anterior se utilizan como entradas para el identificador, con la finalidad de minimizar el error, $e_v(k) = v_{ref}^d(k) - v_{ref}^L(k)$, donde $v_{ref}^L(k)$ es la salida del identificador. La minimización de e_v permite ajustar los parámetros (K_P, K_I y K_D) de la red neuronal que se actualizan en el controlador, en cada iteración. El procedimiento descrito anterior se aplica de manera similar para controlar la velocidad angular ω , considerando la minimización del error $e_\omega(k) = \omega_{ref}^c(k) - \omega_{ref}^d(k)$, donde $\omega_{ref}^c(k)$ es la salida de la acción de control del controlador cinemático, y $\omega_{ref}^d(k)$ es la acción de control calculada por el controlador dinámico. Las variables de control son las velocidades del motor (2.44), por lo que, es necesario calcular la velocidad de las ruedas izquierda Ω_L y derecha Ω_R , con base en los valores de las acciones de control $[v_{ref}^d(k) \ \omega_{ref}^d(k)]^T$ [36], [39].

$$\begin{aligned}\Omega_L &= \frac{2v_{ref}^d(k) - d\omega_{ref}^d(k)}{2r} \\ \Omega_R &= \frac{2v_{ref}^d(k) + d\omega_{ref}^d(k)}{2r}\end{aligned}\quad (2.44)$$

2.2.4.2.2 Control PID + NN

El controlador dinámico basado en el Controlador PID + Red Neuronal (PID+NN) tiene con fundamento el trabajo de los controladores en paralelo [34], [35]. Para este trabajo en base a que el tiempo muerto es muy pequeño como se ha visto en la identificación del modelo, se lo ha despreciado y se ha trabajado con un sistema de primer orden discreto sin retardo como se observa en la Ecuación (2.45).

$$\frac{y}{o_i} = \frac{K/z}{1 - a/z} \quad (2.45)$$

$$y = ayz^{-1} + Ko_i z^{-1} \quad (2.46)$$

Discretizando el modelo de primer orden (2.47).

$$y(k) = ay(k-1) + Ko_i(k-1) \quad (2.47)$$

Despejando la ley de control $o_i = O(k)$

$$o(k) = \frac{1}{K}y(k+1) - \frac{a}{K}y(k) \quad (2.48)$$

Luego de obtener la ley de control y los parámetros a ser identificados se hace un cambio de variables $b = \frac{1}{K}$ y $c = \frac{a}{K}$

$$o(k) = by(k + 1) - cy(k) \quad (2.49)$$

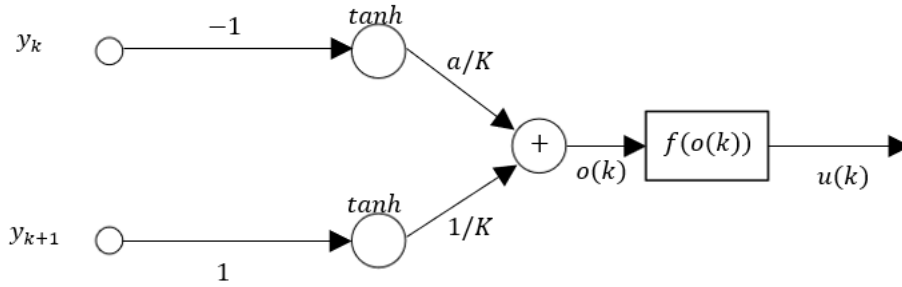


Figura 2.6. Red neuronal PID+NN [25]

En la Figura 2.6., se puede observar una red neuronal haciendo uso de los parámetros del sistema de primer orden (2.48), donde, el vector de entradas es las salidas del sistema $y(k + 1)$ y $y(k)$, los pesos son las constantes b y c que se ajustarán dentro del controlador (2.49). Y cada una de las neuronas tiene una función de activación tangente hiperbólica, ya que, debido a la naturaleza de la planta, el robot móvil puede ir en sentido horario como en sentido antihorario $[1, -1]$.

La salida $o(k)$ también es sometida a una función de activación como se observa en la Ecuación (2.50), donde δ permite tener valores mayores a $[1, -1]$.

$$f(o(k)) = \delta \tanh\left(\frac{o(k)}{\delta}\right) = u(k) \quad (2.50)$$

Para el aprendizaje y ajuste de los pesos, se propaga el error hacia atrás, haciendo uso la regla delta [25], para ello se plantea la derivada del error a minimizar [36] (2.51) respecto a los pesos que permitirá converger el error a cero, ajustando el valor de las constantes mediante una derivación en cadena (2.52).

La referencia y la salida del modelo dinámico son usados como entradas, descrito como $e(k) = x_{ref}(k) - x(k)$, donde x representa la velocidad lineal v o angular ω y la salida del controlador dinámico es $o(k)$.

$$E(k) = \frac{1}{2}(x_{ref}(k) - x(k))^2 \quad (2.51)$$

$$\frac{\partial E(k)}{\partial b} = \frac{\partial E(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o(k)} \frac{\partial o(k)}{\partial b} \quad (2.52)$$

$$\frac{\partial E(k)}{\partial c} = \frac{\partial E(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o(k)} \frac{\partial o(k)}{\partial c}$$

Realizando la derivación:

- Del error respecto a la salida de la planta se tiene:

$$\frac{\partial E(k)}{\partial x(k)} = -e(k) \quad (2.53)$$

- De la función de activación (2.50), donde:

$$\frac{\partial u(k)}{\partial o(k)} = [1 - f^2(o(k))] \quad (2.54)$$

Donde $f(x) = \tanh(x)$ y $f'(x) = 1 - \tanh^2(x)$.

- De la salida del controlador respecto a al peso b :

$$\frac{\partial o(k)}{\partial b} = y(k + 1) \quad (2.55)$$

- De la salida del controlador respecto a al peso c :

$$\frac{\partial o(k)}{\partial c} = y(k) \quad (2.56)$$

Obteniendo las constantes del PID en el instante $k + 1$, donde η es la tasa de aprendizaje y θ , son b, c .

$$\theta(k + 1) = \theta(k) - \eta \frac{\partial E(k)}{\partial \theta} \quad (2.57)$$

En el esquema propuesto visto en la Figura 2.7, el bloque de aprendizaje en línea (Identificador Neuronal 1) tiene como entradas; la variable medida $v(k)$ y su estado anterior, para minimizar el error $e_v(k) = v_{ref}^d(k) - v_{ref}^N(k - 2)$, donde $v_{ref}^N(k)$ es la salida del identificador y $v_{ref}^d(k)$ es la sumatoria la acción de control Neuronal $v_{NN}^C(k)$ y el controlador PID $v_{NN}^C(k)$.

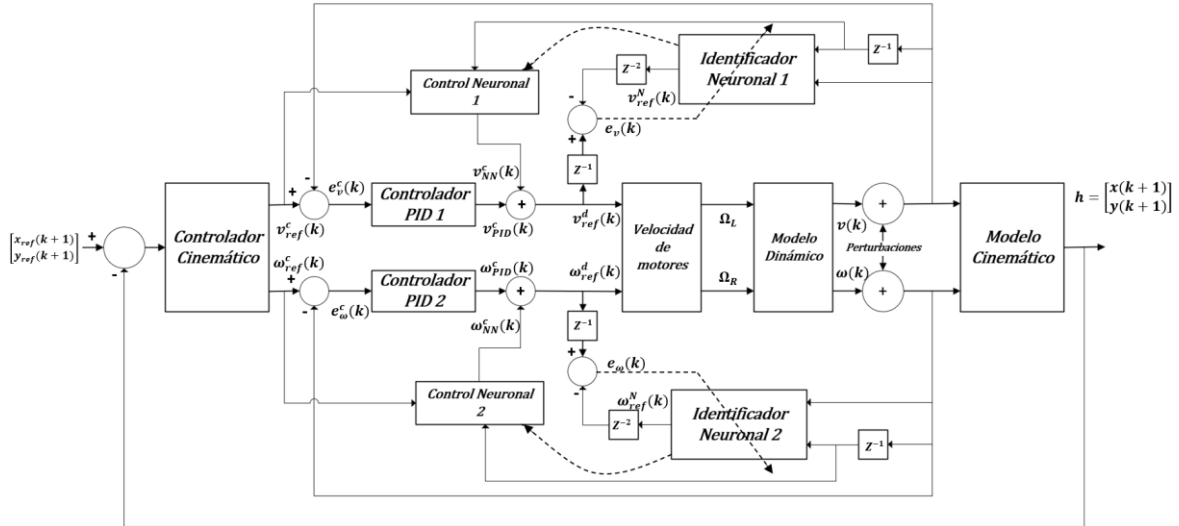


Figura 2.7. Esquema de control PID+NN para seguimiento de trayectoria de un robot móvil usando redes neuronales

La minimización del error e_v sigue ajustando los parámetros del modelo Neuronal que es configurado en el controlador $v_{NN}^c(k)$, en cada tiempo de muestreo, y $e_v^c(k)$ es el error de velocidad lineal, de $v_{ref}^c(k)$ y $v(k)$. El procedimiento descrito anterior se aplica de manera similar para controlar la velocidad angular ω , considerando la minimización del error $e_\omega(k) = \omega_{ref}^d(k-1) - \omega_{ref}^N(k-2)$, donde $\omega_{ref}^N(k)$ es la salida del identificador de la red neuronal y $\omega_{ref}^d(k)$ es la acción de control calculada por el control dinámico, el error de velocidad angular es $e_\omega^c(k) = \omega_{ref}^c(k) - \omega(k)$. Las variables de control son las velocidades del motor (2.44), por lo que, es necesario calcular la velocidad de las ruedas izquierda Ω_L y derecha Ω_R , con base en los valores de las acciones de control $[v_{ref}^d(k) \ \omega_{ref}^d(k)]^T$.

Los controladores propuestos se prueban en un robot Pioneer 3DX dentro de la plataforma virtual VREP, que permite simular sistemas robóticos considerando su cinemática, dinámica y la física del medio ambiente. La versatilidad de este software está ligada a la disponibilidad de complementos para conectarse con otras herramientas computacionales, como Matlab, donde los algoritmos han sido programados [35]. En la Figura 2.8., se muestra el interfaz de V-REP enlazado con el entorno de Matlab, para el control del robot Pioneer 3DX.

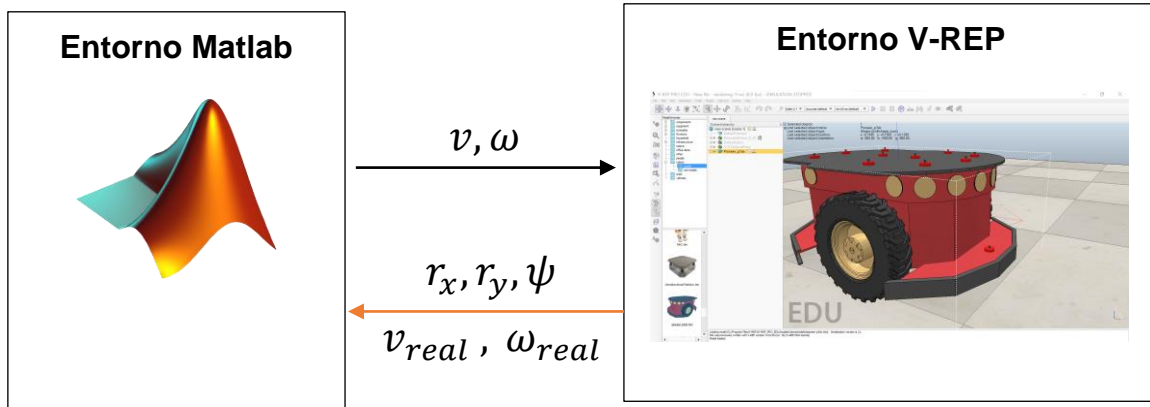


Figura 2.8. Enlace entre el entorno Matlab y V-REP [34]

2.3 Interfaz Gráfica

En la Figura 2.9. se muestra la interfaz gráfica desarrollada, donde se configuran los parámetros para generar las gráficas que serán mostradas en los resultados. Los parámetros por configurar son:

- Trayectoria: Se puede seleccionar 4 tipos de trayectorias. Luego de seleccionado la trayectoria se puede seleccionar el lado o radio dependiendo de la figura, y finalmente se escoge el tiempo de la trayectoria.
- Tipo de controlador: Se selecciona uno de los tres controladores, hasta que todos hayan sido probados.
- Tipo de plano: Se selecciona el plano para determinar si se afectará la dinámica o no del robot.

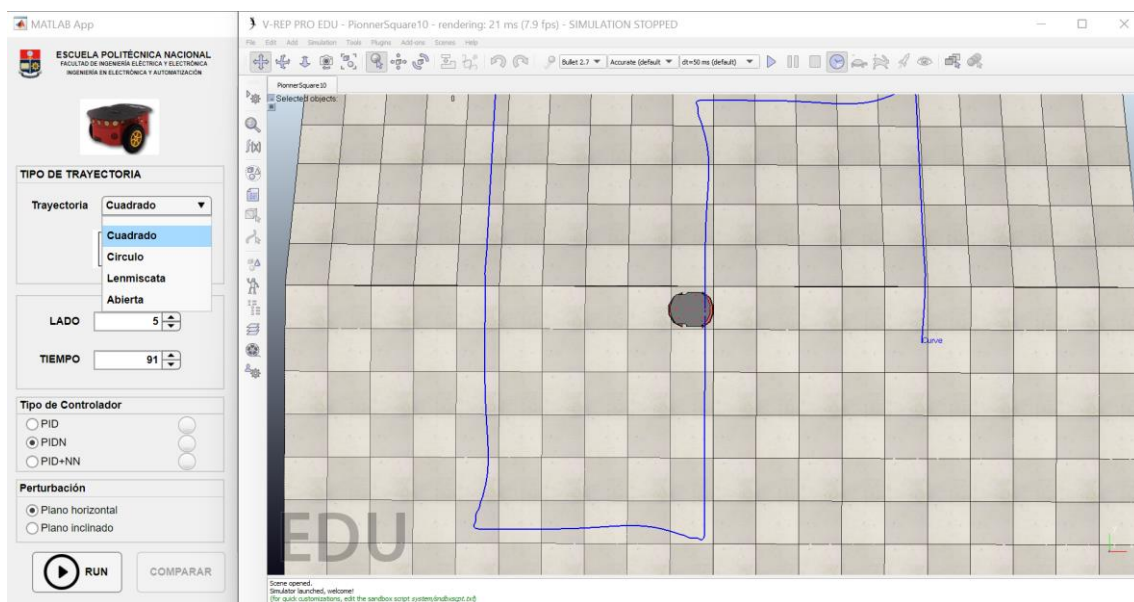


Figura 2.9. Interfaz Gráfica

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En el siguiente capítulo se realizará la sintonización de los controladores, tanto cinemático como dinámico, para realizar la fase de pruebas mediante la interfaz gráfica de cada una de las trayectorias planteadas.

3.1 Sintonización de controladores

El control cinemático fue sintonizado mediante las ecuaciones descritas en el Sección 2.2.4.1. realizando pequeños ajustes para cada una de las trayectorias, donde $a = 0,12$ siendo la distancia entre el punto de referencia del robot y el punto centro del eje de las llantas, $k_x, k_y = 0,07$, son ganancias que permiten minimizar el error de posición, las constantes $l_x, l_y = 0,1$, que permiten la saturación de la velocidad lineal y velocidad angular del robot, y el tiempo de muestreo igual a $T_s = 0,1 \text{ seg}$.

El controlador dinámico fue sintonizado para encontrar las constantes del controlador PID clásico haciendo uso del método de Dahlin que permite obtener las constantes del control, el cual propone [12]:

- K_p es encargada de aumentar la velocidad de respuesta y disminuir el error del sistema, y se calcula con (3.1):

$$K_p = \frac{1}{2K} \left(\frac{t_0}{\tau} \right)^{-1} \quad (3.1)$$

- K_D es encargada de aumentar la respuesta del sistema, y se calcula con (3.2):

$$K_D = K_p \left(\frac{t_0}{2} \right) \quad (3.2)$$

- K_I es encargada de disminuir el error del sistema en régimen estable y aumentar de manera moderada velocidad del sistema y se calcula con (3.3):

$$K_I = \frac{K_p}{\tau} \quad (3.3)$$

Tabla 3.1. Parámetros del controlador PID

Parámetros	Velocidad lineal	Velocidad angular
Constante proporcional (K_p)	1,288	0,9165
Constante integral (K_I)	3,8522	6,0019
Constante derivativa (K_D)	0,0835	0,03871

Los parámetros de controlador PID clásico son utilizados en los controladores inteligentes PIDN y PID+NN. En el controlador PIDN son usados como valores iniciales para iniciar el aprendizaje. En el controlador PID+NN son usadas en el mismo controlador PID. En ambos casos, se ha utilizado la misma tasa de aprendizaje para el controlador de velocidad lineal igual a 0,009 y angular igual a 0,00005 (valores calibrados heurísticamente en base a experimentación), con el objetivo de comparar el rendimiento entre ellos y respecto al PID de manera justa.

3.2 Prueba de seguimiento de trayectorias

En esta Sección, se pretende comparar el funcionamiento controlador PID clásico y los controladores Neuronales para el seguimiento de trayectoria (cuadrada, circular, lemniscata y abierta) del robot móvil.

3.2.1 Trayectoria Cuadrada

La trayectoria cuadrada tiene como características con cambios bruscos de orientación en las esquinas, por lo que, la acción de control de velocidad lineal debe ser más rápida, mientras que la velocidad angular debe ser cero en las rectas de la trayectoria. En la Figura 3.1. se observa la respuesta del seguimiento de trayectoria de una figura cuadra en V-REP y la respuesta para analizarse en la interfaz gráfica en MatLab.

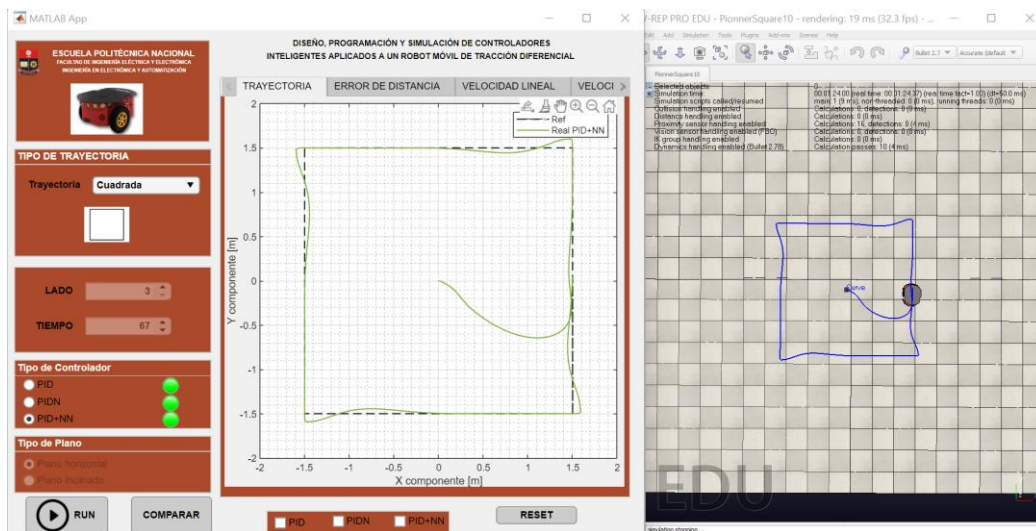


Figura 3.1. Creación de trayectoria mediante el interfaz de gráfica

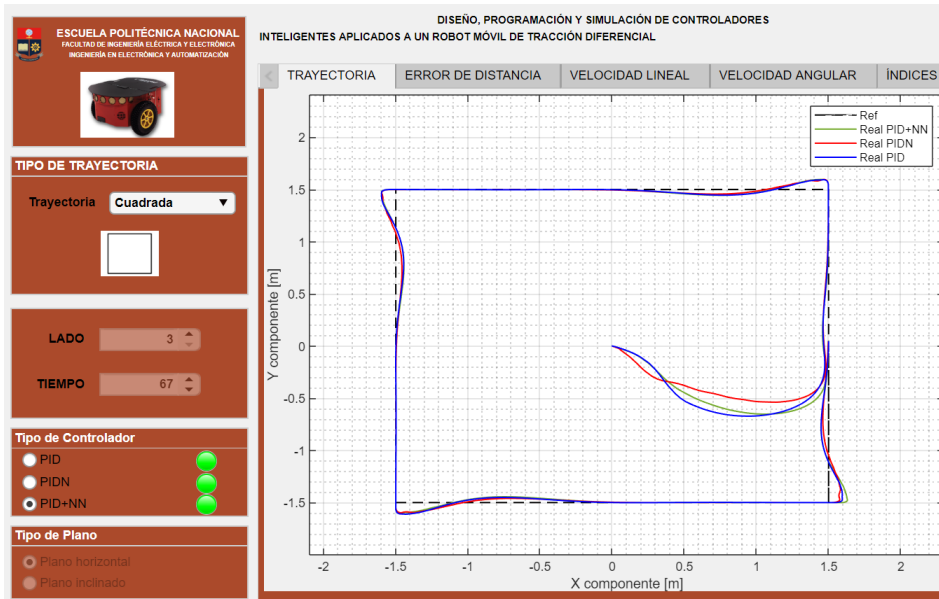


Figura 3.2. Seguimiento de trayectoria cuadrada de 3 m de lado

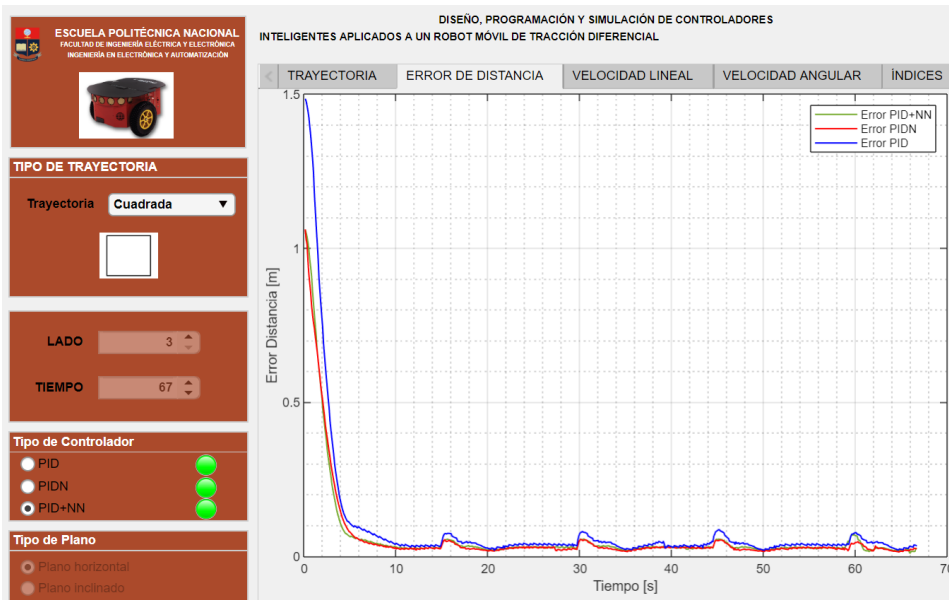


Figura 3.3. Error cuadrático instantáneo de posición del seguimiento de trayectoria cuadrada de 3 m de lado a una velocidad lineal de 0,2 m/s

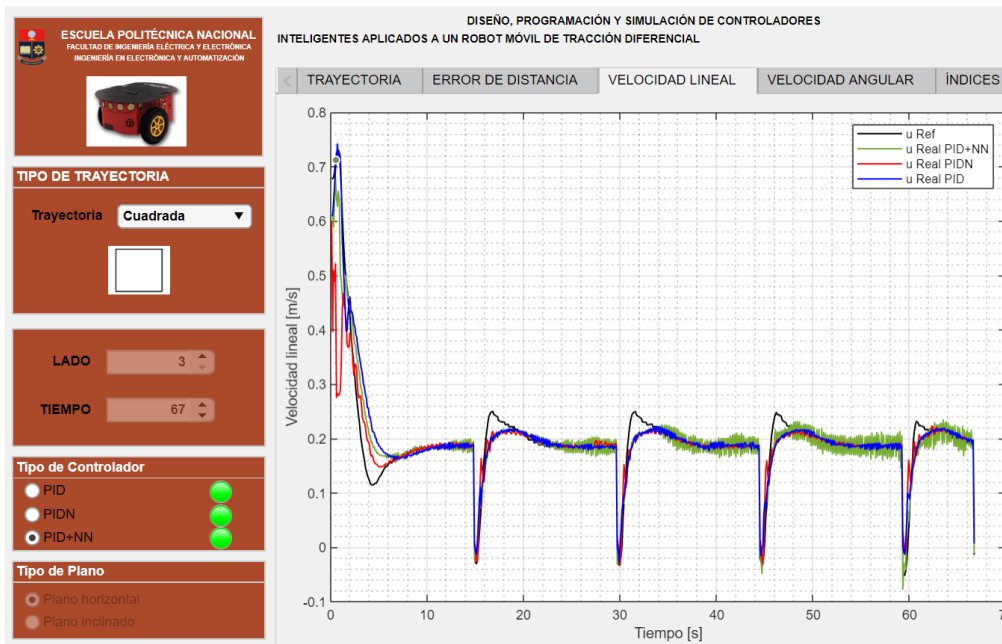


Figura 3.4. Velocidad lineal de $0,2 \text{ m/s}$ para el seguimiento de trayectoria cuadrada de 3 m de lado

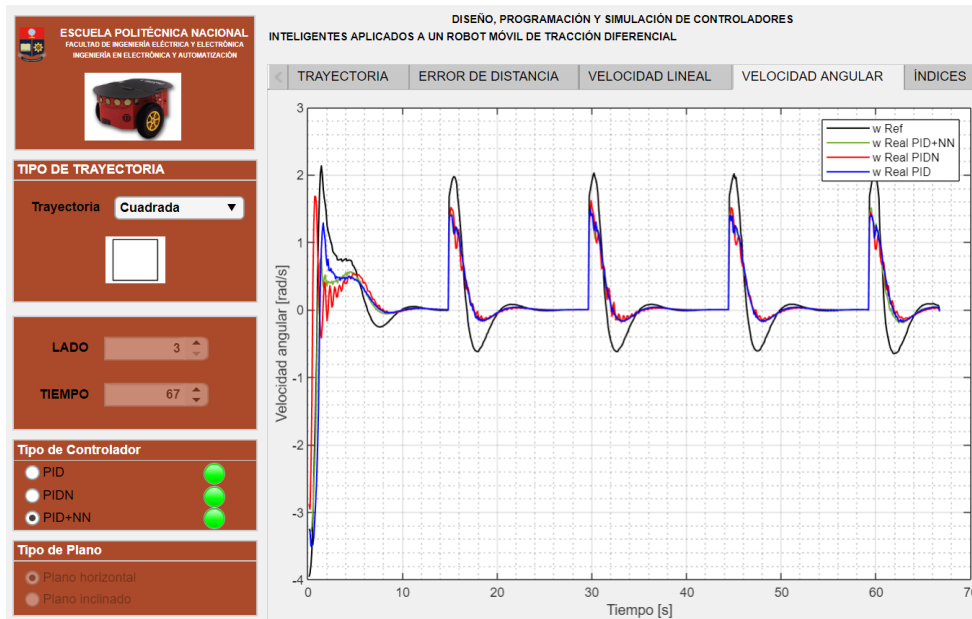


Figura 3.5. Velocidad angular para el seguimiento de trayectoria cuadrada de 3 m de lado para una velocidad lineal de $0,2 \text{ m/s}$

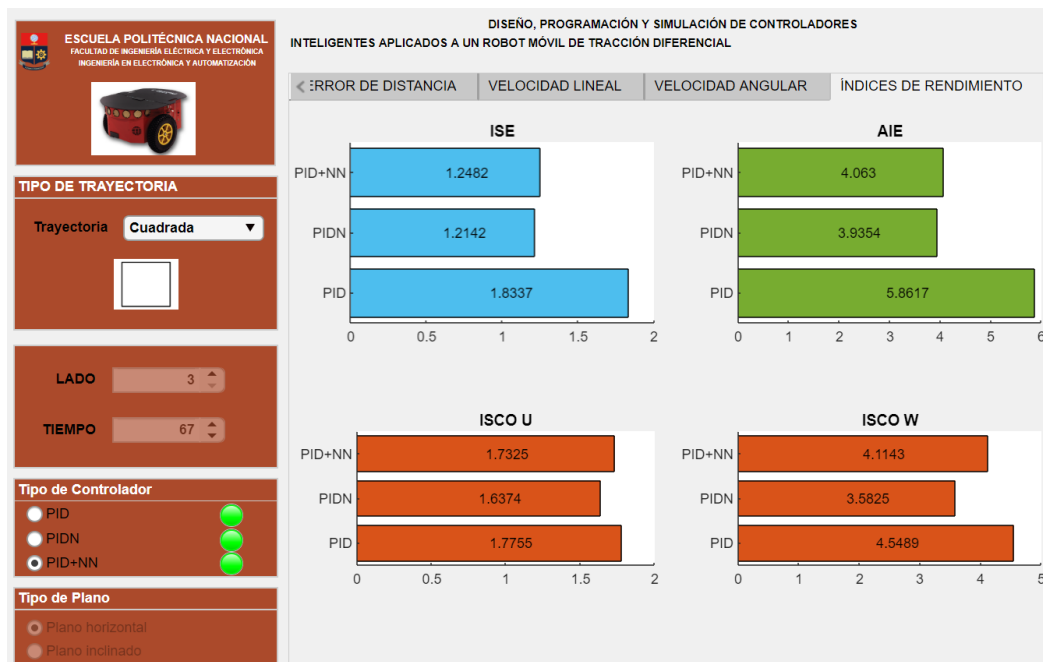


Figura 3.6. Comparación cuantitativa para figura cuadrada

Las Figuras 3.1 y 3.2, muestran el seguimiento del robot móvil en una trayectoria cuadrada. Los resultados observados muestran cualitativamente un error de distancia cuadrático instantáneo promedio de 2 cm (la Figuras 3.3). Este error de posición se da debido a que los controladores se están probando en un entorno aproximado a la realidad (simulación de V-REP) donde generan puntos que se envían al robot y los controladores programados en Matlab responden con las acciones de control que tienden a que el robot siga la trayectoria deseada, además el robot hace lo posible para alcanzar la referencia cuando un nuevo punto llega al controlador, esto para evitar que el robot se detenga o tenga un funcionamiento con aceleración y desaceleración constantes. En el caso de las esquinas, el error no alcanza un valor mayor a 9 cm , siendo corregido por el controlador cinemático para alcanza la referencia (trayectoria).

La velocidad lineal es alrededor de $0,2\text{ m/s}$ (mostradas en la Figura 3.4), donde la acción de control (para el modelo dinámico) es más agresiva en las esquinas para mantener la referencia.

El error de velocidad angular es igual a cero (Figura 3.5), debido a que es una recta con giros de 90° en las esquinas.

El análisis cuantitativo realizado en base a los índices ISE, IAE e ISCO (Figura 3.6), muestran que el controlador PIDN tiene el menor error al realizar el seguimiento de trayectoria cuadrada lo cual evidencia su mejor comportamiento. Para analizar los posteriores resultados se colocó las Figuras de las comparaciones de los controladores.

3.2.2 Trayectoria Circular

La trayectoria circular no presenta grandes cambios de orientación, por lo que, la acción de control debe mantener la referencia de velocidad angular.

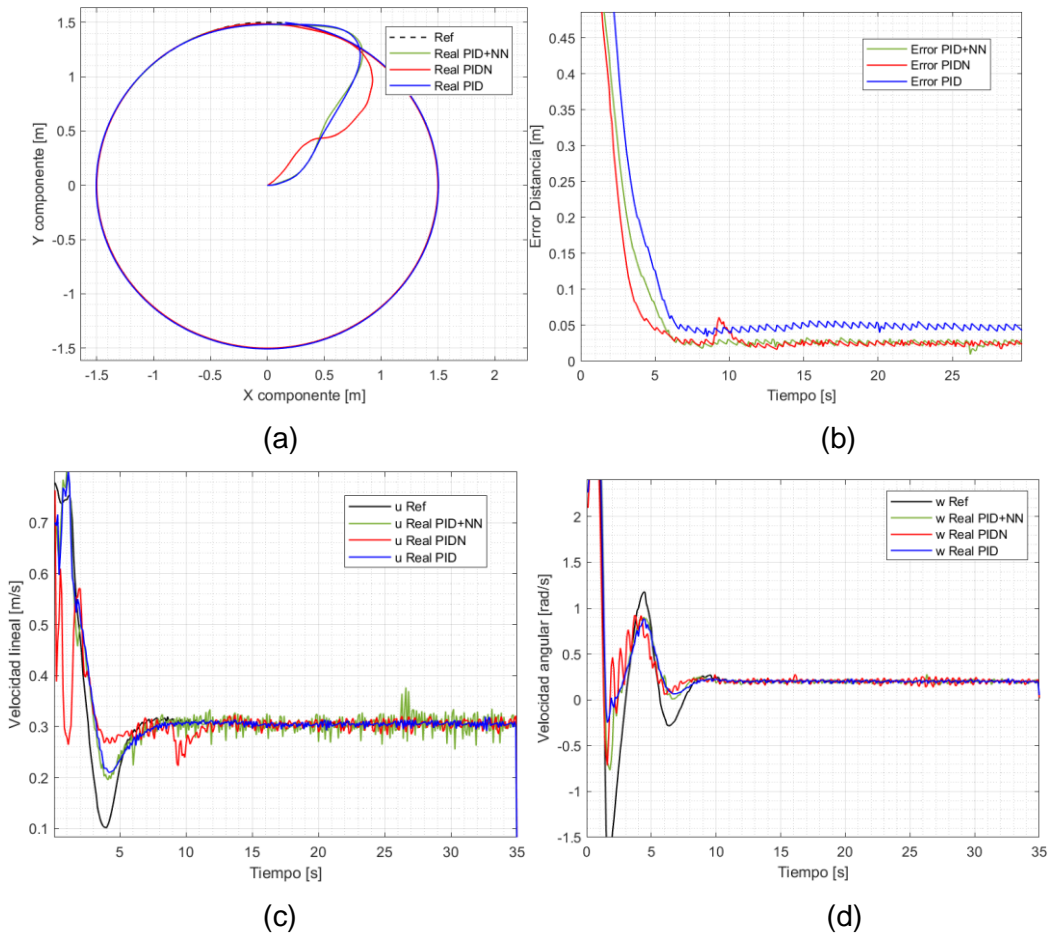


Figura 3.7. (a) Seguimiento de trayectoria circular de 1,5 m de radio, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular

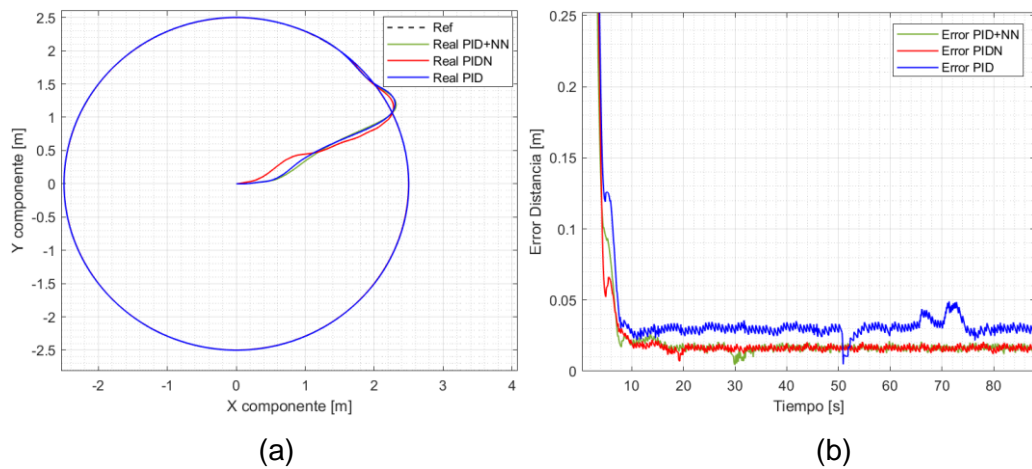


Figura 3.8. (a) Seguimiento de trayectoria circular de 2,5 m de radio, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,2 m/s

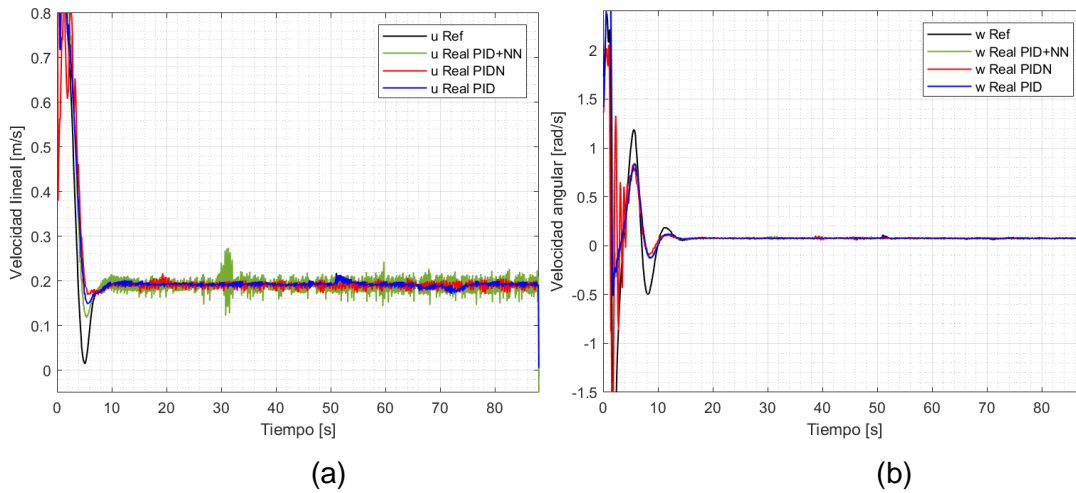


Figura 3.9. Seguimiento de trayectoria circular de 2,5 m de radio, velocidad (a) lineal de 0,2 m/s y (b) angular

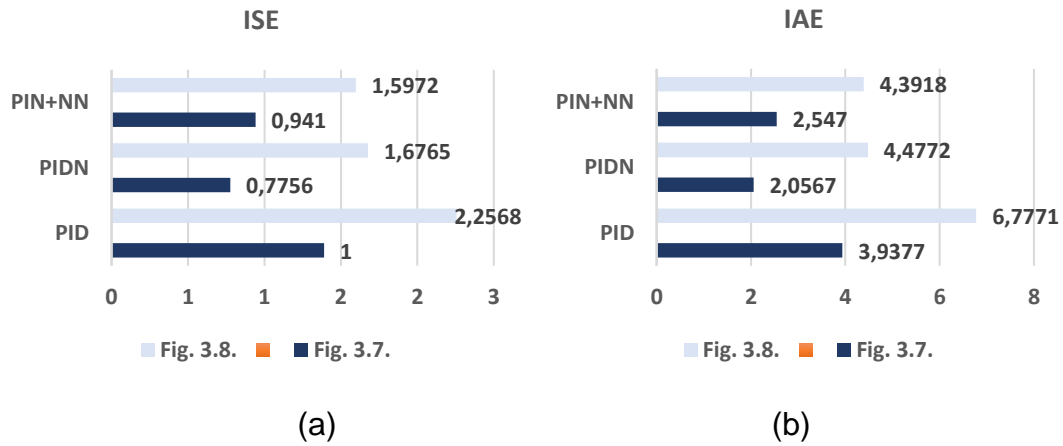


Figura 3.10. Comparación cuantitativa para figura circular (a) ISE, (b) IAE

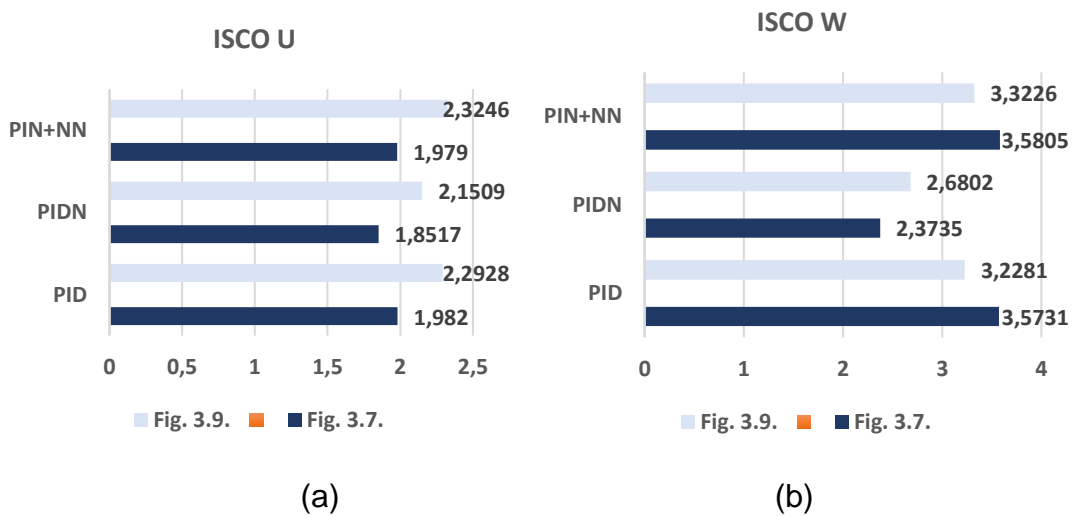


Figura 3.11. Comparación cuantitativa para figura circular (a) ISCO U, (b) ISCO W

Las Figuras 3.7a y 3.8a, muestran el seguimiento del robot móvil en una trayectoria circular. Los resultados observados por causa del control cinemático muestran

cuantitativamente un error de distancia cuadrático instantáneo promedio no mayor a 2 cm (Figura 3.8b) y en el último caso alrededor de 3 cm (Figura 3.7b), debido al aumento de velocidad lineal de 0.2 m/s (mostradas en la Figura 3.9a) y 0.3 m/s (en la Figura 3.7c). Los errores como se ha indicado previamente se dan por el funcionamiento del controlador cinemático y la interconexión entre VREP y Matlab.

Además, se puede apreciar que la velocidad angular es igual a 0.2 rad/s (Figura 3.7d) y 0.1 rad/s (Figura 3.9b), por el cambio de orientación constante, sin embargo, los controladores son capaz de seguir las referencias sin problemas y adecuadamente.

El análisis cuantitativo de la trayectoria circular realizado en base a los índices ISE, IAE e ISCO (Figura 3.10 y 3.11), muestran que el controlador PIDN y PID+NN tienen los mejores desempeños al seguir la trayectoria deseada.

3.2.3 Trayectoria Lemniscata

La trayectoria lemniscata tiene la característica de que constantemente cambia de orientación, provocando que la acción de control de la velocidad lineal y angular sea variable y se muestra en las siguientes Figuras.

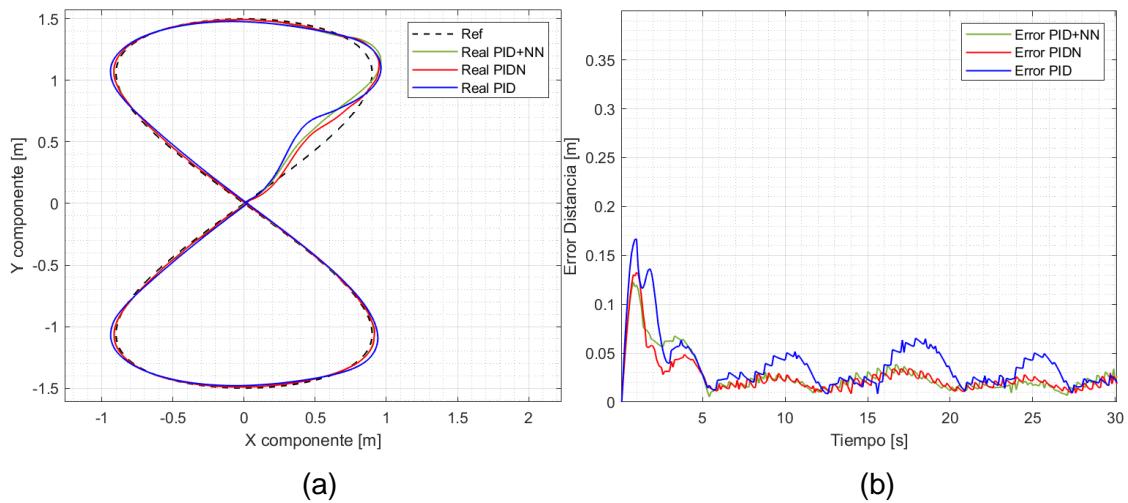


Figura 3.12. (a) Seguimiento de trayectoria lemniscata de $1,5\text{ m}$ de radio, (b) error cuadrático instantáneo de posición a velocidad lineal de $0,3\text{ m/s}$

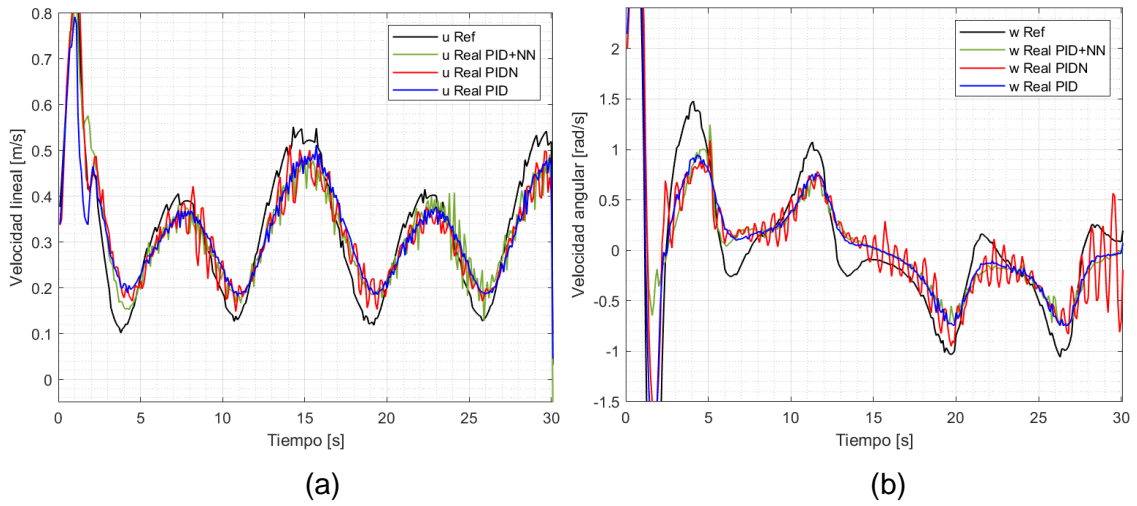


Figura 3.13. Seguimiento de trayectoria lemniscata de 1,5 m de radio velocidad (a) lineal de 0,3 m/s y (b) angular

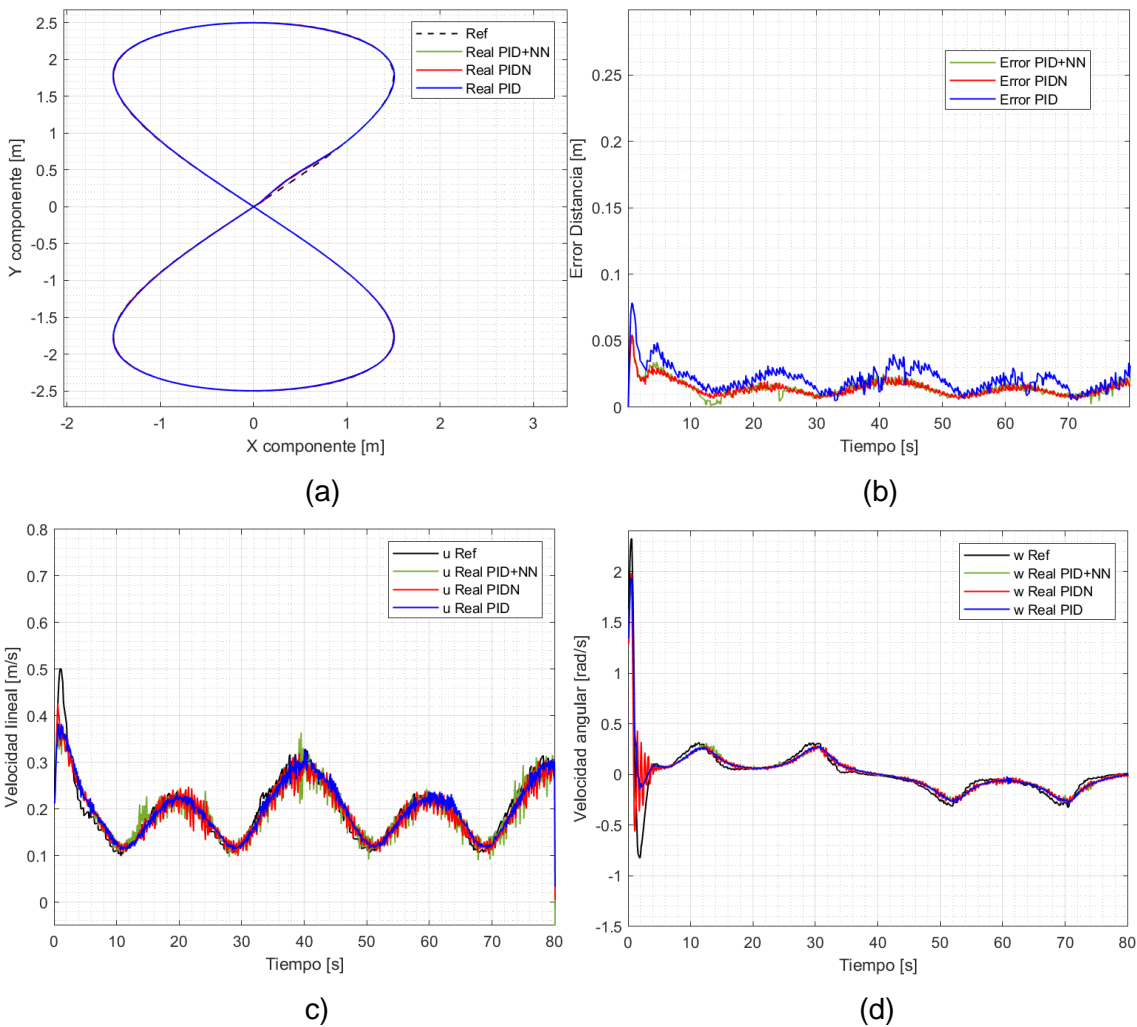
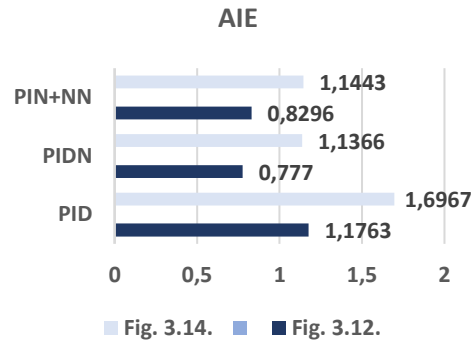
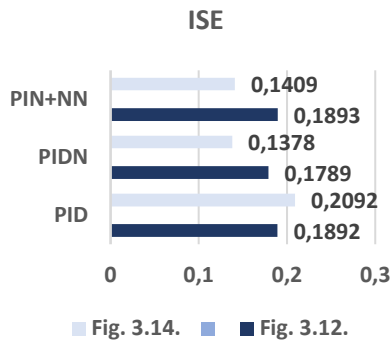


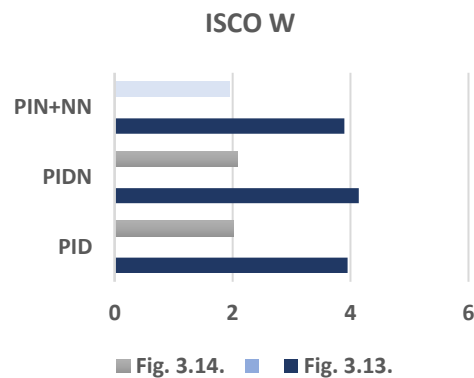
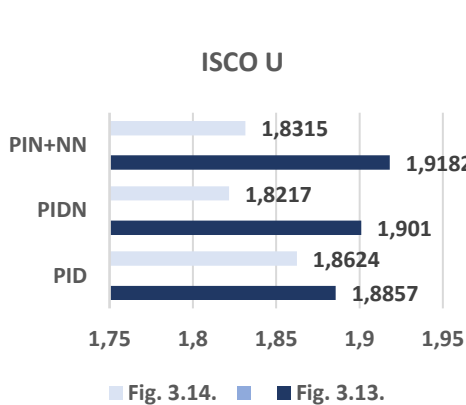
Figura 3.14. (a) Seguimiento de trayectoria lemniscata de 2,5 m de radio, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular



(a)

(b)

Figura 3.15. Comparación cuantitativa en figura lemniscata (a) ISE, (b) IAE



(a)

(b)

Figura 3.16. Comparación cuantitativa en figura lemniscata (a) ISCO U, (b) ISCO W

Las Figuras 3.12a y 3.14a, muestran el seguimiento del robot móvil en una trayectoria lemniscata. Los resultados observados a causa del control cinemático y conexión con V-REP muestran cualitativamente un error oscilante de distancia cuadrático instantáneo promedio no mayor a 2 cm (Figura 3.14b) y a 3 cm (Figura 3.12b).

La velocidad lineal oscilante alrededor de 0.2 m/s (mostradas en la Figura 3.14c) y 0.3 m/s (en la Figura 3.13a) respectivamente, permite al robot acercarse a la referencia debido a una buena acción de control. La velocidad angular (Figura 3.13b y 3.14d) es seguida por cada propuesta de control de manera correcta con pequeñas oscilaciones propias de este tipo de sistemas robóticos.

El análisis cuantitativo de la trayectoria lemniscata realizado en base a los índices ISE, IAE e ISCO véase en la Figuras 3.15 y 3.16, muestran que el controlador PIDN tiene el menor error al realizar el seguimiento de esta trayectoria.

3.2.4 Trayectoria Abierta

La trayectoria abierta presenta un punto de partida y llegada diferente.

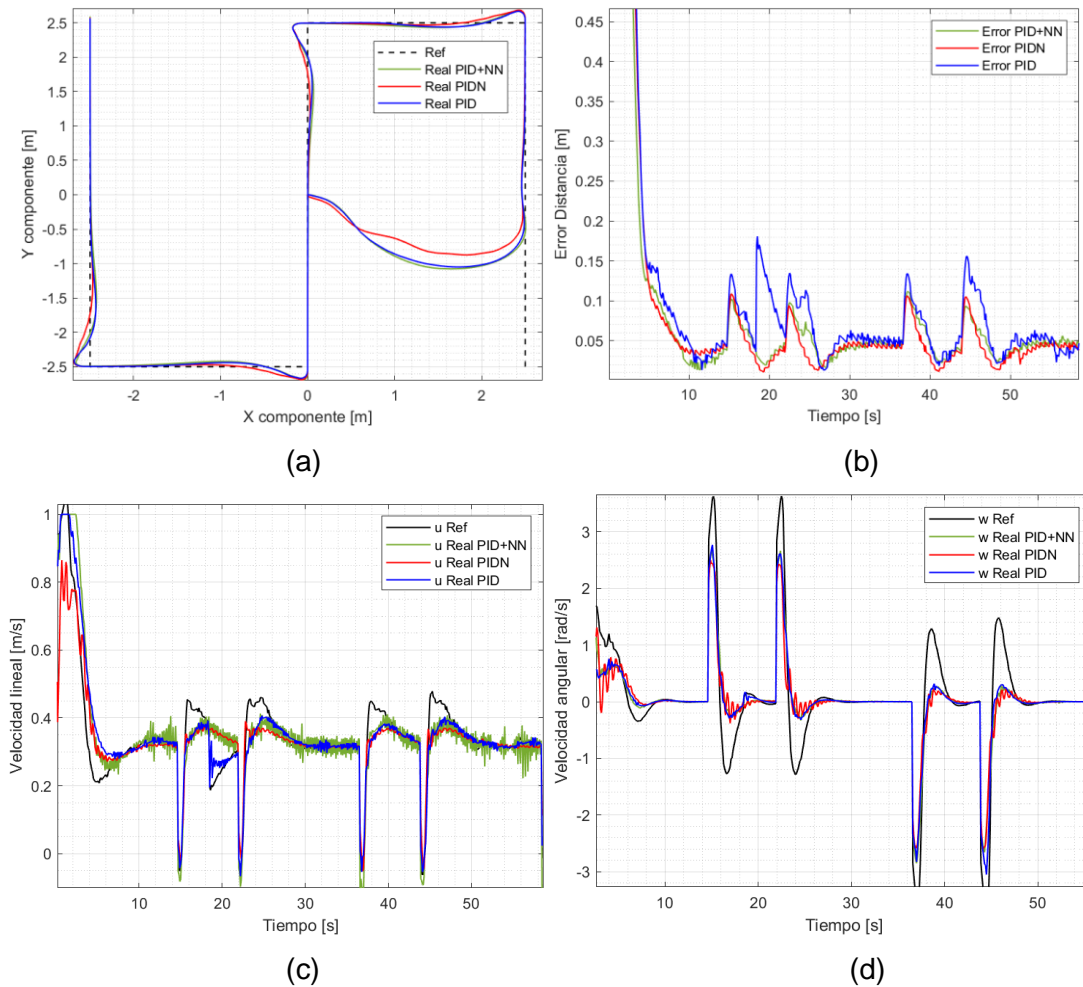


Figura 3.17. (a) Seguimiento de trayectoria abierta, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de $0,3 \text{ m/s}$ y (d) angular

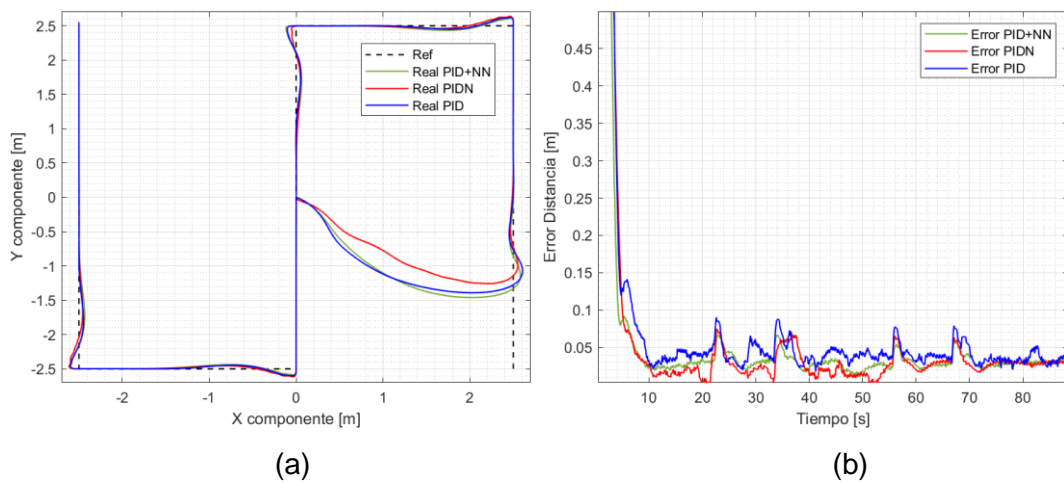


Figura 3.18. (a) Seguimiento de trayectoria abierta, (b) error cuadrático instantáneo de posición a una velocidad lineal de $0,2 \text{ m/s}$

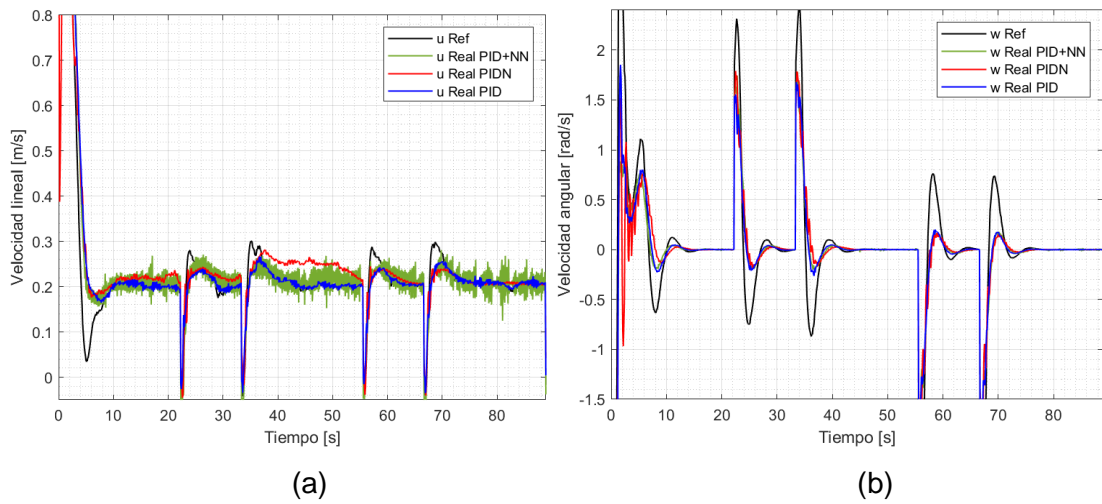


Figura 3.19. Seguimiento de trayectoria abierta a velocidad (a) lineal de 0,2 m/s y (b) angular

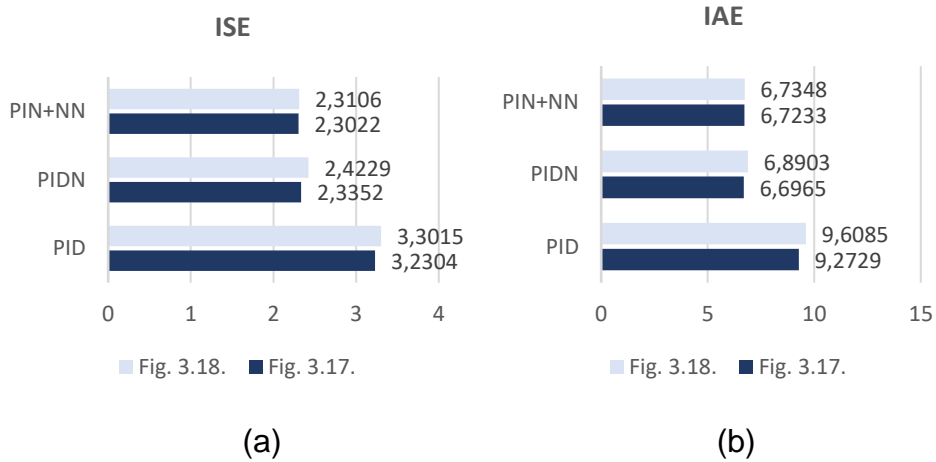


Figura 3.20. Comparación cuantitativa (a) ISE, (b) IAE, en la trayectoria abierta

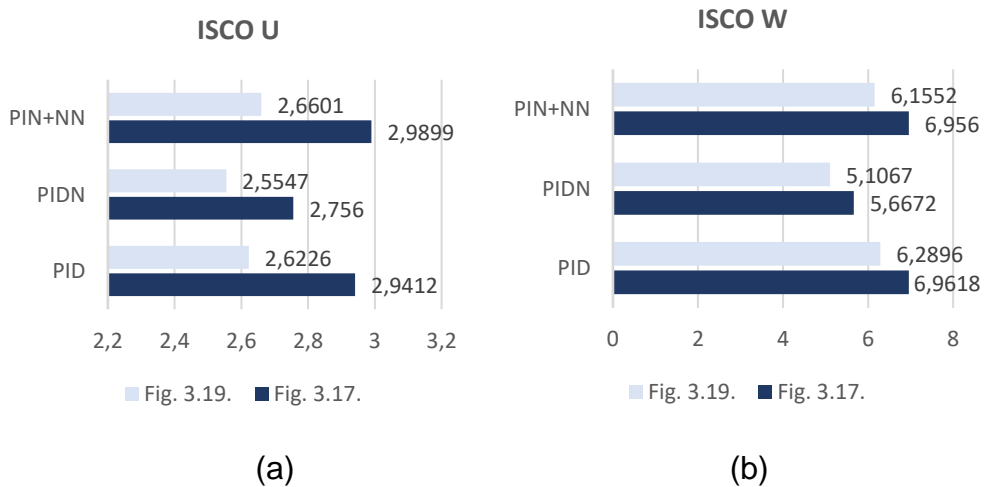


Figura 3.21. Comparación cuantitativa (a) ISCO U, (b) ISCO W, en la trayectoria abierta

Las Figuras 3.17a y 3.18a, muestran el seguimiento del robot móvil en una trayectoria abierta. Los resultados observados a causa del control cinemático muestran cualitativamente un error de distancia cuadrático instantáneo promedio no mayor a 3 cm (Figura 3.18b) y a 5 cm (Figura 3.17b), debido a la conexión con V-REP, donde en los cambios abruptos de orientación el error no sobrepasa los 16 cm , lugares donde los controladores deben actuar de manera abrupta.

La velocidad lineal 0.2 m/s (mostradas en la Figura 3.19a) y 0.3 m/s (Figura 3.17c), son alcanzadas por todas las propuestas de controladores pero particularmente los neuronales presentan un mejor comportamiento presentando un error igual a cero cuando el sistema logra alcanzar la recta, de igual modo se puede observar que el error de velocidad angular (visto en las Figuras 3.17d y 3.19b) es igual a cero en todos los casos pero los controladores neuronales alcanzan la referencia más rápidamente.

El análisis cuantitativo de la trayectoria circular realizado en base a los índices ISE, IAE e ISCO (Figuras 3.20 y 3.21), muestran que nuevamente el controlador PIDN presenta el menor error al realizar el seguimiento de trayectoria abierta especialmente mejorando de manera considerable el comportamiento del PID convencional.

3.3 Pruebas ante perturbaciones

Para analizar cómo actúa el sistema ante perturbaciones, el robot se traslada a través de un plano inclinado en diferentes instantes de tiempo como se observa en la Figura 3.22, lo cual va a afectar la dinámica del robot móvil.

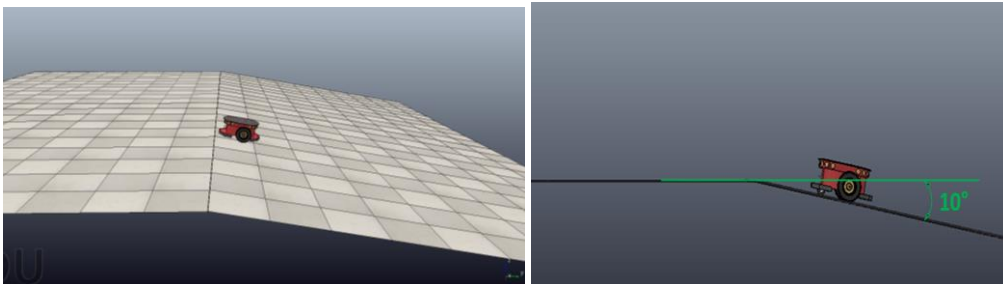
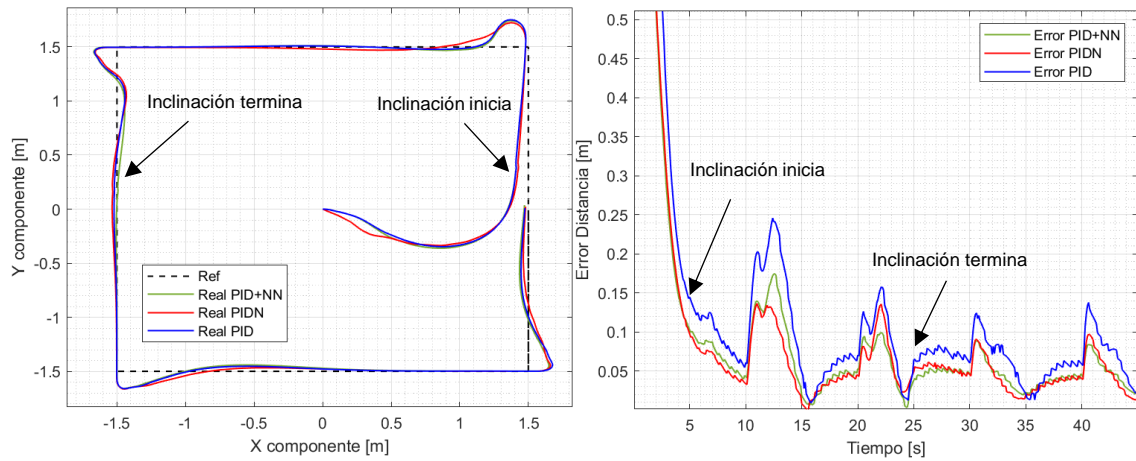


Figura 3.22. Plano inclinado

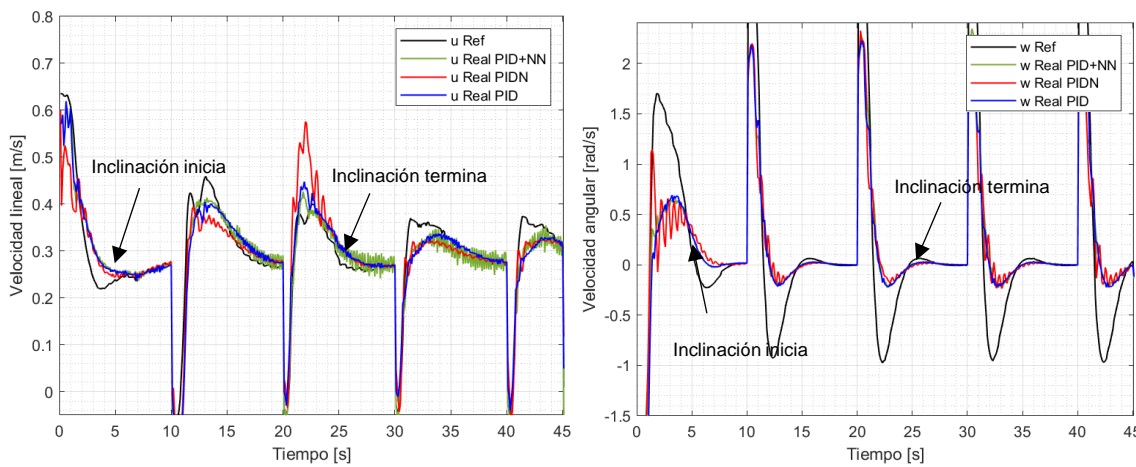
3.3.1 Trayectoria Cuadrada con Plano Inclinado

La inclusión de un plano inclinado hace que tenga el robot mayor dificultad de tomar las curvas durante la curva de la trayectoria cuadrada.



(a)

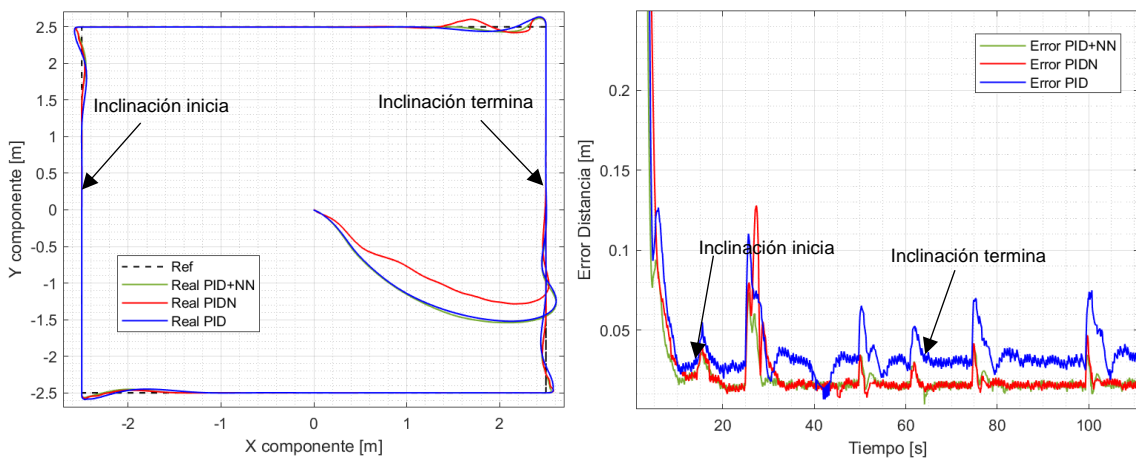
(b)



(c)

(d)

Figura 3.23. (a) Seguimiento de trayectoria cuadrada de 3 m de lado en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular



(a)

(b)

Figura 3.24. (a) Seguimiento de trayectoria cuadrada de 5 m de lado en plano inclinado, (b) error cuadrático instantáneo de posición a velocidad lineal de 0,2 m/s

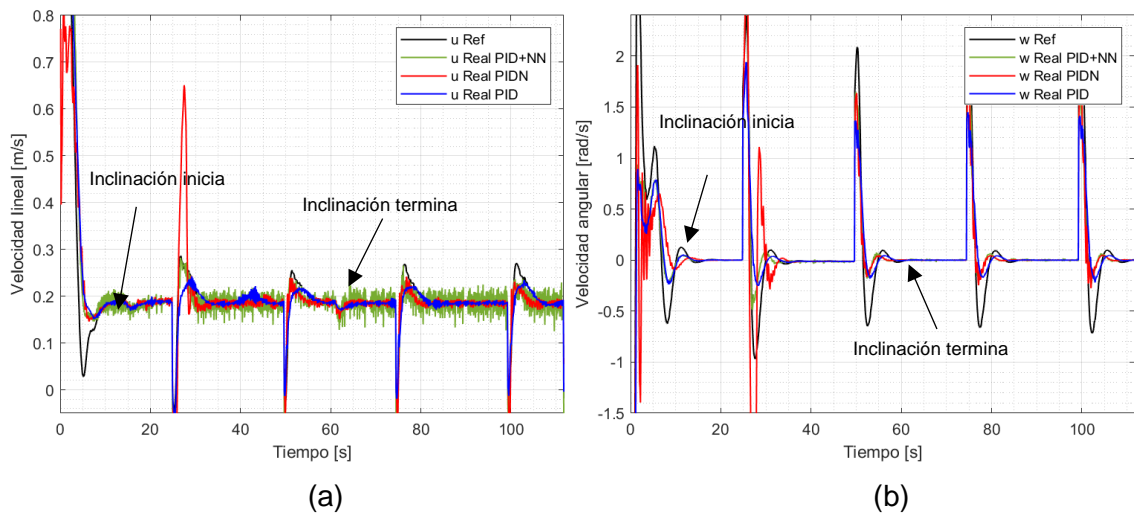


Figura 3.25. Seguimiento de trayectoria cuadrada de 5 m de lado en plano inclinado, velocidad (a) lineal de 0,2 m/s y (b) angular

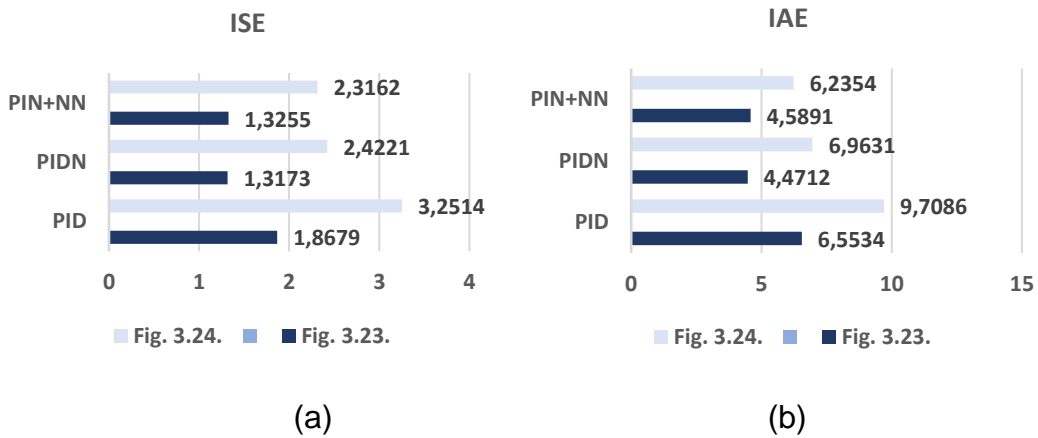


Figura 3.26. Comparación cuantitativa (a) ISE, (b) IAE, en trayectorias cuadradas en plano inclinado.

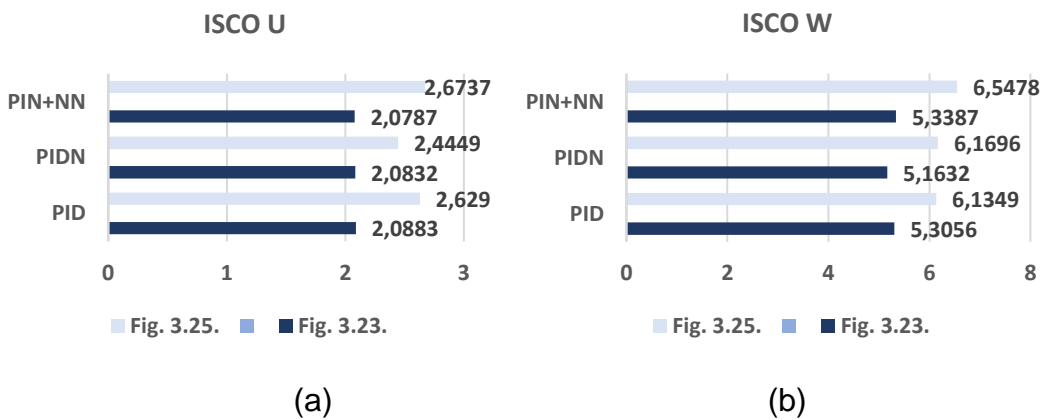


Figura 3.27. Comparación cuantitativa (a) ISCO U, (b) ISCO W, en trayectorias cuadradas en plano inclinado.

Las Figuras 3.23a y 3.24a, muestran el seguimiento del robot móvil en una trayectoria cuadrada con inclinación del plano. Los resultados observados muestran cualitativamente que el error de distancia cuadrático promedio instantáneo causado por el controlador cinemático es no mayor a 2 cm (Figura 3.24b) y a 4 cm (Figura 3.23b). En el caso de las esquinas, se observa que existe mayor error en dos primeras esquinas, lugar donde se realiza el cambio dinámico al robot y gracias al controlador dinámico el error no alcanzar un valor mayor a 15 cm en los cambios bruscos de orientación.

La velocidad lineal es alrededor de 0.2 m/s (mostradas en la Figura 3.25a) y 0.3 m/s (en la Figura 3.23c), con una acción de control (de los controladores basados en redes neuronales) agresiva alcanzado de manera más rápida la referencia deseada.

En el caso de la velocidad angular, se muestra como el controlador reduce el error a cero durante las rectas de la trayectoria (Figuras 3.23d y 3.25b).

El análisis cuantitativo que se ha realizado en base a los índices ISE, IAE e ISCO (Figura 3.26 y 3.27), muestra que el controlador PIDN tiene una respuesta más agresiva disminuyendo el valor de índice de desempeño, respecto al resto de controladores.

3.3.2 Trayectoria Circular con Plano Inclinado

El seguimiento de robot en una trayectoria circular con un plano inclinado tiene un aumento de velocidad lineal al entrar o salir del mismo.

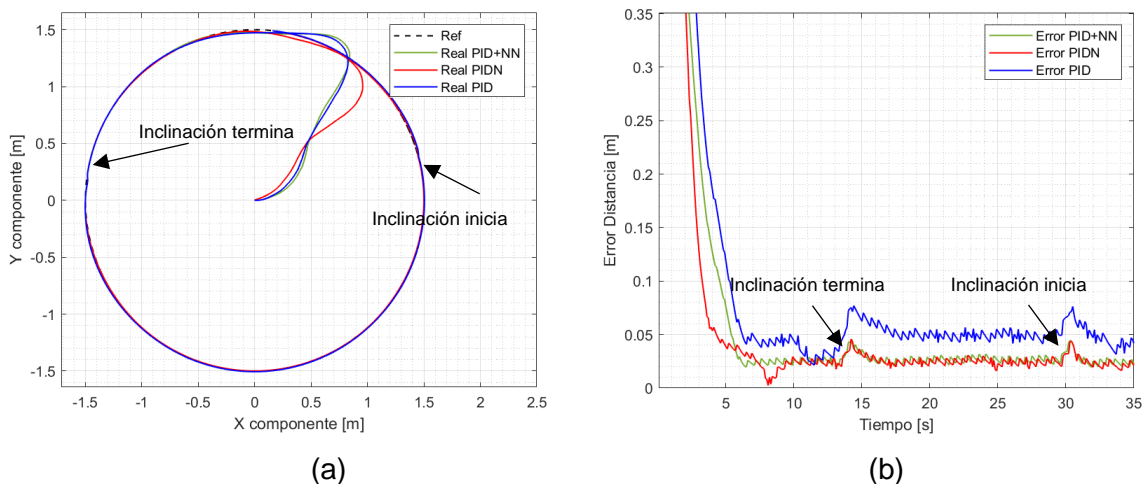


Figura 3.28. (a) Seguimiento de trayectoria circular de $1,5\text{ m}$ de radio en plano inclinado, (b) error cuadrático instantáneo de posición, a velocidad lineal de $0,3\text{ m/s}$

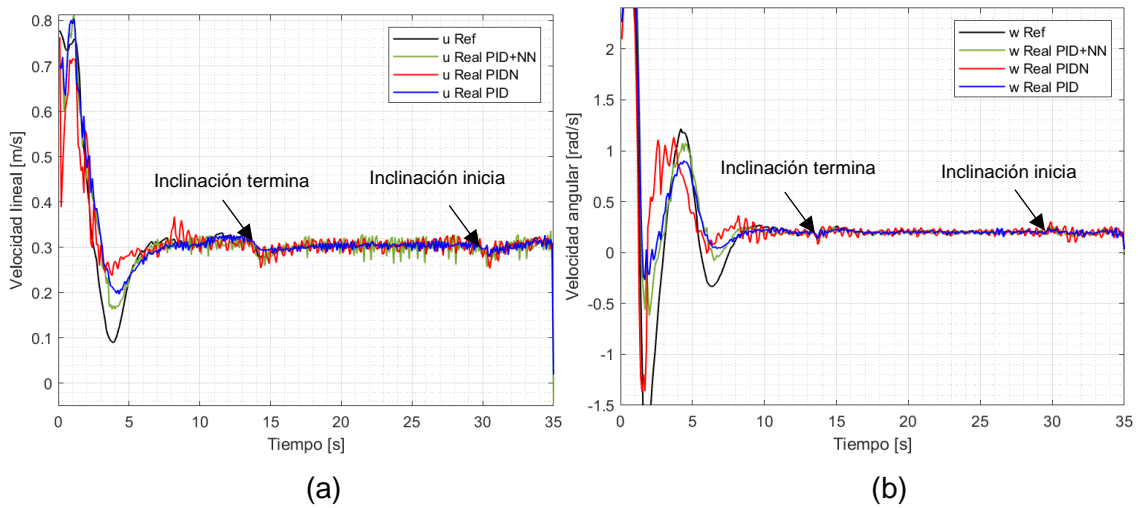


Figura 3.29. Seguimiento de trayectoria circular de 1,5 m de radio en plano inclinado, velocidad (c) lineal de 0,3 m/s y (d) angular

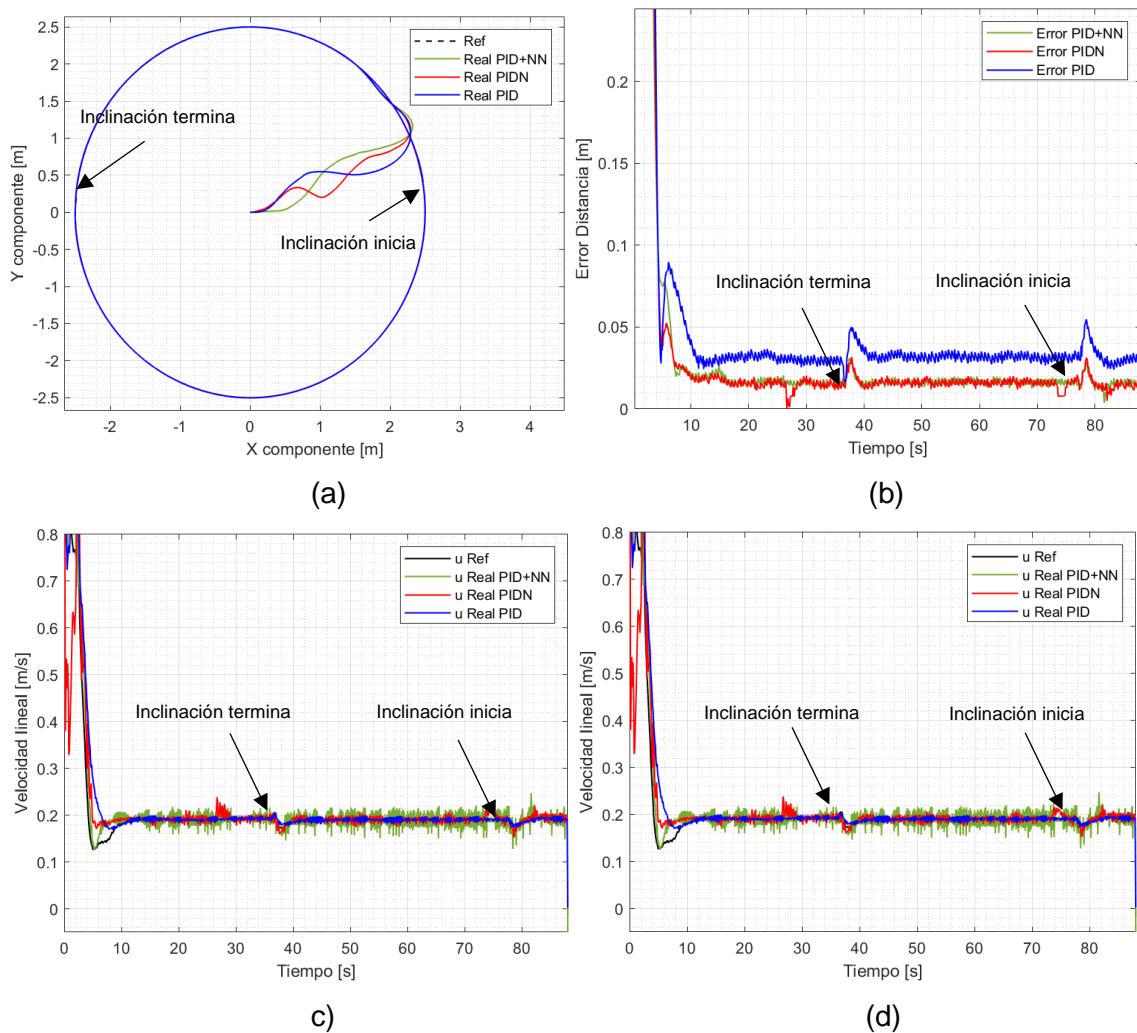


Figura 3.30. (a) Seguimiento de trayectoria circular de 2,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular

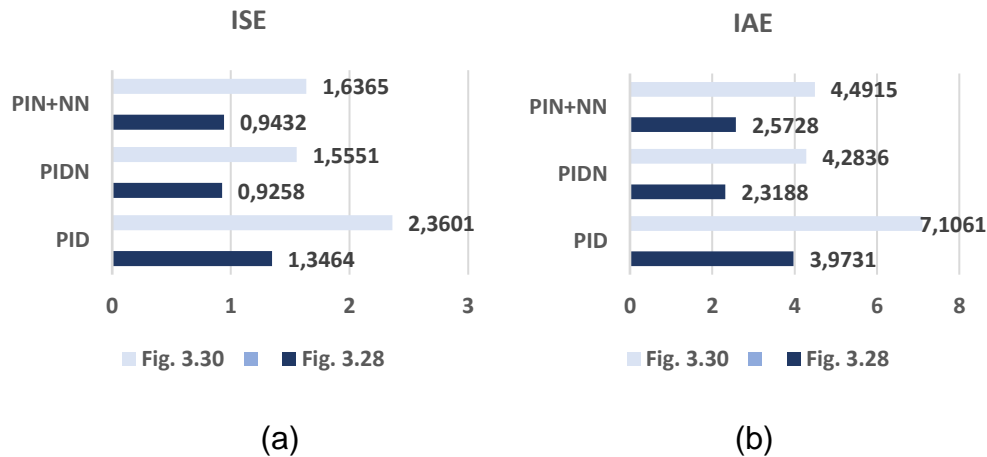


Figura 3.31. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria circular con plano inclinado

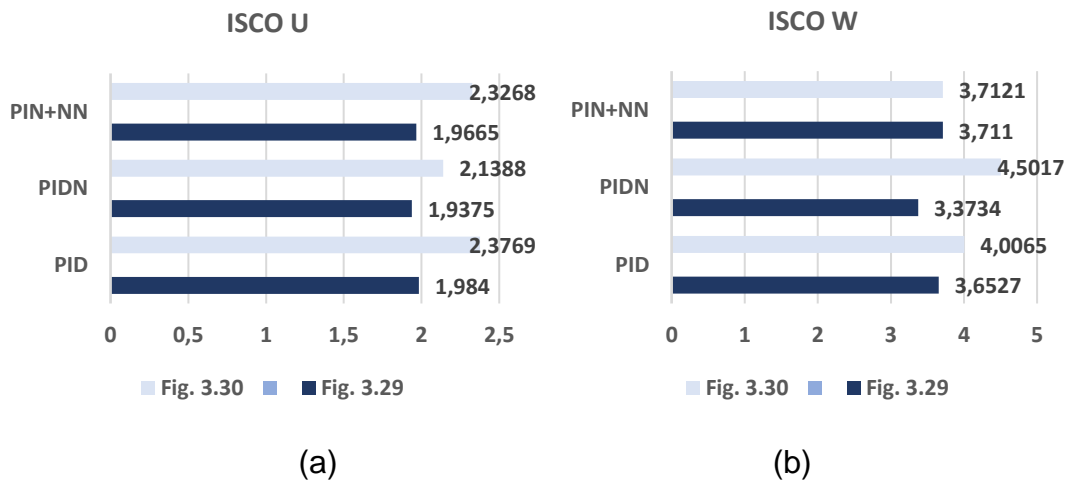


Figura 3.32. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria circular con plano inclinado

Las Figuras 3.28a, 3.30a, muestran el seguimiento del robot móvil en una trayectoria circular en un plano inclinado. Los resultados observados muestran cualitativamente el error de distancia cuadrático instantáneo causado por el controlador cinemático que es no mayor a 2 cm (ver Figura 3.30b) y a 4 cm (ver Figura 3.28b), donde se observa que al entrar o salir del plano inclinado el error de posición aumenta en un 50% sin embargo por los controladores inteligentes de manera rápida antes de que el sistema se torne inestable.

La velocidad lineal es alrededor de 0.2 m/s (mostradas en la Figura 3.30c) y 0.3 m/s (en la Figura 3.29a), con una acción de control (basados en redes neuronales) más agresiva para llegar a la referencia más rápido que con el controlador PID lo cual mejora los índices de desempeño.

En el caso de la velocidad angular (vista en las Figuras 3.29b y 3.30d), la acción de control (basada en redes neuronales) es más agresiva llegando al estado estable en menor tiempo, lo cual es deseado en sistemas que requieren controladores en lazo cerrado para control de robots.

El análisis cuantitativo se ha realizado en base a los índices ISE, IAE e ISCO (Figuras 3.31 y 3.32), muestra que el controlador PIDN disminuye el error de posición y pero tiene mayor error en la acción de control debido a que esta es más agresiva para llegar a la referencia.

3.3.3 Trayectoria Lemniscata con Plano Inclinado

La trayectoria lemniscata con plano inclinado influye a la salida del robot, por lo que se observa un aumento de velocidad lineal.

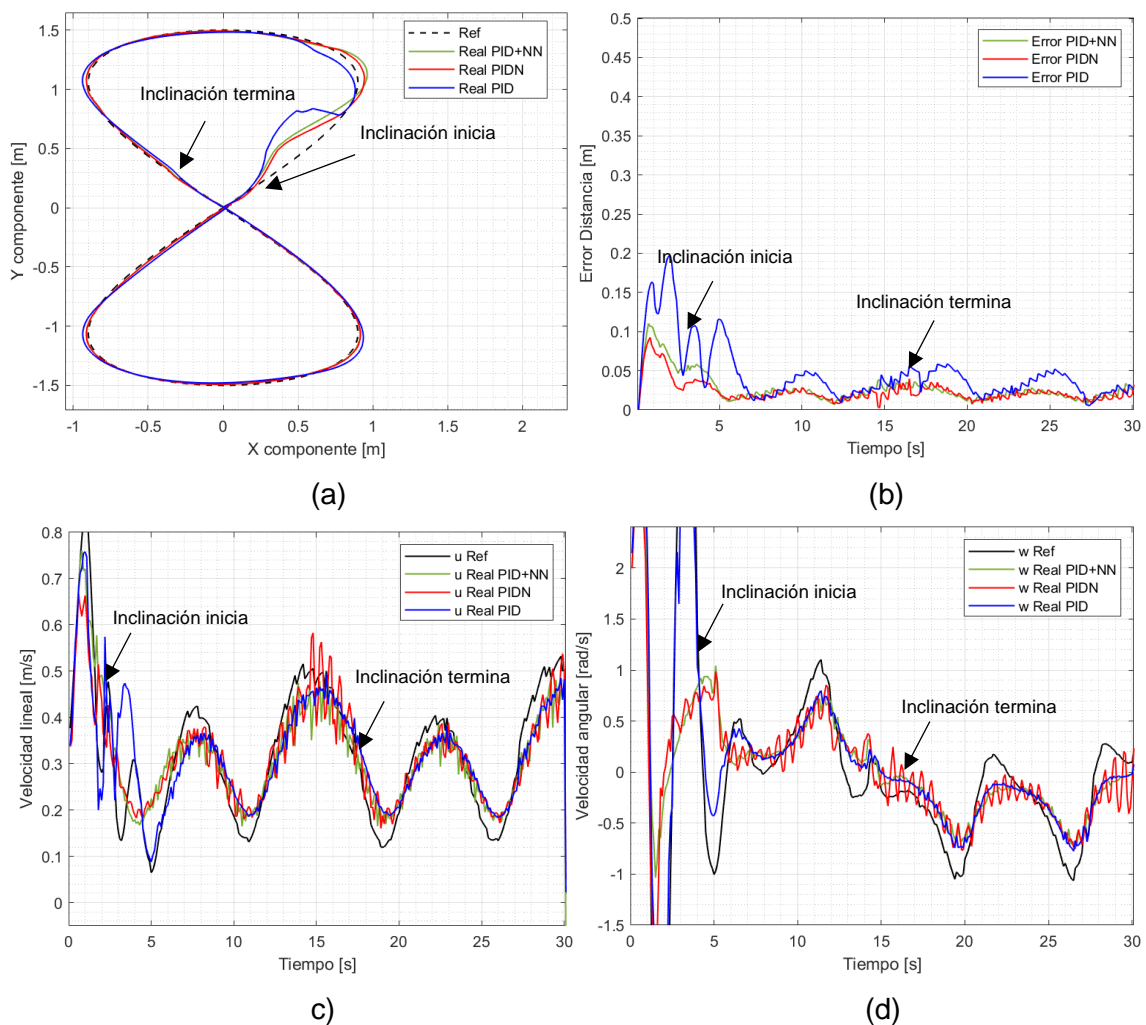


Figura 3.33. (a) Seguimiento de trayectoria lemniscata de 1,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,3 m/s y (d) angular

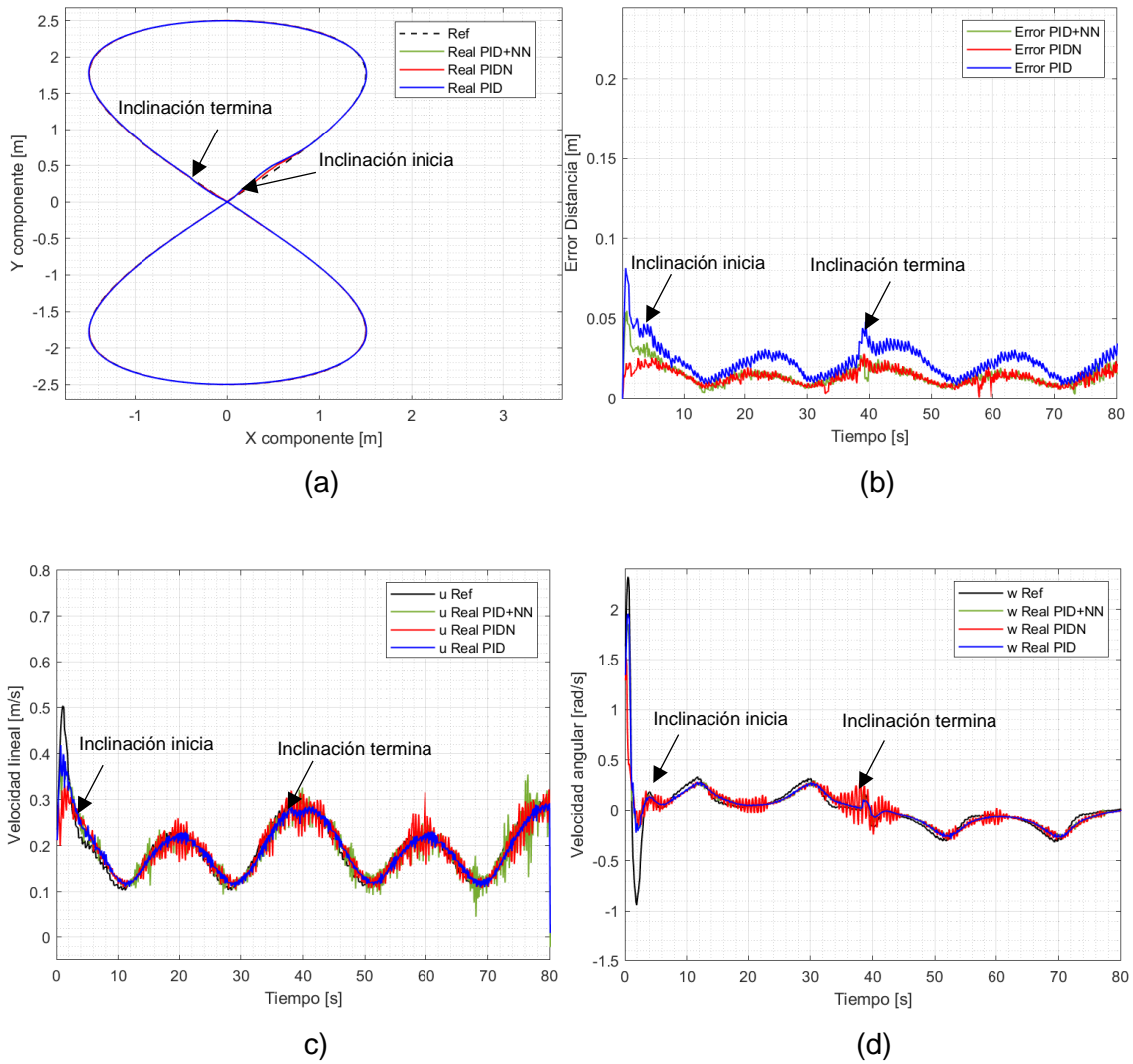


Figura 3.34. (a) Seguimiento de trayectoria lemniscata de 2,5 m de radio en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de 0,2 m/s y (d) angular

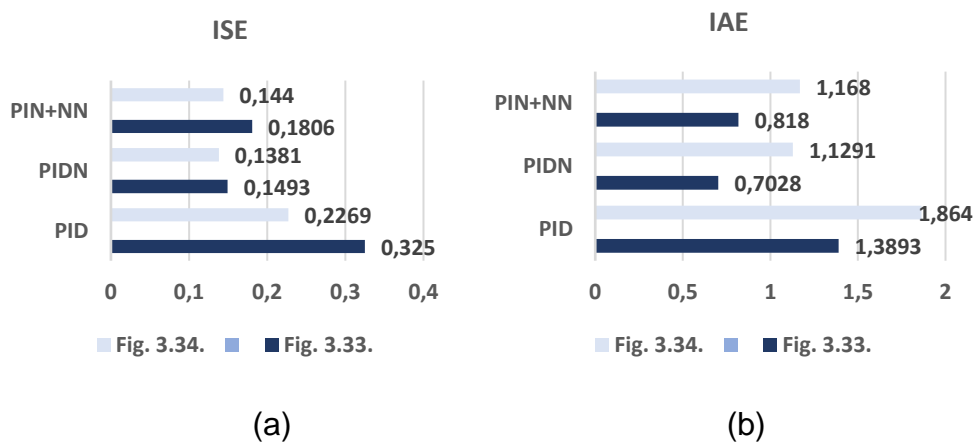


Figura 3.35. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria lemniscata con plano inclinado

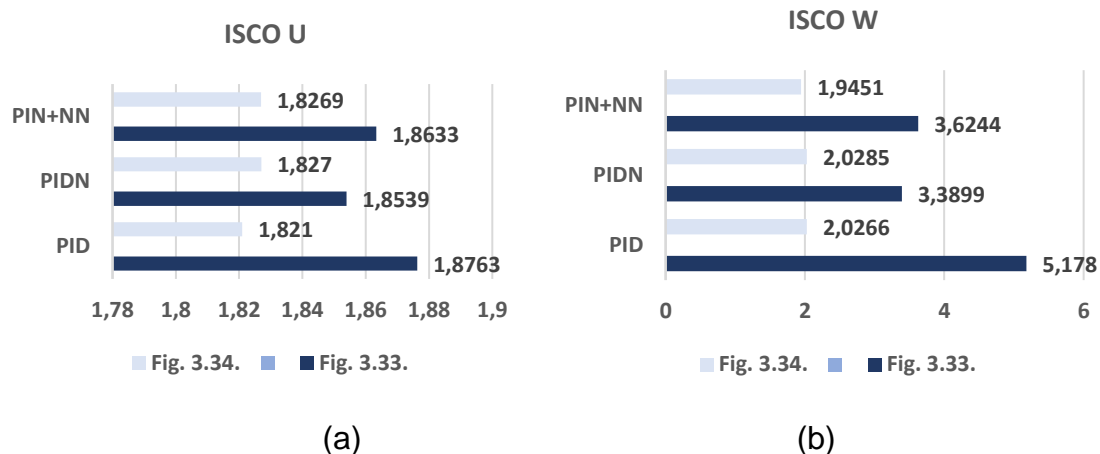


Figura 3.36. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria lemniscata con plano inclinado

Las Figuras 3.33a y 3.34a, se muestra el seguimiento del robot móvil en una trayectoria lemniscata en plano inclinado. Los resultados observados muestran cualitativamente que el error de distancia cuadrático instantáneo a causa del controlador cinemático es no mayor a 2 cm (Figuras 3.34b) y a 3 cm (Figura 3.33b), un valor aceptable debido a que el error es oscilando y se debe a su generación de trayectoria, y se observa que en los puntos de entrada y salida del plano inclinado el error aumenta.

La velocidad lineal es alrededor de 0.2 m/s (mostrada en las Figura 3.34c) y 0.3 m/s (en la Figura 3.33c), donde los controladores dinámicos alcanzan la referencia, siendo la acción de control más agresiva con los controladores inteligentes permitiendo llegar más rápido a la referencia con respecto al controlador PID clásico.

En el caso de la velocidad angular, se muestra las acciones de control (vistas en las Figuras 3.33d y 3.34d) más agresivas y debido a esto más oscilantes para los controladores inteligentes.

El análisis cuantitativo que se ha realizado en base a los índices ISE, IAE e ISCO (Figuras 3.35 y 3.36) muestra un mayor valor en los índices de rendimiento para el control PID clásico, debido a que al principio de la trayectoria tiene mayor dificultad de ser seguida por los cambios constantes de orientación y los cambios de dinámica, mientras que los controladores inteligentes presentan la ventaja de su capacidad de adaptación de sus ganancias cuando existen cambios permanentes en dinámica y orientación.

3.3.4 Trayectoria Abierta con Plano Inclinado

En las Figuras 3.37, 3.38 y 3.39, se muestran los Puntos 1 y 3, que son los inicios de la inclinación y el Punto 2, es la terminación de la inclinación del plano.

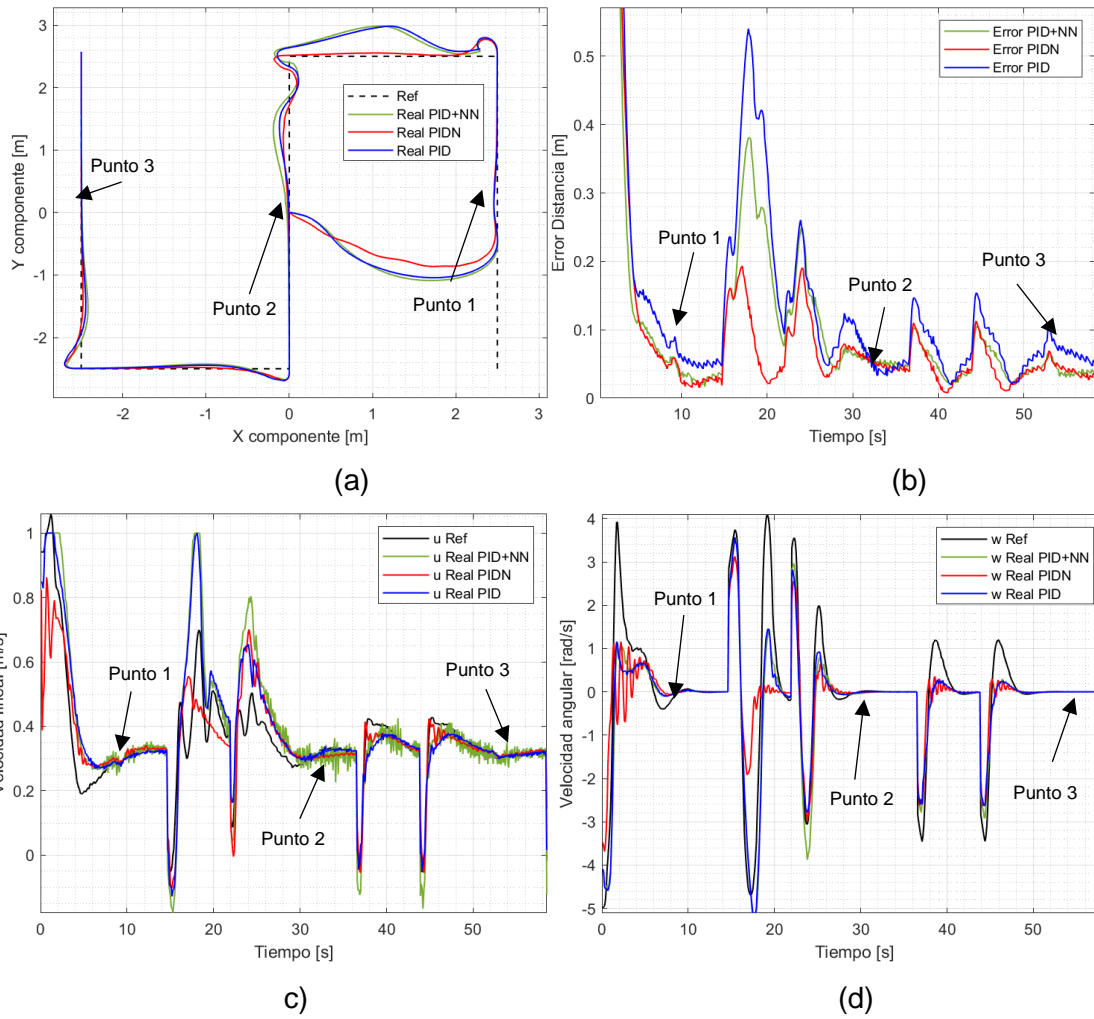


Figura 3.38. (a) Seguimiento de trayectoria abierta en plano inclinado, (b) error cuadrático instantáneo de posición, velocidad (c) lineal de $0,3 \text{ m/s}$ y (d) angular

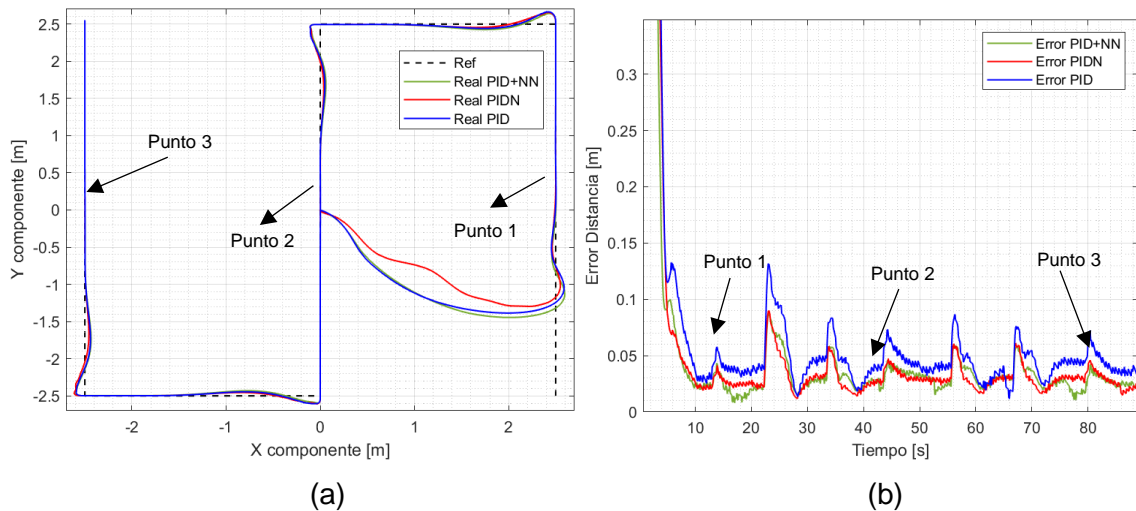


Figura 3.39. (a) Seguimiento de trayectoria abierta en plano inclinado, (b) error cuadrático instantáneo de posición a velocidad lineal de $0,2 \text{ m/s}$

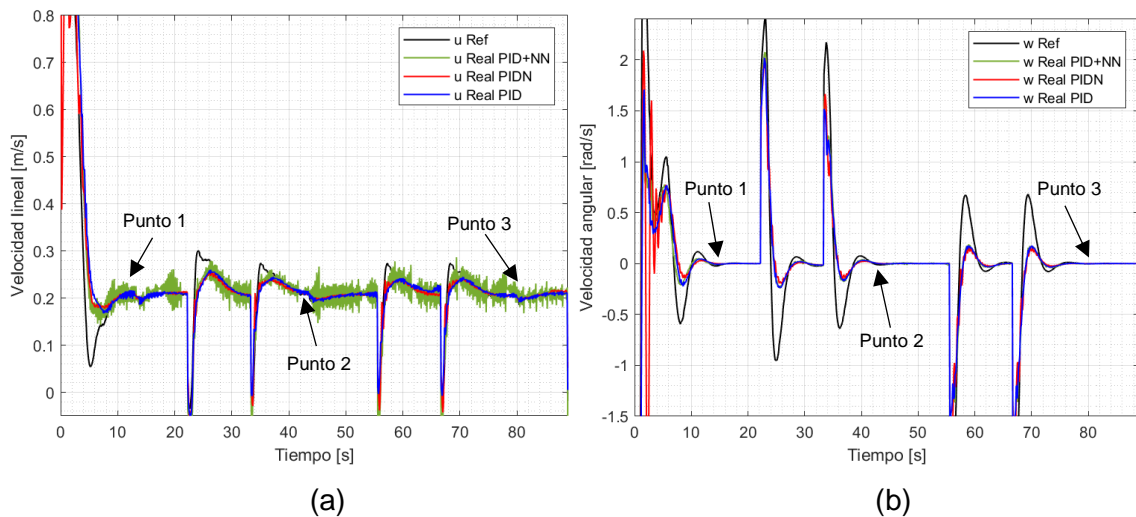


Figura 3.40. Seguimiento de trayectoria abierta en plano inclinado, velocidad (a) lineal de 0,2 m/s y (b) angular

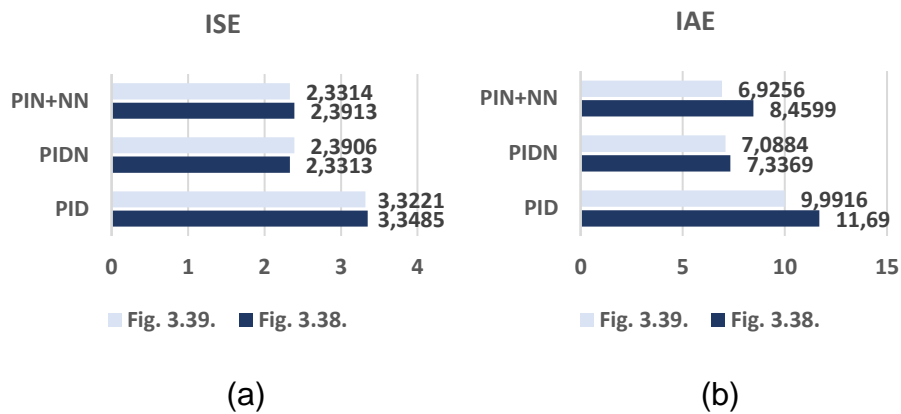


Figura 3.41. Comparación cuantitativa (a) ISE, (b) IAE, trayectoria abierta con plano inclinado

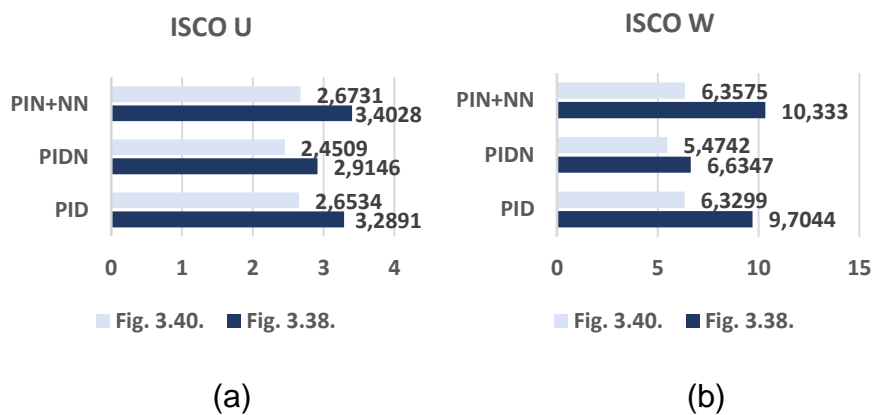


Figura 3.42. Comparación cuantitativa (a) ISCO U, (b) ISCO W, trayectoria abierta con plano inclinado

En Figuras 3.38a y 3.39a, muestran el seguimiento del robot móvil en una trayectoria abierta con plano inclinado. Los resultados observados muestran cualitativamente que tiene un error de distancia cuadrático instantáneo no mayor a 3 cm (ver Figuras 3.39b) y a 5 cm (Figura 3.38b) en la parte estable, debido a la conexión con V-REP y al plano inclinado, donde se puede observar un aumento de error en los cambios de orientación abruptos, sin embargo, se ha podido determinar visualmente que el controlador PIDN toma de mejor manera las curvas cuando la velocidad lineal es de 0.3 m/s, lo que lo hace un controlador más rápido.

La velocidad lineal es alrededor de 0.2 m/s (mostradas en la Figura 3.40a) y 0.3 m/s (en la Figura 3.38c), siendo más agresiva la señal de control (de los controladores basados en redes neuronales) para llegar en menor tiempo al estado estacionario.

En el caso del error de velocidad angular se muestra que es igual a cero en los tramos rectos de la trayectoria (Figuras 3.38d y 3.40b).

El análisis cuantitativo se ha realizado en base a los índices ISE, IAE se muestra en las Figuras 3.41 y 3.42, donde se evidencia que el error de posición es menor al hacer uso de controladores basados en redes neuronales, y para en este caso específico, el controlador PIDN toma de mejor los cambios de orientación abruptos, teniendo una respuesta más suave, pero una acción de control más agresiva respecto a los otros dos controladores.

3.4 CONCLUSIONES

En este trabajo se ha diseñado controladores basados en redes neuronales (PIDN, PID+NN) aplicados a un robot móvil de tracción diferencial para seguimiento de trayectoria, validando las respuestas de los controladores en el entorno V-REP de manera satisfactoria en cada una de las pruebas realizadas.

Se seleccionó el robot móvil de tracción diferencial Pioneer 3DX dentro de la librería del software VREP, ya que, durante años se ha utilizado el robot para investigaciones debido a la versatilidad, fiabilidad y durabilidad. Con la facilidad que el software presenta para realizar mapeo/localización del robot, es sencillo obtener los datos de posición para realizar el seguimiento de las diferentes trayectorias implementadas en este trabajo.

Se ha implementado cuatro tipos de trayectorias; cuadrada, circular, lemniscata y abierta. La trayectoria cuadrada y abierta tiene la característica de poseer esquinas, donde los cambios de orientación son abruptos. La trayectoria circular tiene una orientación uniforme, mientras que la trayectoria lemniscata tiene cambios de

orientación constantes. En todas estas trayectorias donde se probó el controlador cinemático, las respuestas han tenido los resultados esperados, dando el menor error de posición posible y presentando un rendimiento mejor que el del PID.

La estructura de los controladores implementados para el sistema cuenta con un controlador cinemático en el lazo externo basado en el modelo cinemático del robot móvil y un controlador dinámico en el lazo interno basado en redes neuronales (PID y PID+NN), que permitió mantener la velocidad lineal y angular constante ante la existencia de perturbaciones (plano inclinado), llegando a la referencia en el menor tiempo posible con respecto al controlador PID clásico.

Aunque en el objetivo del trabajo se ha planteado presentar un controlador neuronal para el control dinámico del robot, se ha optado por desarrollar dos configuraciones de controladores basadas en redes neuronales, las cuales toman como punto de partida el PID clásico. En la primera se busca optimizar las ganancias del controlador, mientras que en la segunda se hace una identificación en línea de la planta para posteriormente trabajar en paralelo con el controlador PID, las dos propuestas presentan mejoras considerables del sistema en lazo cerrado en general lo cual se ha demostrado a través de simulaciones de manera cualitativa y cuantitativa.

Dentro de los diferentes tipos de controladores inteligentes basados en redes neuronales se optó por: el controlador PID Neuronal y el controlador PID+NN, trabajando los dos casos de forma on-line, esto permitió que los controladores puedan adaptarse ante la existencia de perturbaciones. El controlador PID Neuronal permite ajustar los parámetros de un controlador PID clásico (K_P, K_I y K_D) para mantener la trayectoria ante una perturbación (plano inclinado), tomando de mejor manera los cambios de orientación. El controlador PID+NN realiza el ajuste de los parámetros del sistema identificado ($b = \frac{1}{K}$ y $c = \frac{a}{K}$), que son estimados por una red de identificación que toma información del modelo dinámico de primer orden, adicionando la salida del controlador, a la ley de control del PID, permitiendo mejorar el rendimiento del controlador clásico.

La simulación del robot móvil para seguimiento de trayectorias mediante los programas VREP y Matlab tuvo un desempeño, ya que el software VREP permite simular todas las variables del robot y el entorno físico (cambiando la dinámica del robot al colocar un plano inclinado a diez grados), mientras Matlab permitió procesar los datos para generar las acciones de control y poder realizar un análisis de los

resultados como: trayectoria seguida, error de posición, velocidad lineal, velocidad angular e índices de rendimiento.

Al realizar las comparaciones cualitativas entre los controladores, se comprobó una mejoría al hacer uso de controladores inteligentes basados en redes neuronales respecto al controlador PID clásico. Demostrando que el controlador PIDN tiene mejor respuesta que el controlador PID+NN, ya que, trabajando con la misma tasa de aprendizaje la respuesta de la ley de control es más agresiva sin provocar un sobreimpulso considerable, permitiendo llegar al valor deseado en menor tiempo, además de contar con menor cantidad de oscilaciones en la acción de control las cuales pueden causar daños en los actuadores. En todos los casos, las acciones de control no sobrepasaron los límites físicos y mecánicos admitidos por el robot móvil. Y el análisis cualitativo mediante los indicadores de desempeño demostró que los controladores basados en redes neuronales; tiene menor error de posición, mejoría en la acción de control tanto en velocidad lineal, como en velocidad angular, respecto al controlador PID clásico.

3.5 RECOMENDACIONES

Como recomendación para un trabajo futuro, se propone mejorar el rendimiento del controlador neuronal PID+NN, haciendo uso de un sistema de primer orden con retardo o de orden superior, esto permitiría tener una mejor aproximación del robot móvil con tracción diferencial.

Los controladores basados en redes neuronales permiten tener un mejor rendimiento de la respuesta del sistema no lineales, por lo que, se puede intentar validar el uso de este método aplicando con sistemas de control como lo pueden ser: LAMDA, Fuzzy, SMC o ANFIS, esto con el propósito de mejorar el método de calibración de los parámetros de ajustes de ganancias de la planta.

La calibración para el controlador PID Neuronal se debe realizar partiendo de un punto donde el sistema es estable, ya que, debido a la complejidad del modelo dinámico, el algoritmo de ajuste puede no ser suficientemente eficiente para llegar a la referencia de la trayectoria volviendo al sistema inestable. Para evitar esto, se toma como punto de partida el PID clásico calibrado para proceder al ajuste de las ganancias mediante la red neuronal.

Finalmente, se propone para controladores neuronales disminuir la tasa de aprendizaje lo cual incrementa el tiempo de convergencia a una solución que minimiza el error, pero tiene una mayor precisión al obtener los parámetros de aprendizaje, en un aprendizaje adaptativo como los utilizados en este trabajo.

4. BIBLIOGRAFÍA

- [1] Z. Hui and S. Ji-hong, "Neural Network Robust Control of Ship Trajectory Tracking," *International Conference on Mechatronics and Automation*, Aug. 2014.
- [2] S. Miah, F. Shaik, and H. Chaoui, "Universal Dynamic Tracking Control Law for Mobile Robot Trajectory Tracking," 2017.
- [3] B. Sarabia, "Diseño y Simulación de tres técnicas de control clásicas y robustas aplicadas al seguimiento de trayectorias ante la presencia de retardo fijos para la plataforma robótica Pioneer 3DX," *Escuela Politécnica Nacional*, Nov. 2017.
- [4] A. Vasickaninova, M. Bakosova, and A. Meszaros, "Control of Heat Exchangers in Series Using Neural Networks," in *2021 23rd International Conference on Process Control (PC)*, Jun. 2021, pp. 237–242. doi: 10.1109/PC52310.2021.9447484.
- [5] Wang Ting, Wang Heng, and Xie Hao-fei, "Networked synchronization control method by the combination of RBF neural network and genetic algorithm," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Feb. 2010, pp. 9–12. doi: 10.1109/ICCAE.2010.5451837.
- [6] V. Barrientos and J. García, "Robots Móviles: Evolución y Estado del Arte," *redalyc.org*, pp. 12–17, 2007.
- [7] Zhengguang Wu, Hongye Su, Jian Chu, and Wuneng Zhou, "Improved Delay-Dependent Stability Condition of Discrete Recurrent Neural Networks With Time-Varying Delays," *IEEE Trans Neural Netw*, vol. 21, no. 4, pp. 692–697, Apr. 2010, doi: 10.1109/TNN.2010.2042172.
- [8] Runqi Chai, Lijing Dong, and Baihai Zhang, "Design and stability analysis of nonlinear networked control systems with random delay based on Neural Network," in *2012 24th Chinese Control and Decision Conference (CCDC)*, May 2012, pp. 523–528. doi: 10.1109/CCDC.2012.6244079.
- [9] Dong Wei, Xinghua Pan, and Minglian Zhang, "Neural-Network- Based Multiple Stage Optimal Control of Variable-Air- Volume Systems," in *2006*

- 6th World Congress on Intelligent Control and Automation*, 2006, pp. 8804–8808. doi: 10.1109/WCICA.2006.1713701.
- [10] F. Ulloa, “Diseño e implementación de un sistema virtual para un sistema multivariable no lineal de cuatro tanques,” Escuela Politécnica Nacional, Quito, 2018.
- [11] A. Cumbajin, “Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el transporte de materiales,” Universidad de las Fuerzas Armadas, Quito, 2020.
- [12] O. Camacho, A. Rosales, and F. Rivas, *Control de Procesos*, Primera. Escuela Politécnica Nacional, 2020.
- [13] S. MARMOUCH, T. AROUI, and Y. KOUBAA, “APPLICATION OF STATISTICAL NEURONAL NETWORKS FOR DIAGNOSTICS OF INDUCTION MACHINE ROTOR FAULTS,” *17th international conference on Sciences and Techniques of Automatic control & computer engineering* -, Dec. 2016.
- [14] D. Matich, “Redes Neuronales: Conceptos Básicos y Aplicaciones,” *Grupo de Investigación Aplicada a la Ingeniería Química*, 2001.
- [15] R. Wang, Z. Zhou, and G. Qu, “Fuzzy Neural Network PID Control Based on RBF Neural Network for Variable Configuration Spacecraft,” in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Oct. 2018, pp. 1203–1207. doi: 10.1109/IAEAC.2018.8577860.
- [16] W. Qian, J. Cao, D. Xu, R. Nie, Z. Guan, and R. Zheng, “CNN-Based Embroidery Style Rendering,” *Intern J Pattern Recognit Artif Intell*, vol. 34, no. 14, p. 2059045, Dec. 2020, doi: 10.1142/S0218001420590454.
- [17] A. Swief, A. El-Zawawi, M. El-Habrouk, and A. N. Eldin, “Approximate Neural Network Model for Adaptive Model Predictive Control,” in *2020 16th International Computer Engineering Conference (ICENCO)*, Dec. 2020, pp. 135–140. doi: 10.1109/ICENCO49778.2020.9357373.
- [18] Y. Li, X. Wang, H. Zhang, and X. Liu, “An Attitude Control Method of Aircraft Based on Neural Network Batch Training Strategy,” in *2020 Chinese*

- Automation Congress (CAC)*, Nov. 2020, pp. 1788–1793. doi: 10.1109/CAC51589.2020.9327655.
- [19] Z. Li, Y. Xia, C.-Y. Su, J. Deng, J. Fu, and W. He, “Missile Guidance Law Based on Robust Model Predictive Control Using Neural-Network Optimization,” *IEEE Trans Neural Netw Learn Syst*, vol. 26, no. 8, pp. 1803–1809, Aug. 2015, doi: 10.1109/TNNLS.2014.2345734.
- [20] X. SHEN and W. SHI, “Adaptive Trajectory Tracking Control of Wheeled Mobile Robot,” *National Science Foundation of China under Grant*, 2019.
- [21] J. Fei, Y. Chen, L. Liu, and Y. Fang, “Fuzzy Multiple Hidden Layer Recurrent Neural Control of Nonlinear System Using Terminal Sliding-Mode Controller,” *IEEE Trans Cybern*, vol. 52, no. 9, pp. 9519–9534, Sep. 2022, doi: 10.1109/TCYB.2021.3052234.
- [22] S. A. Shirokov and A. J. Rypalov, “Dispatching method of incoming events for ERP system based on analysis of emails using artificial intelligent technique,” *J Phys Conf Ser*, vol. 1694, no. 1, p. 012034, Dec. 2020, doi: 10.1088/1742-6596/1694/1/012034.
- [23] adept, “Pioneer 3-DX,” *Adept Technology, Inc.* 2011.
- [24] M. Soori and R. Dastres, “Artificial Neural Network Systems,” 2021. [Online]. Available: <https://www.researchgate.net/publication/350486076>
- [25] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*. Prentice Hall/Pearson, 2009.
- [26] L. R. R. dos Santos, F. R. Durand, A. Goedel, and T. Abrão, “Auto-Tuning PID Distributed Power Control for Next-Generation Passive Optical Networks,” *Journal of Optical Communications and Networking*, vol. 10, no. 10, p. D110, Oct. 2018, doi: 10.1364/JOCN.10.00D110.
- [27] J. Nevot, “Diseño de un Contralador Avanzado basado en Redes Neuronales para la gestión de la mezcla aire-gasolina en un motor alternativo,” *Instituto de Organización y Control de Sistemas Industriales*, pp. 141–218, Dec. 1999.
- [28] T. Wang, H. Gao, and J. Qiu, “A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked Industrial

- Process Control,” *IEEE Trans Neural Netw Learn Syst*, vol. 27, no. 2, pp. 416–425, Feb. 2016, doi: 10.1109/TNNLS.2015.2411671.
- [29] T. B. Demarse and K. P. Dockendorf, “Adaptive Flight Control With Living Neuronal Networks on Microelectrode Arrays.”
- [30] Y. Che, B. Liu, H. Li, Y. Qin, and C. Han, “Neural network H-infinity synchronization control for time delay chaotic neuronal systems,” *IEEE*, 2016.
- [31] Y. Su and Y. Liang, “Research on Double Closed-loop Trajectory Tracking Control Algorithm of Wheeled Mobile Robot Based on Global Stability,” in *Proceedings - 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2020*, Aug. 2020, vol. 1, pp. 180–183. doi: 10.1109/IHMSC49165.2020.00048.
- [32] L. Gou, Z. Zhou, Y. Shen, W. Shao, and X. Zeng, “Predictive Control of Turbofan Engine Model Based on Improved Elman Neural Network,” in *2019 Chinese Control Conference (CCC)*, Jul. 2019, pp. 8842–8847. doi: 10.23919/ChiCC.2019.8866686.
- [33] X. Fu and S. Li, “Control of Single-Phase Grid-Connected Converters with LCL Filters Using Recurrent Neural Network and Conventional Control Methods,” *IEEE Trans Power Electron*, pp. 1–1, 2015, doi: 10.1109/TPEL.2015.2490200.
- [34] L. Morales, M. Herrera, O. Camacho, P. Leica, and J. Aguilar, “LAMDA Control Approaches Applied to Trajectory Tracking for Mobile Robots,” *IEEE Access*, vol. 9, pp. 37179–37195, 2021, doi: 10.1109/ACCESS.2021.3062202.
- [35] L. Benalcázar and C. Bonilla, “Diseño y simulación de tres técnicas de control basada en SMC con no linealidades y un controlador PID basado en SMC para el seguimiento de trayectorias en el Robot Pioneer 3DX utilizando CoppeliaSIM-Simulink,” *Escuela Politécnica Nacional*, 2022.
- [36] F. G. Rossomando, C. Soria, and R. Carelli, “Sliding mode neuro adaptive control in trajectory tracking for mobile robots,” *Journal of Intelligent and*

Robotic Systems: Theory and Applications, vol. 74, no. 3–4, pp. 931–944, 2014, doi: 10.1007/s10846-013-9843-5.

- [37] R. Hedjar, M. Alsulaiman, and K. Almutib, “Approximated nonlinear predictive control for trajectory tracking of a wheeled mobile robot,” in *Proceedings - 1st International Conference on Robot, Vision and Signal Processing, RVSP 2011*, 2011, pp. 296–299. doi: 10.1109/RVSP.2011.21.
- [38] M. ASAI, G. CHEN, and I. TAKAMI, “Neural network trajectory tracking of tracked mobile robot,” *International Multi-Conference on Systems, Signals & Devices*, 2019.
- [39] L. Morales, “FORMALIZACIÓN DE CONTROLADORES DIFUSOS LAMDA,” Escuela Politécnica Nacional, Quito, 2021.

5. ANEXOS

ANEXO I

MANUAL DE USUARIO DE INTERFAZ

Para realizar un análisis de los controladores propuestos se ha utilizado una interfaz gráfica implementada en Matlab, con la finalidad de facilitar la visualización de la trayectoria seguida, error de posición, velocidad lineal, velocidad angular e índices de desempeño del robot móvil.

En la Figura I.1. se observa la ventana de inicio la cual indica los datos de la tesis implementada, haciendo clic en el botón de “Iniciar”, se presenta la pantalla de Interfaz hombre máquina para el control del robot móvil en la Figura I.2.



Figura I.1. Pantalla de inicio

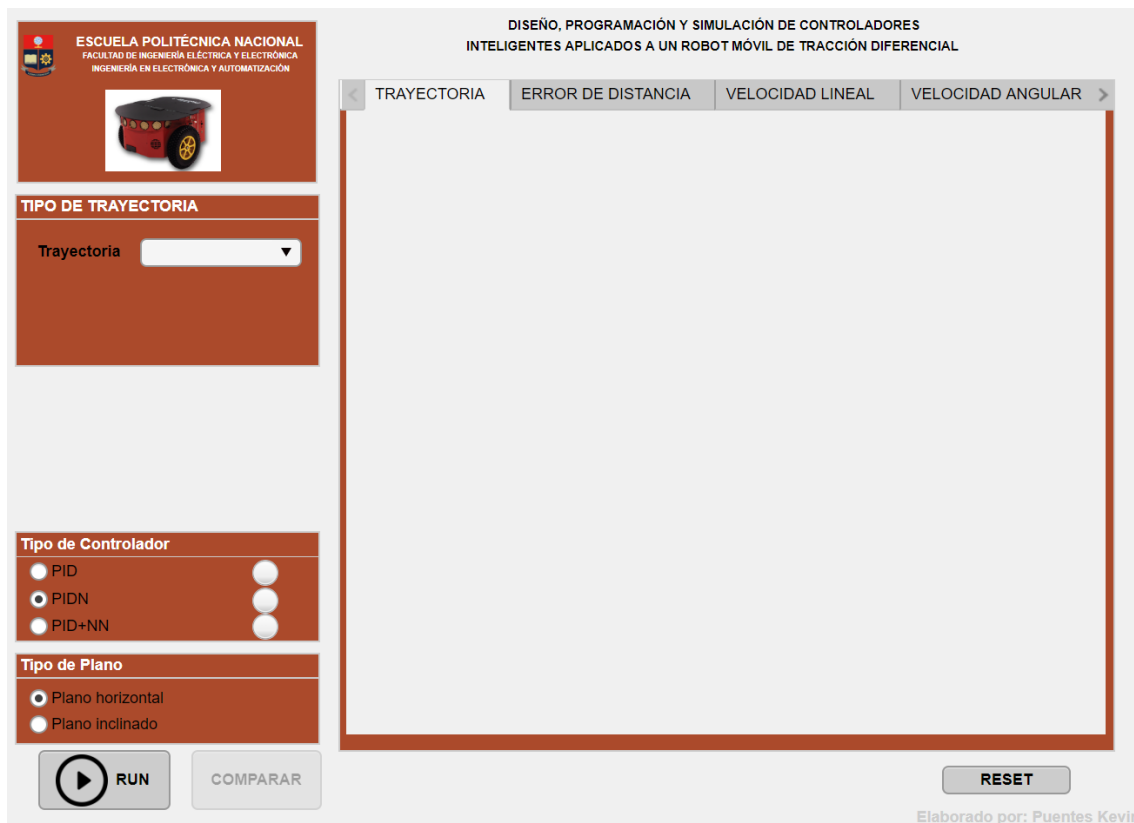


Figura I.2. Interfaz de Usuario

Dentro del Interfaz de Usuario (ver Figura I.2), en la sección izquierda se tiene un menú que permite escoger el tipo de trayectoria, el tiempo y la amplitud de la trayectoria (lado o radio), el tipo de controlador y el tipo de plano.

El usuario comienza seleccionando el tipo de trayectoria: Cuadrado, Circulo, Lemniscata o abierta, como se observa en la Figura I.3.

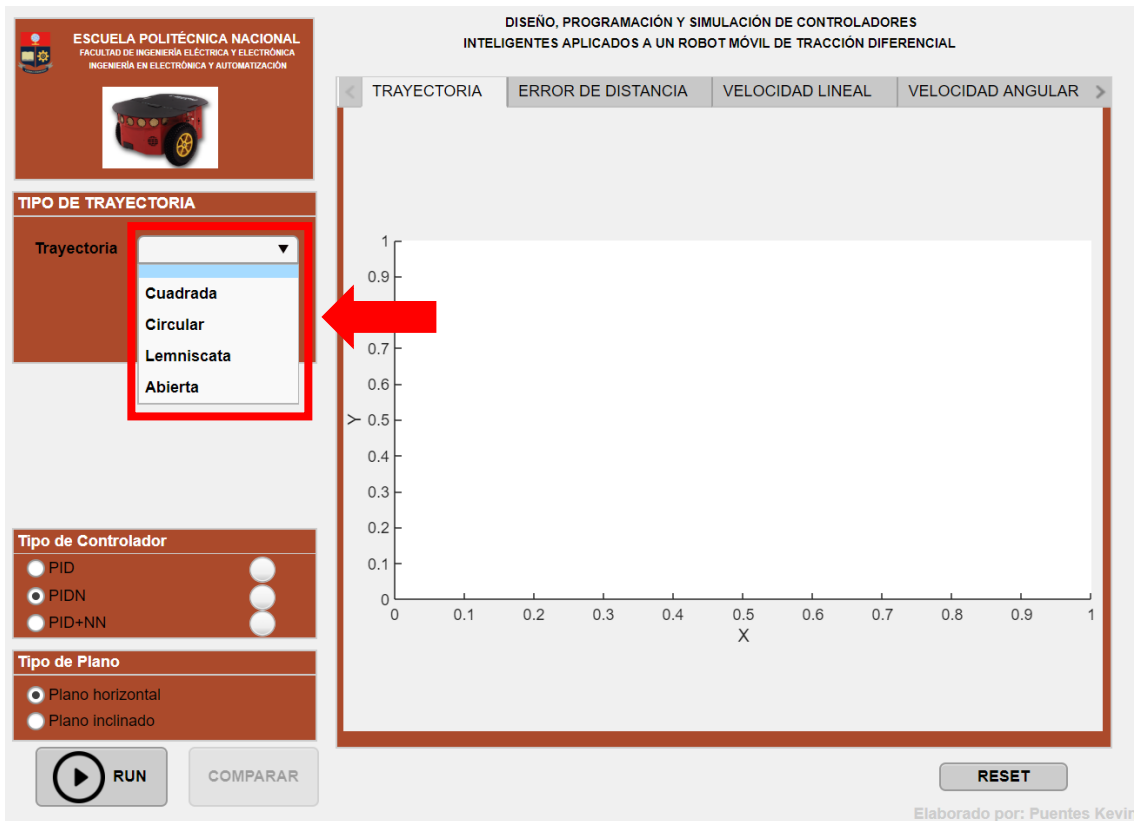


Figura I.3. Selección de trayectoria

Luego, se selecciona el tamaño de la figura y el tiempo de recorrido del robot móvil como se observa en la Figura I.4.

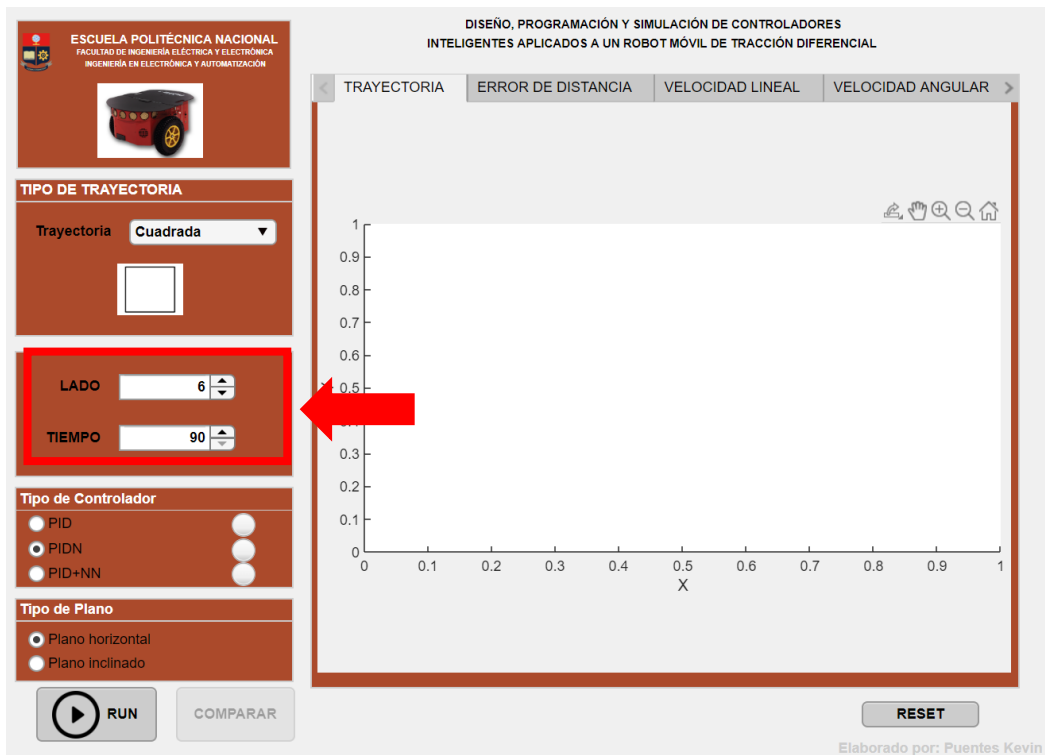


Figura I.4. Tamaño y tiempo de trayectoria

Una vez elegidos los parámetros de la trayectoria, se escoge el Tipo de Controlador.

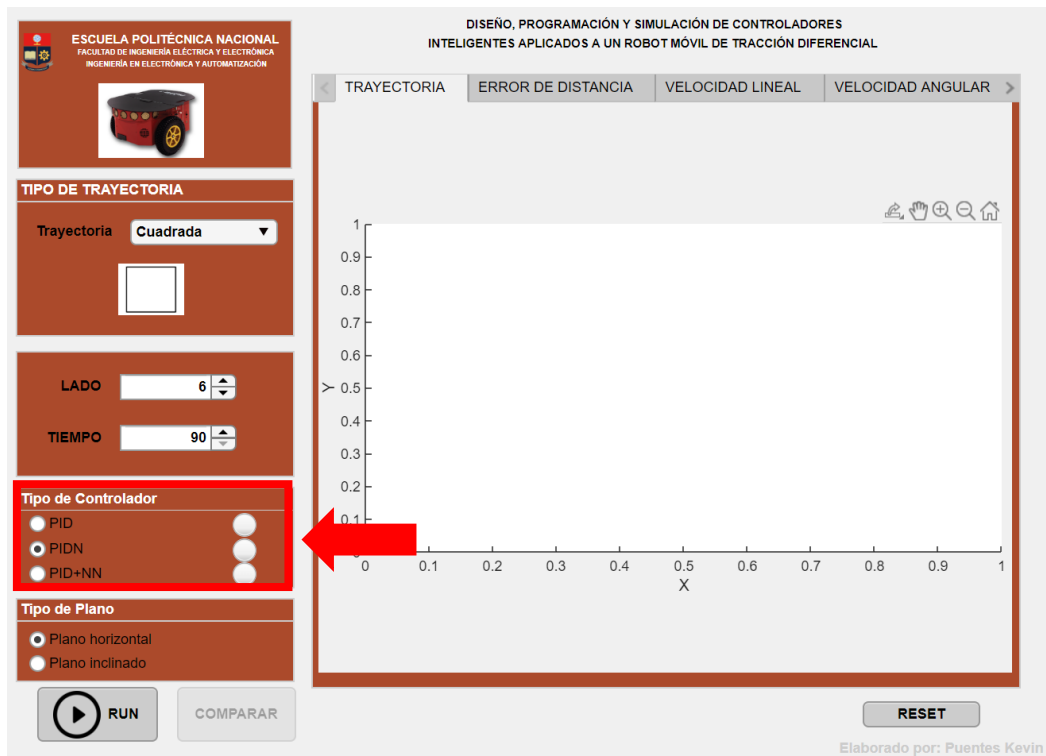


Figura I.5. Selección del controlador

En la Figura I.6. se selecciona el Tipo de Plano, horizontal (sin perturbación al sistema) e inclinado (con perturbación al sistema).

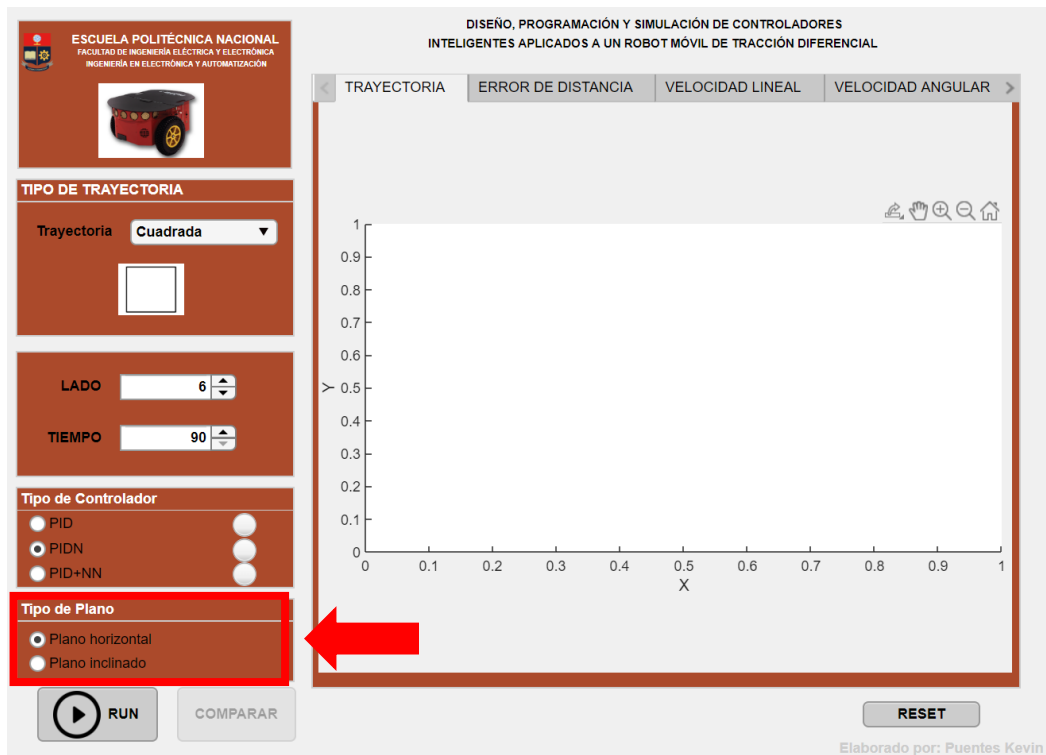


Figura I.6. Selección del Plano

Elegidos todos los parámetros se prosigue a correr el interfaz de V-REP.

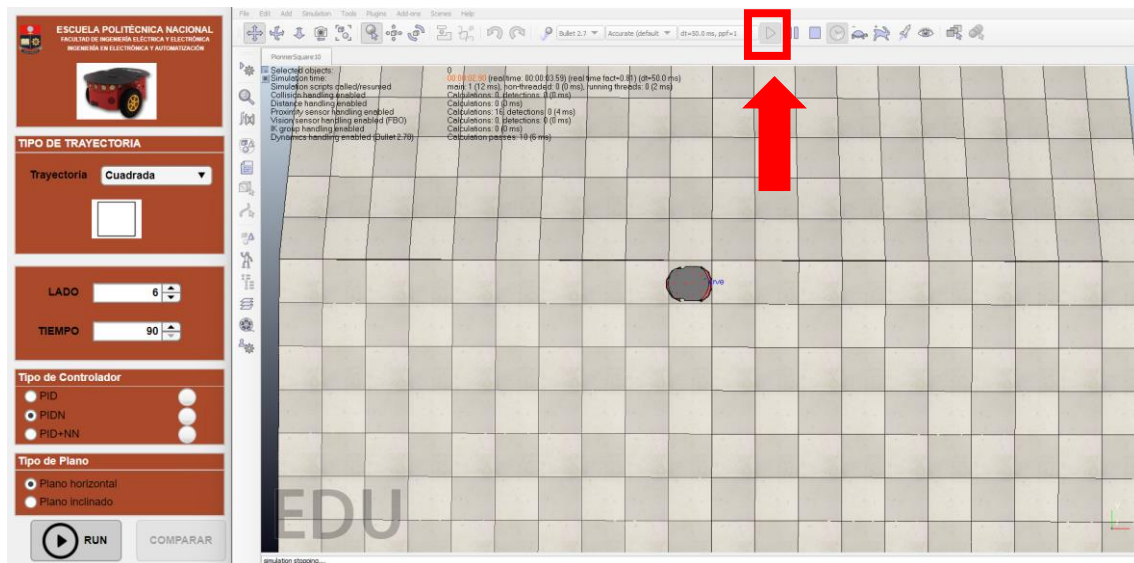


Figura I.7. Simulación V-REP

Luego se prosigue a simular el interfaz dando en el botón “RUN”.

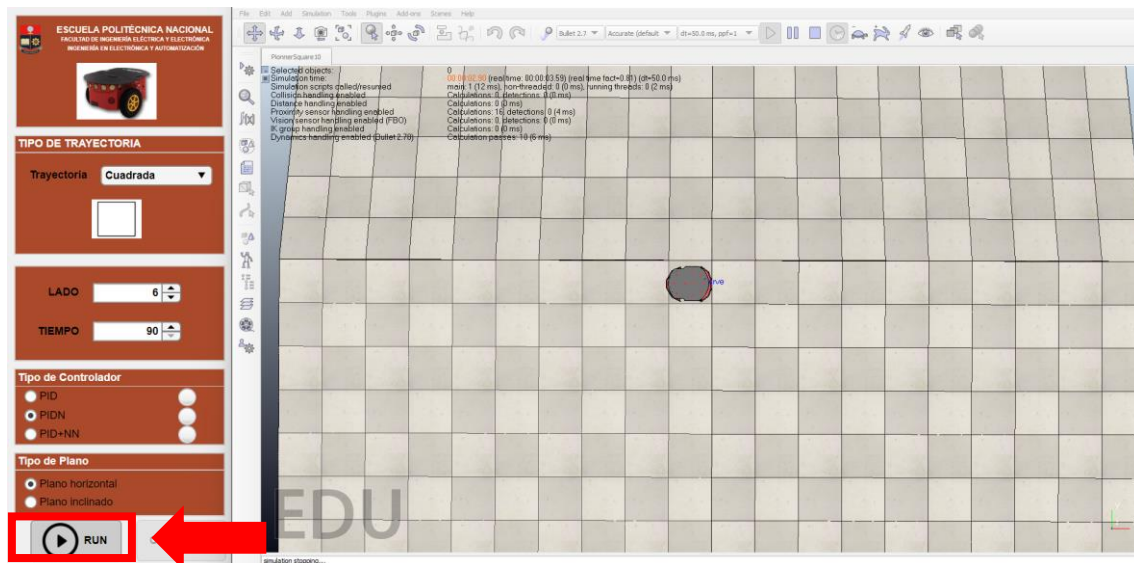


Figura I.8. Simulación Interfaz de Usuario

Luego de generada la trayectoria seleccionada se para la Simulación de V-REP (Ver Figura I.9.), y se observa en la Figura I.10. que las funciones de Tipo de trayectoria, Tipo de plano se deshabilitan, se enciende una luz verde del controlador utilizado, y se grafican los resultados.

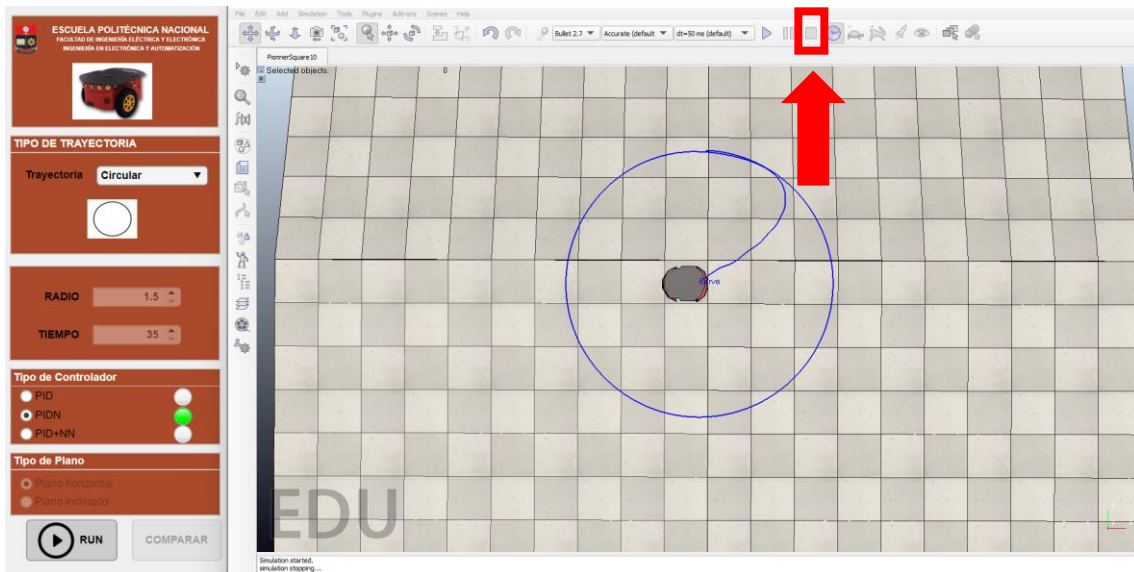


Figura I.9. Simulación detenida V-REP

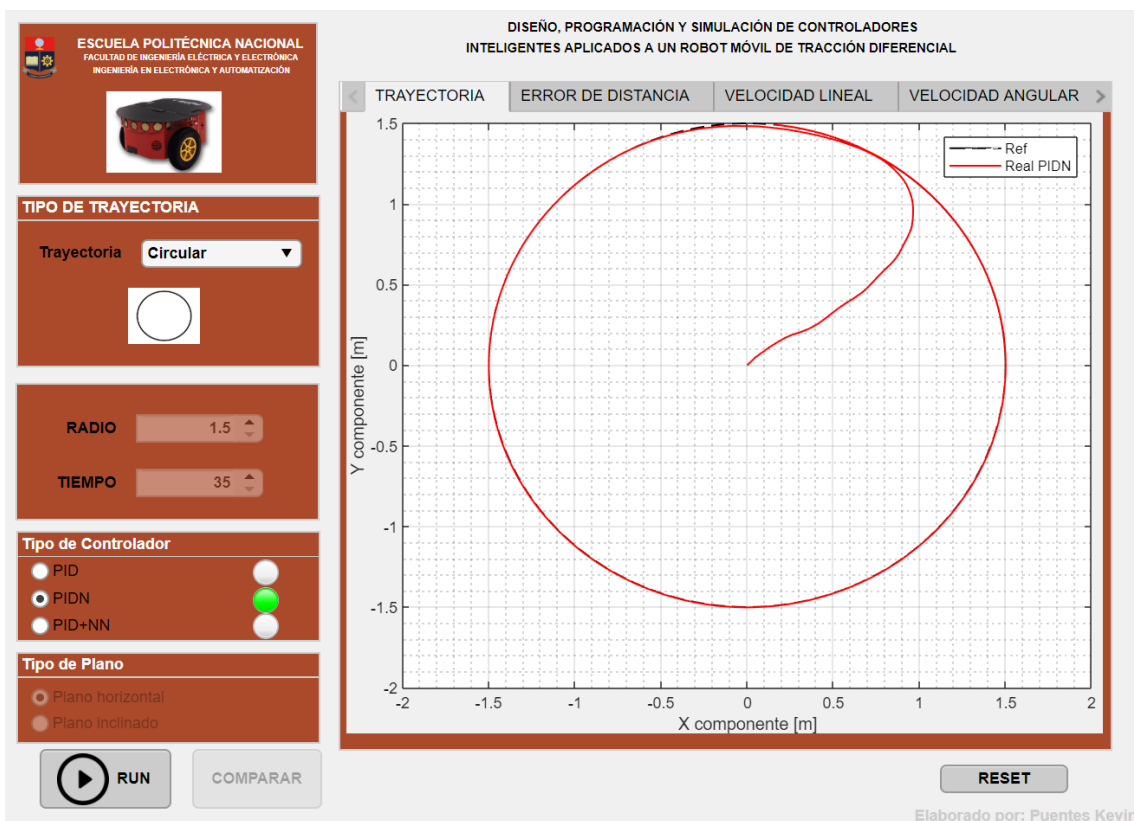


Figura I.10. Gráfica de resultados

Mediante el menú mostrada en la Figura I.11., se puede navegar entre los resultados de la gráfica viendo: trayectoria creada, error de distancia, velocidad lineal, velocidad angular y los índices de desempeño.

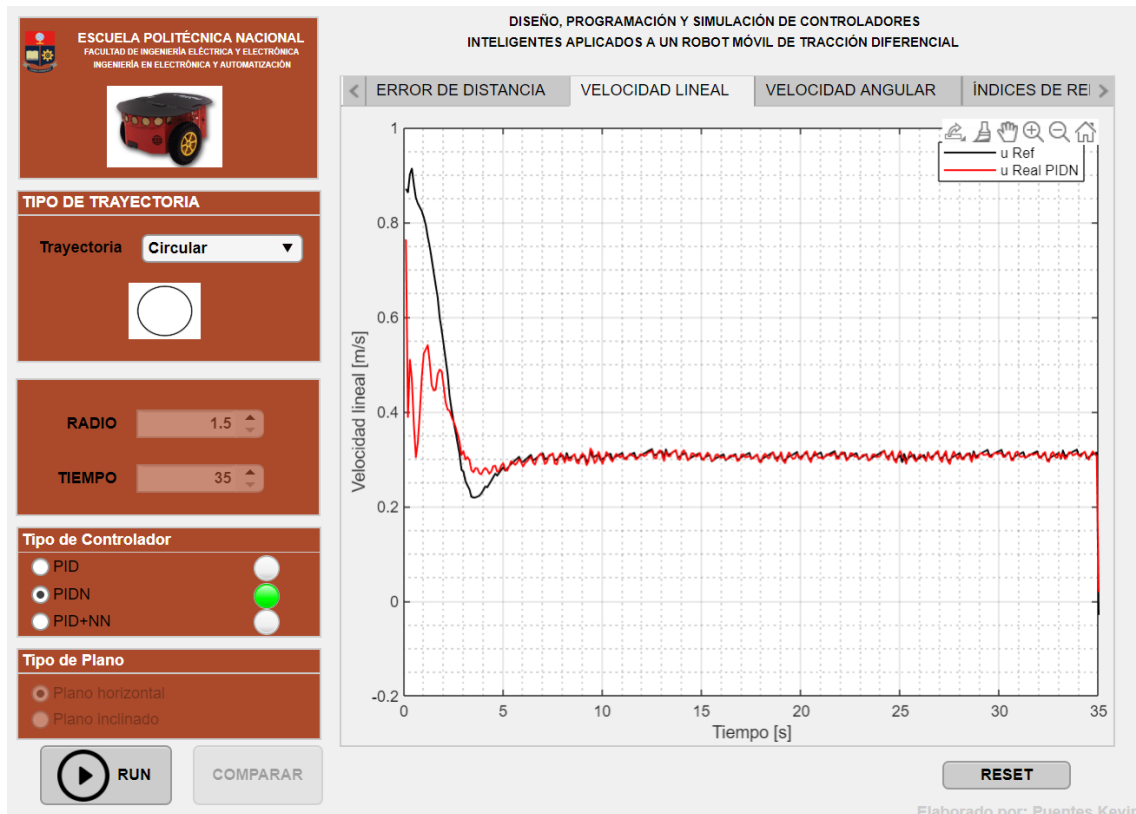


Figura I.11. Navegación de resultados

Ahora solo se puede seleccionar otro tipo de controlador. Se procede a simular V-REP y luego el Interfaz, obteniendo los resultados del nuevo controlador seleccionado, y se realiza el mismo procedimiento para el controlador sobrante, una vez que los tres controladores hayan sido simulados se encienden tres luces y se habilita el botón “Comparar” y un menú en la parte inferior de la interfaz como se observa en la Figura I.12.

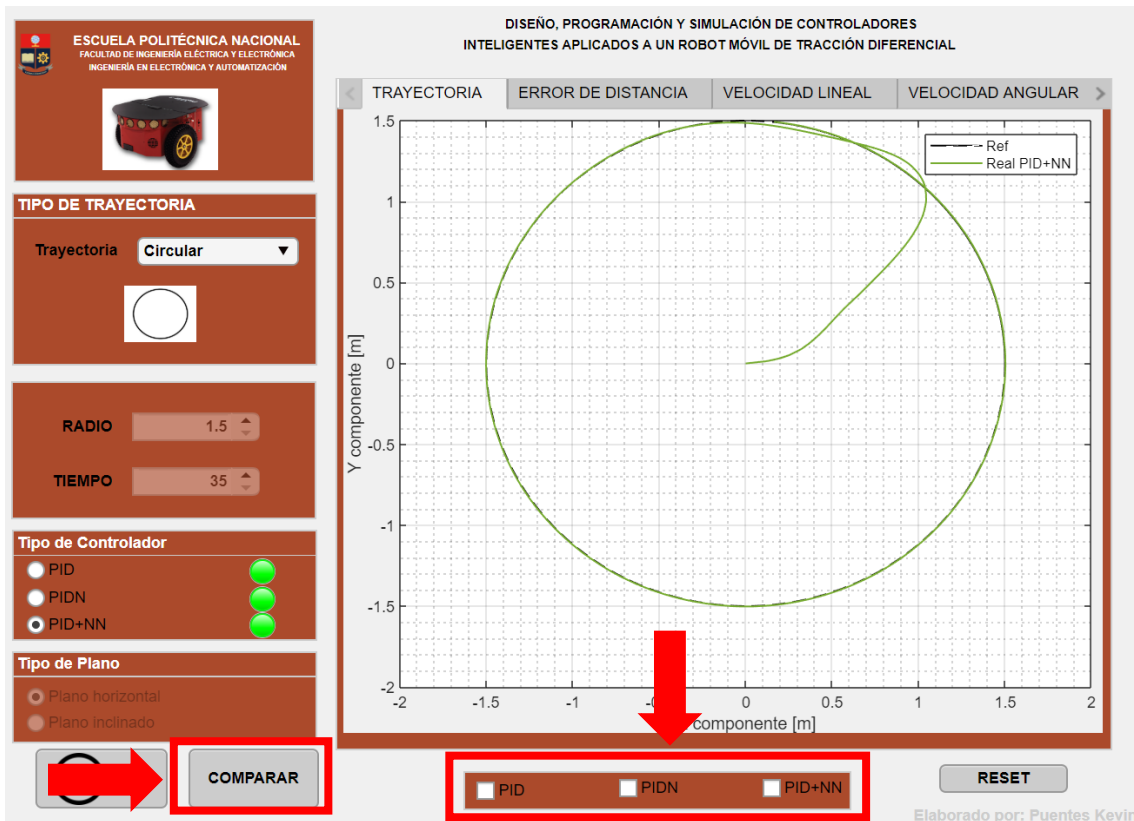


Figura I.12. Habilitación de comparación de controladores

Para analizar los controladores, se selecciona los controladores a comparar y se presiona el botón “Comparar”, graficando los resultados (Ver Figura I.13).

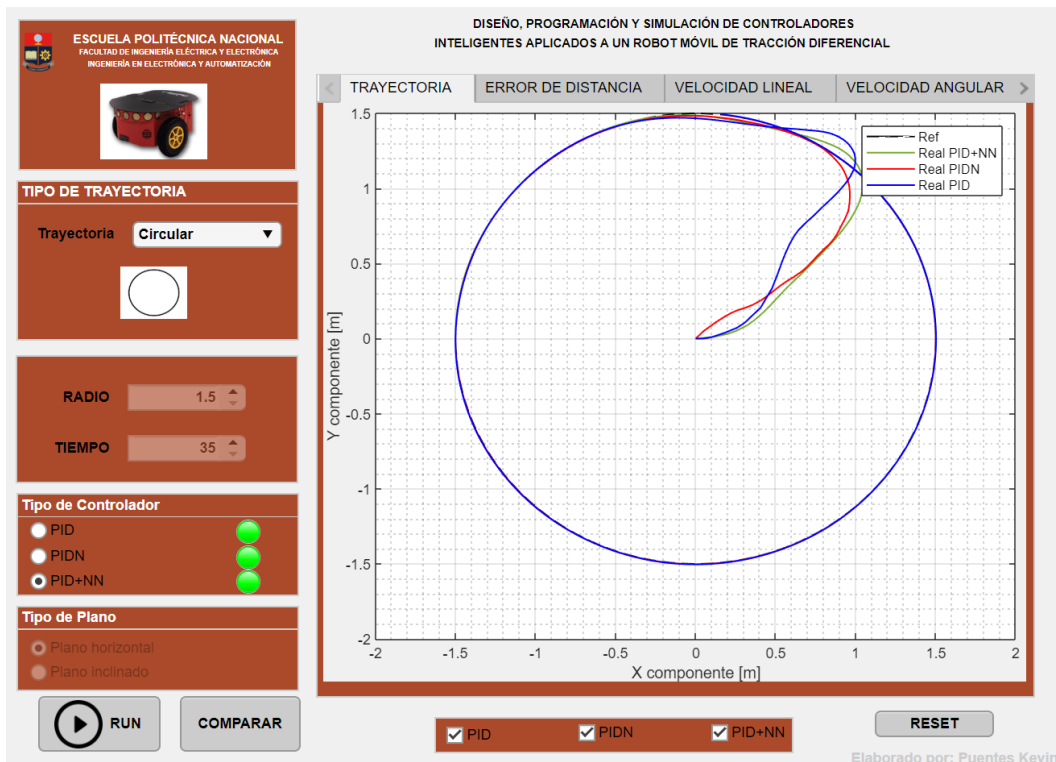


Figura I.13. Comparación de resultados de controladores

Revisados los resultados mediante el menú, se puede dar en el botón “Reset” para seleccionar otra trayectoria (Figura I.14).

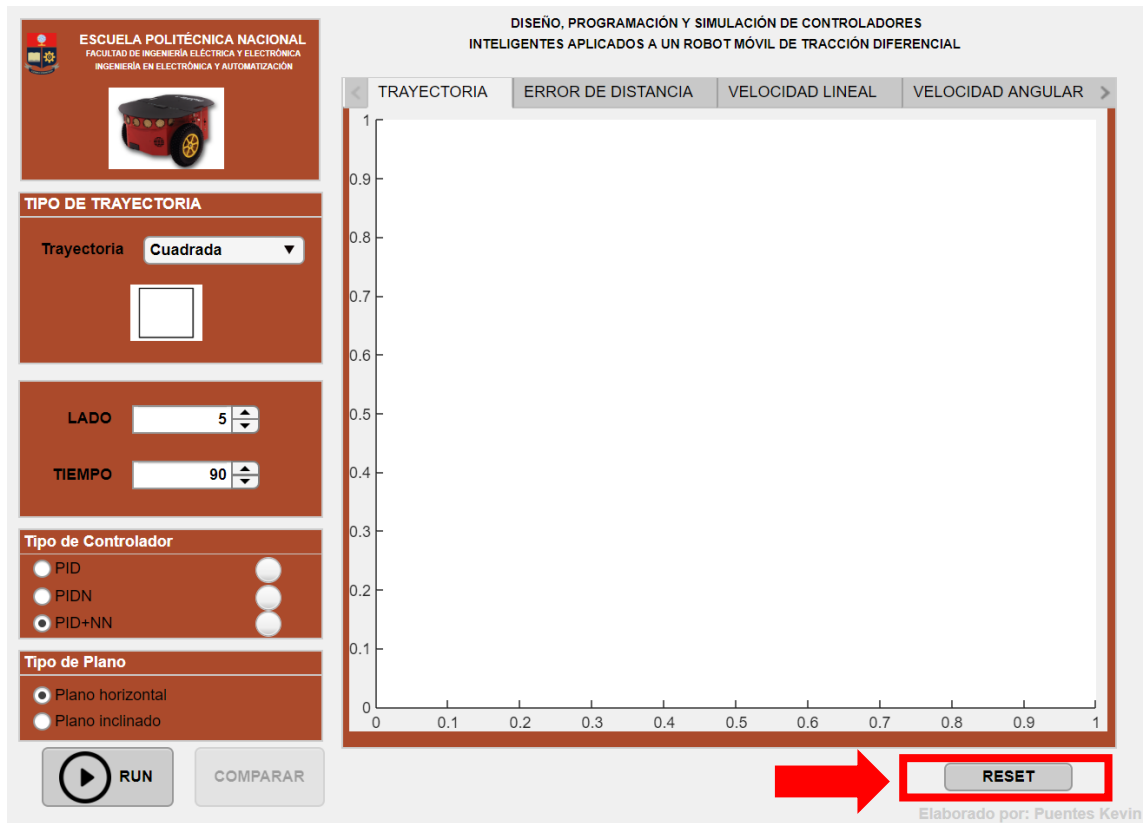


Figura I.14. Generación de nueva trayectoria