

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DISEÑO Y SIMULACIÓN DE DOS CONTROLADORES, PID NO
LINEAL Y PID DUAL NO LINEAL, PARA EL SEGUIMIENTO DE
TRAYECTORIA DEL ROBOT DIFERENCIAL UNICICLO PIONEER
3DX MEDIANTE SIMULINK Y COPPELIA-SIM (TOMO II)**

**DISEÑO Y SIMULACIÓN DEL CONTROLADOR PID DUAL NO
LINEAL PARA EL SEGUIMIENTO DE TRAYECTORIA DEL ROBOT
PIONEER 3DX**

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

HANS JHANDRY LEÓN CHAMBA

hansjhl@outlook.com

DIRECTOR: ING. GEOVANNY DANILO CHÁVEZ GARCÍA, PhD

danilo.chavez@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, HANS JHANDRY LEÓN CHAMBA 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.

HANS JHANDRY LEÓN CHAMBA

Certifico que el presente trabajo de integración curricular fue desarrollado por HANS JHANDRY LEÓN CHAMBA, bajo mi supervisión.

ING. GEOVANNY DANILO CHÁVEZ GARCÍA, PhD.
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) 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.

HANS JHANDRY LEÓN CHAMBA

GEOVANNY DANILO CHÁVEZ GARCÍA

DANIEL ENRIQUE MORALES MARTÍNEZ

DEDICATORIA

Primeramente, dedico este trabajo a mi familia, cuyo apoyo incondicional me ha permitido culminar de manera exitosa mis estudios superiores y ser la persona que hoy soy, principalmente a mi madre, que cada día se esfuerza por brindarnos un mejor porvenir a mi hermano y a mí.

También dedico este trabajo a todas esas personas que me apoyaron durante mi formación universitaria, sin ustedes nada de esto hubiera sido posible.

AGRADECIMIENTO

A mi madre, Fanny, por ser ese pilar fundamental dentro de mi vida y por haber guiado mi camino con su ejemplo de lucha y superación.

A Jaime y Steven, por ser más que mi familia, por estar ahí cuando los necesitaba y por su apoyo ciego e incondicional.

A la gloriosa Escuela Politécnica Nacional y a la Facultad de Ingeniería Eléctrica y Electrónica, principalmente a mis maestros quienes compartieron su conocimiento y experiencias lo que me permitió crecer día a día profesionalmente.

A mi tutor, el Dr. Chávez, cuya guía me permitió culminar de manera exitosa este trabajo, asimismo, al Dr. Camacho por compartir conmigo su conocimiento y por haberme apoyado durante la realización de este trabajo.

A mi amigo, casi hermano, Daniel, por realizar junto a mi persona este trabajo, lo logramos.

A mis amigos: Elizabeth, Steven, Esteban, Danny, Alex, Joselo y Jonathan, por aportar cada uno con su granito de arena en mi vida, gracias a ustedes hoy estoy por culminar mi carrera, por todas esas palabras de aliento y por esos momentos compartidos.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT.....	VIII
1.INTRODUCCIÓN.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	3
1.4.1 Robótica	3
1.4.2 Robótica móvil.....	4
1.4.3 Plataforma Robótica Pioneer 3DX.....	5
1.4.4 Modelo cinemático del robot Pioneer 3DX.....	6
1.4.5 Modelos de primer orden aproximados	8
1.4.6 Índice de desempeño ISE	8
2 METODOLOGÍA.....	9
2.1 Seguimiento de trayectorias con el robot Pioneer 3DX	9
2.2 Generación de trayectorias	10
2.3 Aproximación del sistema a un modelo de primer orden	13
2.4 Controlador cinemático	17
2.5 Introducción al controlador PID no Lineal.....	18
2.6 Diseño del controlador PID Dual no Lineal	19
2.7 Comunicación entre Simulink y Coppelia SIM	21
2.8 Desarrollo de la interfaz gráfica en App Designer	22
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	214
3.1 Resultados de la simulación con la trayectoria cuadrada	24
3.2 Resultados de la simulación con la trayectoria circular	249
3.3 Resultados de la simulación con la trayectoria lemniscara de Bernoulli ..	34
3.4 Comparación entre el PID Dual no lineal y el PID clásico.....	39

3.5	Resultados de la simulación del seguimiento de trayectorias en Coppelia Sim	41
3.6	Conclusiones	43
3.7	Recomendaciones	43
4	REFERENCIAS BIBLIOGRÁFICAS.....	44
5	ANEXOS	46
	ANEXO I	46
	ANEXO II	47

RESUMEN

El seguimiento de trayectorias es un área de la robótica móvil la cual esta en constante crecimiento, los métodos que se usan tanto para la generación de las trayectorias como para el seguimiento de estas son muy variados [1], pero es importante que se continúe realizando investigaciones con diversos tipos de controladores los cuales aporten características de robustez, sencillez y eficacia.

Con respecto a lo anterior, en el presente trabajo se describe el diseño y simulación de un controlador de tipo PID NO LINEAL cuya principal característica es que es de naturaleza DUAL, este controlador se diseña para su aplicación en el seguimiento de trayectorias del robot de tracción diferencial Pioneer 3DX, este tipo de controladores no han sido desarrollados para este tipo de aplicaciones por lo cual se busca analizar las posibles ventajas o desventajas que presente el controlador frente a su contraparte, el controlador PID clásico desarrollado en trabajos anteriores [2].

El ambiente de simulación se desarrolla comunicando síncronamente dos programas, el primero es Simulink de Matlab en el cual se generan las referencias correspondientes a una trayectoria específica además del algoritmo de control, el otro programa es Coppelia Sim en el cual se encuentra el modelo real del robot Pioneer 3DX y este además proporciona mediciones de parámetros necesarios que servirán para la lógica de control en Simulink.

Se cuenta con una interfaz gráfica en la cual se pueden visualizar de manera sencilla cada uno de los resultados ante la simulación de diferentes trayectorias con el controlador propuesto.

PALABRAS CLAVE: Controlador PID Dual no Lineal, Simulink, Coppelia Sim, Robot Pioneer 3DX, seguimiento de trayectorias.

ABSTRACT

Trajectory tracking is an area of mobile robotics which is constantly growing, the methods used for the generation of trajectories and their tracking are very varied [1], but it is important to continue conducting research with various types of controllers which provide characteristics of robustness, simplicity, and efficiency.

Regarding the above, this paper describes the design and simulation of a NON-LINEAR PID type controller whose main characteristic is that it is of a DUAL nature, this controller is designed for its application in tracking the trajectories of the differential traction robot Pioneer 3DX, this type of controllers have not been developed for this type of applications, which is why we seek to analyze the possible advantages or disadvantages that the controller presents compared to its counterpart, the classic PID controller developed in previous works [2].

The simulation environment is developed by synchronously communicating two programs, the first is Matlab's Simulink in which the references corresponding to a specific trajectory are generated in addition to the control algorithm, the other program is Coppelia Sim in which the real model of the Pioneer 3DX and it also provides measurements of necessary parameters that will be used for the control logic in Simulink.

Additionally, there is a graphical interface in which each one of the results can be visualized in a simple way before the simulation of different trajectories with the proposed controller.

KEYWORDS: Dual Non-linear PID Controller, Simulink, Coppelia Sim, Pioneer 3DX Robot, trajectory tracking.

1 INTRODUCCIÓN

En la actualidad el uso de los robots móviles dentro y fuera de la industria ha crecido de manera considerable debido a que brindan grandes ventajas frente a la mano de obra humana, entre estas ventajas están el aumento de la productividad, mayor flexibilidad y mejores condiciones de seguridad [3], sabiendo esto, es necesario que se continúe con las investigaciones que buscan mejorar la metodología de control de este tipo de robots, por lo cual en este trabajo se abarca el diseño y simulación de un controlador PID Dual no Lineal con el objetivo de obtener mejores resultados en cuanto a seguimiento de trayectorias frente a otros tipos de controladores.

En el primer capítulo se abarcan todos los conceptos teóricos necesarios para entender de manera correcta el objetivo general de este trabajo, además de conocer las características fundamentales del robot a usarse como sus límites operativos y su modelo cinemático, también se analizan conceptos como los modelos aproximados de primer orden los cuales son fundamentales para la sintonización del controlador y los índices de rendimiento que nos sirven para realizar las respectivas comparaciones entre los resultados del controlador diseñado y los resultados de controladores clásicos como el PID.

En el segundo capítulo se abarca el diseño completo del controlador descrito anteriormente, para ello se inicia con el planteamiento de seguimiento de trayectorias y la generación de las mismas, luego se analizan características del robot como su modelo de primer orden con retardo para la debida sintonización de las constantes de un PID clásico, que es el punto de partida del controlador PID Dual no Lineal, se analiza también el controlador cinemático que es el que se encarga de pasar las señales provenientes del generador de trayectorias en señales de velocidad lineal y angular, también se explica de manera detallada la comunicación que tienen el software Simulink con el software Coppelia Sim y la utilización de cada uno de los parámetros realimentados desde este último, finalmente se desarrolla la interfaz gráfica que nos permite visualizar de mejor manera cada uno de los resultados obtenidos en las simulaciones.

En el tercer capítulo se muestran los resultados del controlador diseñado en el capítulo dos, dentro de estos resultados se muestran cada una de las trayectorias de referencia como las trayectorias reales que generó el seguimiento del robot, las velocidades lineales y angulares, los errores obtenidos en el seguimiento de la trayectoria, las posiciones en ambos ejes geométricos y los índices de rendimiento obtenidos, también se compara los resultados obtenidos con cada uno de los controladores mediante el análisis del índice de desempeño ISE.

1.1 Objetivo general

Diseñar y simular dos controladores, PID No Lineal y PID Dual No Lineal para el seguimiento de trayectoria del robot diferencial unicycle Pioneer 3DX mediante Simulink y Coppelia-Sim.

1.2 Objetivos específicos

1. Realizar una recopilación bibliográfica de los diferentes esquemas utilizados para el control de trayectoria en robots móviles, los fundamentos de control para PID y para sus variaciones propuestas.
2. Diseñar y sintonizar un esquema de control por PID no lineal y PID dual no lineal para el seguimiento de trayectorias de robots móviles.
3. Simular el seguimiento de trayectorias mediante el uso un robot móvil Pioneer 3DX mediante un esquema Software in Loop, utilizando los softwares CoppeliaSim y Simulink
4. Probar y comparar los esquemas de control propuestos frente a un controlador PID clásico, realizando una cuantificación por índices de desempeño, y estudiando el estado transitorio de las respuestas.
5. Desarrollar una interfaz gráfica en App Designer de Matlab para poder seleccionar el controlador, la trayectoria y visualizar los datos en pantalla.

1.3 Alcance

1. Se investigará sobre modelos cinemáticos del Pioneer 3DX aplicables al seguimiento de trayectoria, con el objetivo de obtener un modelo de primer orden con retardo.
2. Se estudiará la base teórica sobre los controladores PID no lineal y PID dual no lineal, para su aplicación en el control de trayectoria de un robot móvil.
3. Se diseñará un esquema de control por PID no lineal y otro por PID dual no lineal a partir del modelo de primer orden con retardo para su aplicación en el modelo del robot.
4. Se verificará el funcionamiento del sistema propuesto mediante simulación con diferentes trayectorias (cuadrada, circular, lemniscata) con los esquemas de control propuestos. Para lo cual se ocupará Simulink y Coppelia-Sim.

5. Se realizarán comparaciones entre los controladores propuestos mediante su cuantificación de índices de desempeño, y la respuesta en estado transitorio de cada controlador, observando máximo sobrepico y tiempo de establecimiento.
6. Se desarrollará una interfaz gráfica con el software App Designer para seleccionar el controlador, la trayectoria y observar resultados en pantalla, junto con los índices de rendimiento.

1.4 Marco teórico

Este proyecto se centra fundamentalmente en el diseño y simulación de un controlador PID Dual no Lineal para el seguimiento de trayectorias usando el robot Pioneer 3DX, para ello es necesario tener claros ciertos conceptos básicos, a continuación, se describen los distintos componentes que nos ayudarán a comprender de mejor manera el trabajo realizado.

1.4.1 Robótica

La robótica esta definida como una ciencia en la cual intervienen algunas ramas de la tecnología actual como la ingeniería y la informática, cuyo principal objetivo es el diseño de máquinas con características específicas que tengan la capacidad de realizar tareas de automatización o sustituir la mano de obra humana, esta capacidad se vera limitada por lo componentes del robot. [4]

Actualmente es común el uso de robots dentro del mundo industrial, podemos decir que los robots se han vuelto fundamentales debido a que están presentes en la automatización de procesos y en la ejecución de tareas que son peligrosas, agotadoras o de alta dificultad para el ser humano.

Los robots presentan algunas ventajas considerables frente al ser humano, las más notables son descritas a continuación:

- Los robots no tienen un limite de energía por lo cual no se cansan y les permite trabajar prácticamente todos los días del año a toda hora.
- La precisión de un robot es mucho mayor a la del ser humano.
- Los robots permiten que se aumente la productividad y que se reduzcan los costes ya que presentan un mejor uso de los recursos y realizan las tareas en tiempos menores.

- Fácil manejo y supervisión por parte de los operadores. [5]

Existe una gran variedad de aplicaciones en las que los robots pueden ser utilizados como podemos corroborar en [6], esto se da debido a que actualmente hay muchos tipos de robots, sin embargo, para el desarrollo de este trabajo nos enfocaremos en un tipo de robot específico el cual es el robot móvil.

1.4.2 Robótica móvil

Un robot móvil es un sistema electromecánico cuya característica principal es su capacidad de movilizarse de un punto a otro de manera autónoma [7], cabe recalcar que el robot necesita de un medio físico para realizar dicho movimiento y que este desplazamiento se da gracias a que cuenta con dispositivos de locomoción.

Generalmente los robots móviles vienen equipados con sensores que les permiten monitorear y realimentar los datos de posición, velocidad, etc. según se requiera y así lograr completar la tarea asignada.

La clasificación de los robots móviles se da de acuerdo con el sistema de locomoción con el que cuentan, existen tres formas de que el robot se movilice: por ruedas (Figura 1.1a), por orugas (Figura 1.1b) y por patas (Figura 1.1c).

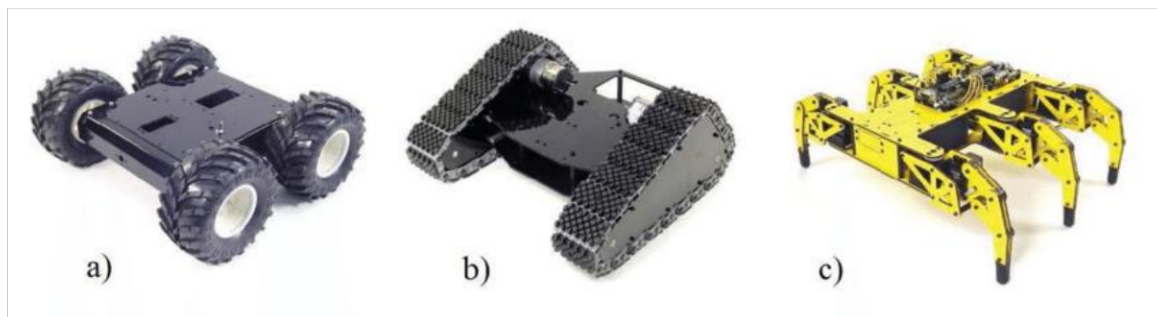


Figura 1.1 Clasificación de los robots móviles [8]

Aunque se ha estudiado a fondo todos estos sistemas de locomoción debemos recalcar que el que ha logrado mejores resultados dentro de la industria es el robot móvil con ruedas debido a que presenta algunas ventajas frente a los otros sistemas de locomoción como, por ejemplo, su sencillez, mayor eficiencia energética, menor desgaste de la superficie, fácil construcción, etc.

Dentro de la temática de robots móviles con ruedas podemos encontrar diversas configuraciones cinemáticas las cuales dependen de la aplicación en la que va a ser usado

el robot, sin embargo, nos concentraremos únicamente en la Tracción diferencial debido a que el robot utilizado en este proyecto de titulación tiene esta configuración.

Tracción diferencial es un tipo de direccionamiento que se caracteriza debido a que cada una de las ruedas del robot es fija y tiene una velocidad diferente de la otra [9], es decir, cada rueda es propulsada y controlada de manera individual con lo cual el direccionamiento dependerá de la velocidad que tenga cada una de las ruedas, adicionalmente se debe contar con una o más ruedas locas las cuales únicamente sirven de soporte, una mejor apreciación de este tipo de tracción la podemos observar en la Figura 1.2.

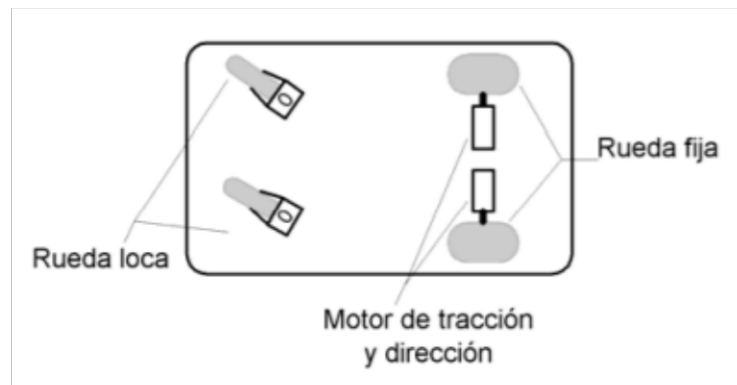


Figura 1.2 Direccionamiento por tracción diferencial [3]

1.4.3 Plataforma Robótica Pioneer 3DX

Una vez nos hemos adentrado en los conceptos de robótica, específicamente en robots móviles, vamos a describir el robot que se usa en este trabajo, se trata del robot Pioneer 3DX, este pequeño robot de tracción diferencial es el más indicado para trabajos de investigación y en el ámbito educativo debido a su diseño y resistencia, además de que puede ser actualizado y personalizado a conveniencia, cuenta con dos ruedas cada una con un motor individual y con los sensores necesarios para obtener los parámetros requeridos para el control del mismo [10]. En la Figura 1.3 podemos apreciar las dimensiones de esta plataforma robótica y en la tabla 1.1 los límites de operación de este robot.

Tabla 1.1. Límites de operación del Robot Pioneer 3DX [10]

Parámetro	Valor máximo
Velocidad lineal (avance o retroceso)	1,2 m/s
Velocidad de rotación	300°/s

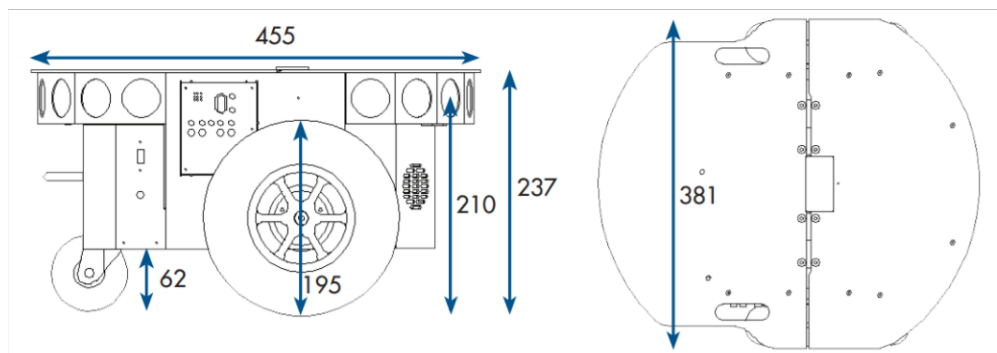


Figura 1.3 Dimensiones en mm del Pioneer 3DX [10]

1.4.4 Modelo cinemático del robot Pioneer 3DX

Dentro del concepto del modelo cinemático del robot se aborda toda la matemática que representa el movimiento del robot, pero sin considerar aspectos externos como alguna fuerza que pueda estar siendo ejercida sobre el robot.

Antes de continuar con este modelo debemos tener en cuenta que este robot uniciclo tiene restricción no holonómica [2], es decir que, no es posible que el robot se mueva de manera lateral. Esto lo podemos apreciar de mejor manera en la Figura 1.4.

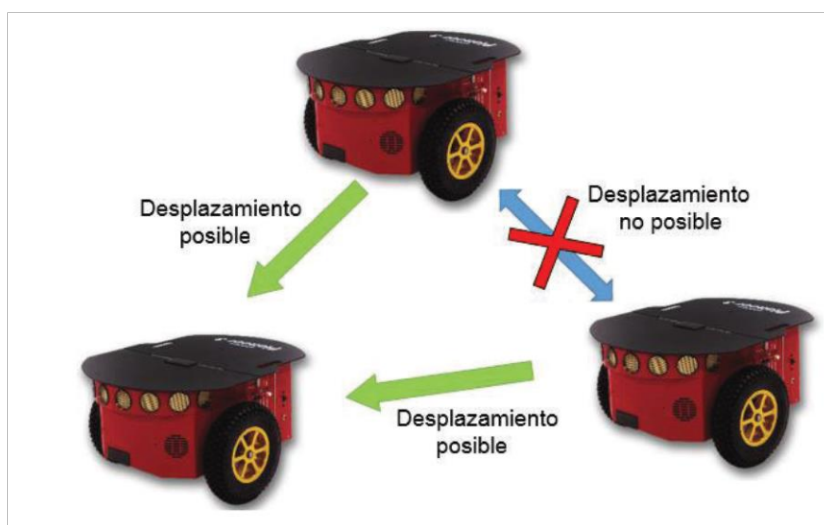


Figura 1.4 Desplazamientos posibles de un robot con restricción no holonómica [2]

La causa de que el robot tenga restricción no holonómica se da debido a que el punto de interés se encuentra entre las dos ruedas del robot [11], este punto P lo podemos observar en la Figura 1.5, con el objetivo de mejorar esta característica del robot en este proyecto se trabajará con el punto G el cual está a una distancia a del punto P , esta distancia corresponde a 16.47cm, a este nuevo modelo se lo conoce como modelo cinemático con restricción no holonómica mejorada el cual será analizado más adelante.

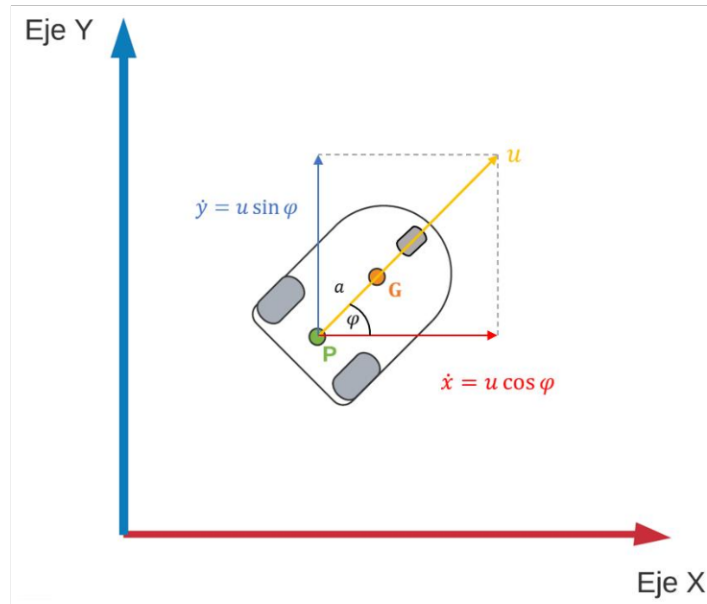


Figura 1.5 Análisis geométrico del movimiento del robot Pioneer 3DX

Para obtener el modelo cinemático del robot debemos analizar cuidadosamente la Figura 1.5, en este caso el robot se desplaza en línea recta a una velocidad lineal constante u , también se aprecia que la línea por la cual se desplaza el robot está desfasada un ángulo φ con respecto al eje X, podemos entonces descomponer la velocidad lineal con respecto al punto P y obtener las ecuaciones que componen el modelo cinemático con restricción no holonómica.

$$\dot{x} = u \cos \varphi \quad (1.1)$$

$$\dot{y} = u \sin \varphi \quad (1.2)$$

$$\dot{\varphi} = \omega \quad (1.3)$$

Como se explicó anteriormente este modelo puede ser mejorado moviendo el punto de interés una distancia a del punto P , con este cambio se obtiene el modelo cinemático con restricción no holonómica mejorada cuyas ecuaciones son las siguientes:

$$\dot{x} = u \cos \varphi - a\omega \sin \varphi \quad (1.4)$$

$$\dot{y} = u \sin \varphi + a\omega \cos \varphi \quad (1.5)$$

$$\dot{\varphi} = \omega \quad (1.6)$$

Colocando estas ecuaciones en variables de estados queda de la siguiente manera:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi & -a \sin \varphi \\ \sin \varphi & a \cos \varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (1.7)$$

1.4.5 Modelos de primer orden aproximados

Debido a que el modelo cinemático del robot (1.7) tiene características no lineales se complica de sobremanera la sintonización de los diferentes controladores ya que requieren de una matemática muy avanzada, en estos casos lo que se realiza es una aproximación del modelo real de la planta, en este caso el robot, a un modelo de orden menor para simplificarlo y obtener una respuesta de tipo lineal la cual debe ser similar a la respuesta real [12].

En este trabajo se aproximará la planta a un modelo de primer orden con retardo también conocido como FOPDT (First Order Plus Death Time) cuya ecuación representativa puede ser observada en (1.8), dentro de los métodos que están basados en la curva de reacción del proceso se encuentra el método de dos puntos en el que solamente se tienen en cuenta dos puntos del proceso con los cuales se puede encontrar los parámetros de la ecuación de primer orden, estamos hablando de la constante de tiempo, la ganancia del sistema y el respectivo retardo [13].

$$G(s) = \frac{k_c e^{-t_m s}}{\tau s + 1} \quad (1.8)$$

Donde:

k_c : ganancia del sistema en estado estable

t_m : retardo o tiempo muerto del sistema

τ : constante de tiempo del sistema

En el capítulo dos se analiza a profundidad el método de dos puntos con el que se calcularán los parámetros para aproximar tanto la velocidad lineal como la velocidad angular a un FOPDT.

1.4.6 Índice de desempeño ISE

El índice ISE o también conocido como índice Integral del error cuadrático no es más que un valor que se calcula a partir del error presente en el proceso cuando se cierra el lazo de control, en otras palabras, el error se encuentra restando la referencia de la variable a controlar y el valor real de dicha variable, al valor absoluto del error se lo eleva al cuadrado y se obtiene el índice ISE [14], la ecuación que representa a este índice la podemos observar en (1.9).

$$ISE = \int_0^T e^2(t) dt \quad (1.9)$$

2 METODOLOGÍA

En el presente capítulo se aborda todas las temáticas relacionadas con el control que requiere el robot Pioneer 3DX para el seguimiento de trayectorias, empezamos definiendo que es en si el seguimiento de trayectorias y el esquema de control necesario para controlar a nuestro robot, luego se detallará la generación de las diferentes trayectorias propuestas en los objetivos del trabajo de titulación así como también el diseño completo del controlador propuesto (PID Dual no Lineal) y otros controladores necesarios para el seguimiento de trayectorias del robot Pioneer 3DX, también se analizará la conexión de los programas que permiten realizar las respectivas simulaciones, en este caso Simulink y Coppelia Sim, por último se revisará la aplicación en donde se desarrolló la interfaz grafica la cual permite que se aprecien de manera más sencilla cada uno de los resultados del seguimiento de trayectoria obtenidos en las simulaciones del robot utilizando todo el esquema de control diseñado.

2.1 Seguimiento de trayectorias con el robot Pioneer 3DX

Para adentrarnos al seguimiento de trayectorias primero se debe tener en claro que es una trayectoria, según [12] una trayectoria es una sucesión de posibles posiciones del robot a medida que se desplaza en un plano cartesiano o de referencia que nos muestra el estado inicial, los estados intermedios y el estado final del robot en un intervalo de tiempo definido, en palabras más simples es el recorrido por el cual se desplaza el robot en el espacio.

El seguimiento de trayectorias por parte de un robot es fundamental en el mundo de la robótica móvil ya que así no existe la necesidad de la intervención humana cuando el robot esta ejecutando una tarea, esta característica de autonomía se da siempre y cuando se conozca el entorno en el que trabajará el robot, ya que si existen obstáculos en su trayectoria el robot debe contar con algún algoritmo de control más complejo.

Para el seguimiento de trayectorias en este trabajo se escogió al robot Pioneer 3DX por lo que se hace necesario conocer la representación que tiene este robot en cuanto a entradas y salidas para la respectiva planificación de referencias, basándonos en su modelo cinemático revisado en el capítulo anterior, en especial en su representación en variables de estado (1.7), podemos representar a nuestro robot como una caja negra con ciertas entradas y salidas, tal como se observa en la Figura 2.1, las entradas corresponden a la velocidad lineal y angular de referencia mientras que las salidas son la velocidad lineal actual, la velocidad angular actual, las posiciones en el plano XY y el ángulo de orientación actuales.

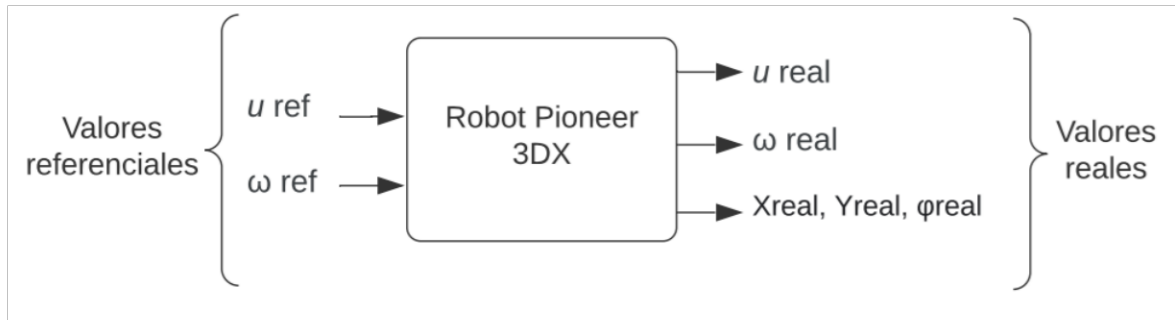


Figura 2.1 Esquema del robot Pioneer 3DX simplificado

Si tenemos en cuenta las variables de entrada y salida del robot mostrados en la figura anterior podemos obtener el esquema de control completo necesario para el seguimiento una trayectoria específica, este esquema lo podemos observar en la Figura 2.2, inicialmente se debe contar con un generador de trayectorias el cual genera posiciones en el eje cartesiano y las envía al controlador cinemático el cual se encarga de generar el conocido “robot virtual” que será la referencia que debe seguir nuestro robot real, esta referencia es de velocidad lineal y angular, el controlador que se propone diseñar más adelante se encarga de enviar las acciones de control al robot real con el objetivo de que este se desplace siguiendo las referencias de velocidad lineal y angular, mediante los sensores presentes en el robot real este realimenta los parámetros necesarios tanto al controlador cinemático como al controlador principal.

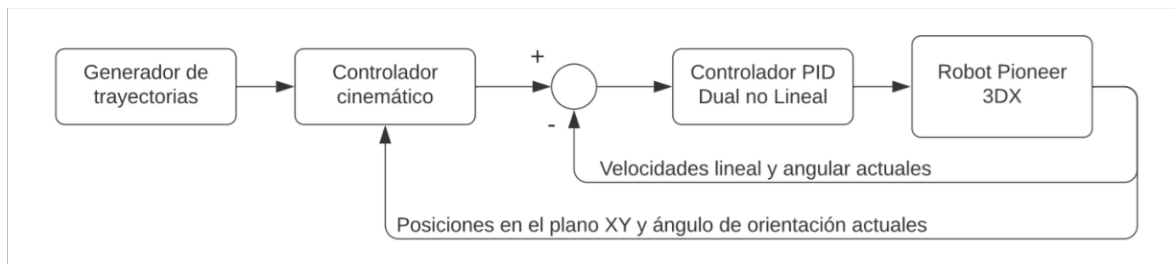


Figura 2.2 Esquema de control del robot Pioneer 3DX para el seguimiento de trayectorias

2.2 Generación de trayectorias

Como se analizó en el apartado anterior el primer punto es la generación de las distintas trayectorias que el robot va a tener que seguir, para este trabajo se generarán tres trayectorias, la cuadrada, la circular y la Lemniscata de Bernoulli, se debe tener en cuenta que se trabajará en tiempo continuo y con una velocidad lineal de referencia (u_{ref}) que en este caso será de 0.3m/s.

Para la trayectoria cuadrada las ecuaciones son generadas en cada uno de los intervalos de tiempo correspondientes a los lados del cuadrado, las ecuaciones resultantes son las siguientes:

Intervalo 1 $0 < t < t_1$

$$x_{ref} = u_{ref} t \quad (2.1)$$

$$y_{ref} = 0 \quad (2.2)$$

Intervalo 2 $t_1 < t < t_2$

$$x_{ref} = l \quad (2.3)$$

$$y_{ref} = -l + u_{ref} t \quad (2.4)$$

Intervalo 3 $t_2 < t < t_3$

$$x_{ref} = 3l - u_{ref} t \quad (2.5)$$

$$y_{ref} = l \quad (2.6)$$

Intervalo 4 $t_3 < t < t_4$

$$x_{ref} = 0 \quad (2.7)$$

$$y_{ref} = 4l - u_{ref} t \quad (2.8)$$

Donde:

u_{ref} : velocidad lineal de referencia

l : lado del cuadrado

En la Figura 2.3 podemos observar la gráfica de la trayectoria generada con las ecuaciones antes descritas, el valor de l en este caso es de 3 metros, se debe tener en cuenta que para que la generación de esta trayectoria sea correcta los intervalos de tiempos deben ser los mismos para cada uno de los lados del cuadrado, es decir que:

$$t_1 = t_2 = t_3 = t_4$$

En este caso el valor de estos intervalos de tiempo es de 20 segundos.

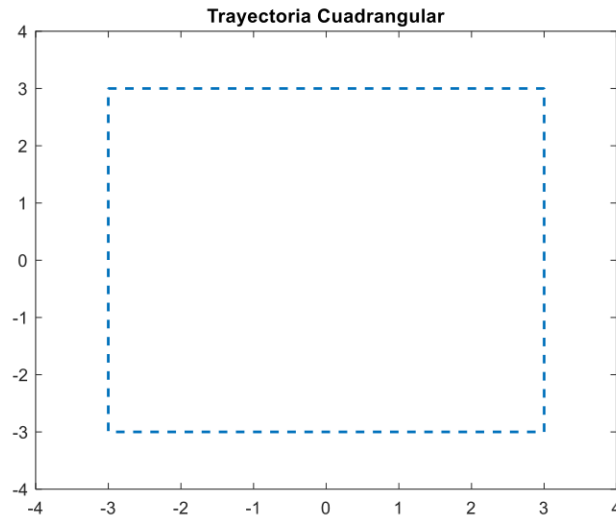


Figura 2.3 Generación de la trayectoria cuadrada

Por otro lado, para la generación de la trayectoria circular es necesario especificar el radio (R) del círculo que se quiere generar, además se debe tomar en cuenta que la velocidad angular con la que se genera la trayectoria corresponde a la siguiente ecuación:

$$\omega = \frac{u_{ref}}{R} \quad (2.9)$$

Teniendo esto en cuenta las ecuaciones para la generación de la trayectoria circular son las siguientes [16]:

$$x_{ref} = R \cos \omega t \quad (2.10)$$

$$y_{ref} = R \sin \omega t \quad (2.11)$$

En la Figura 2.4 se puede apreciar la trayectoria generada con ecuaciones (2.10) y (2.11).

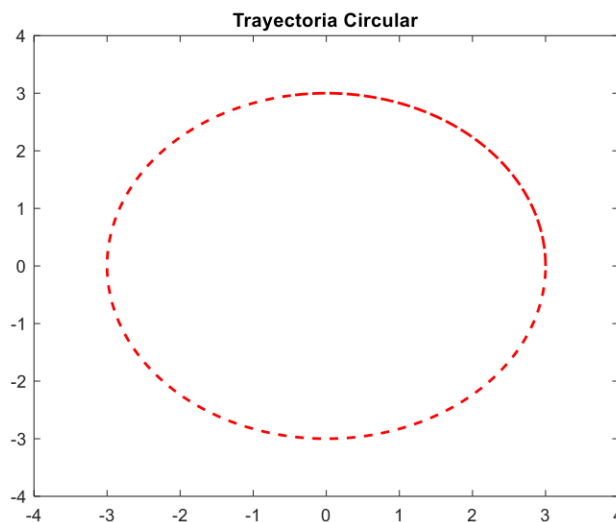


Figura 2.4 Generación de la trayectoria circular

Como se esperaba la gráfica es un perfecto círculo cuyo radio es de 3m, dependiendo del tiempo de simulación se llegará a completar o no el círculo, en este caso con 80 segundos es más que suficiente para la generación del círculo completo.

Por último, tenemos la trayectoria Lemniscata de Bernoulli, para esta trayectoria también debemos conocer el radio (R) de las cuasi circunferencias que se crean, la trayectoria se genera con las siguientes ecuaciones [16]:

$$x_{ref} = \frac{R \cos \omega t}{1 + \sin^2(\omega t)} - \frac{R}{2} \quad (2.12)$$

$$y_{ref} = \frac{R \sin(\omega t) \cos(\omega t)}{1 + \sin^2(\omega t)} \quad (2.13)$$

En la Figura 2.5 podemos apreciar la trayectoria que se genera utilizando las ecuaciones (2.12) y (2.13), en este caso el valor de R es de 3 metros, al igual que en la trayectoria circular el tiempo de simulación tiene que ser el suficiente para que se genere toda la trayectoria, por lo que se simuló un total de 80 segundos para que la gráfica este totalmente completa.

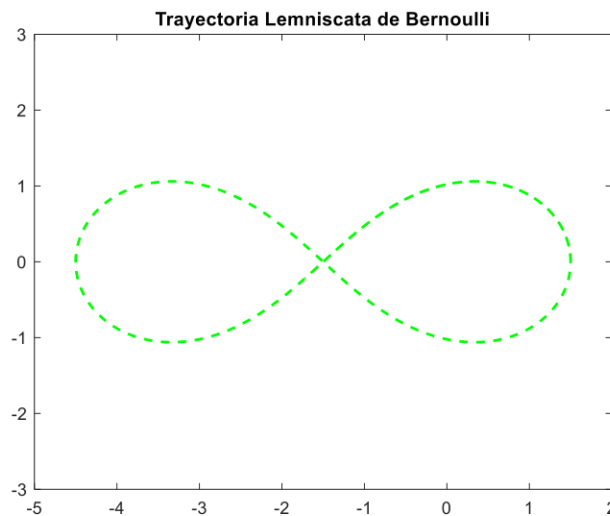


Figura 2.5 Generación de la trayectoria lemniscata de Bernoulli

2.3 Aproximación del sistema a un modelo de primer orden

Como se mencionó en el apartado 1.4.5 de este trabajo, es necesario que se realice la aproximación del modelo real del robot a uno de orden reducido (FOPDT) con el objetivo de facilitar la sintonización de los controladores, sabemos que las entradas de nuestro robot son la velocidad lineal y la velocidad angular generadas por el controlador, en otras palabras, son las velocidades de referencia, mientras que las salidas del sistema vendrían a ser las velocidades que los sensores del robot miden.

Para la aproximación del sistema a uno de primer orden con retardo simplemente se debe analizar el comportamiento de la salida del sistema ante una entrada de tipo paso, en este caso se deben analizar por separado tanto la velocidad lineal como la velocidad angular debido a que son independientes una de la otra.

Una consideración que se debe tener en cuenta es que el robot se encuentra en reposo en el instante de tiempo cero por lo que es necesario no empezar con velocidades nulas, esto para obtener una respuesta que no se vea afectada por la fricción de las ruedas del robot con la superficie o las posibles vibraciones que pueden introducir por la rueda de equilibrio.

En la Figura 2.6 se puede observar la respuesta de la velocidad lineal ante una entrada paso, esta respuesta será analizada para obtener los valores de tiempo que nos permitirán aproximar la respuesta de velocidad lineal a una de primer orden utilizando el método de dos puntos.

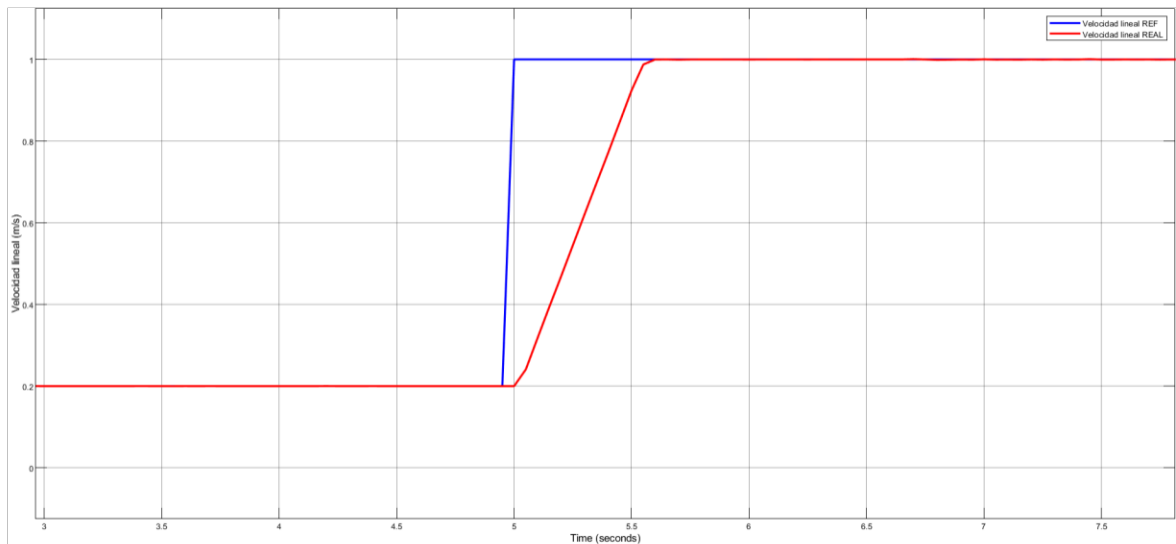


Figura 2.6 Respuesta de la velocidad angular ante una entrada paso

El método de dos puntos consiste en la selección de dos instantes de tiempo en los cuales la respuesta alcance puntos específicos en la curva de reacción del proceso [13], esos dos valores de tiempo serán utilizados para calcular los parámetros de un modelo de primer orden con retardo como el de la ecuación (1.8), las ecuaciones con las que se calcula dichos parámetros se muestran a continuación:

$$\tau = a t_1 + b t_2 \quad (2.14)$$

$$t_m = c t_1 + d t_2 \quad (2.15)$$

Donde:

a, b, c, d : constantes que dependen del método de identificación del sistema de primer orden más retardo

τ : constante de tiempo del sistema

t_m : retardo o tiempo muerto del sistema

En este caso vamos a utilizar el método de Alfaro cuyos valores de las constantes son las siguientes:

$$a = -0.910$$

$$b = 0.910$$

$$c = 1.262$$

$$d = -0.262$$

El método de Alfaro dicta que se deben medir los instantes de tiempo en los cuales la respuesta haya alcanzado el 25 y el 75% de su valor máximo, con ayuda de los cursores de Simulink obtenemos estos valores de tiempo:

$$t_1 = 0.1540$$

$$t_2 = 0.4191$$

Reemplazando en la ecuación (2.14) y (2.15) se obtienen los valores de la constante de tiempo y del retardo del sistema respectivamente, estos valores son:

$$\tau = 0.2412$$

$$t_m = 0.0845$$

Por otro lado, para obtener la ganancia en estado estable del modelo de primer orden con retardo es necesario aplicar la siguiente ecuación:

$$k_c = \frac{\Delta Y}{\Delta U} \quad (2.16)$$

Donde:

ΔY : máxima variación de la respuesta

ΔU : máxima variación de la entrada

Como se observa en la Figura 2.6 la variación de la entrada es la misma que la variación de la respuesta, por ende, el valor de la ganancia será igual a 1.

Reemplazando los valores calculados en la ecuación (1.8) obtenemos el modelo de primer orden más retardo de la velocidad lineal:

$$G(s) = \frac{e^{-0.0845s}}{0.2412s+1} \quad (2.17)$$

Se procede a la validación del modelo encontrado simulando su respuesta ante la misma entrada paso, en la Figura 2.7 podemos observar tanto la respuesta del modelo real como la respuesta del modelo aproximado, como se aprecia el modelo aproximado es correcto debido a que su respuesta se asemeja a la respuesta del modelo real.

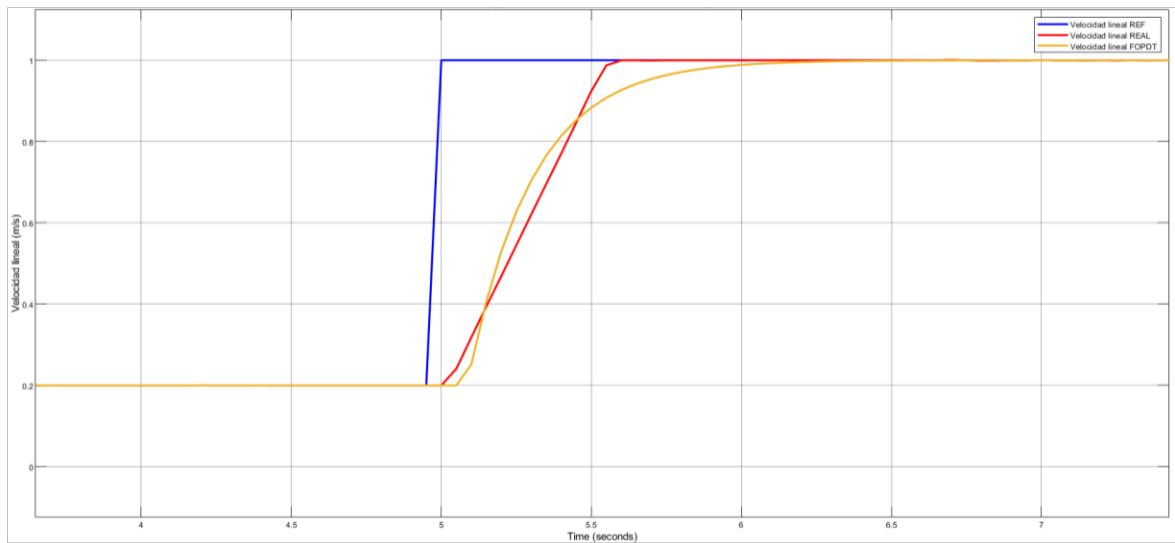


Figura 2.7 Respuesta del modelo aproximado y del modelo real de la velocidad lineal ante una entrada paso

Se realiza un procedimiento similar para la obtención del modelo de primer orden más retardo de la velocidad angular, los parámetros obtenidos los podemos apreciar a continuación:

$$k_p = 1.08$$

$$t_1 = 0.0828$$

$$t_2 = 0.2162$$

Reemplazando en la ecuación (2.14) y (2.15) se obtienen los valores de la constante de tiempo y del retardo del sistema respectivamente, estos valores son:

$$\tau = 0.1214$$

$$t_m = 0.0478$$

Reemplazando los valores calculados en la ecuación (1.8) obtenemos el modelo de primer orden más retardo de la velocidad angular:

$$G(s) = \frac{e^{-0.0478s}}{0.1214s+1} \quad (2.18)$$

Se procede a realizar la validación del modelo de primer orden encontrado, el resultado se puede apreciar en la Figura 2.8, analizando esta grafica podemos concluir que el modelo aproximado es correcto debido a que su respuesta se asemeja a la respuesta del modelo real.

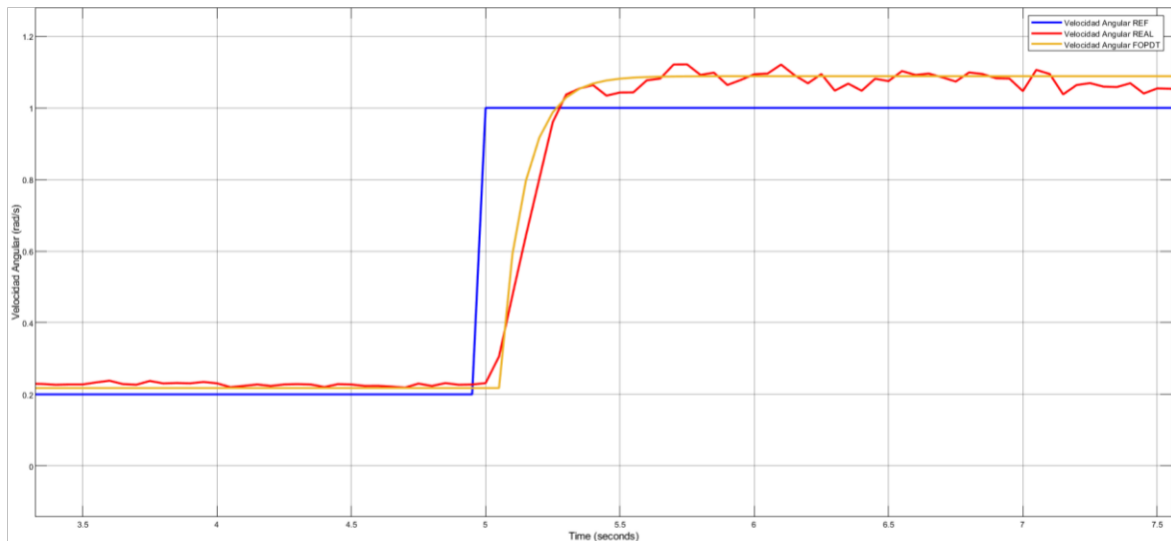


Figura 2.8 Respuesta del modelo aproximado y del modelo real de la velocidad angular ante una entrada paso

2.4 Controlador cinemático

El controlador cinemático es aquel cuya función principal es el cálculo de las referencias de velocidad lineal y angular que serán enviadas al controlador principal partiendo de los datos de posición proporcionados por el generador de trayectorias, las posiciones actuales y el ángulo de orientación actual del robot.

Este controlador se basa en el modelo cinemático del robot Pioneer 3DX revisado en el apartado 1.4.4 del primer capítulo, específicamente en la ecuación (1.7).

Como necesitamos que las entradas sean las posiciones en el plano cartesiano y las salidas la velocidad lineal y angular únicamente vamos a tener en cuenta solamente las dos primeras variables de estado del modelo cinemático obteniendo el siguiente modelo:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \varphi & -a \sin \varphi \\ \sin \varphi & a \cos \varphi \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (2.19)$$

Procedemos a calcular el modelo de cinemática inversa:

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\frac{1}{a} \sin \varphi & \frac{1}{a} \cos \varphi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.20)$$

Teniendo en cuenta la ecuación anterior la ley de control del controlador cinemático nos queda de la siguiente manera [17]:

$$\begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\frac{1}{a} \sin \varphi & \frac{1}{a} \cos \varphi \end{bmatrix} \begin{bmatrix} \dot{x}_{ref} + l_x \tanh\left(\frac{k_x}{l_x} E_x\right) \\ \dot{y}_{ref} + l_y \tanh\left(\frac{k_y}{l_y} E_y\right) \end{bmatrix} \quad (2.21)$$

Donde:

E_x : error de posición actual en el eje X ($x_{ref} - x$)

E_y : error de posición actual en el eje Y ($y_{ref} - y$)

k_x, k_y : ganancias del controlador

l_x, l_y : constantes de saturación

2.5 Introducción al controlador PID no Lineal

El controlador PID no Lineal es una variación del controlador PID clásico, en este controlador las ganancias del controlador (proporcional, integral, derivativa) ya no son constantes sino más bien dependen de algún parámetro que varíe en el tiempo, en este caso el error del sistema, esto le proporciona un mejor rendimiento al sistema [18], se utiliza el error ya que si este crece el controlador tiende a tener ganancias mayores y con esto una acción de control más agresiva con el objetivo de estabilizar el sistema de manera rápida, mientras que cuando el error es menor las ganancias se mantienen en un valor próximo a las sintonizadas tradicionalmente [19], la ecuación que describe este comportamiento es la siguiente:

$$K_p(e) = K_{po} + K_{NL}|e| \quad (2.22)$$

Donde:

$K_p(e)$: ganancia proporcional dependiente del error en el tiempo

K_{po} : ganancia proporcional lineal

K_{NL} : constante de no linealidad, obtenida de manera experimental

e : error del sistema

La ecuación (2.22) también podría ser aplicada a las otras ganancias del controlador obteniendo así tres ecuaciones que dependan del error, pero por facilidad se utiliza la forma ideal del controlador en la cual la ganancia proporcional de (2.22) afecta a las otras ganancias del sistema tal como se observa en la siguiente ecuación:

$$u(t) = (K_p(e)) * (e(t) + \frac{1}{T_i} \int e dt + T_d \frac{de(t)}{dt}) \quad (2.23)$$

Entonces la función de transferencia resultante de un controlador PID no lineal de forma ideal es:

$$C_{PID NL} = K_p(e) * (1 + \frac{1}{T_i s} + T_d s) \quad (2.24)$$

2.6 Diseño del controlador PID Dual no Lineal

El concepto de dualidad sugiere la implementación de más de un controlador con el objetivo de mejorar la respuesta en el seguimiento de trayectorias tal como se describe en [20], en este caso se implementarán dos controladores, un PID no lineal y un PI no lineal.

Existen dos condiciones fundamentales las cuales sirven como indicador para saber cuándo se debe cambiar de un controlador a otro, estos se describen a continuación:

- 1) El valor absoluto del error supera cierto límite.
- 2) Existe un cambio en la referencia.

En el seguimiento de trayectorias los cambios de referencia son constantes, es decir, siempre se está generando nuevas referencias tanto de velocidad lineal como de velocidad angular por lo cual el segundo criterio no podría ser aplicado, por ende, debemos utilizar el criterio del valor absoluto del error para saber cuándo cambiar de controlador.

Sabemos que el PID no lineal tiene una respuesta más rápida y menos agresiva que el controlador PI no lineal por lo cual cuando se tenga errores elevados trabajará el controlador PID no lineal mientras que cuando el error disminuya se trabajará con el controlador PI no lineal, esto para evitar que el sistema tenga oscilaciones prominentes cuando se tenga errores elevados y conseguir así un control más suave pero eficaz y por otro lado, no tener error de posición en el estado estable gracias a la acción del controlador PI no lineal.

La literatura [21] sugiere que el valor máximo permisible para realizar el cambio de controlador es el 5% del error máximo, es decir, cuando la magnitud del error sea mayor al

5% del error máximo se trabajará con el controlador PID no lineal, mientras que cuando se tenga una magnitud de error menor al 5% se trabajará con el controlador PI no lineal.

Para la sintonización de los controladores debemos partir de la sintonización del PID lineal clásico, para esto se utiliza los parámetros principales de los modelos de primer orden aproximados encontrados en el apartado 2.3. (ganancia del sistema K_c , constante de tiempo del sistema τ y retardo del sistema t_m), el método de sintonización de las constantes del PID es el de Chien-Hrones-Reswick (CHR) [22].

Las ecuaciones para la sintonización del controlador PI son:

$$K_{po} = \frac{0.6\tau}{K_c t_m} \quad (2.25)$$

$$T_i = \tau \quad (2.26)$$

Las ecuaciones para la sintonización del controlador PID son:

$$K_{po} = \frac{0.95\tau}{K_c t_m} \quad (2.27)$$

$$T_i = 1.36\tau \quad (2.28)$$

$$T_d = 0.47t_m \quad (2.28)$$

Una vez que se ha sintonizado los controladores PID tradicionales se procede a la sintonización mediante experimentación de la constante o factor de no linealidad (K_{NL}) para que los controladores pasen de ser PID's lineales a no lineales, esto se lo realiza comparando los distintos valores del índice ISE que arrojen cada uno de los resultados con cada controlador. En el Anexo I se encuentra el código para el cálculo de los parámetros de los controladores antes descritos.

Los valores de K_{NL} obtenidos para cada controlador se describen en la tabla 2.1.

Tabla 2.1. Factores de no linealidad sintonizados para cada controlador

Controlador	Factor de no linealidad (K_{NL})
PID no lineal (Velocidad lineal)	1
PI no lineal (Velocidad lineal)	2.375
PID no lineal (Velocidad angular)	1.8
PI no lineal (Velocidad angular)	2

Se debe tener en consideración otro aspecto importante al momento de encontrar experimentalmente los valores del factor no lineal y es que las acciones de control

provenientes de cada controlador estén dentro de los límites operativos del robot revisados en la tabla 1.1, esto con el objetivo de que el robot funcione sin ningún problema.

2.7 Comunicación entre Simulink y Coppelia SIM

Para la correcta simulación del seguimiento de trayectoria del Pioneer 3DX es necesaria la utilización de dos programas cuya comunicación debe ser síncrona, el primero es Simulink, este software de Matlab nos permite simular los controladores que se necesitan para el seguimiento de trayectorias así como también para la generación de las mismas, como se vio en 2.1 se necesita de un controlador cinemático, del controlador diseñado (PID Dual No Lineal) y de un generador de trayectorias, todo esto se lo realiza en Simulink mediante diagramas de bloques [23]; el otro programa es Coppelia SIM, este software nos permite simular el robot Pioneer 3DX como si fuera el modelo real (ver Figura 2.9), dentro de las librerías de este software podemos encontrar este robot y muchísimos otros modelos, además se cuenta con un sistema de sensores los cuales nos permiten obtener los parámetros reales como velocidad lineal, velocidad angular, posición, etc. [24], para la respectiva realimentación hacia el software Simulink, esto quiere decir que los dos programas están trabajando en lazo cerrado.



Figura 2.9 Robot Pioneer 3DX en Coppelia SIM

Como se mencionó anteriormente la comunicación tiene que ser obligatoriamente síncrona debido a que para el seguimiento de trayectorias es necesario conocer la posición y otros parámetros del robot en cada instante de tiempo para así generar la acción de control que se enviará al robot, para esto es necesario que los tiempos de simulación de cada uno de

los programas sean los mismos, en este caso se trabaja con 50ms, en la Figura 2.10 se puede apreciar un grafico que describe de mejor manera cual es la función de cada uno de los programas que intervienen en la simulación del seguimiento de trayectoria del Pioneer 3DX.

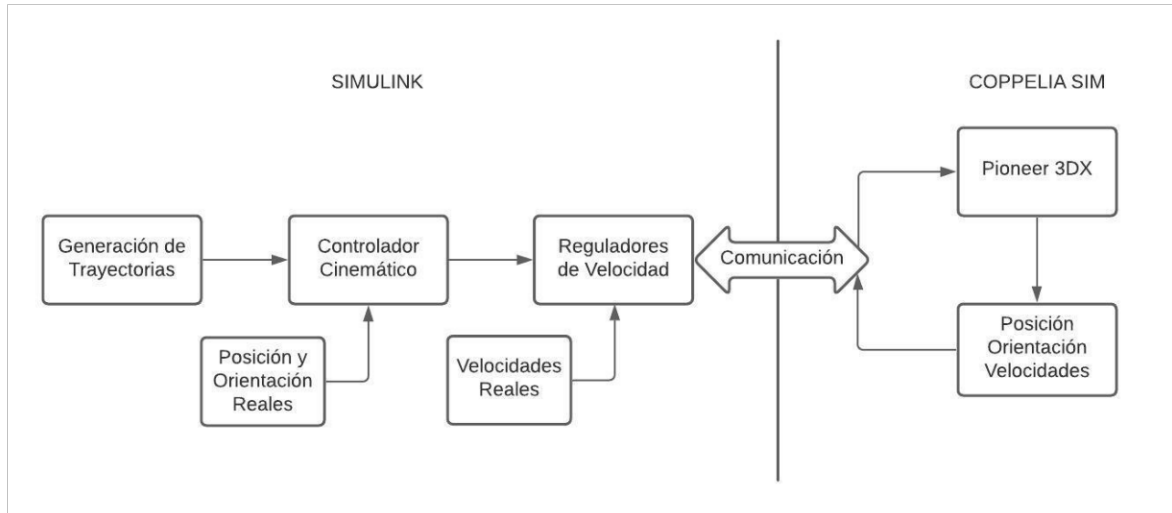


Figura 2.10 Diagrama representativo de la comunicación entre Simulink y Coppelia SIM

2.8 Desarrollo de la interfaz gráfica en App Designer

Para la realización de la interfaz grafica se utilizó App Designer, esta aplicación de Matlab permite el diseño de interfaces graficas de manera simple y rápida [25], cuenta con varios componentes los cuales permiten al usuario visualizar de mejor manera los resultados de este trabajo además de seleccionar que parámetros desea visualizar en la interfaz.

Esta aplicación viene incluida en el software Matlab por lo cual no es necesario descargarlo de los complementos que ofrece este software, para ingresar a esta app únicamente es necesario colocar en la ventana de comandos *appdesigner*, como se mencionó anteriormente es sumamente sencillo trabajar con esta aplicación ya que cuenta con un panel en el cual encontraremos todos los elementos necesarios para crear una interfaz casi profesional (ver Figura 2.11), solo es cuestión de arrastrar cada uno de los elementos que queremos añadir a la interfaz y programar el comportamiento que tendrá dentro de esta.

En este caso se trabajará con dos ventanas, la primera actuará como una caratula que nos permitirá seleccionar cual de los controladores podemos simular, para esta únicamente se utilizó textos y botones, además de una imagen de fondo. (Ver Figura 2.12)

En la segunda ventana encontraremos la interfaz que le permite al usuario seleccionar entre las diferentes trayectorias que puede seguir el robot y que parámetro desea observar,

entre estos están la velocidad lineal, velocidad angular, error de posición y los índices de rendimiento respectivos, finalmente cuenta con dos planos gráficos en los cuales se observará los resultados de las simulaciones. (Ver Figura 2.13)

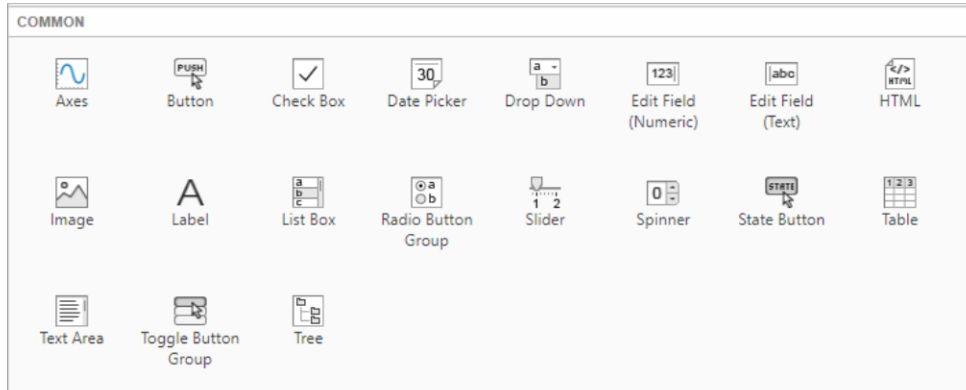


Figura 2.11 Elementos comunes de la librería de componentes de App Designer



Figura 2.12 Ventana de selección del controlador

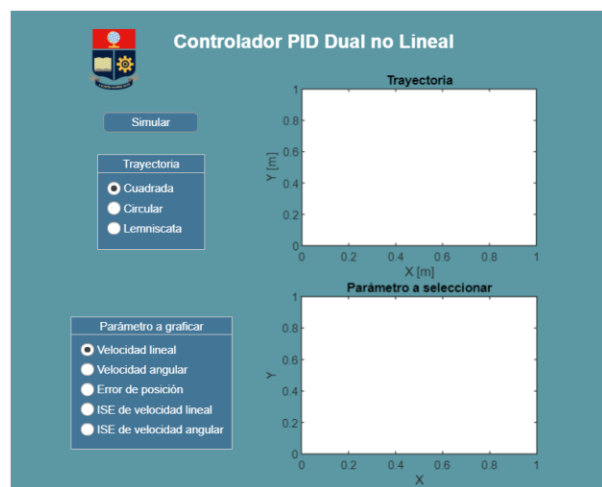


Figura 2.13 Ventana de selección de la trayectoria y otros parámetros a visualizarse

En el Anexo II se encuentra una guía para la correcta utilización de la interfaz diseñada.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se analizarán los diferentes resultados obtenidos de la simulación del controlador PID Dual no Lineal diseñado en el capítulo anterior, para el seguimiento de trayectorias usando el robot Pioneer 3DX.

Se realizó la simulación de tres trayectorias específicas cuya generación fue explicada anteriormente, se empieza con el análisis de los resultados de la trayectoria cuadrada, luego se analizarán los resultados con la trayectoria circular y por último se apreciarán los resultados de la trayectoria lemniscata de Bernoulli, todas las simulaciones fueron realizadas con un tiempo de 80 segundos.

Antes del respectivo análisis de cada uno de los resultados de la simulación debemos tener en cuenta que el robot real se encuentra en el origen del eje de coordenadas, es decir, en el punto (0,0) mientras que la generación de la trayectoria empieza a 3m hacia la derecha del punto 0 con respecto al eje X, es decir en el punto (3, 0), al iniciar el desplazamiento existirá un estado transitorio en el cual el robot tratará de seguir la referencia de posición.

3.1 Resultados de la simulación con la trayectoria cuadrada

Al ser una trayectoria cuadrada la mayor dificultad de seguimiento de trayectoria se encuentra en las esquinas del cuadrado debido a que en estos puntos las referencias de la velocidad angular y lineal tienen a cambiar de manera abrupta.

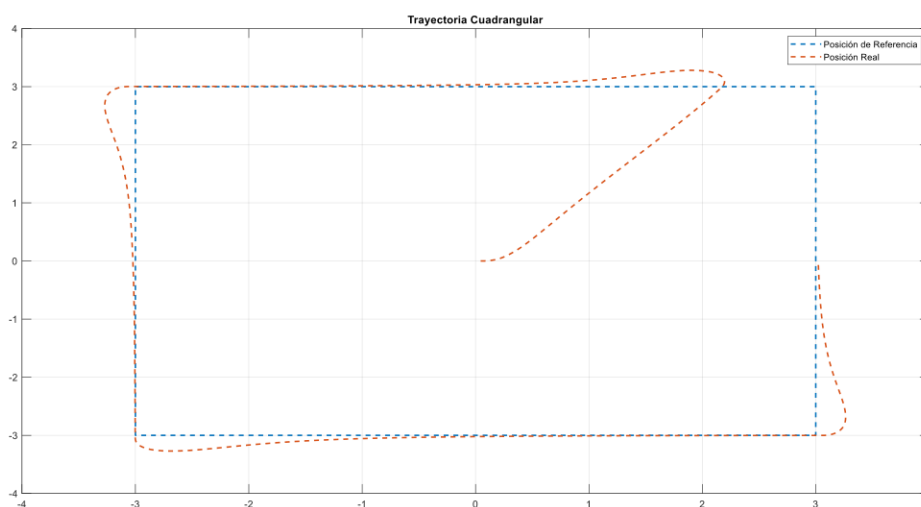


Figura 3.1 Resultado del seguimiento de trayectoria del Pioneer 3DX usando el controlador PID DUAL no Lineal (Trayectoria cuadrada)

Se puede apreciar en la Figura 3.1, como se había mencionado anteriormente, que el seguimiento de trayectoria en las esquinas del robot es particular, esto se debe a que en estos puntos la referencia de velocidad lineal se vuelve muy pequeña mientras que la referencia de velocidad angular crece considerablemente, el robot no puede detenerse y girar por lo cual se sale de la trayectoria pero en el transcurso del tiempo vuelve a seguir correctamente la trayectoria, como sabemos cuando el error de posición es mayor al 5% del valor máximo de error actúa el controlador PID no lineal, en este caso este controlador estaría funcionando al inicio de la simulación debido a que en este punto tenemos el error máximo y también en cada una de las esquinas de la trayectoria ya que el error crece en estos puntos.

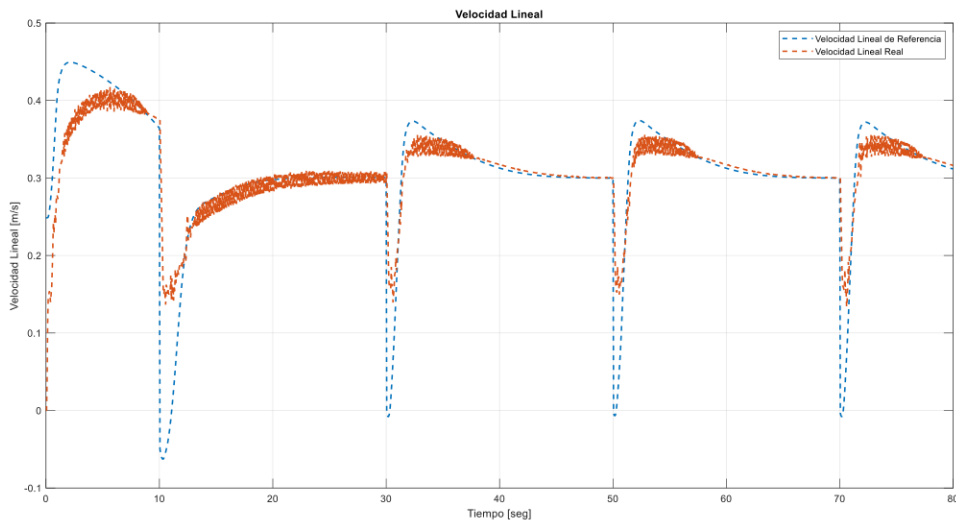


Figura 3.2 Resultado de velocidad lineal de referencia y real (Trayectoria cuadrada)

En la Figura 3.2 podemos apreciar el comportamiento de la velocidad lineal, habíamos mencionado que en las esquinas del robot la velocidad lineal tiende a bajar de manera agresiva y esto es evidenciado en los instantes de tiempo $t=30$, $t=50$ y $t=70$ seg, en estos tiempos la velocidad lineal real disminuye de manera considerable pero no baja hasta 0, es decir el robot no se detiene, luego de haber conseguido pasar la esquina del cuadrado la velocidad lineal vuelve a tratar de mantenerse en su valor constante de 0.3m/s. En $t=10$ seg no se tiene una esquina de la trayectoria, en este instante el robot tiene que girar para dejar su trayectoria inicial desde el origen y comenzar a seguir la trayectoria de referencia, por lo que la referencia de velocidad lineal en este instante es menor, otro aspecto importante es que al inicio la referencia de velocidad lineal es elevada debido a que el error es mayor y se necesita que el robot acelere hasta llegar a la referencia.

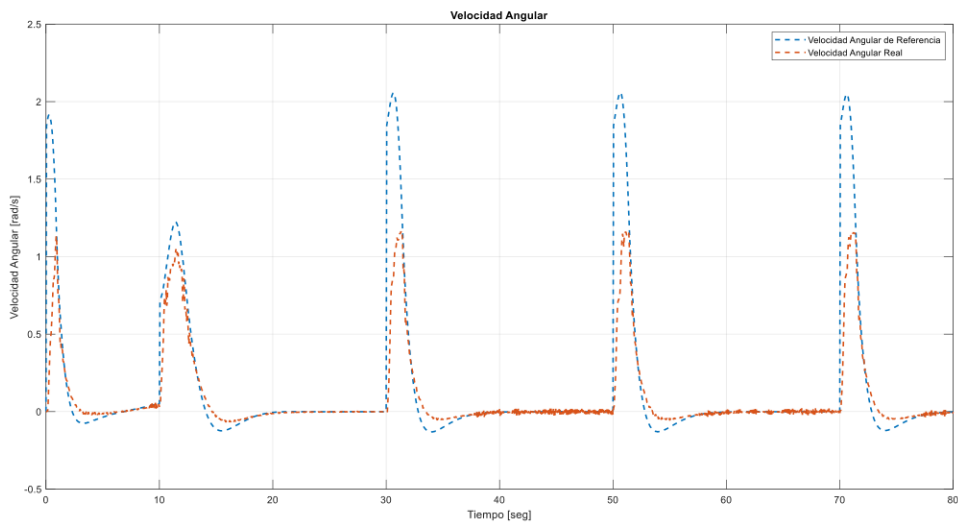


Figura 3.3 Resultado de velocidad angular de referencia y real (Trayectoria cuadrada)

En la Figura 3.3 tenemos el resultado de la velocidad angular, en ciertos instantes de tiempo la velocidad angular crecerá debido a que se necesita que el robot gire (esquinas del cuadrado), se puede apreciar que la referencia es mucho más elevada en las esquinas del cuadrado que cuando el robot comienza a seguir la trayectoria en $t=10\text{seg}$, se debe tener en cuenta que la velocidad angular se mantiene entre los límites de construcción del robot para evitar que este se dañe, las velocidades angulares negativas se dan debido a que el robot tiene que estabilizarse cada vez que acabar de girar.

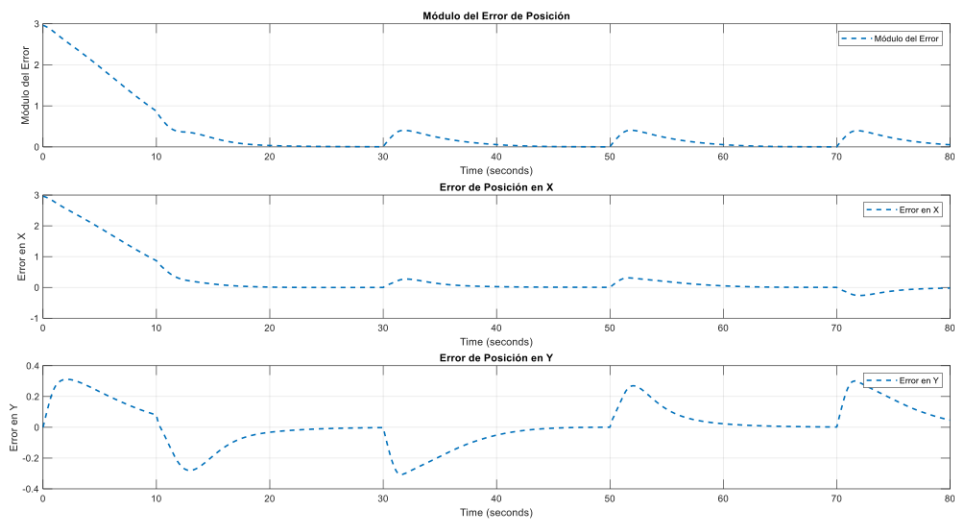


Figura 3.4 Errores de posición del robot (Trayectoria cuadrada)

En la Figura 3.4 tenemos tres gráficos, el primero representa el modulo del error de posición que tiene el robot durante toda la trayectoria, al inicio se tiene un mayor error debido a que

el robot empieza en el origen de coordenadas mientras que la referencia a 3m de distancia del origen por lo cual el error máximo tiene un valor de 3, luego se tienen errores cada vez que el robot gira, es decir, en las esquinas de la trayectoria cuadrada y cuando el robot intenta alcanzar la referencia por primera vez, cuando el robot alcanza la referencia podemos observar que el error tiende a cero, las otras dos graficas representan los errores de posición tanto en X como Y, podemos apreciar que existen mayores errores en el eje X debido a la posición inicial del robot, el comportamiento durante el resto de la trayectoria es similar a la primera grafica.

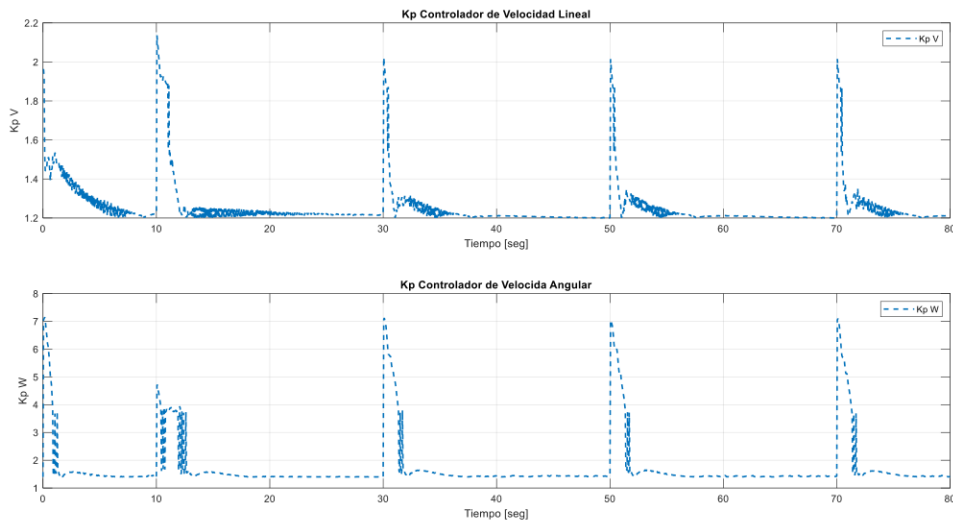


Figura 3.5 Resultado del parámetro Kp de los controladores de velocidad lineal y angular (Trayectoria cuadrada)

En la Figura 3.5 se muestra la evolución que tiene el parámetro Kp de ambos controladores, el de velocidad lineal y el de velocidad angular, durante el seguimiento de la trayectoria, sabemos que al inicio se trabaja con el controlador PID Dual no Lineal por lo cual en estos instantes de tiempo el valor de Kp depende del valor de la constante de no linealidad sintonizada para este controlador así como del error, como se observa en la grafica debido a que el error es alto se obtiene una Kp elevada, mientras que cuando el error disminuye entra a trabajar el controlador PI no Lineal el cual tiene otra constante de no linealidad por lo cual se nota el cambio de controlador considerablemente en ambas gráficas, en especial en la del Kp de la velocidad angular debido a que existe una mayor diferencia entre los valores de las ganancias en este controlador.

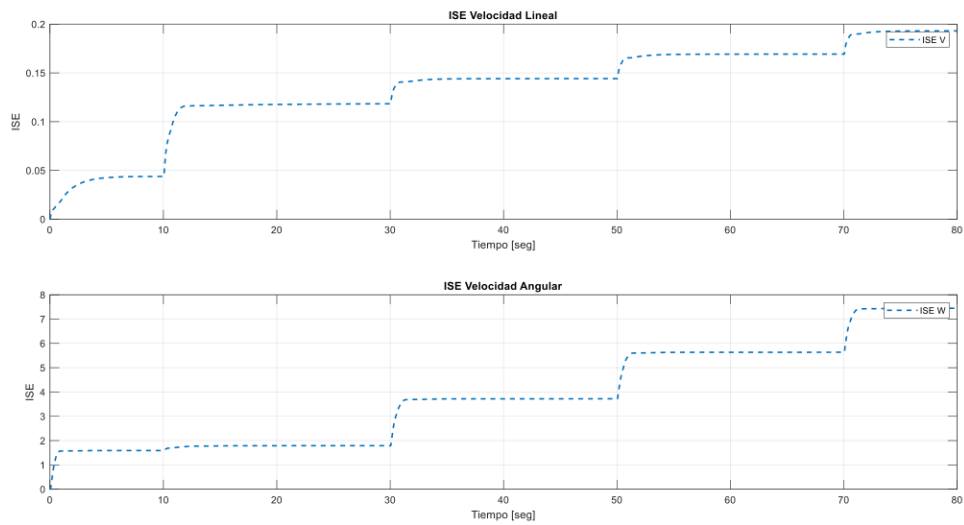


Figura 3.6 Índices ISE de velocidad lineal y velocidad angular (Trayectoria cuadrada)

El análisis de los índices ISE se los puede realizar observando la Figura 3.6 en la cual podemos observar que en la velocidad lineal se tiene un ISE máximo de 0.193 y como se esperaba el ISE aumenta cada vez que el robot se encuentra con un giro en la trayectoria debido a que en estos puntos el error es mayor tanto en posiciones como en velocidades, en cuanto al ISE de la velocidad angular es mucho más considerable debido a que existe un mayor error entre la velocidad angular real y su referencia, el ISE final en este caso es de 7.445.

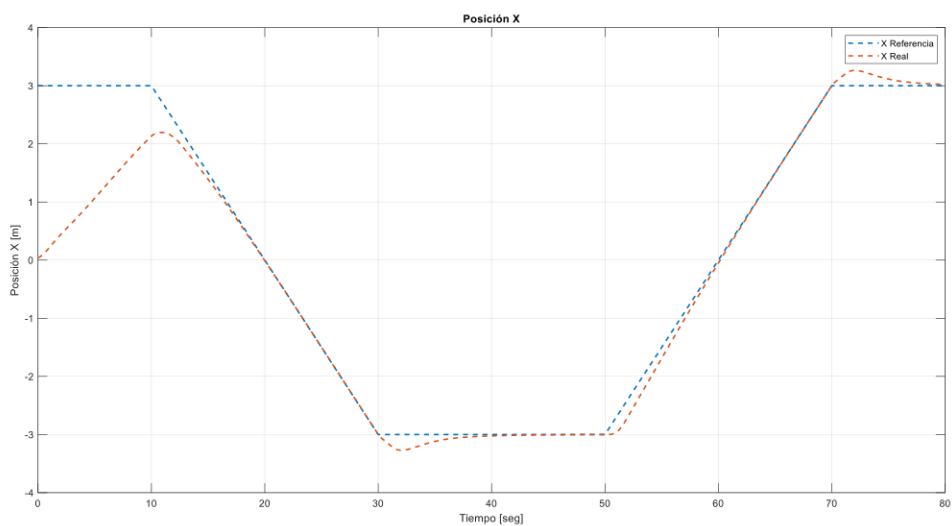


Figura 3.7 Resultado de las posiciones en X tanto de referencia como real (Trayectoria cuadrada)

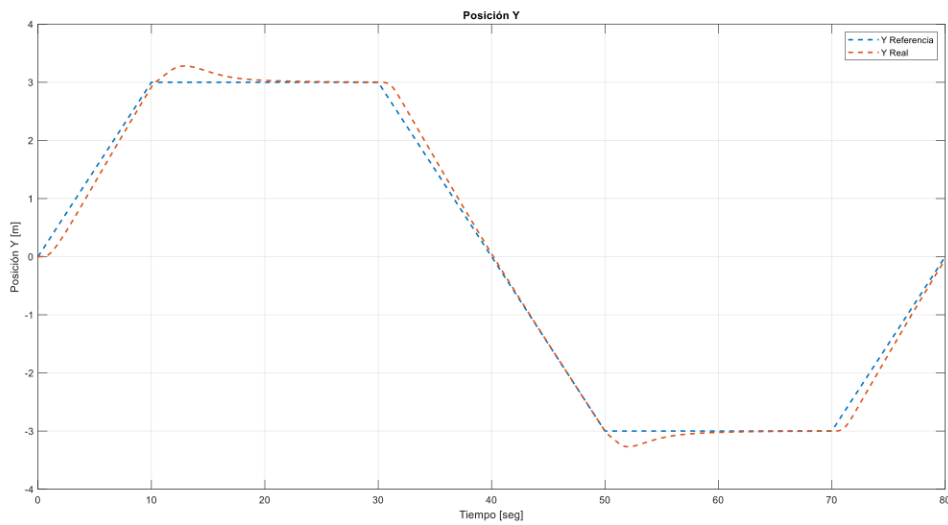


Figura 3.8 Resultado de las posiciones en Y tanto de referencia como real (Trayectoria cuadrada)

En las Figuras 3.7 y 3.8 tenemos los resultados de posiciones reales y posiciones de referencia en los ejes X y Y respectivamente, se puede apreciar que como se ha venido analizando el robot empieza muy separado de su referencia en X por lo que hay un error elevado al inicio, mientras que en Y se empieza sin error de posición, como también se ha analizado anteriormente el robot cumple con sus posiciones en los instantes de tiempo que no esta girando debido a las esquinas del cuadrado.

3.2 Resultados de la simulación con la trayectoria circular

La segunda trayectoria de la que analizaremos los resultados es la circular, en esta se espera que inicialmente el robot alcance la referencia debido a que empieza lejos de esta y cuando ya la alcance se normalice y se vuelva constante tanto la velocidad lineal como la angular, la referencia fue generada con ambos valores de velocidad constantes, una velocidad lineal de 0.3m/seg y una velocidad angular de 0.1 rad/seg.

El radio del círculo es de 3m por lo cual existirá el mismo error de posición inicial en X que en la anterior trayectoria, a diferencia de la anterior trayectoria en esta el controlador PID no Lineal solo actuará al inicio de la trayectoria debido al alto error, cuando el error sea menor al 5% actuará el controlador PI no Lineal hasta el final de la trayectoria.

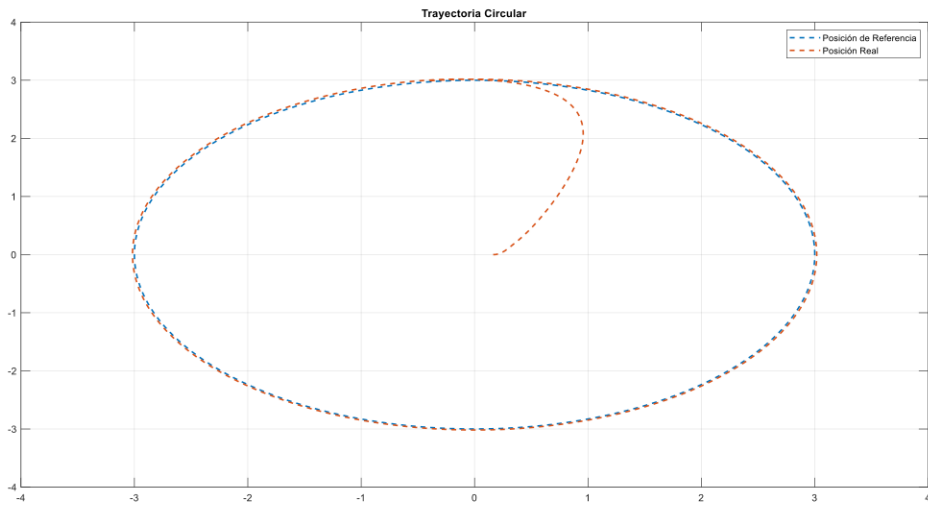


Figura 3.9 Resultado del seguimiento de trayectoria del Pioneer 3DX usando el controlador PID DUAL no Lineal (Trayectoria circular)

Como se puede observar en la Figura 3.9 se obtiene un muy buen resultado en cuanto al seguimiento de esta trayectoria por parte del robot, como se mencionó anteriormente el robot empieza en el origen de coordenadas y se demora algunos segundos en llegar a la trayectoria de referencia por lo cual el error será mayor en este intervalo de tiempo, luego el error disminuirá llegando a tender a 0.

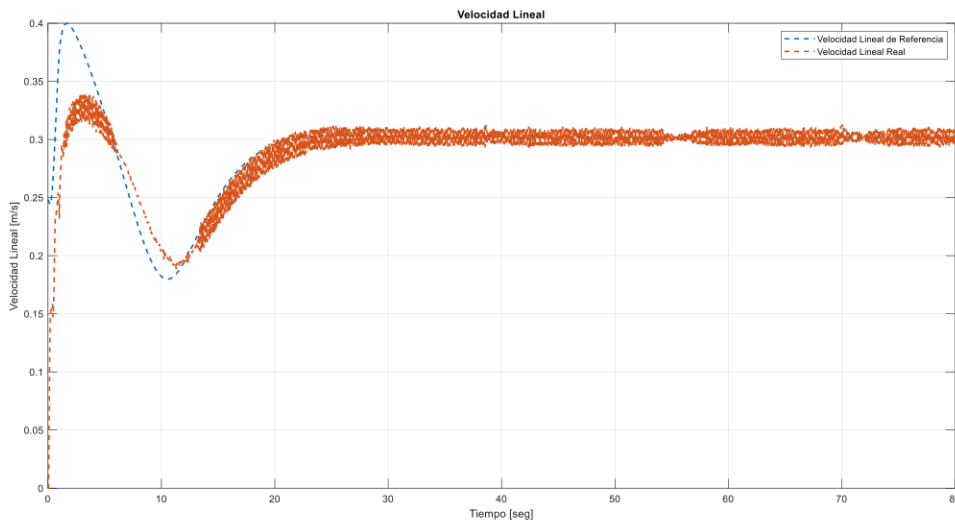


Figura 3.10 Resultado de velocidad lineal de referencia y real (Trayectoria circular)

Como podemos apreciar en la Figura 3.10 en un principio la referencia de velocidad lineal es relativamente alta y la velocidad real trata de seguir la referencia, esto se da debido a que el error es alto en este punto por lo cual la constante no lineal también es alta haciendo

al controlador más agresivo con el objetivo de alcanzar rápidamente la referencia de trayectoria, una vez se llegue a alcanzar la referencia la velocidad lineal se vuelve relativamente constante ya que existe una oscilación permanente, esto se debe a que se trata de un control PI no lineal cuya naturaleza es oscilatoria.

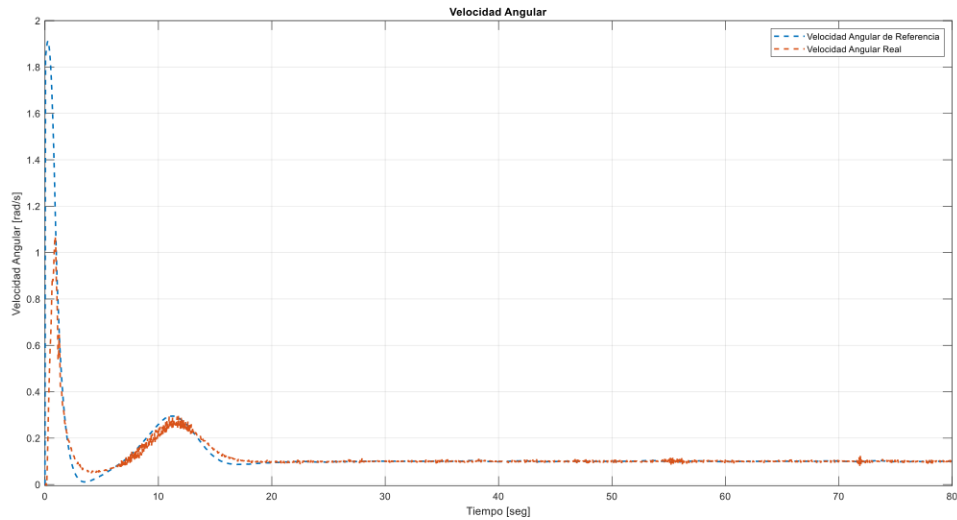


Figura 3.11 Resultado de velocidad angular de referencia y real (Trayectoria circular)

En este caso podemos analizar la velocidad angular resultante en la Figura 3.11, como habíamos mencionado al inicio debido al alto error la referencia de velocidad angular es elevada, la velocidad real trata de llegar a la referencia, pero no puede sobrepasar el límite constructivo de velocidad angular, luego del estado transitorio se obtiene una respuesta muy buena con oscilaciones casi imperceptibles.

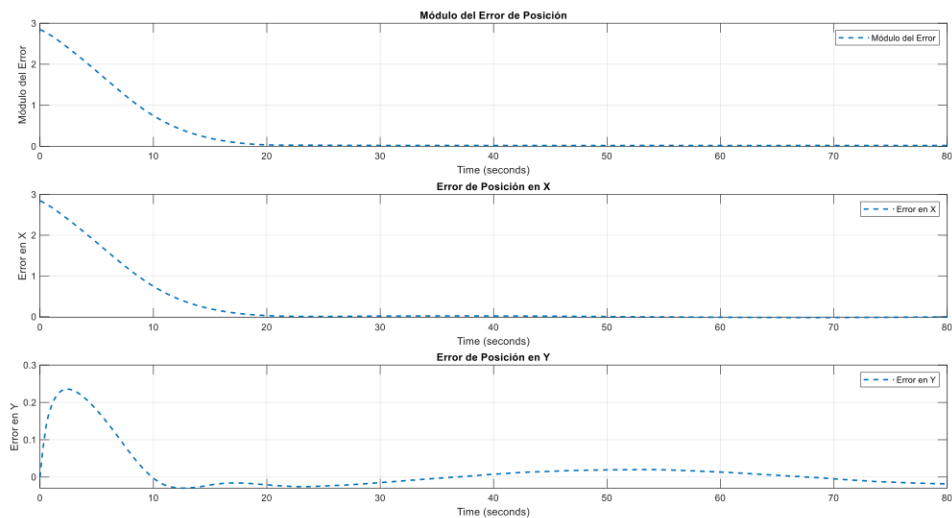


Figura 3.12 Errores de posición del robot (Trayectoria circular)

En el grafico 3.12 tenemos los errores de posición en ambos ejes además del modulo total del error, en esta trayectoria podemos apreciar que el error más alto se da al inicio de la trayectoria, posteriormente el error tiende a cero y se mantiene en estos valores, esto se da debido a que una vez que el robot alcanza la trayectoria comienza a tener valores de velocidad lineal y angular constantes, en la grafica de los errores de la posición en Y se tienen pequeños picos pero debido a que son muy bajos pueden ser despreciados.

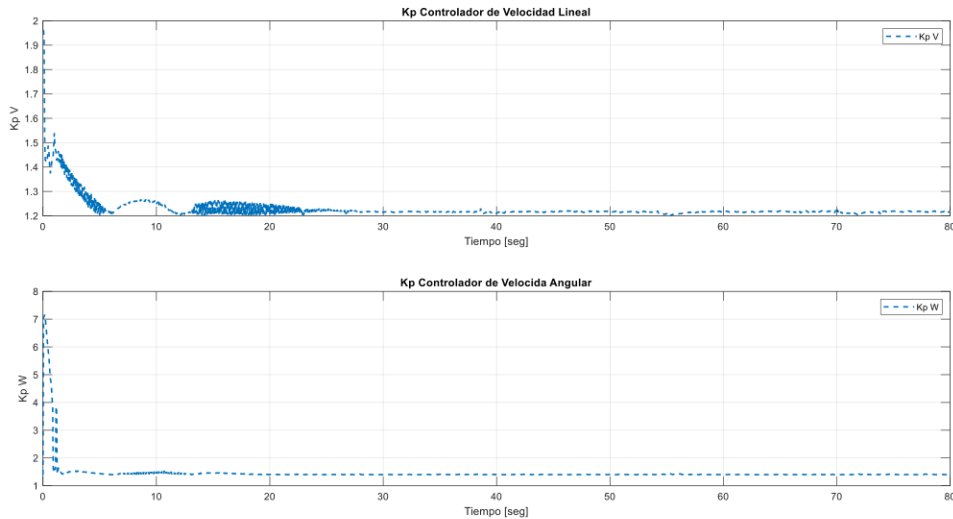


Figura 3.13 Resultado del parámetro Kp de los controladores de velocidad lineal y angular (Trayectoria circular)

Para analizar el parámetro Kp de cada controlador debemos observar la Figura 3.13, en esta podemos apreciar tanto el Kp del controlador de la velocidad lineal como el de la velocidad angular, al igual que con la anterior trayectoria el cambio de controladores se nota mucho más en el Kp de la velocidad angular debido a que las constantes no lineales tienen mayor diferencia en este controlador. En este caso el valor de Kp es alto únicamente en los instantes de tiempo en los que el error es alto, es decir, al inicio de la trayectoria cuando el robot trata de llegar a la trayectoria de referencia, luego de esto el valor del Kp baja y se mantiene en un valor cercano a el valor de la ganancia de cada controlador debido a que el error tanto en la velocidad lineal como en la velocidad angular tiende a 0.

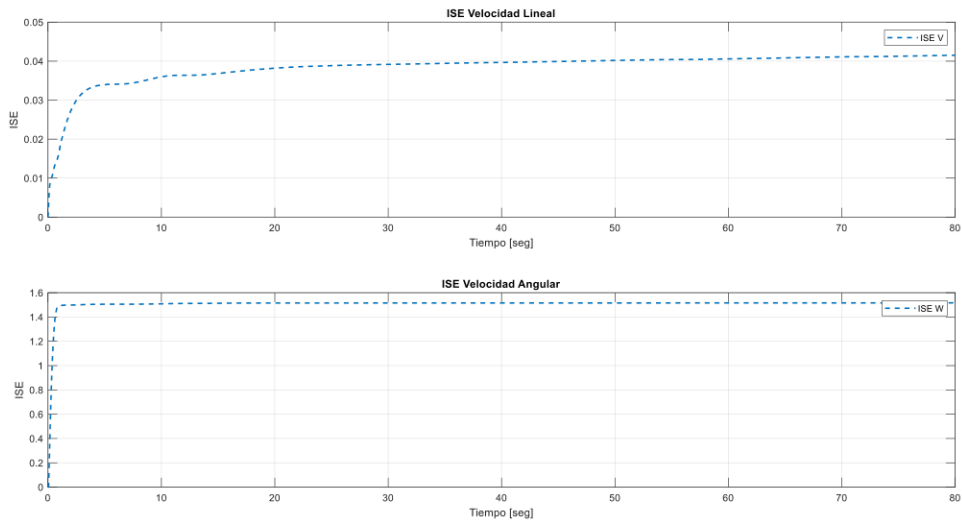


Figura 3.14 Índices ISE de velocidad lineal y velocidad angular (Trayectoria circular)

En la Figura 3.14 se encuentran los índices ISE tanto de la velocidad lineal como de la velocidad angular, en este caso a diferencia de la anterior trayectoria, el ISE crece en su mayoría en los instantes iniciales cuando hay un error elevado en la velocidad lineal y en la velocidad angular, en el caso de la velocidad lineal el ISE continua creciendo poco a poco con el paso del tiempo hasta un valor final de 0.0415 mientras que en el caso de la velocidad angular el ISE crece en un intervalo de tiempo muy pequeño (menos de 5 segundos) llegando a un valor máximo de 1.516, valor considerablemente menor que el ISE de la velocidad angular obtenido en la trayectoria cuadrada.

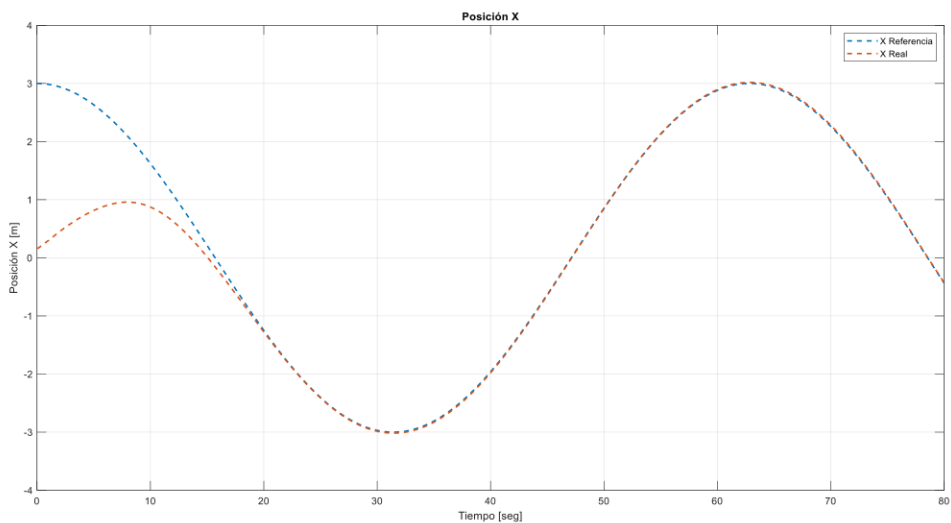


Figura 3.15 Resultado de las posiciones en X tanto de referencia como real (Trayectoria circular)

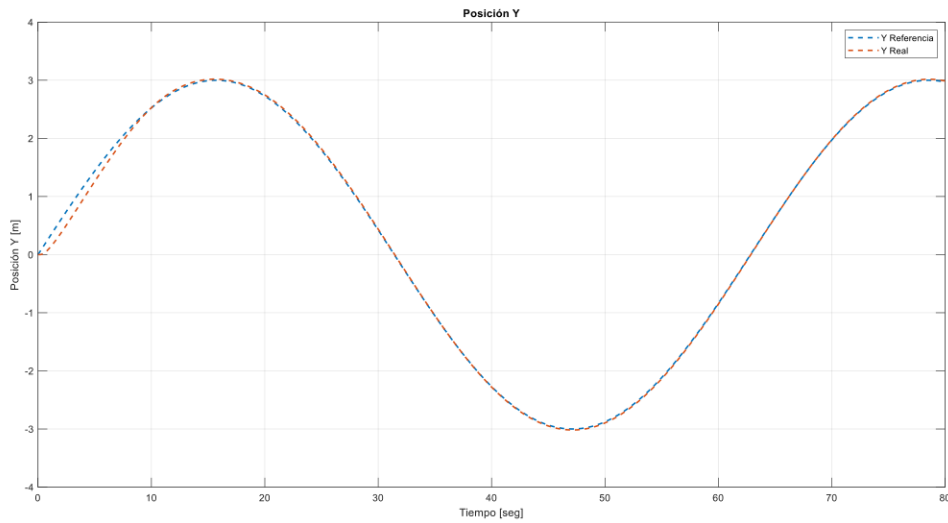


Figura 3.16 Resultado de las posiciones en Y tanto de referencia como real (Trayectoria circular)

En las Figuras 3.15 y 3.16 tenemos los resultados de posiciones reales y posiciones de referencia en los ejes X y Y respectivamente, en las posiciones en X observamos que debido a que el robot empieza su desplazamiento en el origen existe una diferencia de posiciones igual al radio del círculo, por otro lado, la posición en Y tiene un error menos considerable en el estado transitorio debido a que empiezan en el mismo valor de Y, una vez se llega a la referencia las posiciones tienen un error de posición cercano a 0.

3.3 Resultados de la simulación con la trayectoria lemniscata de Bernoulli

La última trayectoria por analizar es la lemniscata de Bernoulli, esta interesante trayectoria nos permite apreciar el comportamiento de nuestro controlador Dual ante cambios de referencias tanto de la velocidad lineal como la angular, de manera creciente y decreciente, como ya veremos más adelante.

Los parámetros de la trayectoria de referencia tienen una alta similitud con los vistos en la anterior trayectoria, ya que se busca mantener constantes los valores de la velocidad lineal y angular en ciertos intervalos de tiempo, estos tienen un valor de 0.3m/s y 0.1rad/s respectivamente.

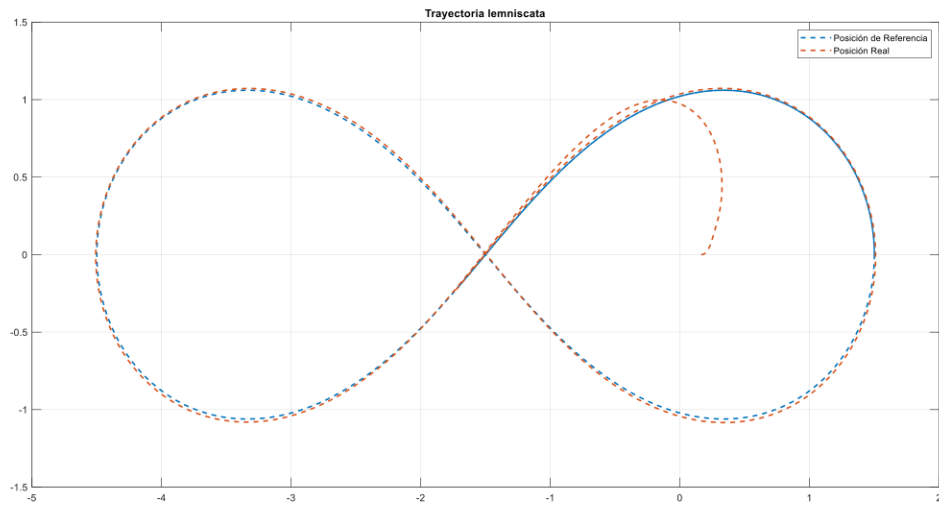


Figura 3.17 Resultado del seguimiento de trayectoria del Pioneer 3DX usando el controlador PID DUAL no Lineal (Trayectoria lemniscata)

Como se aprecia en la Figura 3.17 el robot sigue muy bien la referencia de la trayectoria, podemos apreciar que hay pequeños errores de posición sobre todo en las partes circulares de la trayectoria, al inicio al igual que en las otras trayectorias el robot se encuentra en el origen por lo cual tiene que movilizarse hasta llegar a la referencia, durante este periodo de tiempo trabaja el controlador PID no Lineal, mientras que cuando se alcanza la referencia trabaja el controlador PI no Lineal debido a que el error disminuye por debajo del 5% del error total.

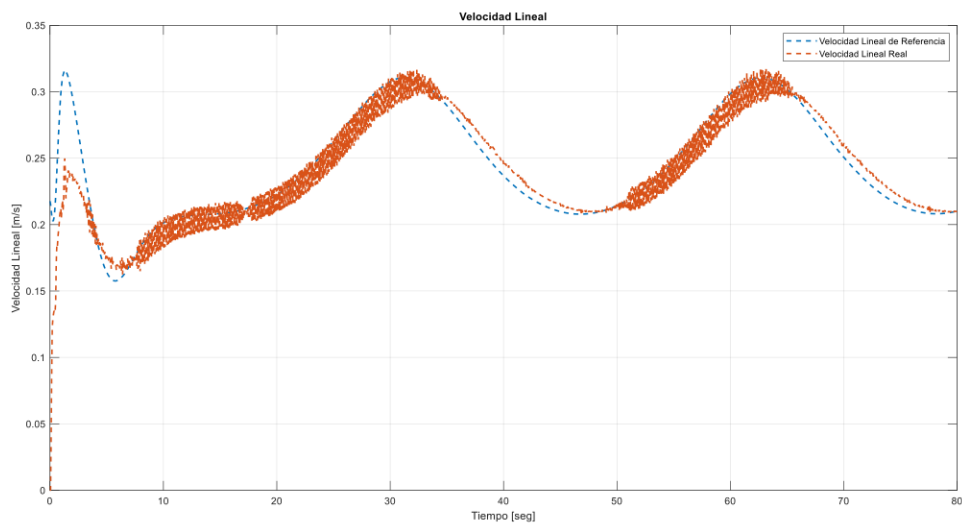


Figura 3.18 Resultado de velocidad lineal de referencia y real (Trayectoria lemniscata)

Podemos apreciar en la Figura 3.18 que la velocidad lineal tiene muchos periodos de tiempo en los que es oscilatoria tal como pasaba en la trayectoria circular, esto se debe a la naturaleza del controlador ante trayectorias de este tipo, vemos que existen periodos de tiempo de aceleración y desaceleración por parte del robot, cuando el robot sigue una línea recta este acelera sin sobrepasar su límite de velocidad máxima, mientras que cuando está tratando de seguir una curva su velocidad lineal tiende a ser constante y cercano al valor descrito anteriormente para lo cual debe desacelerar y esto es lo que se aprecia en el gráfico.

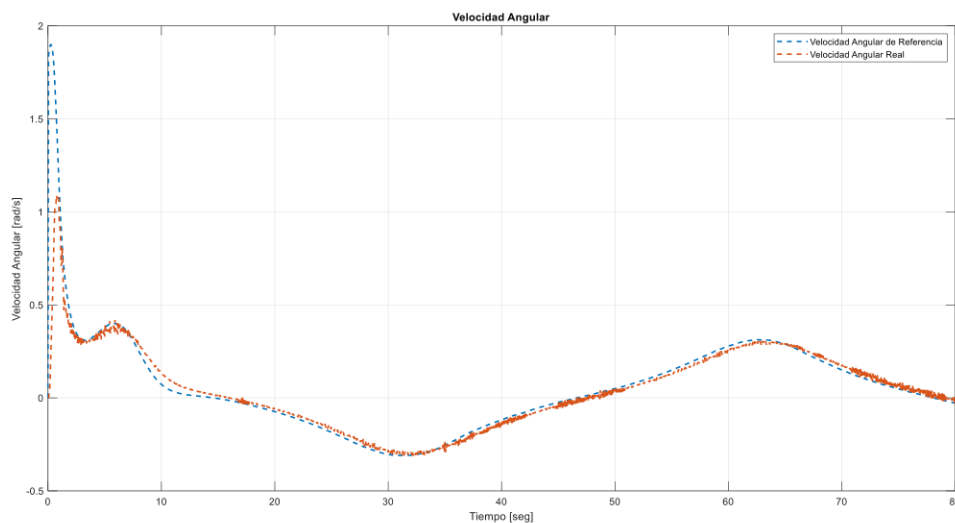


Figura 3.19 Resultado de velocidad angular de referencia y real (Trayectoria lemniscata)

En cuanto a la velocidad angular, los resultados los tenemos en la Figura 3.19, al igual que en la trayectoria circular no se tienen oscilaciones en la respuesta de la velocidad angular; debido a que el robot tiene que girar al inicio para dirigirse a la trayectoria de referencia se tiene una velocidad angular de referencia muy alta por lo que se trata de alcanzar este valor sin sobrepasar el límite máximo de velocidad angular, podemos apreciar que en este caso se cuenta con velocidades angulares negativas, esto se debe a que el robot pasa de estar girando hacia la izquierda a girar hacia la derecha, esto se puede apreciar de mejor manera en la Figura 3.17, este cambio de dirección de giro se traduce en pasar del eje Y positivo al eje Y negativo en la gráfica de la velocidad angular, en los intervalos de tiempo cuando el robot está sobre una línea recta la velocidad angular tiende a ser cero, se aprecia que esto solo sucede durante un instante de tiempo ya que el robot no se mantiene en línea recta.

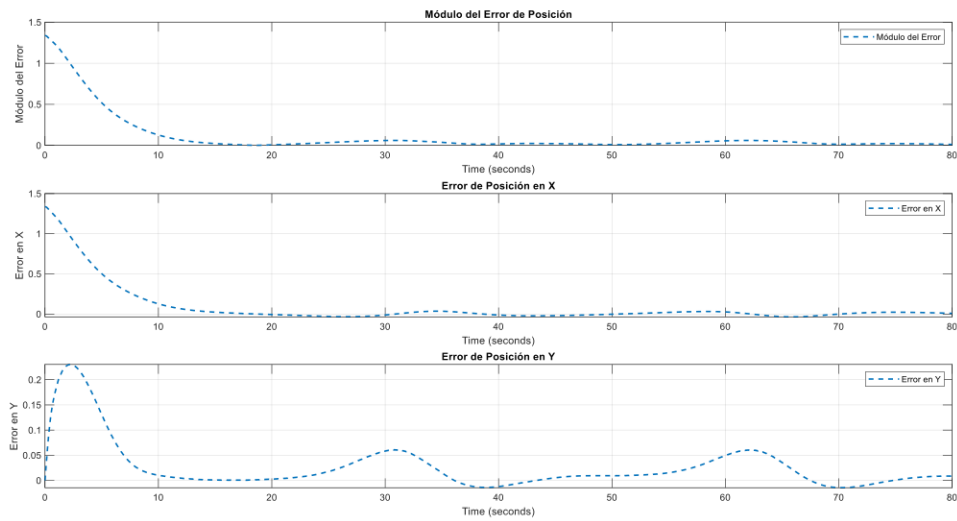


Figura 3.20 Errores de posición del robot (Trayectoria lemniscata)

Los errores los observamos en la Figura 3.20, al igual que en las anteriores trayectorias se tiene un error elevado al inicio de la simulación, este error decrece mientras el robot se acerca a la trayectoria de referencia llegando a valores cercanos a 0, al igual que en la anterior trayectoria el error de posición en X tiene un comportamiento similar al del módulo del error de posición, mientras que el error de posición en Y es mucho menos considerable.

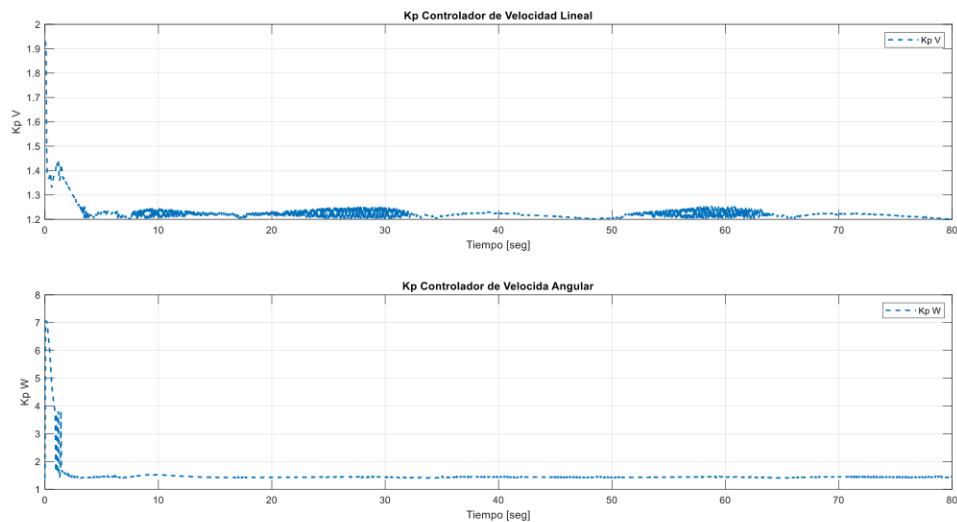


Figura 3.21 Resultado del parámetro Kp de los controladores de velocidad lineal y angular (Trayectoria lemniscata)

Para el análisis del parámetro Kp cuando se trabaja con la trayectoria lemniscata debemos acudir a la Figura 3.21, como se mencionó anteriormente el cambio de controladores se da cuando el error decrece por debajo del 5% del error máximo, esto se aprecia de mejor

manera en el K_p de la velocidad angular tal y como pasaba en las anteriores trayectorias, otro aspecto similar es que debido al alto error inicial se tiene un mayor valor en el parámetro K_p en ambos casos, velocidad lineal y velocidad angular.

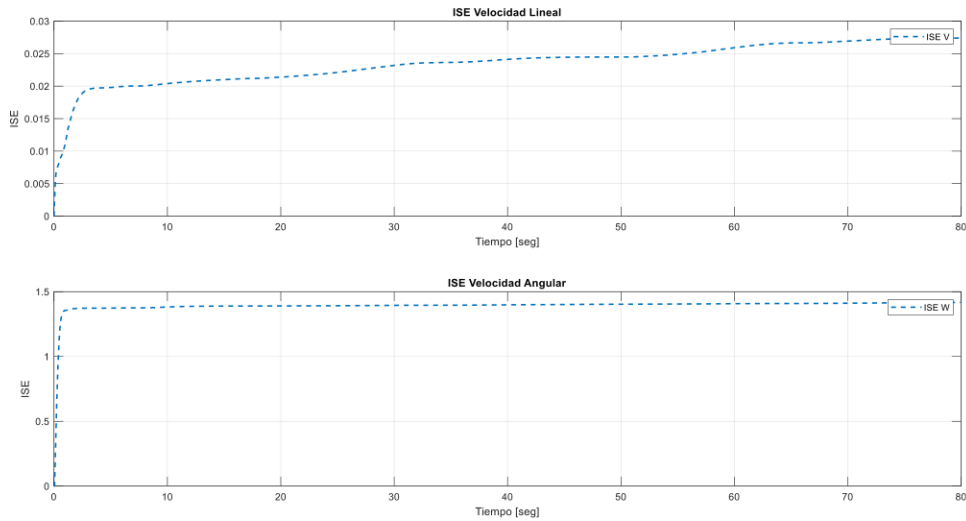


Figura 3.22 Índices ISE de velocidad lineal y velocidad angular (Trayectoria lemniscata)

En la Figura 3.22 observamos los índices ISE de los controladores de la velocidad lineal y angular, tienen un comportamiento muy parecido a los de la trayectoria circular, en este caso los valores máximos del ISE son 0.027 en velocidad lineal y 1.417 en velocidad angular, valores menores comparados a los obtenidos en las otras trayectorias.

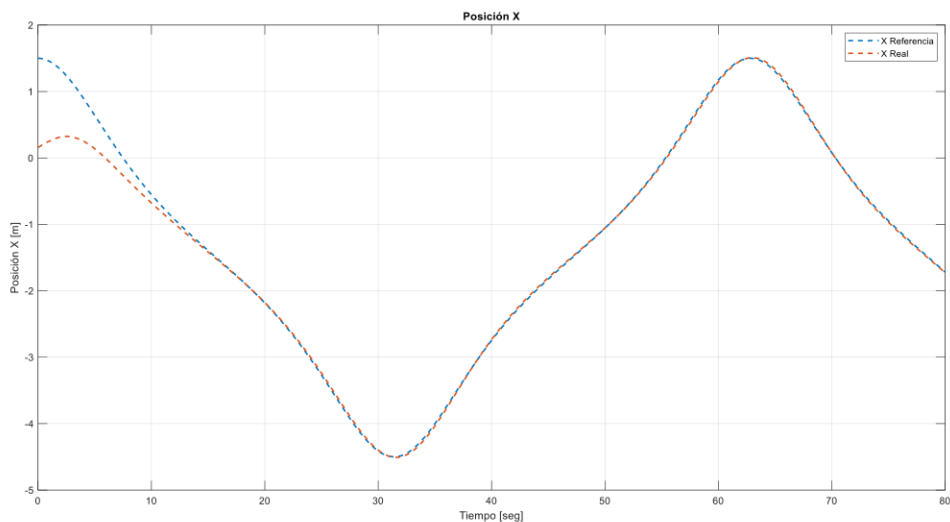


Figura 3.23 Resultado de las posiciones en X tanto de referencia como real (Trayectoria lemniscata)

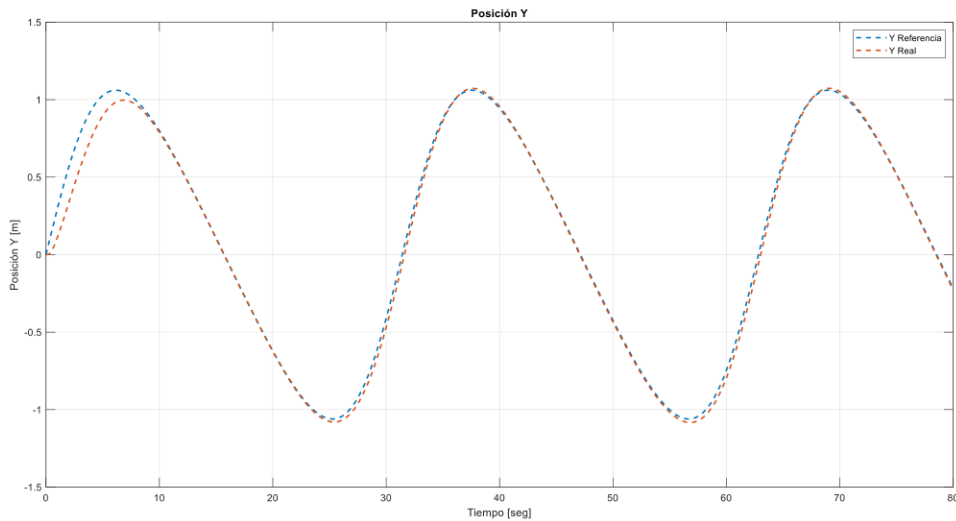


Figura 3.24 Resultado de las posiciones en Y tanto de referencia como real (Trayectoria lemniscata)

En las Figuras 3.23 y 3.24 tenemos los resultados de posiciones reales y posiciones de referencia en los ejes X y Y respectivamente, al igual que en las anteriores trayectorias se tiene un error de posición en el eje X debido al posicionamiento inicial del robot y el punto de inicio de la generación de la trayectoria de referencia, por otro lado, en las posiciones en Y no se tiene este inconveniente; en general los errores de posición en el estado estable son muy pequeños por lo que se tiene un seguimiento de trayectoria bastante bueno tanto en X como en Y.

3.4 Comparación entre el PID Dual no lineal y el PID clásico

Con el objetivo de comprobar el correcto funcionamiento del controlador diseñado se lo comparará con el controlador PID clásico mediante los índices de desempeño ISE.

En la Figura 3.25 se aprecia el resultado de cada uno de los controladores en el seguimiento de la trayectoria cuadrada, se observa que el PID clásico tiene un mayor error en cuanto al seguimiento de trayectoria en el estado estable mientras que el PID Dual no lineal tiene un comportamiento menos agresivo en cada una de las esquinas del cuadrado.

Por otro lado, en las Figura 3.26 y 3.27 podemos apreciar los índices ISE tanto de la velocidad lineal como de la velocidad angular respectivamente, analizando esta grafica podemos concluir que en cuanto al ISE de la velocidad lineal el controlador PID clásico tiene un valor menor por lo que su error es menor, mientras que en la velocidad angular el controlador PID Dual no Lineal tiene un menor valor de ISE y, por ende, un menor error en esta velocidad.

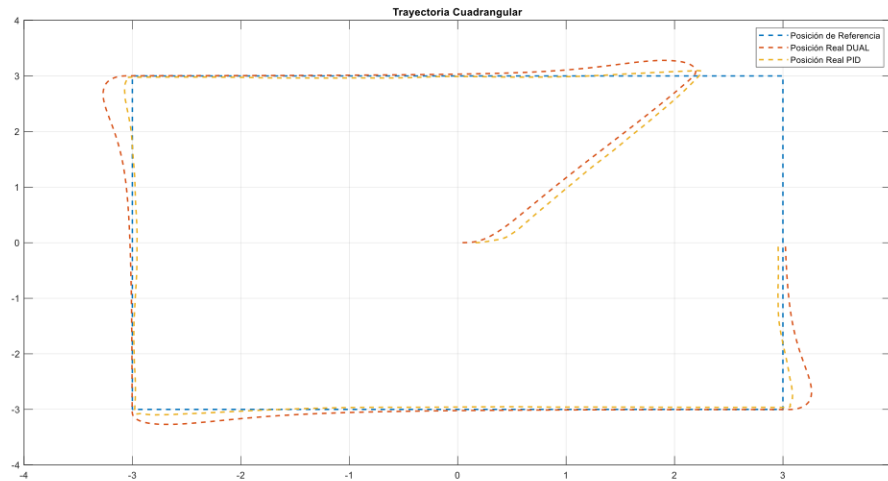


Figura 3.25 Resultado del seguimiento de trayectoria del Pioneer 3DX usando el controlador PID DUAL no Lineal y el controlador PID clásico (Trayectoria cuadrada)

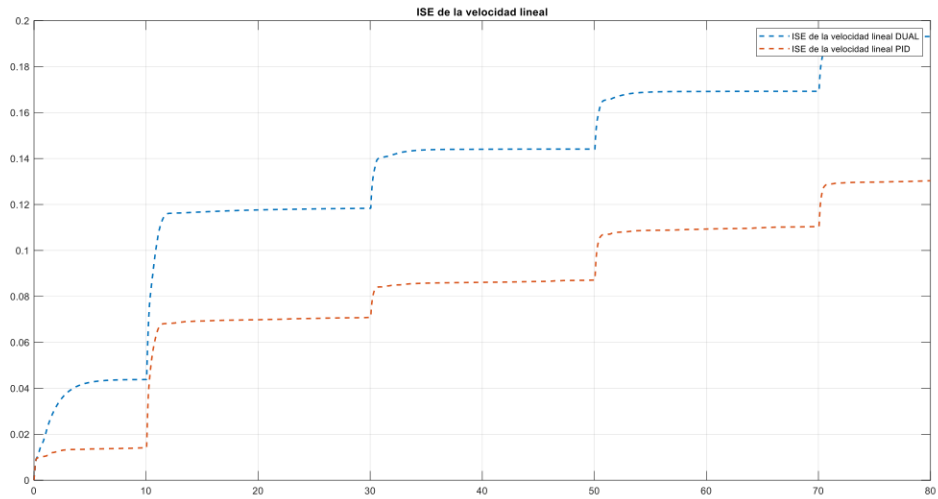


Figura 3.26 Índices ISE de la velocidad lineal (Trayectoria cuadrada)

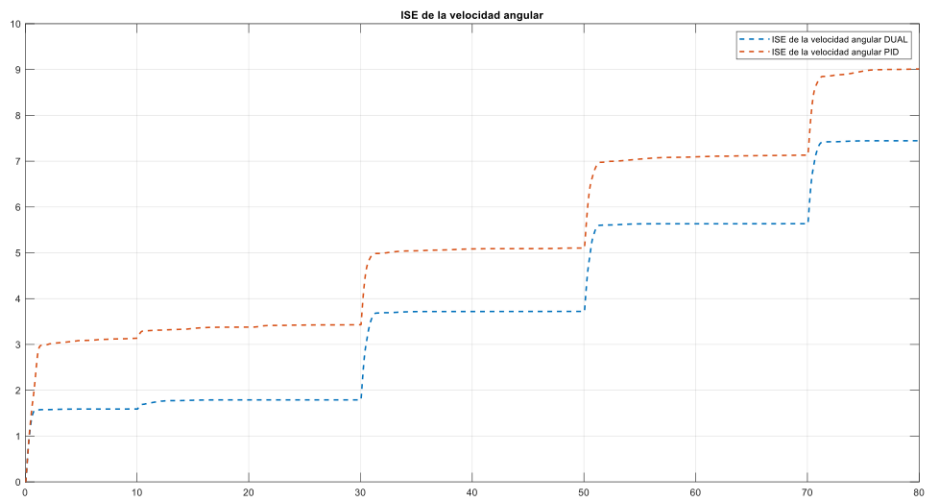


Figura 3.27 Índices ISE de la velocidad angular (Trayectoria cuadrada)

En la tabla 3.1 podemos analizar los índices de desempeño ISE de cada una de las trayectorias con cada uno de los controladores, PID Dual no lineal y PID clásico.

Tabla 3.1. Índices ISE de la velocidad lineal y angular de cada controlador para cada trayectoria simulada.

Parámetro	ISE DUAL NL PID	ISE PID
<i>Trayectoria cuadrada</i>		
Velocidad lineal	0.193	0.13
Velocidad angular	7.445	9.01
<i>Trayectoria circular</i>		
Velocidad lineal	0.0415	0.0218
Velocidad angular	1.516	3.15
<i>Trayectoria lemniscata de Bernoulli</i>		
Velocidad lineal	0.027	0.0175
Velocidad angular	1.417	2.397

Los resultados obtenidos en la tabla anterior permiten concluir que el controlador de velocidad lineal tiene un menor error con el controlador PID clásico, sin embargo, la diferencia con el controlador PID Dual no Lineal no es muy considerable, en cambio, si hablamos del ISE de la velocidad angular se puede concluir que el controlador PID Dual no Lineal tiene un mejor desempeño y por ende mucho menos error, esto se ve reflejado en la diferencia existente entre los índices ISE de cada controlador.

3.5 Resultados de la simulación del seguimiento de trayectorias en Coppelia Sim

En este apartado se presentan los resultados en cuanto al seguimiento de trayectoria que realiza el robot Pioneer 3DX en el software Coppelia Sim, se ha colocado una línea de color azul para una mejor visualización de la trayectoria que el robot genera durante el seguimiento de cada una de las referencias.

En la figura 3.28 se aprecia el resultado del seguimiento de la trayectoria cuadrada, este es el mismo resultado que se muestra en el apartado 3.1 debido a que estos datos de posición son los que se realimentan hacia Simulink.

Por otro lado, en las figuras 3.29 y 3.30 se pueden observar los resultados de las trayectorias circular y lemniscata de Bernoulli respectivamente, al igual que la grafica de la figura 3.28 estos corresponden a resultados que se analizaron en apartados anteriores ya que son los mismos datos de posición que se envían al software Simulink.

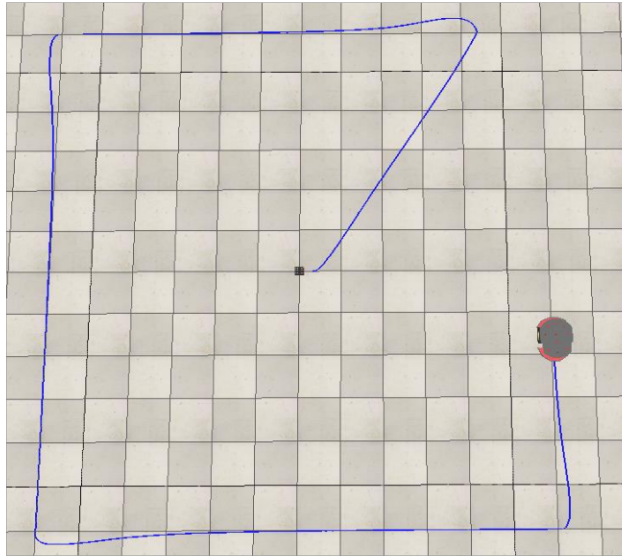


Figura 3.28 Trayectoria cuadrada real en Coppelia SIM

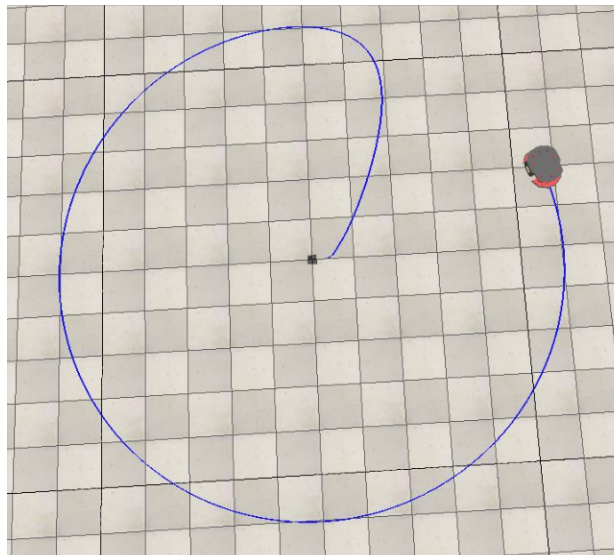


Figura 3.29 Trayectoria circular real en Coppelia SIM

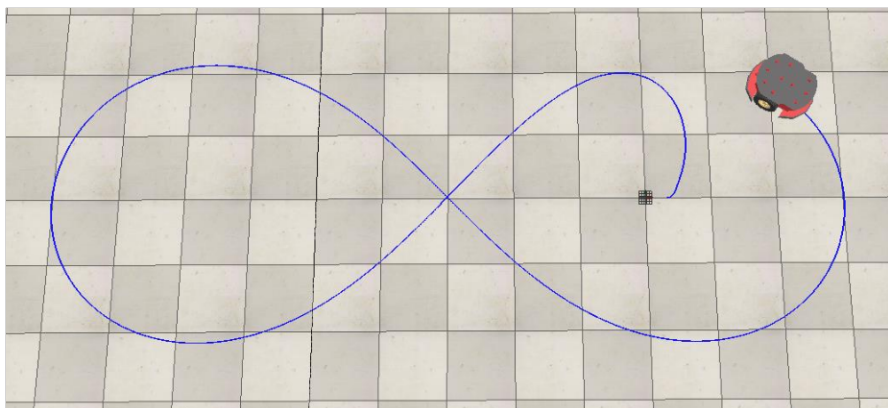


Figura 3.30 Trayectoria lemniscata de Bernoulli real en Coppelia SIM

3.6 Conclusiones

- Se investigó varios de esquemas de control para el seguimiento de trayectorias dentro del mundo de la robótica móvil haciendo énfasis en controladores basados en variaciones del controlador PID.
- Se diseñó el controlador propuesto basándose en la aproximación del modelo real del robot a uno de primer orden lo que permitió que la sintonización de los controladores tanto de la velocidad lineal como angular sean simples.
- Se implementó el esquema de comunicación entre los programas necesarios para la simulación del robot en cuanto a seguimiento de trayectorias de manera síncrona obteniendo un enlace estable y confiable entre estos.
- Se obtuvo resultados satisfactorios con la implementación del controlador diseñado en el seguimiento de trayectorias en comparación, mediante índices de desempeño (ISE), a controladores clásicos presentados trabajos previos.
- Se diseñó una interfaz gráfica que facilita la visualización de los resultados y que permite que el usuario elija tanto la trayectoria que desea simular, así como los diferentes parámetros a analizarse.

3.7 Recomendaciones

- Tener en cuenta los límites operativos del robot ya que si se realiza una implementación real este podría malograrse debido a acciones de control fuera de dicho límite.
- Existen varios métodos de sintonización de modelos de primer orden con retardo por lo que se debe analizar las respuestas que se obtienen con cada uno de estos y elegir el que tenga una similitud mayor al modelo real.
- El tiempo de muestreo con el que se realiza la simulación debe ser el adecuado ya que si se eligen tiempos demasiado pequeños esto representaría una carga computacional elevada y podría estar fuera del rango operativo del robot.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] L. E. Solaque Guzmán, M. A. Molina Villa, and E. L. Rodríguez Vásquez, "Seguimiento de trayectorias con un robot móvil de configuración diferencial," Ingenierías USBMed, pp. 26–34, 2014.
- [2] L. J. Capito Ruiz and P. A. Proaño Chamorro, "Seguimiento de Trayectorias mediante cuatro Técnicas de Control utilizando una Plataforma Robótica Pioneer 3DX y el Sistema Operativo Robótico ROS," Quito: Escuela Politécnica Nacional, 2015.
- [3] I. Bambino, "Una introducción a los robots móviles", 2008. [Online]. Available: http://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.
- [4] Revista de robots, "Robótica. ¿Qué es la robótica y para qué sirve?," diciembre 2021. [Online]. Available: <https://revistaderobots.com/robots-y-robotica/que-es-la-robotica/>.
- [5] EDS Robotics, "¿Qué es la robótica?," abril 2021. [Online]. Available: <https://www.edsrobotics.com/blog/que-es-la-robotica/>.
- [6] J. Pérez Turiel, J. C. Fraile Marinero and J. R. Perán Gonzales, "Aplicaciones de la robótica: últimas tendencias y nuevas perspectivas," DYNA, abril 2002, vol. 77, no. 3, pp.61-68.
- [7] V. R. Barrientos, J. R. García and R. Silva, "Robots Móviles: Evolución y Estado del Arte," Polibits, 2007, no. 35, pp. 12-17.
- [8] R. García and M. Arias Montiel, "Prototipo virtual de un robot móvil multi-terreno para aplicaciones de búsqueda y rescate," 2016.
- [9] J. A. Cook Meneses, "Diseño e implementación de un sistema de generación de trayectorias para un robot móvil utilizando control odométrico," 2012.
- [10] Adept, "Robot Pioneer 3DX", mobilerobots, 2011.
- [11] F. N. Martins, "Modelo dinámico y controlador de seguimiento para robots móviles tipo unicycle," V Jornadas Argentinas de Robótica-JAR08, 2008.
- [12] B. F. Sarabia Morales, "Diseño y simulación de tres técnicas de control clásicas y robustas aplicadas al seguimiento de trayectorias ante la presencia de retardos fijos para la plataforma robótica Pioneer 3DX," Tesis de Licenciatura, Quito 2017.
- [13] V. M. Alfaro, "Identificación de procesos sobre amortiguados utilizando técnicas de lazo abierto," Revista de la Universidad de Costa Rica, 2001, vol. 11, no. 1-2, pp. 11-25.

- [14] T. Kealy and A. O'dwyer, "Analytical ISE calculation and optimum control system design," 2003.
- [15] A. Yandún and N. Sotomayor. "Planeación Y Seguimiento De Trayectorias Para Un Robot Móvil," Revista politécnica, Quito, 2011.
- [16] O. González, J. Rosero and A. Rosales, "Control Predictivo Aplicado a un Robot Móvil," Revista Politécnica, 2014.
- [17] F. N. Martins, "Un controlador dinámico adaptable de seguimiento para robots móviles tipo unicycle," V Jornadas Argentinas de Robótica-JAR08, 2008.
- [18] L. F. Cerecero, "Comparación experimental de controladores PID clásico, PID no lineal y PID difuso para el caso de regulación," Res. Comput. Sci., 2017, vol. 135, pp. 173-187.
- [19] V. Mazzone, "Controladores pid," Quilmes: Universidad Nacional de Quilmes, 2002.
- [20] F. G. Shinskey, "Process control systems," McGraw-Hill, 1979.
- [21] C. Obando, D. Chávez, P. Leica and O. Camacho, "Sliding Mode Controller Based on a Hybrid Surface for Tracking Improvement of Non-Linear Processes," IFAC, Volume 53, Issue 2, 2020, Pages 11747-11752.
- [22] W. L. Torres, I. B. Q. Araujo, J. B. Menezes Filho and A. G. Costa Junior, "Mathematical Modeling and PID Controller Parameter Tuning in a Didactic Thermal Plant," in IEEE Latin America Transactions, vol. 15, no. 7, pp. 1250-1256, 2017.
- [23] Mathworks, "Simulink para el diseño basado en modelos," [Online]. Available: <https://la.mathworks.com/products/simulink.html>
- [24] Coppelia Robotics, "Robot Simulator CoppeliaSim," [Online]. Available: <https://coppeliarobotics.com/>
- [25] Mathworks, "MATLAB App Designer," [Online]. Available: <https://la.mathworks.com/products/matlab/app-designer.html>

5 ANEXOS

ANEXO I

El siguiente código es usado para el cálculo de los parámetros del modelo de primer orden con retardo de las velocidades tanto lineal como angular del robot, además de las diferentes constantes de los controladores PI y PID de cada una de las velocidades.

Cálculos para la velocidad lineal:

Modelo de primer orden con retardo (Método de Alfaro)

```
a=-0.91;
b=0.910;
c=1.262;
d=-0.262;
t1=0.154
t2=0.4191
Tao=a*t1+b*t2
tm=c*t1+d*t2
```

Controlador PI (Método de Chien-Hrones-Reswick)

```
K=0.5/0.5
Kp=(0.6*Tao)/(K*tm)
Ti=Tao
Ki=Kp/Ti
```

Controlador PID (Método de Chien-Hrones-Reswick)

```
Kc=(1/K)*(1.35*(Tao/tm)+0.25)
Ti=Tao*((2.5*(tm/Tao)+0.46*(tm/Tao)^2)/(1+0.61*(tm/Tao)))
Td=(0.37*tm)/(1+0.2*(tm/Tao))
Kd=Td*Kc
Ki=Kc/Ti
```

Cálculos para la velocidad angular:

Modelo de primer orden con retardo (Método de Alfaro)

```
a=-0.91;
b=0.910;
c=1.262;
d=-0.262;
t1=0.0828
t2=0.2162
Tao=a*t1+b*t2
tm=c*t1+d*t2
```

Controlador PI (Método de Chien-Hrones-Reswick)

```
K=0.871/0.8
Kp=(0.6*Tao)/(K*tm)
Ti=Tao
Ki=Kp/Ti
```

Controlador PID (Método de Chien-Hrones-Reswick)

```
Kc=(1/K)*(1.35*(Tao/tm)+0.25)
Ti=Tao*((2.5*(tm/Tao)+0.46*(tm/Tao)^2)/(1+0.61*(tm/Tao)))
Td=(0.37*tm)/(1+0.2*(tm/Tao))
Kd=Td*Kc
Ki=Kc/T
```

ANEXO II

En este anexo se describe una guía completa de la correcta utilización de la interfaz gráfica para la simulación de cada uno de los controladores propuestos en este trabajo

Para abrir la interfaz grafica debemos buscar el archivo “Ventana Principal” en la carpeta “Trabajo Final Entregable”, el tipo de archivo es “MATLAB App”, esto lo podemos apreciar de mejor manera en la figura 5.1.

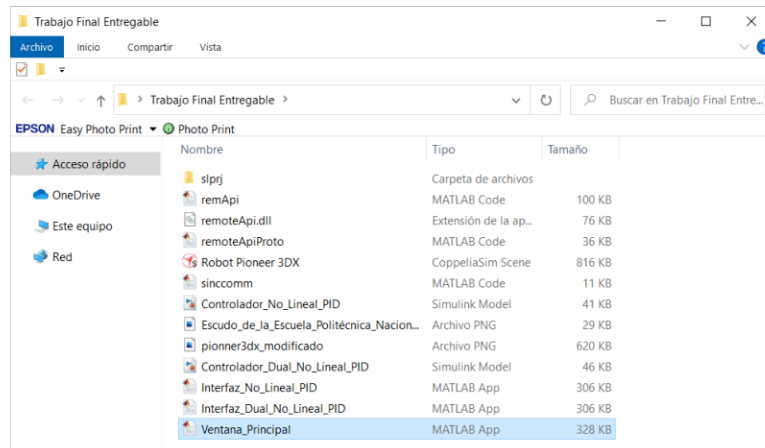


Figura 5.1 Archivo que abre la ventana principal de la interfaz gráfica

Una vez hayamos abierto este archivo se inicializará automáticamente el software Matlab y se mostrará la siguiente ventana: (Ver Figura 5.2)



Figura 5.2 Ventana principal de la interfaz gráfica

En esta ventana podemos seleccionar entre dos controladores, en este trabajo se trató el desarrollo del controlador PID Dual no Lineal por lo cual nos interesa observar los resultados que se obtienen de su simulación por lo que procedemos a seleccionar la opción “Controlador Dual No Lineal”, una vez demos click en esta opción se abrirá automáticamente el archivo de Simulink en donde se encuentra la generación de trayectorias y los respectivos controladores (ver figura 5.3), es necesario que se cambie el directorio de Matlab y se coloque en la dirección de la carpeta “Trabajo Final Entregable” para que se realice la simulación sin ningún problema (ver figura 5.4).

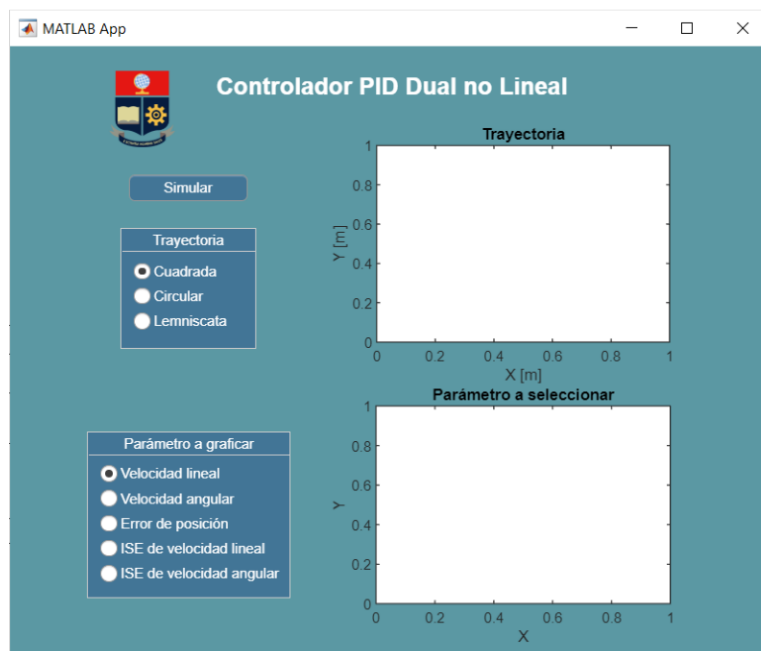


Figura 5.3 Ventana de presentación de resultados

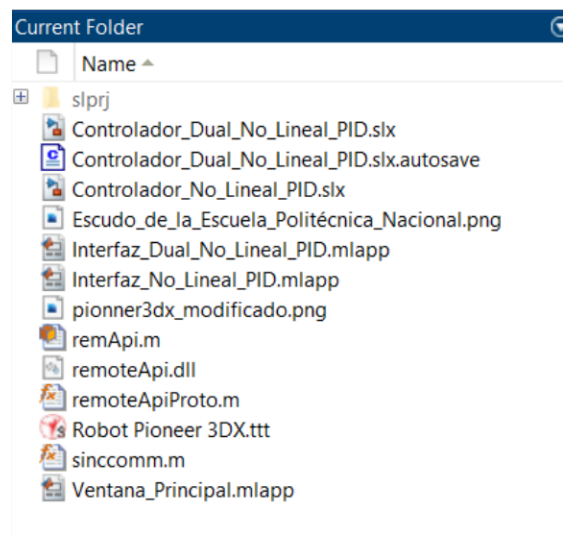


Figura 5.4 Cambio del directorio de Matlab para una correcta simulación

El siguiente paso es la apertura manual del archivo de Coppelia SIM, sabemos que este software contiene el modelo real del robot, el nombre de este archivo es “Robot Pioneer 3DX” (ver figura 5.5), este archivo es de tipo “CoppeliaSim Scene”, para abrir este archivo es necesario tener instalada la versión 4.0.0.

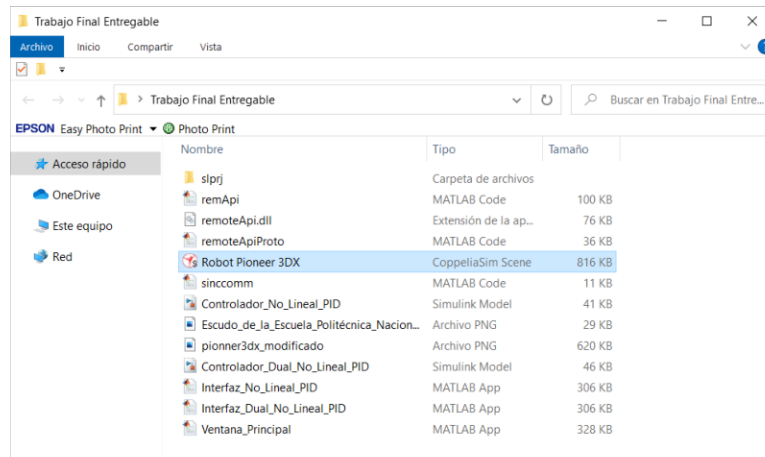


Figura 5.5 Archivo que abre el entorno en Coppelia SIM (Robot Pioneer 3DX)

Una vez tengamos abierto el archivo de Coppelia SIM ya podemos seleccionar la trayectoria que deseamos que se simule además del parámetro que se desea graficar en la ventana de la figura 5.3, para una pequeña demostración se elige la trayectoria Circular y como parámetro a graficar seleccionamos la velocidad angular, procedemos a dar click en el botón “Simular”, podemos observar en el software Coppelia Sim que el robot empieza a moverse, se ha añadido una línea que representa la trayectoria por la cual se movilizó el robot (ver figura 5.6).

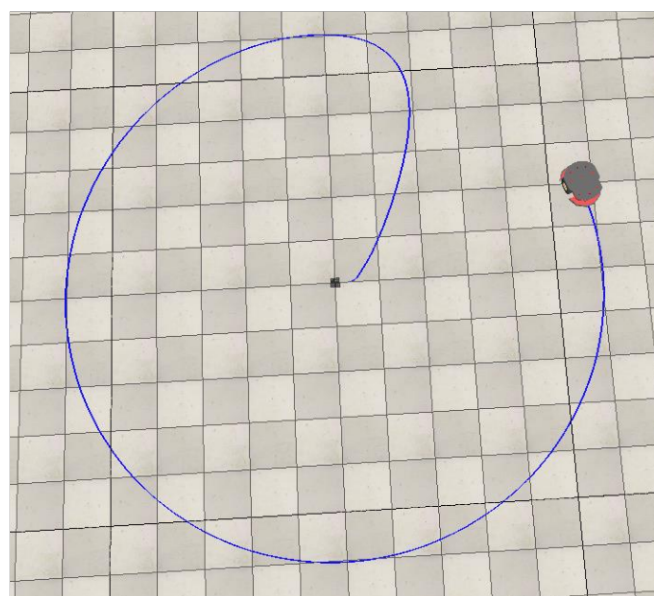


Figura 5.6 Trayectoria realizada por el robot en Coppelia Sim

Cuando haya finalizado la simulación se actualizan los gráficos de la interfaz y se pueden apreciar los resultados (ver figura 5.7), si se desea observar algún otro parámetro de la misma trayectoria basta con seleccionarlo y se actualizará la gráfica, pero si se desea observar otra trayectoria y los parámetros de esta es necesario que se vuelva a simular con la trayectoria deseada.

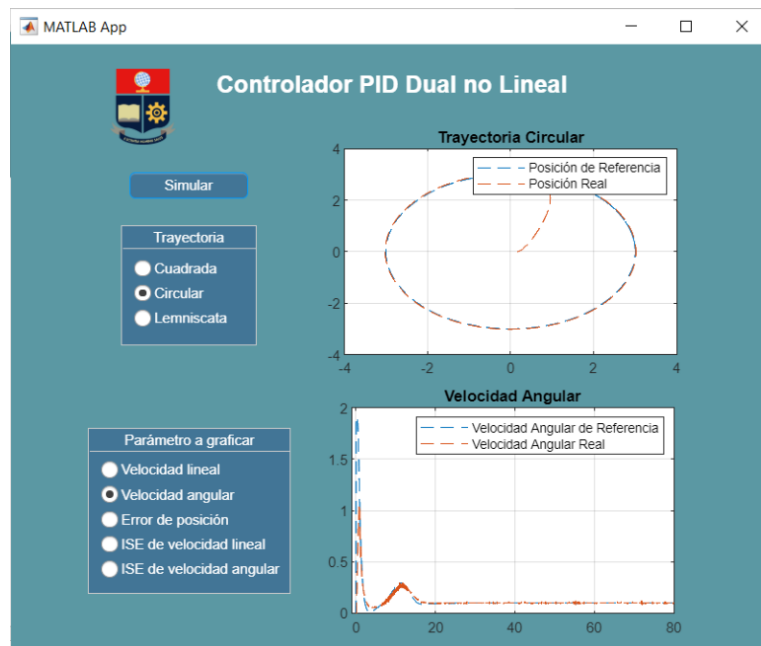


Figura 5.7 Resultados obtenidos para la simulación de la trayectoria circular