



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño, construcción y
control de sistema de plato y
bola mediante técnicas de
control lineal de orden
fraccionario**

TESIS

Que para obtener el título de
Ingeniero Eléctrico-Electrónico

P R E S E N T A

Enrique Pérez Martínez

DIRECTOR DE TESIS

Dr. Marcos Ángel González Olvera



Ciudad Universitaria, Cd. Mx., 2023

Índice general

0.1. Agradecimientos	4
1. Introducción	6
1.1. Objetivo	6
1.2. Motivación	6
1.3. Objetivos Principales	7
1.4. Contribuciones	7
2. Estado del Arte	8
2.1. Control PID	8
2.2. Cálculo de Orden Fraccionario	9
2.3. Control PID Fraccionario	9
2.4. Sistema de Plato y Bola	10
3. Descripción del sistema	11
3.1. Modelado	11
3.1.1. Variables de Estado	11
3.2. Modelado del sistema de Plato y Bola	12
3.2.1. Modelado por Ecuaciones de Euler-Lagrange	13
3.2.2. Modelo del sistema de Plato y Bola	13
3.3. Puntos de equilibrio	15
3.3.1. Puntos de equilibrio del sistema de Plato y Bola	16
3.4. Sistema Lineal	16
3.4.1. Linealización	16
3.4.2. Linealización del sistema de Plato y Bola	17
3.5. Transformada de Laplace	18
3.6. Función de Transferencia	19
3.7. Plano Complejo	20
3.8. Lugar Geométrico de las Raíces	20
3.9. Relación Función de Transferencia-Variables de Estado	21
3.10. Función de Transferencia del Sistema de Plato y Bola	21
3.11. Lugar Geométrico de las raíces del sistema de Plato y Bola	22
3.12. Respuesta en Frecuencia	23
3.13. Diagramas de Bode	23
3.13.1. Diagramas de Bode del Sistema de Plato y Bola	24
3.14. Estabilidad	24
3.14.1. Estabilidad en el Sentido de Lyapunov (ESL)	25
3.14.2. Método Directo de Lyapunov	25
3.14.3. ESL para Sistemas Lineales	26
3.14.4. Estabilidad del sistema de Plato y Bola	26
4. Plataforma Experimental	28
4.1. Diseño	28
4.2. Construcción	28
4.3. Software de Control	31
4.4. Diseño Electrónico	32

4.4.1. Arduino Mega 2560	32
4.5. Motores	34
4.5.1. Señal de Control	35
4.6. Sensor	36
4.6.1. Procesamiento de imagen para webcam	37
4.7. Planta Real	41
5. Diseño de Controladores Lineales	45
5.1. Controlabilidad	45
5.1.1. Controlabilidad del Sistema de Plato y Bola	45
5.2. Lazo de Control	46
5.3. Control PID	47
5.3.1. Control PD para el Sistema de Plato y Bola	48
5.3.2. Control PID para el Sistema de Plato y Bola	51
5.4. Cálculo Fraccionario	54
5.4.1. Definición de Grunwald-Letnikov	54
5.4.2. Definición de Riemann-Liouville	54
5.4.3. Definición de Caputo	55
5.5. Control PID Fraccionario	55
5.5.1. Control PD Fraccionario para el Sistema de Plato y Bola	56
5.5.2. Control PID Fraccionario para Sistema de Plato y Bola	58
6. Control en Tiempo Discreto	60
6.1. Transformada Z	61
6.2. Correspondencia entre el Plano s y z	62
6.3. Lugar Geométrico de las Raíces en Tiempo Discreto	62
6.3.1. Retén de Orden Cero	62
6.3.2. Lugar Geométrico de las Raíces en Tiempo Discreto del Sistema de Plato y Bola	63
6.4. Aproximación Rectangular o de Euler	64
6.4.1. Control PD con Aproximación Rectangular	65
6.4.2. Control PD con Aproximación Rectangular para el Sistema de Plato y Bola	66
6.4.3. Control PID con Aproximación Rectangular	68
6.4.4. Control PID con Aproximación Rectangular para el Sistema de Plato y Bola	68
6.5. Aproximación Trapezoidal o de Tustin	70
6.5.1. Control PD con Aproximación de Tustin	71
6.5.2. Control PD con Aproximación de Tustin para el Sistema de Plato y Bola	72
6.5.3. Control PID con Aproximación de Tustin	73
6.5.4. Control PID con Aproximación de Tustin para el Sistema de Plato y Bola	74
6.6. Aproximación de Controladores de Orden Fraccionario	76
6.6.1. Control PD Fraccionario con Aproximación de Oustaloup	77
6.6.2. Control PD Fraccionario Discretizado para el Sistema de Plato y Bola	78
6.6.3. Control PID Fraccionario con Aproximación de Oustaloup	80
6.6.4. Control PID Fraccionario Discretizado para el Sistema de Plato y Bola	81
7. Implementación de Controladores al Sistema de Plato y Bola	83
7.1. Código en MatLAB	84
7.2. Control PD con Aproximación Rectangular Implementado	89
7.3. Control PID con Aproximación Rectangular Implementado	95
7.4. Control PD con Aproximación de Tustin Implementado	102
7.5. Control PID con Aproximación de Tustin Implementado	104
7.6. Control PD Fraccionario Implementado	104
7.7. Control PID Fraccionario Implementado	111
7.8. Comparación de controladores PD y PD fraccionario	111

8. Conclusiones y Bibliografía	114
8.1. Conclusiones	114
8.2. Bibliografía	116
Referencias	116

0.1. Agradecimientos

La tesis además de la intención de aportar algo a la teoría de control PID, va también dedicada a todas las personas e instituciones que me han apoyado día a día a lo largo de la carrera y que sin ellos no sería posible la elaboración de esta tesis. Va dedicado principalmente y en forma de agradecimiento a mis padres, Natalia Martínez Casiano y el Ing. Enrique Pérez Caballero. La tesis aquí presentada es un reflejo del apoyo y dedicación de ambos durante mi formación como persona y durante la carrera para mi formación profesional, el apoyo emocional por seguir adelante, además del apoyo económico a lo largo de la carrera por lo que ese esfuerzo también se ve reflejado en esta tesis y por todo esto solo me queda decir gracias a ambos.

A mi hermano Marco Antonio Pérez Martínez quien también influyó a lo largo de mi carrera, siempre he tenido la motivación de seguir adelante para poderle dar un buen ejemplo como hermano mayor, para convertirme en un modelo a seguir, tanto como persona y como profesional. Esto hace que trate siempre de mejorar como persona, esperando que el día de mañana el cumpla sus sueños y sus metas, para que se convierta en una persona mucho mejor que su hermano. En general a toda mi familia les agradezco por motivarme a seguir adelante, en especial a mi tía Alejandra, a todos mis primos, en particular al Ing. Abraham Cruz, José Luis Flores, Guillermo Guerrero, así como mis tíos y primos que poco o mucho han aportado a mi formación y en general a toda mi familia le agradezco por el apoyo a lo largo de la carrera.

Después de mi familia a quien más tengo que agradecer es a la Universidad Nacional Autónoma de México, con mención especial a la Facultad de Ingeniería. Es un hecho que sin la UNAM jamás tendría la oportunidad de adquirir todos los conocimientos que aprendí durante toda la carrera, pero más importante, la ética profesional que he adquirido y no solo las bases técnicas de un ingeniero, poder adquirir esa mentalidad y los valores que motiva la misma universidad para ser mejor persona y aportar algo a la sociedad como profesional. El hecho de tener los salones, las diferentes bibliotecas, laboratorios y todos los espacios para estudiar; al igual que las actividades culturales y deportivas que implementa la UNAM, creo que aportan a los estudiantes para el desarrollo de ingenieros y profesionistas en general, que compitan con cualquier otro, siempre con la motivación a cada estudiante para continuar día a día en la carrera así como, a ampliar sus conocimientos en la rama en que se desempeñan y de cualquier rama de interés.

En mi opinión lo mejor que tiene la UNAM son los profesores, con sus excepciones, pero al menos muchos de los profesores que llevé a lo largo de la carrera son de lo mejor que hay, no hay duda que la formación que da cada profesor que tuve durante toda la carrera, motiva a los estudiantes a ser ingenieros como ellos y en cierta forma, su esfuerzo también se ve reflejado aquí y por ello debo agradecerles a todos. Debo dar la gracias a muchos profesores, por mencionar a algunos al Ing. Yukihiro, Ing. Ocariz, Fis. Pedro Manny, al Ing. Juan Aguilar Pascual, que en paz descanse, Ing. Rolando Peralta, Ing. Benjamín Ramírez, Dr. Gerardo Espinoza, Mtro. Ing. Lauro, Ing. Mandujano, Mtro. Ing. Enrique Díaz Cerón profesor de ética, Mtro. Ing. Mauro López, Dr. Saúl Nieves, Mtro. Ing. José Castillo, Dr. Hoover Mujica, Mtra. Ing. Gloria Correa, Dr. Paul Maya, Mtro. Ing. Salva Calleja, entre muchos otros profesores que día a día se esfuerzan en que sus alumnos aprendan y se conviertan en mejores ingenieros.

Entre tantos profesores debo dar un agradecimiento especial al Dr. Miguel Ángel Bañuelos Saucedo por la oportunidad de realizar mi servicio social en el Instituto de Ciencias Aplicadas y Tecnología (ICAT), al igual que a todos los integrantes del laboratorio de instrumentación científica en el área de electrónica, el Mtro. Ing. José Castillo, Mtro. Ing. Ricardo Zamacona y Mtro. Ing. Sergio Thierry, gracias a todos por lo que aprendí durante mi servicio social y permitirme usar su laboratorio durante mi servicio social y gracias al Dr. Bañuelos por esa oportunidad y por todo lo que me enseñó a lo largo del servicio en el ICAT.

El profesor que creo merece más mi agradecimiento es al Dr. Marcos Ángel González Olvera por permitirme elaborar esta tesis bajo su asesoramiento, al igual que permitirme usar por poco tiempo su laboratorio en la UACM y sobre todo darme un poco más de tiempo al realizar esta tesis. También le agradezco al Dr. Marcos Ángel por enseñarme lo que es un ingeniero de control, mucho de lo que aprendí en esta área fue gracias a él y esto cambio por completo mi idea de lo que es un ingeniero de control. Le agradezco por siempre resolver mis dudas durante su clase y sobre todo su disposición y tiempo hacia mí, eso genero la motivación para elaborar esta tesis con usted como asesor y por eso le doy las gracias.

La UNAM no solo se define por los salones y profesores, un elemento fundamental es la gente que estudia ahí, todas las personas que conocí durante la carrera, aportaron algo a la persona que soy hoy en día, a mis amigos de la carrera de ingeniería eléctrica-electrónica que a lo largo de la carrera nos apoyamos y motivamos para seguir adelante en este camino y es un hecho que no pude acabar la carrera sin su ayuda y a todos ellos les agradezco a Abraham Rangel, Rommel García, Yael Gómez, Erick Vázquez, Ernesto Méndez, Gustavo Chávez, Alexis Vilchis y Jonathan Alcalde. Entre muchos otros futuros ingenieros de otras carreras que conocí a lo largo de la carrera como en computación a Jesús Martínez, Daniel Saturnino e Inri Jiménez, de mecatrónica a Hugo Fernández y por supuesto, a mis amigos de ingeniería civil a Luis Estévez, Ricardo López y Eduard García que también aprendí mucho con ustedes. A lo largo de la carrera entendí que se puede aprender mucho de otras carreras, sin duda, no todo el conocimiento se encuentra en ingeniería y conocer de otras carreras también aporta a uno como persona, por ello le agradezco a Alex Lozano, a Abigail y Susana, por ayudarme a entender esto. Sé que no puedo mencionar a todos, pero a todas las personas que aportaron algo mi formación académica y no mencione, les debo una disculpa, puesto que es difícil recordar a todos y aunque no están mencionados de igual forma les agradezco a todos.

Capítulo 1

Introducción

1.1. Objetivo

El objetivo de esta tesis es aplicar diferentes técnicas de control lineal a un sistema físico real, con la finalidad de observar y comparar las respuestas del sistema a estas técnicas de control. Obtener un análisis completo desde el punto de vista matemático de las teorías de control y del sistema en sí, que para esta tesis es el sistema electromecánico de plato y bola, a través de los modelos matemáticos para obtener toda la información posible y poder generar las simulaciones necesarias para después implementar las teorías de control al sistema real y poder comparar estas teorías de control con el objetivo de obtener una conclusión para este sistema.

La tesis desarrolla paso a paso todos los elementos que implica aplicar las técnicas de control a un sistema como este, desde el diseño hasta la construcción del sistema físico, la instrumentación del mismo, es decir, los sensores que se requieren para obtener información del sistema, así como generar el movimiento de la bola en el plato con los actuadores que sean capaces de aplicar físicamente cualquier señal de control y por último, realizar la implementación de las diferentes técnicas de control para obtener una respuesta del sistema tanto de forma simulada como de forma experimental que dependerá de la técnica de control lineal implementada. Una de las técnicas de control a utilizar es tal vez, la teoría más usada para sistemas físicos en cualquier rama de control y es el control PID. La otra técnica de control surge a partir de la teoría del cálculo fraccionario que en conjunto con la teoría del control PID se genera un PID de orden fraccionario. El objetivo también es mostrar una comparación entre estas técnicas y al final obtener una conclusión de los resultados obtenidos y poder aportar una pequeña parte al conocimiento de esta teoría de control lineal.

1.2. Motivación

El desarrollo de la tesis partió con la finalidad de aportar un poco al conocimiento en la teoría del control PID, con la adición de la teoría del cálculo de orden fraccionario, una teoría no muy común y generar un PID con la opción de una integral y derivada de orden fraccionario haciendo que el controlador tenga más grados de libertad y se ajuste mejor al sistema real, un concepto que no es nuevo, pero no es muy común verlo en la práctica. Al ver la importancia del control PID que, hoy en día sigue siendo fundamental en cualquier rama de control, principalmente por su facilidad de implementar a cualquier sistema y que a pesar de que existen muchas otras técnicas de control, un PID sigue resolviendo muchos problemas de control y por el otro lado está la teoría del cálculo fraccionario, una teoría no muy común que complementa la teoría del control PID para poder generar una herramienta más y aportar una solución diferente a problemas de control que tal vez un PID normal no pueda resolver.

Para cualquier sistema resulta interesante comparar un PID normal con uno fraccionario y ver cómo responden al ser aplicados a diferentes sistemas físicos, sin embargo, para esta tesis solo se utilizó un solo sistema físico que es el sistema de plato y bola en el que se probaron ambas teorías de control.

El sistema de plato y bola es un sistema electromecánico que consiste en estabilizar una bola en un punto de equilibrio del plato, por lo que el control implementado deberá ser capaz de soportar perturbaciones y mantener la posición de la bola en el plato el tiempo que se requiera, así como tener la capacidad de generar trayectorias de la bola en el plato. Durante el desarrollo de la tesis y después de modelar el sistema a controlar, resultó ser un sistema que no es tan fácil de controlar además de ser inestable, también que depende considerablemente de los parámetros en su diseño físico, sin embargo, es controlable; además de que al tener esa opción de diseñar el propio sistema se puede aprender mucho del propio sistema y mejorar continuamente el diseño para construir un modelo lo suficientemente robusto para implementar cualquier técnica de control y esto hace aún más interesante ver la comparación entre las técnicas de control en un sistema tan particular como este pero la motivación principal es tratar de mostrar una herramienta más para resolver problemas de control.

1.3. Objetivos Principales

De manera puntual los objetivos a desarrollar en esta tesis son los siguientes:

- Diseñar y construir el modelo físico del plato y bola.
- Instrumentar el sistema con base en las variables físicas a obtener, que en este caso es la posición de la bola en el plato.
- Proveer al sistema físico los actuadores necesarios para poder utilizar cualquier técnica de control y permita mover el sistema libremente.
- Generar el modelo matemático del sistema de plato y bola, así como el análisis del sistema, desde el punto de vista matemático, sus variables de estado, puntos de equilibrio y estabilidad.
- Obtener al menos dos técnicas de control lineal que efectivamente estabilicen el sistema del plato y bola en un punto de equilibrio.
- Analizar técnicas de control lineal basadas en el control lineal PID y en control lineal de orden fraccionario PID fraccionario.
- Obtener el desempeño del sistema y los controladores mediante diferentes simulaciones de cada control.
- Implementar físicamente los controladores de forma experimental en el sistema del plato y bola.

1.4. Contribuciones

A continuación, se presentan las contribuciones de esta tesis:

- El diseño y la elaboración del sistema electromecánico de plato y bola con una construcción robusta para implementar cualquier tipo de control.
- La adecuación del sistema para integrar un sensor por webcam y la elaboración del algoritmo para obtener la posición exacta de la bola en el plato.
- La elaboración de una aproximación polinomial para integrar cualquier señal de control y se adapte a la señal eléctrica que responden los actuadores instalados.
- La implementación de controladores PID con diferentes aproximaciones en tiempo discreto y el análisis del rendimiento por controlador del sistema de plato y bola.
- La implementación de manera sistemática de controladores PID fraccionarios y su aproximación en tiempo discreto al igual que el análisis en el rendimiento del controlador y del sistema de plato y bola.
- Análisis de la comparación entre controladores PID y PID de orden fraccionario para llegar a una conclusión en cuando al rendimiento y pros y contras de cada controlador.

Capítulo 2

Estado del Arte

La tesis aborda principalmente la teoría del control PID y el PID fraccionario implementados en el sistema de plato y bola. A continuación, se muestra un poco de los avances y de lo que es hoy en día el control PID y PID fraccionario en el área de control en general; al igual que las principales formas en que el sistema de plato y bola se utiliza, que en su mayoría es para la investigación en diversas teorías de control.

2.1. Control PID

El control PID se creó alrededor del siglo XX, específicamente en 1911, aunque ya desde antes existía el control proporcional (P), el integral (I) y el derivativo (D) por separado para controlar la velocidad de conducción de las máquinas de vapor. Fue Elmer Sperry quien desarrolló el control PID para la armada de Estados Unidos con la finalidad de automatizar la dirección de los barcos como si fuera un timonel capaz de compensar las variaciones en el mar. Hoy en día el control PID resuelve los problemas de control en un noventa por ciento de todos los sistemas dinámicos, al menos en la rama industrial y esto se debe a la relativa facilidad de implementar el control PID a cualquier microcontrolador; hoy en día, es raro ver un controlador industrial sin un PID integrado en su propia programación. El control PID es muy útil para problemas de regulación de cualquier variable física, sea temperatura, presión, volumen, etc. Prácticamente en cualquier lazo cerrado, es común el uso del PID para controlar procesos industriales. Muchos controladores lógico-programables como los PLC ya traen un bloque de control PID ya desarrollado, basta con colocar el setpoint o la referencia, el sensor de la variable física a controlar, así como el actuador que se utilice y se sintoniza en función a los requerimientos del sistema. Por esta razón el control PID es relativamente amigable para implementar en procesos con lazo de control simples, el detalle del control PID es la sintonización de sus ganancias K_p , K_i y K_d , que más adelante se abordará la teoría matemática de estas ganancias, pero en muchos casos la sintonización se hace in situ ya con el sistema listo para controlar y una vez que se aplica el control PID, se va sintonizando en función a los requerimientos del proceso. (Ogata, 1996).

La teoría del control PID sigue teniendo avances en la rama de la investigación, el hecho de tener tanto tiempo de que se creó no implica que se haya quedado solo en la teoría básica del PID, se han generado muchas teorías de sintonización con una metodología y una base matemática para obtener las mejores ganancias en función al rendimiento del sistema. Hoy en día, hay controladores PID que resuelven el detalle de la sintonización teniendo auto-sintonización o auto-tuning, que se han implementado en sistemas reales, tanto para la investigación como para controladores industriales. También se sigue probando el control PID en sistemas dinámicos como sistemas neumáticos, hidráulicos, mecánicos, electromecánicos y en casos muy particulares vehículos aéreos, tanto para control por regulación como control por seguimiento, en trayectorias de brazos de robots industriales y también, hay investigación en sistemas de redes neuronales con controladores PID, entre muchas otras investigaciones que siguen ampliando la teoría del control PID. El planteamiento de esta tesis es probar el control PID en el sistema de plato y bola y comparar, tanto el PID tradicional como el PID fraccionario para seguir ampliando esta teoría del control PID.

2.2. Cálculo de Orden Fraccionario

En matemáticas, física, química e ingeniería es común encontrar la notación del operador diferencial como $\frac{d}{dx}$ y $\frac{d^2}{dx^2}$ en el cálculo diferencial básico, pero porque no puede ser un número no entero. Esta incógnita surgió a partir de una carta a L'Hôpital en 1695 de Leibniz; donde Leibniz le pregunta si puede generalizarse el significado de las derivadas con orden entero y no entero, esto hizo que L'Hôpital sintiera cierta curiosidad y le respondiera con la siguiente pregunta: "¿Qué pasaría si el orden fuera $1/2$?", Leibniz en una carta del 30 de septiembre de 1695 le respondió: "llevará a una paradoja, de la que un día se obtendrán consecuencias útiles". Esta pregunta planteada por Leibniz fue un tema de discusión por más de 300 años y lo que hoy en día se conoce como cálculo fraccionario es una generalización del cálculo ordinario, tanto diferenciación como integración con orden arbitrario no entero. (Monje, Chen, Vinagre, Xue, y Feliu-Batlle, 2010).

Con esta declaración de Leibniz se creía que el estudio del cálculo fraccionario era reservado para las mejores mentes en la rama matemática, sin ninguna aplicación real por el momento; Fourier, Euler, Heaviside, Goldman, Laplace, entre muchos otros que incursionaron en el cálculo fraccionario han aportado algo a esta teoría; por ejemplo, Laplace definió una derivada fraccionaria utilizando una integral y la variable s . Fue en 1832 cuando el primer estudio sistemático del cálculo fraccionario por J. Liouville, quien quedó fascinado por los comentarios entre Laplace y Fourier acerca del cálculo fraccionario. Después, en 1990 M. Axtell discutió la aplicación del cálculo fraccionario a sistemas de control utilizando transformadas de Laplace por primera vez. El hecho de tener muchas teorías válidas que se han generado a lo largo de la historia del cálculo fraccionario, a ocasionado que existan muchas definiciones del cálculo fraccionario y todas son válidas, sin embargo, para la implementación del cálculo fraccionario en sistema de control destacan tres, la definición de Grünwald-Letnikov, la definición de Riemann-Liouville y la definición de Caputo. Más adelante se abordará la teoría matemática del cálculo fraccionario de forma muy superficial y con énfasis en las tres teorías que más se aplican para el desarrollo de esta tesis y sobre todo para utilizar estas teorías del cálculo fraccionario y juntarlo con la teoría del PID para obtener un control PID fraccionario. (Shah y Agashe, 2016).

2.3. Control PID Fraccionario

La teoría del cálculo fraccionario se puede utilizar para complementar la teoría del control PID y obtener un PID de orden fraccionario, más adelante se abordará la teoría y el análisis de forma matemática, pero en pocas palabras lo que hace un PID fraccionario es añadir dos grados de libertad, adicional a las ganancias K_p , K_i y K_d . Agregado el orden fraccionario de la derivada, que se define por μ y el orden fraccionario de la integral λ se espera tener un control que se adapte mejor al sistema y que mejore el desempeño tanto del controlador como del sistema. La desventaja de un controlador PID fraccionario es que, en vez de tratar de sintonizar 3 ganancias, se tendrían que sintonizar 5 ganancias y esto ocasiona que sea más complicado la sintonización.

Fue en 1993 que a través de las investigaciones de Alain Oustaloup con su controlador fraccionario CRONE, así como de Igor Podlubny en 1994, quien fue el primero en presentar el PID fraccionario de esta forma: $PI^\lambda D^\mu$ y de la definición de Grünwald-Letnikov que se colocaron las bases para la teoría del control PID fraccionario y que se sigue ampliando esta teoría. (Zamojski, 2018). Es interesante ver que, en un principio, no había mucho interés por los controladores de orden fraccionario; principalmente por la complejidad computacional de los algoritmos de cálculo de orden fraccionario y de las capacidades de las computadoras de ese entonces. Fue a través de los avances en la tecnología informática que permitieron desarrollar mejor los controladores de orden fraccionario y sobre todo con la ventaja de que estos algoritmos ya se podían integrar en cualquier controlador, tanto en la rama de la investigación, como en el ámbito industrial. Desarrollando estos métodos y algoritmos matemáticos, hoy en día se implementan controladores PID de orden fraccionario en sistemas de todo tipo, similar al control PID tradicional o en algunos casos para comparar su respuesta con un PID tradicional y ver si el rendimiento tanto del sistema, como del controlador mejoran, como es el caso de esta tesis. Además de que ya se cuenta con la ventaja de que es posible sintonizar tanto las ganancias K_p, K_i, K_d y añadiendo μ y λ de forma relativamente sencilla en softwares como MATLAB y Simulink para el controlador fraccionario en sistema electromecánicos como en el sistema de plato y bola. (Zamojski, 2018).

2.4. Sistema de Plato y Bola

El sistema de plato y bola o ball and plate, es un sistema electromecánico que comúnmente, se utiliza para probar diferentes teorías de control, principalmente en la rama de la investigación y para propósitos educativos en el área de control. El objetivo de este sistema es controlar el movimiento de la bola en una superficie plana como un plato, ya sea para ubicar la bola en algún punto deseado, como en el centro del plato, por ejemplo y después aplicar perturbaciones al sistema, con la idea de mantener la bola en el punto deseado lo mejor posible. Otra forma de utilizar este sistema, así como probar diferentes teorías de control, es a través de generar trayectorias con la bola formando alguna figura en el plato.

El sistema de plato y bola se cree que se origino en 1990, en el laboratorio Rockwell de la universidad de Checoslovaquia, siendo una extensión del sistema tradicional de bola y viga (en el caso del sistema de bola y viga es controlando la posición de la bola en un solo eje); particularmente que con la configuración del plato y bola, se añade más grados de libertad y se obtiene un sistema con múltiples entradas y múltiples salidas (MIMO), teniendo una situación muy interesante al observar los efectos del acoplamiento cruzado y además, desde el punto de vista de su modelo matemático, un sistema lineal e inestable en lazo abierto, convierten al sistema de plato y bola un desafío para controlar. A pesar de ser un sistema difícil de controlar, hoy en día se implementan controladores a este sistema con teorías de control óptimo, modos deslizantes, controladores no lineales, entre muchas otras teorías. (Debono y Bugeja, 2015).

Así como se han implementado muchas teorías de control en el sistema de plato y bola, también se tienen muchos diseños para este sistema, hay muchas formas de construir el sistema de plato y bola y cada uno depende de las características del control a implementar, pero sobre todo de la instrumentación, tanto para medir la posición de la bola en el plato, como generar el movimiento del mismo. En el caso de esta tesis se consideró el diseño para poder soportar cualquier teoría de control, se construyó con la idea de hacerlo lo más robusto posible y que fuera capaz de integrar cualquier controlador y cualquier teoría de control.

Capítulo 3

Descripción del sistema

3.1. Modelado

Teóricamente cualquier sistema físico se puede representar con un modelo matemático, comúnmente se utiliza el modelo matemático para entender y representar la dinámica de un sistema a través del tiempo. Utilizando técnicas matemáticas se puede conocer ciertas características de dicho sistema, estas características nos ayudan al momento de aplicar alguna técnica de control al sistema e incluso para saber si acaso es posible implementar un control al mismo. Mientras más detallado sea su modelo matemático de cualquier sistema, mejor representa su dinámica, sin embargo, al hacerlo más detallado, se hace cada vez más complicado de analizar, desde el punto de vista matemático. Debido a este problema, en la mayoría de los casos en que se requiere obtener un modelo matemático de un sistema, se usa modelos más simplificados que se puedan utilizar y analizar mucho más fácil y que, aun así, representen la mayor parte de la dinámica del sistema.

El modelo matemático surge a partir de los elementos que forman dicho sistema; los sistemas reales se pueden separar dependiendo de la naturaleza del mismo, ya sea un sistema mecánico, eléctrico, hidráulico, térmico o híbrido, para cualquier caso se utilizan las leyes de física y química que rigen la dinámica de dicho sistema, por ejemplo, en sistemas mecánicos, las leyes de Newton ayudan a desarrollar un modelo matemático que represente la dinámica del sistema. Una vez que se identifican las leyes que rigen el sistema, se define una señal de entrada al sistema y de igual forma, se debe definir una señal de salida; los elementos que están dentro del sistema responderán a través de su dinámica a la entrada definida para obtener una salida también definida. Para modelar algunos sistemas, en algunos casos se requiere información adicional del sistema, por ejemplo, en el caso de sistemas causales, se debe conocer información del sistema en el pasado y en el presente.

De esta forma se obtiene una expresión matemática que representa la dinámica de un sistema; en muchas ocasiones, por lo mismo que se utilizan las leyes de la física y química que rigen al sistema para generar el modelo, se obtiene un modelo no lineal que, aunque representa mejor la dinámica de dicho sistema, en la mayoría de los casos se complica mucho el análisis del modelo por estas no linealidades. Para solucionar esto, existen formas de modelar a un sistema no lineal y poder obtener información del modelo, con el objetivo de poder controlar el sistema más adelante, uno de los modelos más conocidos es el modelado en variables de estado.

3.1.1. Variables de Estado

Una forma de obtener el modelo matemático para sistemas físicos es a través de su modelo en variables de estado; este tipo de modelo permite que con la mínima cantidad de información del sistema y al aplicar una entrada, se pueda encontrar la dinámica del sistema en cualquier instante de tiempo para obtener una respuesta en su modelo matemático. La entrada como la salida del sistema la define quien genera el modelo de dicho sistema; además, el modelado en variables de estado permite un mejor análisis matemático de sistemas no lineales y con múltiples entradas y salidas. El modelado en variables de estado genera un número finito de ecuaciones diferenciales ordinarias de primer orden que dependen de las variables de estado, definidas por el que genera el modelo, todas las entradas al sistema y el tiempo t . (Hendricks, Jannerup, y Sørensen, 2008).

La estructura básica de un sistema en variables de estado se representa de la siguiente forma:

$$\dot{x}(t) = f(x(t), u(t), t) \quad (3.1)$$

$$y(t) = h(x(t), u(t), t) \quad (3.2)$$

Donde $\dot{x}(t)$ es un vector que representa la primer derivada de las variables de estado dadas por el vector $x(t)$, a este vector se le llama vector de variables de estado, donde $x \in R^{n \times 1}$; así como del vector $u(t)$, que se le llama vector de entradas, donde $u \in R^{m \times 1}$ y del tiempo t ya sea de forma implícita o explícita. A las funciones algebraicas del vector f se le llaman ecuaciones de estado, al vector $y(t)$ que de igual manera, depende de los vectores de variables de estado, de las entradas al sistema y del tiempo, se le conoce como vector de salida y a las funciones algebraicas dentro de este vector se les conoce como ecuaciones de salida, el vector de salida $y(t) \in R^{p \times 1}$. (Hendricks y cols., 2008).

La persona que genera el modelo, puede definir las variables de estado a utilizar, sin embargo, se debe tener consideraciones en las variables de estado ya que por definición, una variable de estado $x = [x_1, x_2, \dots, x_n]$ debe cumplir con lo siguiente:

- Que sean algebraicamente independientes entre sí.
- Que sean el conjunto mínimo de variables del sistema, con el cual el estado del mismo en cualquier tiempo $t \geq 0$ pueda conocerse, si se conoce además su condición inicial $x_0 = x(0)$ y la entrada $u(t) \forall t \geq 0$.

Una vez definidas las variables de estado se debe definir las entradas al sistema, así como las salidas del mismo, el proceso de modelado dependerá de las ecuaciones que rigen al sistema, es decir, las leyes físicas o químicas que rigen al sistema, aplicando estas ecuaciones en función a las variables seleccionadas y colocarlas de forma matricial hasta obtener todas las ecuaciones de estado. Una vez que se obtienen las ecuaciones ya se puede tener un modelo matemático con una buena aproximación de la dinámica completa o lo más general posible y poder manipular el modelo de forma relativamente sencilla para obtener información del sistema e implementar alguna técnica de control.

3.2. Modelado del sistema de Plato y Bola

Para obtener el modelo matemático del sistema de plato y bola, se debe analizar las leyes físicas que rigen al sistema. El sistema resulta ser un sistema electromecánico que consiste en poder controlar un objeto redondo en un plato, el plato se mueve de forma libre tanto en el eje x , como el eje y , en el plano de la superficie del plato. Al analizarlo de forma lateral al plato forma un ángulo entre la inclinación de cada eje y la superficie; el ángulo que se aplique en cada eje, generará un movimiento de la bola en la plano x , y de la superficie del plato y de esta forma se podrá controlar la posición de la bola.

Al definir las leyes que rigen el sistema se encontró que el sistema de plato y bola es un sistema no lineal con múltiples entradas y múltiples salidas, de modo que es conveniente modelarlo en variables de estado, sin embargo, para poder generar un modelo matemático que se pueda manipular de forma sencilla, matemáticamente hablando, se hicieron algunas consideraciones. Estas consideraciones se pueden hacer ya que el modelado en variables de estado, permite modelar el sistema físico tan sencillo o complejo a como se requiera, siempre que se respete la definición de variables de estado, así como las leyes constitutivas que rigen el sistema. Las consideraciones para el modelado fueron las siguientes:

- Se definieron como variables de estado a la posición y velocidad de la bola en el plato.
- No se tomó en cuenta la fricción viscosa del sistema.
- No se consideró la inercia del plato.
- La bola siempre debe estar en contacto con el plato es decir no puede botar en el plato o salirse del mismo.
- El sistema fue modelado considerando solamente la posición de la bola y su masa, por lo que la única fuerza que actúa sobre la bola es su propio peso.

- Se tomaron los ángulos aplicados a cada eje como entradas al sistema, debido a que el plato tiene instalado un actuador en cada eje cartesiano del plano x , y .
- La salida del sistema es la posición de la bola en el plato para el eje x , y , así como su velocidad en cada eje del plano del plato respectivamente.

Debido a que cada eje del plato esta actuado de forma independiente y por la forma en que se eligieron las entradas al sistema es posible modelar el sistema en dos subsistemas, a modo de poder asignar las variables de estado para el eje x y para el eje y de forma independiente. Por lo que el vector de variables de estado para cada eje queda de la siguiente forma:

Para x :

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} d_x \\ V_x \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} \quad (3.3)$$

Donde d_x es la posición en el eje x y V_x es la velocidad, como la velocidad es la derivada de la posición también se puede declarar como x y \dot{x} .

Para y :

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} d_y \\ V_y \end{pmatrix} = \begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} \quad (3.4)$$

Donde d_y es la posición en el eje y y V_y es la velocidad, y al igual que el eje x se puede declarar como y y \dot{y} .

3.2.1. Modelado por Ecuaciones de Euler-Lagrange

Para obtener el modelo del sistema de plato y bola se utilizó el método de las ecuaciones de Euler-Lagrange, el cual permite obtener el modelo del sistema en variables de estado de una forma práctica considerando el lagrangiano del sistema. El lagrangiano es la resta de la energía cinética y la energía potencial del sistema, el lagrangiano se utiliza de la siguiente forma para obtener la ecuación de Euler-Lagrange:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} \right) - \frac{\partial L}{\partial x_i} + \frac{\partial D}{\partial x_i} = u_i(t) \quad (3.5)$$

Se considera L como el lagrangiano del sistema, D es el vector de disipación de Rayleigh, u_i es el vector de entradas al sistema y x es el vector de variables de estado. El lagrangiano del sistema se obtiene a partir de la energía del sistema y se representa de la siguiente forma:

$$L = K - U \quad (3.6)$$

Donde K es la energía cinética total del sistema y U es la energía potencial total del sistema. (Hendricks y cols., 2008).

3.2.2. Modelo del sistema de Plato y Bola

Para aplicar el método Euler-Lagrange en el modelo de Plato y Bola primero se debe obtener el lagrangiano del sistema, para ello se debe obtener antes la energía tanto cinética como potencial del sistema, en el caso de la energía cinética del sistema de plato y bola se obtiene a partir de la siguiente ecuación:

$$K = K_l + K_a \quad (3.7)$$

Donde K_l es la energía cinética lineal y K_a es la energía cinética angular.

$$K_l = \frac{1}{2} m (v_x^2 + v_y^2) \quad (3.8)$$

$$K_a = \frac{1}{2} J (w_x^2 + w_y^2) \quad (3.9)$$

$$K = K_l + K_a = \frac{1}{2} m (v_x^2 + v_y^2) + \frac{1}{2} J (w_x^2 + w_y^2) \quad (3.10)$$

Donde w_x y w_y es la velocidad angular de la bola y J es la inercia rotacional. El valor de J ya está definido para una bola solida a través de la siguiente formula:

$$J = \frac{2}{5}mr^2 \quad (3.11)$$

Partiendo de la definición del radián se puede obtener la relación entre velocidad angular y velocidad tangencial con la siguiente ecuación:

$$v = rw \quad (3.12)$$

Usando esta relación entre v y r , podemos obtener K en función solo de las velocidades tangenciales tanto para x como y , obteniendo K de la siguiente forma:

$$K = \left(\frac{m}{2} + \frac{J}{2r^2}\right)(v_x^2 + v_y^2) \quad (3.13)$$

Para obtener la energía potencial, se puede utilizar el ángulo en que el plato visto de forma lateral al plano x , y respecto a la superficie, como se muestra en la Fig. 3.1. Al analizarlo de esta forma se puede ver las fuerzas que actúan sobre la bola y si se descarta la fricción entre la bola y el plato, se tiene que la única fuerza es el peso de la bola tanto para cada eje del plano x como y .

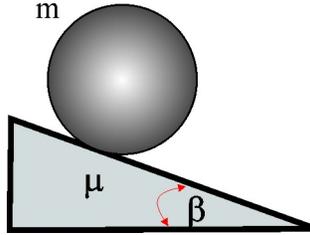


Figura 3.1: Plato y bola visto de forma lateral.

De modo que la energía potencial se obtiene de la siguiente forma:

$$U = U_x + U_y \quad (3.14)$$

$$U_x = mgh_x = mgd_x \sin(\alpha) \quad (3.15)$$

$$U_y = mgh_y = mgd_y \sin(\beta) \quad (3.16)$$

$$U = mg(d_x \sin(\alpha) + d_y \sin(\beta)) \quad (3.17)$$

Donde m es la masa de la bola, g es la aceleración gravitacional, h_x y h_y son la altura en los ejes x y y , d_x y d_y son las posiciones en ambos ejes y α y β son los ángulos de inclinación en cada eje respectivamente.

Al haber obtenido K y U el Lagrangiano resulta de la siguiente forma:

$$L = \left(\frac{m}{2} + \frac{J}{2r^2}\right)(v_x^2 + v_y^2) - mg(d_x \sin \alpha + d_y \sin \beta) \quad (3.18)$$

Sustituyendo por las variables de estado

$$L = \left(\frac{m}{2} + \frac{J}{2r^2}\right)(\dot{x}^2 + \dot{y}^2) - mg(x \sin \alpha + y \sin \beta) \quad (3.19)$$

Una vez calculado el lagrangiano se obtiene la ecuación de Euler-Lagrange (E-L) para ambos ejes. Para x :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} + \frac{\partial D}{\partial x} = u_x(t) \quad (3.20)$$

Donde D es el vector de disipación de Rayleigh y como $D = 0$ y $u_x(t) = 0$ la ecuación E-L queda.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = 0 \quad (3.21)$$

$$\left(m + \frac{J}{r^2} \right) \ddot{x} - (-mg \sin \alpha) = 0 \quad (3.22)$$

Para y :

$$\frac{d}{dt} \left(\frac{dL}{d\dot{y}} \right) - \frac{dL}{dy} + \frac{dD}{dy} = u_y(t) \quad (3.23)$$

Donde D es el vector de disipación de Rayleigh y como $D = 0$ y $u_y(t) = 0$ la ecuación E-L queda.

$$\frac{d}{dt} \left(\frac{dL}{d\dot{y}} \right) - \frac{dL}{dy} = 0 \quad (3.24)$$

$$\left(m + \frac{J}{r^2} \right) \ddot{y} - (-mg \sin \beta) = 0 \quad (3.25)$$

Para obtener el sistema en variables de estado se despeja \ddot{x} y \ddot{y} respectivamente.

$$\ddot{x} = \left(\frac{-mg}{m + \frac{J}{r^2}} \right) \sin \alpha \quad (3.26)$$

$$\ddot{y} = \left(\frac{-mg}{m + \frac{J}{r^2}} \right) \sin \beta \quad (3.27)$$

Definimos a θ como:

$$\theta = \frac{-mg}{m + \frac{J}{r^2}} \quad (3.28)$$

Por último, obtenemos el sistema en variables de estado para cada eje.

$$\dot{x} = \begin{pmatrix} x_2 \\ \theta \sin \alpha \end{pmatrix} \quad (3.29)$$

$$\dot{y} = \begin{pmatrix} y_2 \\ \theta \sin \beta \end{pmatrix} \quad (3.30)$$

Este es el modelo no lineal en variables de estado del sistema de plato y bola para cada eje, con ayuda de este modelo podemos analizar el sistema, desde el punto de vista matemático y obtener información del sistema, un dato muy importante que podemos obtener son los puntos de equilibrio.

3.3. Puntos de equilibrio

Una vez obtenido el modelo en variables de estado, el siguiente paso es obtener los puntos de equilibrio del sistema ya que, para poder controlar el sistema, es necesario definir antes los puntos de equilibrio. Para el modelo en variables de estado un punto de equilibrio se define de la siguiente forma:

$$\dot{x} = f(x(t), u(t)) \quad (3.31)$$

$$y(t) = h(x(t), u(t)) \quad (3.32)$$

Se define el punto de equilibrio como $x = x_{eq}$, $u = u_{eq}$ y $y = y_{eq}$ a aquellos valores constantes para los cuales se cumple que: $x_0 = x_{eq}$ y $u = u_{eq} \forall t \geq t_0$, entonces $x(t) = x_{eq}$ y $y(t) = y_{eq} \forall t \geq t_0$, es decir, que siempre que el estado comience en x_{eq} permanecerá en x_{eq} para todo tiempo futuro, además de que los valores constantes deben cumplir que:

$$0 = f(x_{eq}, u_{eq}) \quad (3.33)$$

$$y_{eq} = h(x_{eq}, u_{eq}) \quad (3.34)$$

Si cumplen esta condición se consideran como puntos de equilibrio. (Hendricks y cols., 2008).

3.3.1. Puntos de equilibrio del sistema de Plato y Bola

Una vez que definimos lo que es un punto de equilibrio, se puede aplicar al sistema de plato y bola, en base a su modelo matemático en variables de estado se aplica la definición de puntos de equilibrio de la siguiente forma:

$$\dot{x} = \begin{pmatrix} x_2 \\ \theta \sin(\alpha) \end{pmatrix} = 0 \quad (3.35)$$

$$\dot{y} = \begin{pmatrix} y_2 \\ \theta \sin(\beta) \end{pmatrix} = 0 \quad (3.36)$$

Se obtiene que $x_2 = 0$ y $y_2 = 0$, de forma que $x_1 = x_{eq}$ y $y_1 = y_{eq}$, debido a que la entrada para ambos ejes es el ángulo se tiene que tanto $\theta \sin(\alpha) = 0$ y $\theta \sin(\beta) = 0$ por lo que al resolver el sistema resulta que $u_{eqx} = 0$ y $u_{eqy} = 0$.

Al hacer un análisis desde el punto de vista del sistema real, suena lógico pensar que si el ángulo en ambos ejes es cero en el plano x, y ; implica que el plato está completamente horizontal y la bola se mantendrá en cualquier punto del plato tal que x_1 y y_1 representan cualquier punto en cada eje, además de que, como x_2 y y_2 se refieren a la velocidad en cada eje y para el punto de equilibrio ambos tienen que valer cero, es coherente pensar que el sistema se mantendrá en equilibrio en cualquier punto del plato, siempre que la velocidad sea cero y el plato este a completamente horizontal, es decir a cero grados en ambos ejes; si esta condición se cumple la bola se mantendrá en ese punto del plato x_{eq} y y_{eq} . También implica que, si la velocidad en cada eje es diferente de cero, por mínima que sea, se pierde el punto de equilibrio, para $x_1 = x_{eq}$ y para $y_1 = y_{eq}$; de igual forma, si el ángulo en cada eje es diferente de cero no cumple con la condición de punto de equilibrio, sin importar donde este la bola en el plato. Una vez que conocemos los puntos de equilibrio podemos trabajar con el concepto de linealización, pero antes se debe definir lo que es un sistema lineal.

3.4. Sistema Lineal

En la representación del sistema en variables de estado, si todas las ecuaciones de estado son lineales, la representación del sistema puede simplificarse mediante un modelo lineal representado de forma matricial. Partiendo del sistema:

$$\dot{x} = f(x(t), u(t)) \quad (3.37)$$

$$y = h(x(t), u(t)) \quad (3.38)$$

Si $f_i, i = 1, 2, \dots, n$ y $h_j, j = 1, 2, \dots, p$ son combinaciones lineales de los elementos $x = x_1, x_2, \dots, x_n$ y $u = u_1, u_2, \dots, u_m$ se puede representar de forma matricial

$$\dot{x} = Ax(t) + Bu(t) \quad (3.39)$$

$$y = Cx(t) + Du(t) \quad (3.40)$$

Donde A se le conoce como matriz del sistema, B se le llama matriz de entrada, C matriz de salida y D como la matriz de alimentación directa, a este modelo se le conoce como Modelo Lineal en variables de estado. (Hendricks y cols., 2008).

3.4.1. Linealización

Se define como linealización cuando un sistema dinámico no lineal se aproxima a un modelo lineal, esto se usa principalmente para simplificar el modelo matemático, ya que en algunos casos resulta más complicado trabajar con un modelo no lineal que con uno lineal. Si tenemos el siguiente modelo en variables de estado:

$$\dot{x} = f(x(t), u(t)) \quad (3.41)$$

$$y = h(x(t), u(t)) \quad (3.42)$$

En torno a un punto de equilibrio (x_{eq}, U_{eq}, y_{eq}) , se aproxima el modelo a un modelo en variables de estado lineal de la siguiente forma.

$$\dot{x}' = Ax'(t) + Bu'(t) \quad (3.43)$$

$$x'(t) = Cx'(t) + Du'(t) \quad (3.44)$$

Donde $x'(t) = x - x_{eq}$, $u'(t) = u - u_{eq}$, $y'(t) = y - y_{eq}$.

La linealización se obtiene si se aproxima cada ecuación de estado y cada ecuación de salida mediante una expansión en series de Taylor en torno al punto de equilibrio del sistema.

$$f_i(x, u) = f_i(x_{eq}, u_{eq}) + \frac{\partial f_i}{\partial x_1}(x_1 - x_{eq}) + \dots + \frac{\partial f_i}{\partial x_n}(x_n - x_{eq}) + \frac{\partial f_i}{\partial u_1}(u_1 - u_{eq}) + \dots + \frac{\partial f_i}{\partial u_m}(u_m - u_{eq}) + \dots \quad (3.45)$$

Más términos de Orden Superior $(\frac{\partial^2 f_i}{\partial x_i^2})$, $(\frac{\partial^3 f_i}{\partial x_i^3})$, etc.

$$h_j(x, u) = h_j(x_{eq}, u_{eq}) + \frac{\partial h_j}{\partial x_1}(x_1 - x_{eq}) + \dots + \frac{\partial h_j}{\partial x_n}(x_n - x_{eq}) + \frac{\partial h_j}{\partial u_1}(u_1 - u_{eq}) + \dots + \frac{\partial h_j}{\partial u_m}(u_m - u_{eq}) + \dots \quad (3.46)$$

Más términos de orden superior $(\frac{\partial^2 h_j}{\partial x_i^2})$, $(\frac{\partial^3 h_j}{\partial x_i^3})$, etc.

Si despreciamos los términos de orden superior es posible agrupar en forma matricial de la siguiente forma:

$$f(x, u) \simeq \frac{\partial f_j}{\partial x} x'(t) + \frac{\partial f}{\partial u} u'(t) \quad (3.47)$$

$$h(x, u) \simeq h(x_{eq}, u_{eq}) + \frac{\partial h}{\partial x} x'(t) + \frac{\partial h}{\partial u} u'(t) \quad (3.48)$$

Podemos aproximar este modelo a un modelo lineal si utilizamos derivadas parciales de la siguiente forma:

$$A = \frac{\partial f}{\partial x}, B = \frac{\partial f}{\partial u} \quad (3.49)$$

$$C = \frac{\partial h}{\partial x}, D = \frac{\partial h}{\partial u} \quad (3.50)$$

De esta forma si en el modelo aproximado, a cada derivada parcial se evalúa en el punto de equilibrio se obtiene una aproximación lineal, a esto se le conoce como linealización. (Hendricks y cols., 2008).

3.4.2. Linealización del sistema de Plato y Bola

Aplicando la teoría de linealización al modelo de plato y bola se puede obtener una aproximación lineal para cada eje. Partiendo del sistema no lineal de la siguiente forma:

$$\dot{x} = \begin{pmatrix} x_2 \\ \theta \sin(\alpha) \end{pmatrix} = 0 \quad (3.51)$$

$$\dot{y} = \begin{pmatrix} y_2 \\ \theta \sin(\beta) \end{pmatrix} = 0 \quad (3.52)$$

Y obtenemos las matrices de la aproximación lineal de la siguiente forma:

Para el eje x:

$$A = \frac{\partial f}{\partial x}, B = \frac{\partial f}{\partial u_1} \quad (3.53)$$

Donde.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}, B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \frac{\partial f_2}{\partial u_1} \end{bmatrix} \quad (3.54)$$

Al resolver las derivadas parciales y evaluar en el punto de equilibrio del sistema obtenemos el siguiente modelo lineal:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \theta \end{bmatrix} \quad (3.55)$$

Y por último agregamos las matrices C y D , cabe señalar que para este sistema, no es posible medir la velocidad y por esta razón se define las matrices C y D para el eje x como:

$$C = [1 \quad 0], D = [0] \quad (3.56)$$

Así obtenemos el modelo lineal del sistema para el eje x . Hacemos lo mismo para el eje y :

$$A = \frac{\partial f}{\partial y}, B = \frac{\partial f}{\partial u_2} \quad (3.57)$$

Donde.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix}, B = \begin{bmatrix} \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_2} \end{bmatrix} \quad (3.58)$$

Al resolver las derivadas parciales y evaluar en el punto de equilibrio del sistema obtenemos el siguiente modelo lineal:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \theta \end{bmatrix} u(t) \quad (3.59)$$

Y definimos de la misma forma que en el eje x las matrices C y D para el eje y como:

$$y(t) = [1 \quad 0] x(t) + [0] u(t) \quad (3.60)$$

De esta forma obtenemos el modelo lineal para el eje y . Con esta herramienta de linealización se obtuvo un modelo lineal que representa la dinámica, lo más general posible, del sistema de plato y bola, con esta herramienta podemos seguir analizando el sistema para que más adelante se pueda implementar las técnicas de control al mismo. Existen muchos otros métodos para modelar un sistema dinámico, un modelo muy común es obteniendo su modelo como función de transferencia a través del modelado entrada/salida, sin embargo, para entender este método es necesario conocer el concepto de transformada de Laplace.

3.5. Transformada de Laplace

El método de la transformada de Laplace es un método que aporta muchas ventajas al momento de resolver ecuaciones diferenciales lineales, la idea general es transformar las ecuaciones en función de t al dominio de una variable compleja s y resolver la ecuación de forma algebraica, con este cambio de variable, resulta más sencillo resolver la ecuación diferencial s que resolverla en el dominio de t . Otra ventaja de es que, una vez resuelta la ecuación, se puede regresar al dominio de t y tener la solución a la ecuación diferencial lineal en el dominio que se requiere. La variable compleja s tiene parte real e imaginaria y se denota como $s = \sigma + j\omega$ donde σ es la parte real y ω es la parte imaginaria. (Ogata y Yang, 2002).

Una función $f(t)$ tal que $f(t) = 0$ para $t < 0$ su transformada de Laplace se obtiene mediante la siguiente ecuación:

$$L[f(t)] = F(s) = \int_0^{\infty} f(t)e^{-st} dt \quad (3.61)$$

El proceso inverso para cambiar del dominio de s a t se le denomina como transformada inversa de Laplace y se obtiene con la siguiente ecuación:

$$L^{-1}[F(s)] = f(t) = \frac{1}{2\pi j} \int_{c-j\omega}^{c+j\omega} F(s)e^{st} ds \quad (3.62)$$

Para $t > 0$. (Ogata y Yang, 2002).

El concepto de la transformada de Laplace para propósito del control nos es de mucha ayuda ya que, si se tiene un modelo dinámico con ecuaciones diferenciales e integrales, con la transformada de Laplace se facilita el análisis en el dominio de s , para hacer el cambio al dominio de s , comúnmente se utilizan estas dos expresiones que parten de la transformada de Laplace.

$$L \left[\frac{d}{dt} f(t) \right] = sF(s) - f(0) \quad (3.63)$$

$$L \left[\int f(t) dt \right] = \frac{F(s)}{s} + \frac{f^{-1}(0)}{s} \quad (3.64)$$

Al utilizar estas dos ecuaciones en una ecuación diferencial e integral que representa la dinámica de un sistema se genera un nuevo modelo matemático, ahora en el dominio complejo de s , a un modelo dinámico en el dominio de s se le conoce como función de transferencia. (Ogata y Yang, 2002).

3.6. Función de Transferencia

Además del modelo en variables de estado, otra forma muy común de describir la dinámica de un sistema físico es con el modelo de entrada/salida o en función de transferencia donde a partir de las ecuaciones constitutivas que rigen el sistema, se genera una sola ecuación que describa la dinámica del sistema en función de una entrada y una salida. (Ogata y Yang, 2002).

Para este modelo se deben considerar algunas cosas:

- 1) Se debe definir cuál es la entrada y cuál es la salida del sistema.
- 2) Las variables del sistema se expresan en función de la entrada y la salida del sistema.
- 3) Que sean el conjunto mínimo de variables del sistema en cualquier tiempo t

La representación de este modelo está definida por ecuaciones diferenciales de la siguiente forma:

$$a_0 \frac{d^n}{dt^n} y(t) + a_1 \frac{d^{n-1}}{dt^{n-1}} y(t) + \dots + a_{n-1} \frac{d}{dt} y(t) + a_n y(t) = b_0 \frac{d^n}{dt^n} u(t) + \dots + b_{m-1} \frac{d}{dt} u(t) + b_m u(t) \quad (3.65)$$

Donde $y(t)$ es la salida del sistema, $u(t)$ es la entrada del sistema y a_i, b_j son coeficientes que dependen de los parámetros del sistema.

Si aplicamos la transformada de Laplace a la ecuación queda de la siguiente forma:

$$a_0 s^n y(s) + a_1 s^{n-1} y(s) + \dots + a_{n-1} s y(s) + a_n y(s) - p_y(s) = b_0 s^m u(s) + b_1 s^{m-1} u(s) + \dots + b_{m-1} s u(s) + b_m u(s) - p_u(s) \quad (3.66)$$

p_y y p_u son polinomios de grado $n - 1$ y $m - 1$ respectivamente que dependen de las condiciones iniciales de $y(t)$ y $u(t)$, si las condiciones iniciales son cero el modelo queda de la siguiente forma:

$$\frac{y(s)}{u(s)} = \frac{b_0 s^m + \dots + b_{m-1} + b_m}{a_0 s^n + \dots + a_{n-1} + a_n} = G(s) \quad (3.67)$$

a $G(s)$ se le conoce como función de transferencia. (Ogata y Yang, 2002).

Utilizando la función de transferencia $G(s)$ se puede obtener información del sistema, a los valores de $s \in C$ que hacen que $G(s) = \infty$ se le conocen como los ceros del sistema, a los valores de $s \in C$ que hacen que $G(s) = 0$ se le conocen como los polos del sistema, tanto a los polos como a los ceros del sistema se les puede hacer un análisis en el plano complejo y se puede definir la respuesta de un sistema a partir de la entrada. (Ogata y Yang, 2002).

3.7. Plano Complejo

Se conoce como plano complejo a una forma de visualizar los números con parte real e imaginaria con una representación cartesiana como se muestra en la Fig 3.2

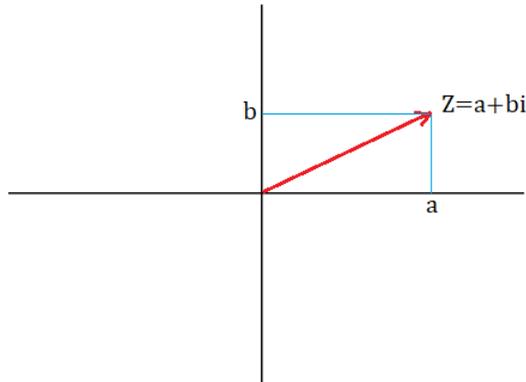


Figura 3.2: Plano Complejo

Para el modelo como función de transferencia, el plano complejo se utiliza para ubicar polos y ceros del sistema con su parte real y su parte imaginaria en la representación cartesiana en el que, la parte real se representa como el eje de las abscisas y la parte imaginaria el eje de las ordenadas.

Para poder colocar los polos de un sistema dinámico se representa con una X en los valores cartesianos que sean el valor numérico del polo tanto en su parte real como imaginaria y para el caso de los ceros se coloca un cero (0) en el lugar cartesiano que representa el valor numérico con su parte real e imaginaria. Una vez colocados los polos y ceros se procede a hacer un análisis para cada uno, a este análisis se le conoce como el lugar geométrico de las raíces.

3.8. Lugar Geométrico de las Raíces

Con apoyo del plano complejo, es posible hacer un análisis de la ubicación de polos y ceros del sistema ya sea en lazo cerrado o en lazo abierto. La respuesta transitoria de cualquier sistema se relaciona con la ubicación de los polos y ceros en el plano complejo; la ubicación dependerá del valor de la ganancia elegida para el sistema a controlar, al modificar esta ganancia en lazo cerrado, implica mover los polos y ceros del sistema y ajustarlos en la posición deseada que permita aplicar control y obtener una respuesta acotada del sistema. Partiendo de la función $G(s)$, de la que se obtiene los polos y ceros del sistema y que por definición los polos se refieren a todos los valores de s que hacen que $G(s) = \infty$ y los ceros son todos los valores de s que hacen que $G(s) = 0$. (Ogata y Yang, 2002).

La cantidad de polos y ceros está definida en la función de transferencia; para el caso de los ceros, habrá tantos ceros en infinito según el grado relativo de la función de transferencia; el grado relativo, es la diferencia entre el grado del denominador menos el grado del numerador. El análisis de los polos del sistema dependerá si las raíces del denominador tienen parte real e imaginaria o solo parte real o solo parte imaginaria, en base a las características de los polos se tendrá una salida diferente y de manera general el análisis de polos se muestra a continuación.

- 1) Si los polos del sistema tienen parte real negativa, la salida es acotada si la entrada es acotada.
- 2) Si algún polo tiene parte real positiva, la salida tiende a infinito sin importar la entrada.
- 3) Al polo más cercano al origen se le conoce como polo dominante y este polo define si el sistema responde lento o rápido, mientras más cerca del origen el sistema responde más lento.
- 4) La parte compleja de los polos define el amortiguamiento que existe en el sistema y se ve reflejado en la salida con mayor o menor amortiguamiento.

Con este análisis de polos y ceros se puede predecir, al menos en el modelo lineal del sistema, el comportamiento dinámico del mismo. (Ogata y Yang, 2002).

Con el plano complejo se puede visualizar la dinámica que tendrá los polos y ceros del sistema a través del modelo en función de transferencia, este modelo junto con el modelo en variables de estado aportan información del sistema, aunque si se desea obtener el modelo en función de transferencia desde cero puede resultar un poco complicado, ya que se deben obtener todas las ecuaciones diferenciales del sistema en función a la entrada y salida definidos, aplicar la transformada de Laplace y obtener la función $G(s)$. Una forma más sencilla para obtener la función de transferencia es a través del modelo en variables de estado.

3.9. Relación Función de Transferencia-Variables de Estado

Existe una forma de obtener la función de transferencia a partir del modelo en variables de estado, cabe señalar que solo aplica para sistemas lineales o para el modelo linealizado. Para poder realizar esto se debe partir de que la condición inicial del sistema es nula, si esto se cumple es posible aplicar al modelo lineal la transformada de Laplace y utilizarla en el dominio de s . (Hendricks y cols., 2008). Sea el modelo lineal:

$$\dot{x} = Ax + Bu \quad (3.68)$$

$$y = Cx + Du \quad (3.69)$$

Aplicando la transformada de Laplace.

$$sx(s) = Ax(s) + Bu(s) \quad (3.70)$$

$$y(s) = Cx(s) + Du(s) \quad (3.71)$$

Definiendo la función de transferencia $H(s)$, como entrada $u(s)$ y salida $y(s)$

$$y(s) = H(s)u(s) \quad (3.72)$$

$$H(s) = [C(sI - A)^{-1}B + D] \quad (3.73)$$

De esta forma se obtiene la función de transferencia a partir del modelo en variables de estado.

3.10. Función de Transferencia del Sistema de Plato y Bola

Utilizando la relación de la función de transferencia a partir de un modelo en variables de estado, es posible obtener la función de transferencia del modelo de plato y bola previamente linealizado de la siguiente forma:

$$\dot{x} = Ax + Bu \quad (3.74)$$

$$y = Cx + Du \quad (3.75)$$

Donde:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \theta \end{bmatrix} \quad (3.76)$$

$$C = [1 \quad 0], D = [0] \quad (3.77)$$

Usando $H(s)$ para obtener la función de transferencia:

$$H(s) = [C(sI - A)^{-1}B + D] \quad (3.78)$$

Al resolver la ecuación se obtiene $H(s)$ como se muestra a continuación:

$$H(s) = \frac{\theta}{s^2} \quad (3.79)$$

Una vez definida la función de transferencia del sistema de plato y bola es posible analizar sus polos y ceros para obtener más información del sistema y que junto con el modelo en variables de estado, permite un mejor análisis del sistema, con ayuda de la función de transferencia se puede hacer un análisis del lugar geométrico de las raíces del sistema de plato y bola.

3.11. Lugar Geométrico de las raíces del sistema de Plato y Bola

Una vez que se tiene la función de transferencia del sistema, es posible analizar el modelo de plato y bola para ubicar los polos y ceros del sistema y a partir de este, comenzar a diseñar las técnicas de control lineal a partir de la ubicación de los polos y ceros de la función de transferencia del sistema de plato y bola. A través de su función de transferencia:

$$H(s) = \frac{\theta}{s^2} \quad (3.80)$$

Al hacer un análisis a la función en lazo abierto, se obtiene que el sistema tiene dos polos en $s = 0$ y dos ceros en infinito de modo que el grado relativo es 2. Al ubicar los polos y ceros del sistema de plato y bola en el plano complejo resulta como se muestra en la Fig 3.3.

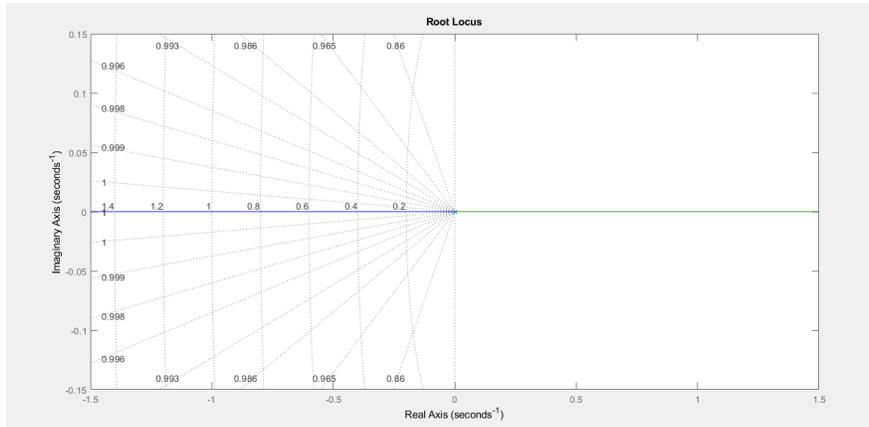


Figura 3.3: Polos y Ceros del sistema de Plato y Bola

Utilizando el análisis del lugar geométrico de las raíces podemos comenzar a diseñar un control, pero antes se debe hacer un análisis del sistema en lazo directo. Se puede hacer una prueba en lazo abierto al aplicar una señal de entrada al modelo de plato y bola como función de transferencia, por ejemplo, una señal escalón. Con apoyo del software MatLAB se puede generar diferentes simulaciones a partir de este modelo. Se realizó la prueba en lazo directo, aplicando una entrada escalón al sistema de manera simulada, la respuesta del sistema se muestra en la Fig 3.4.

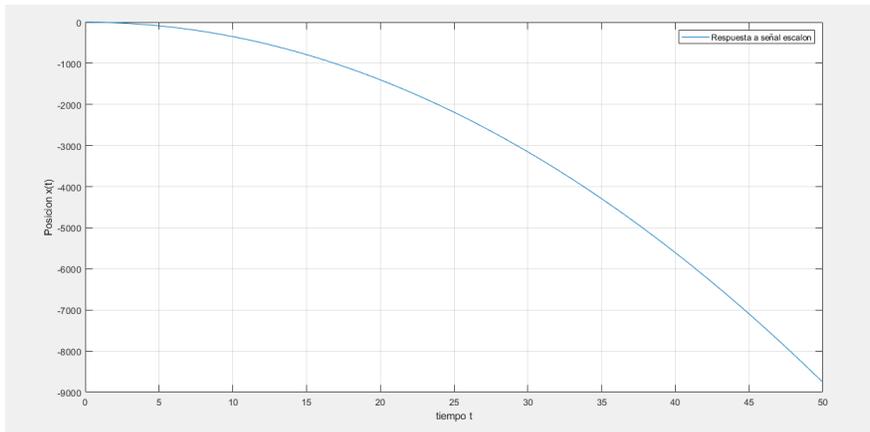


Figura 3.4: Respuesta en lazo directo de Plato y Bola

Al ver la respuesta del sistema a una señal escalón se observa que la salida no es acotada, sin embargo, se puede conocer un poco más de información del sistema si se le aplica otra señal de entrada, por ejemplo, una señal senoidal, si se aplica este tipo de señal al sistema se puede hacer un análisis de su respuesta en frecuencia.

3.12. Respuesta en Frecuencia

El termino de respuesta en frecuencia se refiere a la respuesta del sistema en estado estable a una entrada senoidal, la frecuencia de la señal senoidal se varia en cierto rango para analizar la respuesta resultante, de esta forma permite a través del criterio de estabilidad de Nyquist, averiguar la estabilidad relativa y absoluta, con las características de este análisis, del sistema en lazo abierto. Este método de análisis se utiliza mucho en funciones de transferencia con componentes complicados de analizar, ya que permite diseñar un sistema en el que se desprecian los efectos del ruido, además de que con este método se utilizan los conceptos de magnitud y fase como se muestra a continuación:

$$\frac{y(s)}{x(s)} = G(s) \quad (3.81)$$

Si $x(t) = X \text{sen}(\omega t)$ y el sistema es estable la salida $y(t)$ resulta:

$$y(t) = Y \text{sen}(\omega t + \phi) \quad (3.82)$$

Si utilizamos solo la parte compleja de s es decir $s = j\omega$ obtenemos:

$$Y = X |G(j\omega)| \quad (3.83)$$

$$\phi = \tan^{-1} \left(\frac{\Im[G(j\omega)]}{\Re[G(j\omega)]} \right) \quad (3.84)$$

Esto implica que un sistema estable, lineal e invariante en el tiempo, sujeto a una entrada senoidal, tendrá una salida senoidal de la misma frecuencia que la entrada, pero de manera general, la magnitud y fase de la salida serán diferentes de la entrada. La función de transferencia $G(j\omega)$ es una cantidad compleja y se representa mediante la magnitud y el ángulo de fase, un ángulo de fase negativo se denomina como atraso de fase y un ángulo de fase positivo es un adelanto de fase, de manera general la función de transferencia senoidal se obtiene sustituyendo s por $j\omega$ en la función del sistema. (Ogata y Yang, 2002). De manera gráfica existen tres representaciones que son las más comunes y son las siguientes:

- Trazas o Diagramas de Bode o logarítmicas.
- Trazas o Diagramas de Nyquist o polares.
- Trazas de magnitud logarítmica contra la fase.

Para el sistema de plato y bola solo se hará el análisis con las trazas o diagramas de Bode.

3.13. Diagramas de Bode

La representación en diagramas de Bode se refiere a que por medio de dos graficas distintas, una de magnitud respecto a la frecuencia y otra el ángulo respecto a la frecuencia, ambas graficas se representan de manera logarítmica, la representación más común de la magnitud logarítmica es la siguiente:

$$G(j\omega) = 20 \log |G(j\omega)| \quad (3.85)$$

La base del logaritmo es 10 y la unidad que se usa es el decibel (dB), esta grafica se traza en papel semi-logarítmico donde el eje de las abscisas determina la cantidad de ciclos logarítmicos, la ventaja principal de utilizar los diagramas de Bode es que la multiplicación de magnitudes se convierte en suma. Cuenta con un método simple para trazar una curva aproximada de magnitud logarítmica mediante asíntotas.

Con esta aproximación es más que suficiente para obtener información de la respuesta en frecuencia, para obtener el ángulo de fase es a través de la siguiente ecuación:

$$\phi = -\tan^{-1}(\omega T) \quad (3.86)$$

A partir de estas dos ecuaciones podemos obtener los diagramas de Bode tanto de magnitud como de fase de cualquier sistema a través de su función de transferencia. (Ogata y Yang, 2002). Podemos elaborar los diagramas de Bode del sistema de plato y bola para seguir obteniendo más información del sistema.

3.13.1. Diagramas de Bode del Sistema de Plato y Bola

A través del software de MatLAB se puede obtener los diagramas de bode con un solo comando, al tener la función de transferencia del sistema de plato y bola que es:

$$H(s) = \frac{\theta}{s^2} \quad (3.87)$$

Y definiendo θ de la siguiente forma:

$$\theta = \frac{-mg}{m + \frac{J}{r^2}} \quad (3.88)$$

Se sustituyen los valores de $m = 0.017[kg]$, $r = 0.0085[m]$ y $g = 9.81[\frac{m}{s^2}]$ obtenemos el valor de $\theta = -7.0071$ y aplicamos el comando de MatLAB para obtener los diagramas de Bode del sistema de plato y bola como se muestra en la Fig 3.5.

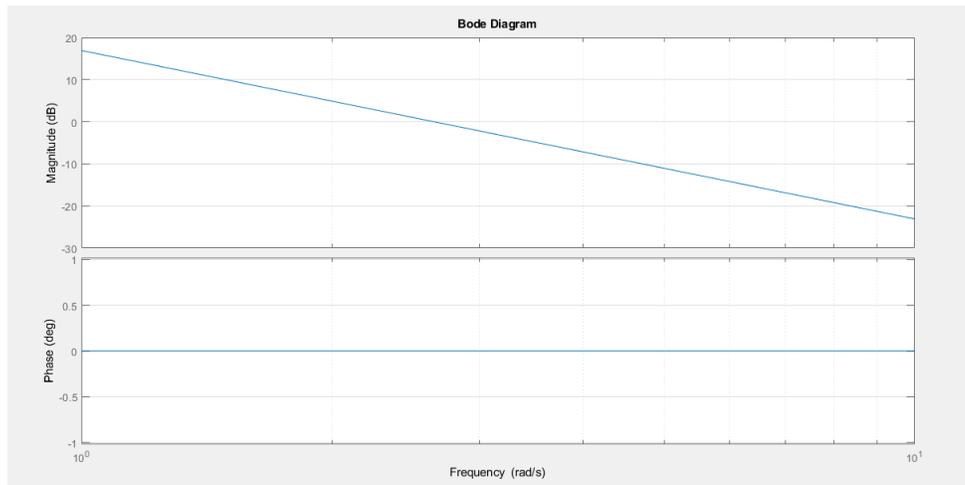


Figura 3.5: Diagramas de Bode (Magnitud y Fase) del sistema plato y bola

El análisis de los diagramas de bode nos ayuda a visualizar su respuesta en frecuencia del sistema de plato y bola, esto nos ayudara más adelante al desarrollar los controladores, por el momento no podemos saber de la estabilidad del sistema con este análisis, el concepto de estabilidad para cualquier sistema dinámico es fundamental para su control y su análisis de modo que se debe definir este concepto de estabilidad.

3.14. Estabilidad

Para poder analizar la dinámica de un sistema es necesario el concepto de estabilidad, la estabilidad se define como una propiedad de los sistemas dinámicos, en el que, si el sistema se coloca en algún punto de equilibrio, seguirá en dicho punto si es afectado por alguna entrada o perturbación del sistema, cabe señalar que, el hecho de que el sistema tenga puntos de equilibrio no garantiza que el sistema sea estable en dichos puntos. (Hendricks y cols., 2008).

3.14.1. Estabilidad en el Sentido de Lyapunov (ESL)

Aleksandr Mijáilovich Lyapunov fue un matemático y físico ruso, entre muchos de sus trabajos definió un método para conocer la estabilidad de un sistema dinámico. Se dice que un estado de equilibrio x_e es estable en el sentido de Lyapunov (ESL) si en una región esférica en el espacio de estados de radio k tal que:

$$\|x - x_e\| \geq k \quad (3.89)$$

En correspondencia con $S(\xi)$ definida por:

$$\|\phi(t, x_0, t_0) - x_e\| \leq \xi \forall t \geq t_0 \quad (3.90)$$

Existe $S(\delta)$ que se define por:

$$\|x_0 - x_e\| \leq \delta \quad (3.91)$$

Tal que las trayectorias que comienzan en $S(\delta)$ no abandonan el espacio $S(\xi)$ conforme t se incrementa indefinidamente. (Slotine, Li, y cols., 1991). Se dice que un estado de equilibrio de sistema $\dot{x} = f(x, t)$ es asintóticamente estable en el sentido de Lyapunov (AESL) si es estable en el sentido de Lyapunov, y todas las soluciones que empiezan dentro de $S(\delta)$ convergen a x_e conforme t se incrementa indefinidamente. (Hendricks y cols., 2008). Si la estabilidad asintótica es válida para todo el espacio de estados se le conoce como estabilidad asintótica global (GAS) y solo existe un solo punto de equilibrio del sistema. (Slotine y cols., 1991).

Visto de otra forma, el análisis de estabilidad en el sentido de Lyapunov consiste en analizar el comportamiento de los estados del sistema modelado en variables de estado, a partir de condiciones iniciales cercanos a un punto de equilibrio y así establecer si ese punto de equilibrio es Asintóticamente Estable en el Sentido de Lyapunov (AESL), Estable en el Sentido de Lyapunov (ESL) o Inestable en el Sentido de Lyapunov (ISL). (Slotine y cols., 1991).

La definición de Estabilidad en el Sentido de Lyapunov es la siguiente. Dada una condición inicial cualquiera x_0 cercana al punto de equilibrio en el origen tal que $\|x_0\| < \delta$ y la trayectoria del sistema $x(t)$ en el espacio de estados se dice que el punto de equilibrio es:

- ESL si $\forall \varepsilon > 0$ tal que $\|x(t)\| < \varepsilon \exists \delta = \delta(\varepsilon)$.
- AESL si el punto de equilibrio es estable y además $\lim_{t \rightarrow \infty} \|x(t)\| = 0$ para algún $\delta > 0$.
- ISL si no es ESL.

3.14.2. Método Directo de Lyapunov

Se utiliza este método para determinar si el origen es un punto de equilibrio AESL o ESL, para comprobar esto se emplean funciones dependientes a los estados que, al menos localmente, cumplan con la condición que sean funciones positivas definidas o positivas semi-definidas, a esta función se le conoce comúnmente como función candidata de Lyapunov $V(x)$. Si se tiene un espacio n dimensional y $x \in \mathbb{R}^{n-1}$, una función escalar $V(x)$ es llamada función localmente positiva para la región $\|x(t)\| \in [0, r)$, $r > 0$ si:

- $V(x) > 0 \forall x \neq 0$.
- $V(0) = 0$.
- $V(x) = 0 \Rightarrow x = 0$.

Si esto se cumple para $r \rightarrow \infty$ entonces $V(x)$ es globalmente positiva definida. Si en lugar $V(x) \geq 0 \forall x \neq 0$ y $V(0) = 0$ entonces la función $V(x)$ es positiva semi-definida.

Teorema. El origen de un sistema $\dot{x} = f(x)$ es ESL si \exists una función $V(x)$ al menos localmente positiva definida (l.p.m) cuya derivada $\dot{V}(x)$ sea al menos localmente negativa semi-definida (l.n.s-d), en ese caso $V(x)$ es llamada función de Lyapunov del sistema, si además $\dot{V}(x)$ es al menos localmente negativa definida (l.n.d), entonces el origen es AESL, si todo lo demas se cumple globalmente, el origen es el único punto de equilibrio del sistema. (Hendricks y cols., 2008).

3.14.3. ESL para Sistemas Lineales

El concepto de estabilidad en el sentido de Lyapunov se puede aplicar estos conceptos a sistemas lineales de la siguiente forma, sea el sistema:

$$\dot{x} = Ax \quad (3.92)$$

Empleamos una función candidata de Lyapunov de la siguiente forma:

$$V(x) = x^T P x \quad (3.93)$$

Donde $P = P^T \in R^{n \times n}$, a esta forma de $V(x)$ se le conoce como forma cuadrática.

Esta forma cuadrática $V(x) = x^T P x$ es positiva definida si todos los valores característicos de P son positivos, por lo que P define el signo de la función $V(x)$, si:

- P es positiva definida ($P > 0$) si $V(x) = x^* P x > 0$.
- P es positiva semi-definida ($P \geq 0$) si $V(x) = x^* P x \geq 0$.
- P es negativa definida ($P < 0$) si $V(x) = -x^* P x > 0$.
- P es negativa semi-definida ($P \leq 0$) si $V(x) = -x^* P x \geq 0$.

$\forall x \in C^{n \times 1}$ Si calculamos $V(\dot{x})$

$$V(\dot{x}) = \dot{x} P x + x^T P \dot{x} \quad (3.94)$$

$$V(\dot{x}) = x^T (A^T P + P A^T) x \quad (3.95)$$

Podemos definir una matriz Q donde:

$$Q = (A^T P + P A^T) \quad (3.96)$$

Y si utilizamos la ecuación de Lyapunov para sistemas lineales:

$$A^T P + P A^T + Q = 0 \quad (3.97)$$

Entonces, existe, y además es única, una matriz $P = P^T > 0 \forall Q > 0$ el origen del sistema $\dot{x} = Ax$ es AESL. (Hendricks y cols., 2008).

Se debe aclarar que el método de Lyapunov para sistemas lineales únicamente demuestra que en el origen $x = 0$ es AESL, si el sistema es ISL basta con que un valor característico sea de parte real positiva, si fuera ESL, este método no permite demostrarlo y para esto es necesario recurrir al método directo y comprobar si es ESL o ISL.

3.14.4. Estabilidad del sistema de Plato y Bola

A partir del modelo lineal del plato y bola.

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \theta \end{bmatrix} u \quad (3.98)$$

$$\dot{y} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \theta \end{bmatrix} u \quad (3.99)$$

Se puede utilizar el concepto de estabilidad de Lyapunov para sistemas lineales de la siguiente forma.

Si se definen las matrices P y Q como:

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.100)$$

Sustituyendo en la ecuación de Lyapunov para sistemas lineales

$$A^T P + P A + Q = 0 \quad (3.101)$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0 \quad (3.102)$$

Resolviendo el sistema de ecuaciones se obtiene el valor de P

$$P = \begin{bmatrix} -p_3 & -1 \\ -1 & p_3 \end{bmatrix} = \begin{bmatrix} p_1 & -1 \\ -1 & -p_1 \end{bmatrix} \quad (3.103)$$

Dado que tanto p_1 como p_3 no está definido su signo no existe matriz P que satisfaga la ecuación de Lyapunov para sistemas lineales, por lo tanto, el sistema NO es AESL.

Como aún no se demuestra la estabilidad por el método directo de Lyapunov se debe analizar los valores característicos del sistema.

Para x:

$$\det(\lambda I - A) = \det\left(\begin{bmatrix} \lambda & -1 \\ 0 & \lambda \end{bmatrix}\right) = \lambda^2 \quad (3.104)$$

Por lo que $\lambda_1 = 0$ y $\lambda_2 = 0$.

Si se obtiene la solución de las ecuaciones de estado con la matriz e^{At} , a partir de una condición inicial x_0 resulta:

$$x(t) = e^{At} x_0 = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (3.105)$$

Por lo que la solución es:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_1(0) + tx_2(0) \\ x_2(0) \end{bmatrix} \quad (3.106)$$

Para y:

$$\det(\lambda I - A) = \det\left(\begin{bmatrix} \lambda & -1 \\ 0 & \lambda \end{bmatrix}\right) = \lambda^2 \quad (3.107)$$

Por lo que $\lambda_1 = 0$ y $\lambda_2 = 0$.

Si obtenemos la solución de las ecuaciones de estado con la matriz e^{At} , a partir de una condición inicial y_0 resulta:

$$y(t) = e^{At} y_0 = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} \quad (3.108)$$

Por lo que la solución es:

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} y_1(0) + ty_2(0) \\ y_2(0) \end{bmatrix} \quad (3.109)$$

Tanto para el eje x como el eje y si $x(0)$ y $y(0) \neq 0$ y $t \rightarrow \infty$ entonces $x(t) \rightarrow \infty$ por lo que el sistema es ISL.

Al hacer un análisis de la estabilidad de Lyapunov cabe hacer la aclaración de que la estabilidad se demuestra para un punto de equilibrio y dado que cualquier punto en el plato en que la bola este ubicada, siempre que la velocidad para ambos ejes sea cero, es decir que $x_2 = 0$ y $y_2 = 0$, será un punto de equilibrio. En el momento en que x_2 y y_2 sean diferente de cero el sistema es ISL y si estas variables aumentan con el tiempo, el sistema sale de la zona lineal y acaba por salir la bola del plato.

Finalizando con el concepto de estabilidad y ya obtenido un análisis completo del sistema con un modelo en variables de estado no lineal y linealizado, así como un modelo como función de transferencia, ubicación de polos en el plano complejo para su análisis del lugar geométrico de las raíces, ya podemos predecir hasta cierto punto el comportamiento dinámico del sistema, también fue posible a través de estos modelos matemáticos obtener los puntos de equilibrio de sistema de plato y bola y definir la estabilidad en el sentido de Lyapunov, con esto se puede tener un concepto claro del sistema de plato y bola desde el punto de vista matemático, el paso siguiente es diseñar y construir la plataforma experimental.

Capítulo 4

Plataforma Experimental

4.1. Diseño

En este capítulo se explicará el diseño y construcción de la plataforma experimental del sistema electro-mecánico de plato y bola paso a paso, cabe señalar que se consideraron algunas características para que el diseño de la plataforma experimental permita aplicar las diferentes técnicas de control a implementar y se muestran a continuación:

- Debido a que las entradas de control deberán ser el ángulo aplicado en cada eje del plano x, y del plato, el diseño debe permitir que el plato se mueva lo más libremente posible y con un actuador tanto en el eje x como en el eje y que permita un movimiento independiente para cada eje.
- Ya que las variables de estado previamente definidas son la posición y velocidad de la bola en cada eje, es necesario poder medir al menos la posición de la bola en el plato, de esta forma se puede tener un lazo de control cerrado con un sensor que mida la posición de la bola en el plato para ambos ejes.
- Al tener un sensor que mida la posición de la bola en el plato es necesario que la forma en que se obtenga la información no interfiera en el movimiento del plato, para que los actuadores apliquen la salida de control del sistema sin ningún problema.
- En el diseño del plato se debe considerar que no permita que la bola salga del plato, ya que el modelo no maneja que la bola salga del plato, de igual forma se debe evitar que la bola rebote en el plato durante el experimento.
- Como no se tomó en cuenta la fricción viscosa, así como la inercia del plato, se debe reducir la fricción en las piezas que conecten los motores con el plato para aproximar lo mejor posible, la plataforma experimental al modelo previamente utilizado.

4.2. Construcción

Una vez definido el diseño con las características principales que debe construir el sistema real, primero se analizó la forma de poder mover el plato y para ello lo más conveniente es colocar un motor para cada eje. El motor debe tener la capacidad de colocar el plato en el ángulo en el que se requiere para cada eje y ajustar la posición de la bola, con estas condiciones los motores que mejor se adaptan a este comportamiento son los servomotores.

Los servomotores tienen la capacidad de girar de 0 a 180 grados, aproximadamente, el eje del motor y mantener el ángulo en ese intervalo; esto se hace por medio de un PWM (Modulación por ancho de pulso) para que de esta forma se puede controlar la posición y sobre todo el ángulo en cada eje. Los motores se deben colocar a modo que puedan aplicar el ángulo deseado en cada eje, su rango de giro es de aproximadamente 180 grados y se debe tener la opción del giro completo del motor en ambos ejes tanto para que el plato se mueva hacia un lado como al otro.

Por esta razón, cada motor se debe fijar a una base, a modo de que solo interfiera la flecha del motor en el plato, para aplicar el ángulo necesario en cada eje. Se debe considerar que el ángulo de cero grados en el plato, es decir, completamente horizontal, es aproximadamente 90 grados para cada motor debido a que el motor puede subir o bajar el plato según se requiera.

La plataforma experimental esta sobre de una base de MDF de 25×25 [cm] de área y 9 [mm] de grosor con los cortes necesarios para la estructura de cada base de los motores. A lo largo de las orillas de esta base principal se colocaron piezas de MDF de 3 [mm] y a modo de cerrar la base como una caja, con la ventaja de que en caso de que la bola caiga del plato, esta no salga fuera de la base principal y sea más fácil recuperar la bola y colocarla de nuevo en el plato. Adicional a las bases de los motores, se colocó una base adicional para el plato, esto debido a que los motores solo intervienen en el movimiento y debe haber una base que permita el libre movimiento en ambos ejes y que sostenga el plato en todo momento, el diseño de la base queda como se muestra en la Fig. 4.1

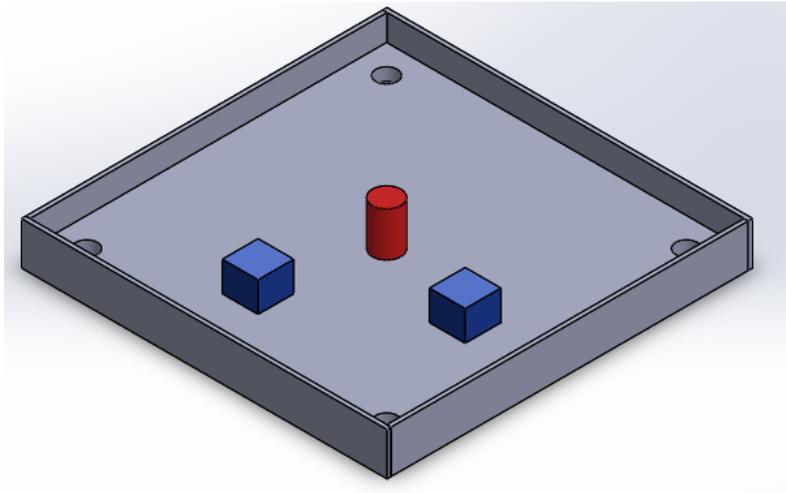


Figura 4.1: Diseño de la base del plato y bola

Para la construcción del plato se utilizó una pieza de MDF de 18×18 [cm] de área y 3 [mm] de grosor como base; el plato debe soportar el ángulo aplicado en cada motor y por el mismo diseño debe ser lo más liso posible, además de que debe poder soportar en todo momento el peso de la bola. De la misma forma que en la base, se colocaron piezas de MDF de 3 [mm] en la orilla del plato, con la finalidad de evitar que la bola salga del plato salvo que por alguna causa bote y salga del plato, si llega a salir del plato la bola caería en la base principal sin riesgo que salga la bola de la base.

Para poder acoplar la flecha de cada motor al plato, se construyó una estructura mecánica que aplique el ángulo necesario a cada eje para el movimiento del plato. La estructura consiste en dos piezas de plástico, armadas de tal forma que hacen un brazo que conecta en un extremo al motor y en el otro un eje del plato, esto se construyó a modo de poder mover la base de arriba a abajo, sin el riesgo de forzar el motor o que los motores se desacoplen del plato. Con esta configuración y como los brazos son rígidos, se genera un ángulo diferente al que se genera en el motor, de cierta forma esto reduce considerablemente la apertura del movimiento del plato; experimentalmente se encontró su punto máximo y mínimo, en que permite el movimiento de la bola para ser ubicada en cualquier punto del plato, aunque si se debe considerar esta diferencia de ángulo al implementar cualquier técnica de control.

Adicional a los brazos que conectan los motores con el plato, se colocó un soporte directamente al plato; esta pieza se construyó para que permita el movimiento libre del plato en ambos ejes, su diseño surgió a modo de simular el movimiento de la articulación en la muñeca, es decir, que pueda rotar libremente a modo tal que no afecte a los motores ni al ángulo en cada eje y que permita el soporte necesario en el plato para generar el ángulo necesario con las piezas acopladas a cada motor, el diseño la base del plato con los soportes resulta como se muestra en la Fig. 4.2

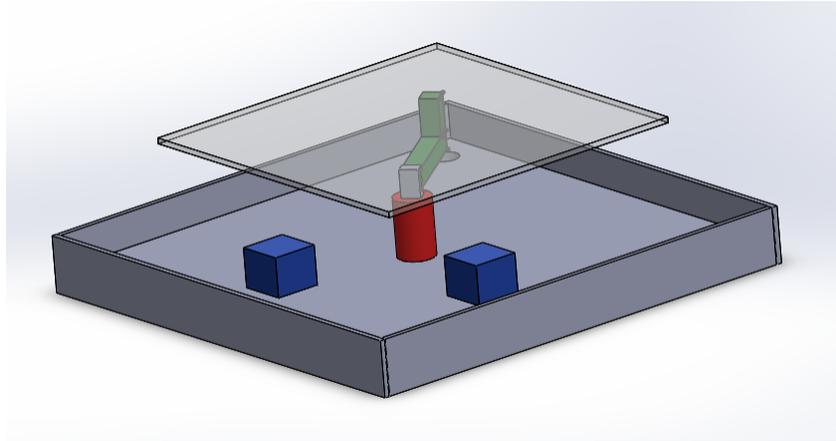


Figura 4.2: Diseño de base con soporte para el plato

Para tener un lazo de control se debe medir la posición o la velocidad de alguna forma ya que son las variables de estado seleccionadas. Realizando una investigación se observó que muchos otros modelos de plato y bola utilizan un touchscreen en el plato como sensor de posición, sin embargo, esto resulta muy costoso por el tamaño del plato, por esta razón se optó por utilizar una webcam como sensor para ubicar la posición de la bola en el plato en ambos ejes. Debido a que escogió la webcam fue necesario diseñar y construir una estructura que permita tener la webcam arriba del plato, lo suficientemente alto para no afectar el movimiento de los motores y obtener una imagen completa del plato.

La base de la cámara debe tener la capacidad de cambiar de posición y fijar la cámara según se requiera, además de poder ajustarla en la posición necesaria para tomar lecturas correctas. La primera estructura que se construyó para sostener la cámara consiste en 4 tubos de PVC de 45 [cm] de largo y 1.54 [cm] de diámetro los cuales estaban fijados a la base principal que se diseñó con huecos en las esquinas con este propósito. Los tubos a su vez sujetan uno palos de madera de aproximadamente 0.5 [cm] de diámetro, formando un arreglo que permite fijar la cámara con otros dos palos colocados de forma perpendicular y sujetos con armellas a los palos de los tubos de PVC. Utilizando esta configuración se tiene todo el espacio de arriba del plato para ajustar la webcam en la posición en que se requiera, como se muestra en la Fig.4.3

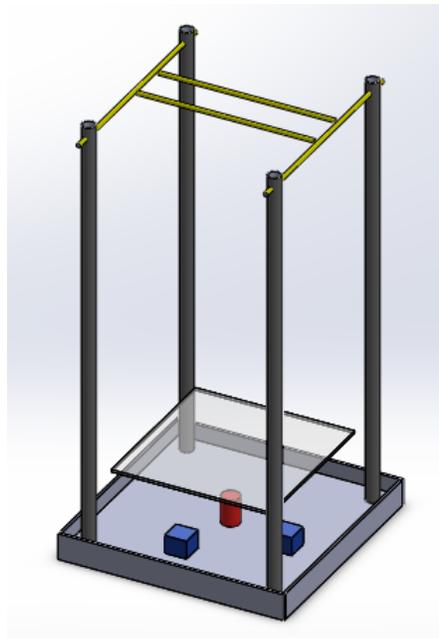


Figura 4.3: Estructura para sujetar la cámara

A medida que se realizaron las pruebas en la plataforma experimental, ya con el control implementado, se observó que la base de la webcam se movía mucho y en algunos casos desenfocaba el plato, ocasionando lecturas incorrectas. Debido a esto se tomó la decisión de cambiar la base por un diseño más robusto, se optó por cambiar los tubos de PVC con varillas roscadas de 3/8 de pulgada como base principal de la webcam; se fijó una tuerca en cada hueco la base de MDF, asignado a cada esquina de la base y se fijaron las varillas de la misma forma que los tubos de PVC. En la parte superior se colocaron más tuercas para soportar una base entre las varillas verticales y con ayuda de unas armellas, tuercas y varillas de 3/16 de pulgada se armaron dos bases de varilla a varilla, para cada varilla vertical se colocó una tuerca tipo mariposa para ajustar de forma precisa la altura de la webcam sobre el plato. Por último, se colocó a la mitad de esta base un tensor en cada varilla horizontal y se colocaron dos tuercas a modo de poder mover horizontalmente la webcam arriba del plato y dejarla fija en un punto; la webcam se colocó en otra varilla de 3/16 de pulgada entre los tensores para rotar la webcam y enfocar lo mejor posible al plato. El objetivo de toda esta configuración fue tener todos los grados de libertad posible para ajustar la webcam según se requiera en los experimentos y tenga la robustez para que una vez fija la webcam en un punto sobre el plato no se mueva bajo ninguna circunstancia, la configuración de la base superior quedó como se observa en la Fig.4.4



Figura 4.4: Base de webcam

Adicional a toda la estructura se observó que durante los experimentos afectaba la luz de la habitación a las lecturas de la webcam, más adelante se explicara a fondo como se obtuvo la posición de la bola a través de la webcam, pero por esta razón se colocó una lámpara cerca de la plataforma experimental para asegurarse de tener la mejor lectura durante los experimentos.

4.3. Software de Control

Una vez construido la plataforma experimental del sistema de plato y bola, es necesario tener un dispositivo que permita obtener la información de la webcam, así como aplicar la señal de control a los motores en cada eje para poder implementar las diferentes técnicas de control en este sistema. Dichas técnicas de control se deben programar en una plataforma donde se genera el lazo de control, que permita la adquisición de datos del sistema y que pueda aplicar la señal eléctrica a los motores; para esto fue necesario una laptop como dispositivo de control con el software que se utilizó para la implementación de las diferentes técnicas de control, el software que se utilizó fue MATLAB. MATLAB (Matrix Laboratory) es un programa para realizar cálculos numéricos con vectores y matrices, se puede utilizar con números reales o complejos y otras estructuras de programación; también puede realizar una gran variedad de gráficos en dos o tres dimensiones según se requiera y a pesar de tener su propio lenguaje de programación, permite código en lenguaje C, C++ y Fortran, así como muchas otras utilidades que tiene este software.

El software de MATLAB se puede utilizar como control discreto por medio de la programación en su propio lenguaje o a través de su herramienta de simulink, además de que tiene una gran cantidad de bibliotecas llamadas toolboxes que ayudan a utilizar MATLAB con diferentes equipos como sensores, microcontroladores, tarjetas de adquisición de datos y muchos otros equipos; permite modelar y simular sistemas y procesos para su análisis e incluso su diseño, por esta razón fue bastante útil la herramienta de MATLAB para implementar las técnicas de control a este sistema, el siguiente paso es integrar el controlador al sistema a través del circuito electrónico del sistema de plato y bola.

4.4. Diseño Electrónico

Teniendo en cuenta que los motores que generan el movimiento del plato son independientes entre sí y que el sensor para obtener la posición de la bola es una webcam, se puede separar cada bloque como un circuito independiente uno del otro, pero todos deben estar conectados al dispositivo de control que en este caso es la laptop con el software de MATLAB. Para la webcam no hubo necesidad de construir un circuito adicional, se conecta directo a la PC con una conexión USB tipo A; para el caso de los motores, si fue necesario un actuador que pueda acoplar los motores al dispositivo de control. El actuador que se utilizó fue un microcontrolador que funciona como driver entre la laptop y cada motor, para que permita aplicar el ángulo necesario según lo requiera la señal de control y para poder colocar la bola en cualquier punto del plato, el microcontrolador que se utilizó fue la tarjeta de desarrollo Arduino Mega 2560.

4.4.1. Arduino Mega 2560

El Arduino Mega 2560 es una tarjeta de desarrollo basada en el microcontrolador ATmega2560 y con su respectivo software, permite programarse para generar desde una tarjeta de adquisición de datos, hasta implementar un control PID a un sistema electromecánico, además de muchas otras aplicaciones. La tarjeta consta de 54 entradas/salidas digitales, dependiendo de lo que se requiera se pueden usar como entrada o como salida; de estas salidas digitales 15 se pueden usar como salida de PWM, además de tener 16 entradas analógicas, 4 UARTs, un cristal de cuarzo de 16 MHz, como reloj, además de una conexión USB tipo A-B para programar el microcontrolador y alimentar a la tarjeta con 5V así como una conexión Jack para alimentación DC independiente si se requiere para trabajar de forma autónoma, es decir sin estar conectada a una PC una vez programada. Sus especificaciones generales son las siguientes:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5V
- Voltaje de entrada: 7-12V
- Voltaje máximo de entrada: 20V
- Pines Entradas/Salidas: 54 (15 salida PWM)
- Entradas analógicas: 16
- Corriente DC en Pines 5V: 20 mA
- Corriente DC en Pines 3.3V: 50 mA
- Memoria Flash: 256 KB
- EEPROM: 4KB
- Velocidad de reloj: 16 MHz
- Dimensiones: 101.52 mm X 53.3 mm

El esquema general de la tarjeta de desarrollo Arduino Mega 2560 se muestra en la Fig.4.5 y la Fig.4.6

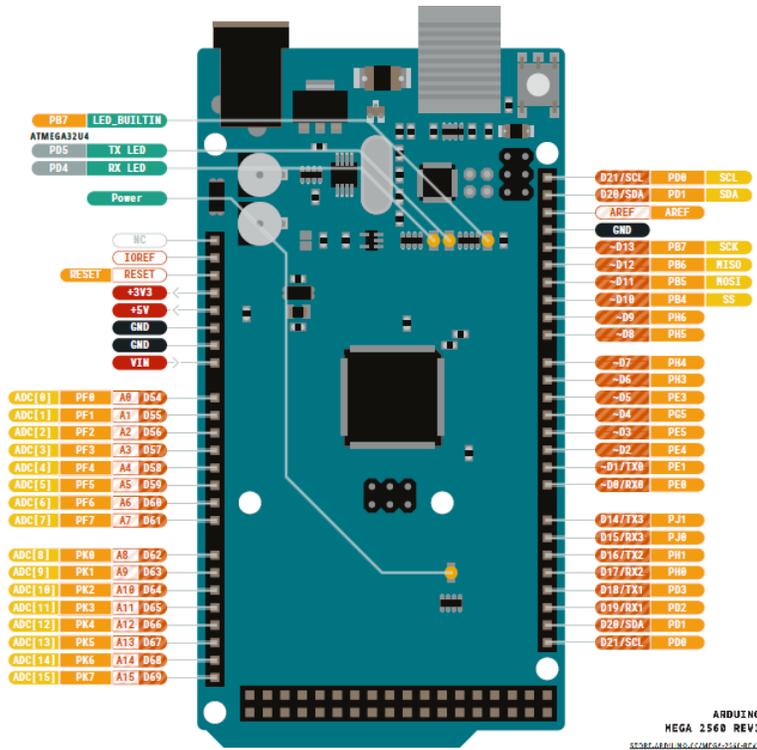


Figura 4.5: Esquema general Arduino Mega 2560 1

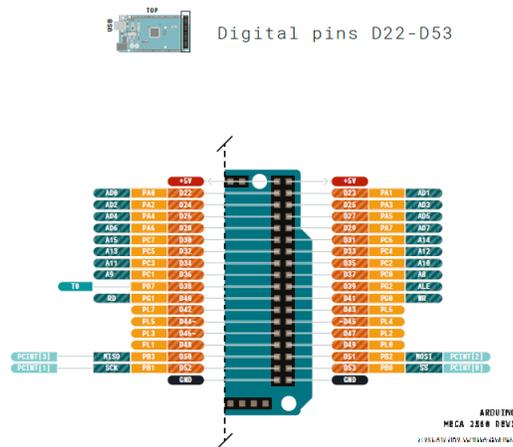


Figura 4.6: Esquema general Arduino Mega 2560 2

Para poder integrar la tarjeta Arduino Mega 2560 fue necesario cargar al software de MATLAB las bibliotecas desarrolladas en el mismo software para utilizar esta tarjeta como driver y tener un enlace de comunicación del software con los motores que controlan la posición del plato, este factor es fundamental para escoger los motores que permitan este enlace entre la tarjeta de desarrollo Arduino y el sistema de plato y bola.

4.5. Motores

Para la elección de los motores se hizo considerando como característica principal del motor, que pudiera aplicar el ángulo a cada eje lo más preciso posible y que no fuera tan costoso, además de tener el suficiente torque para aplicar la señal de control necesaria sin forzarse. Debido a esto se consideró a los servomotores como la mejor opción para este sistema, los servomotores son dispositivos que a base de un motor de DC (corriente directa) y un arreglo de engranes permiten que a partir de una señal de PWM (modulación por ancho de pulso) colocar su flecha de giro en el ángulo en que desee con una precisión considerable y se mantenga hasta que se modifique la señal de PWM.

Los servomotores se utilizarán para hacer girar el plato en cada eje para subir o bajar el plato y colocar la bola en cualquier punto del plato, otro factor a considerar es el peso que soporta cada motor, sin embargo, para este caso el sistema de plato y bola se diseñaron las piezas con MDF considerando que es un material ligero y resistente para que el peso del plato no influya de forma considerable en la elección del servomotor, de modo que se escogió el servomotor Tower Pro-micro servo SG90 (Fig.4.7).



Figura 4.7: Servo SG90

Este servomotor posee algunas propiedades que en base a su hoja de especificaciones resulta bastante útil para este sistema:

- Su rotación es de aproximadamente 180 grados
- No tiene condición de software, es decir que basta con generar el PWM sin alguna librería extra para controlar su posición
- El peso que soporta es de 9 gramos aproximadamente
- Su par de fuerza es de $1.8 Kgfuerza * cm$
- Su voltaje de operación es de 5V

Debido a que su rango de operación es de 5V, no fue necesario acoplar una etapa de potencia a la señal de PWM, esto facilita su conexión con cualquier dispositivo ya que solo se debe conectar a cualquier salida de PWM, que en este caso ya viene integrada en la tarjeta de desarrollo y su respectiva alimentación de 5V de DC, con esta conexión directa es posible accionar cada servomotor. Según la hoja de datos si en la señal de PWM el ciclo de trabajo es de 1.5 ms se posiciona en el ángulo cero del servomotor, dependiendo del ciclo de trabajo cambiara el valor del ángulo, cabe señalar que para el software de MATLAB el ángulo cero que maneja es a mitad del giro es decir que aunque gire 180 grados, en MATLAB tiene un rango de apertura de -90 grados y +90 grados, por lo que en 1.5 ms aproximadamente se ubica a la mitad del giro como se muestra en la Fig.4.8

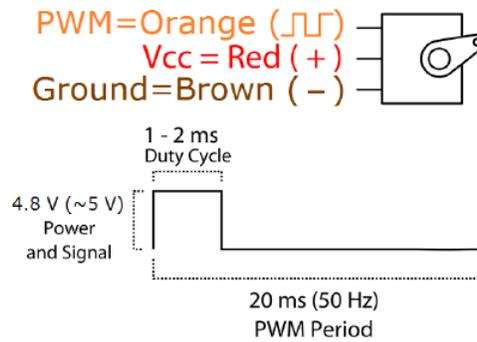


Figura 4.8: Conexión Servo SG90

Las conexiones de cada motor con la tarjeta de desarrollo Arduino quedaron como se muestra en la Fig [4.9](#)

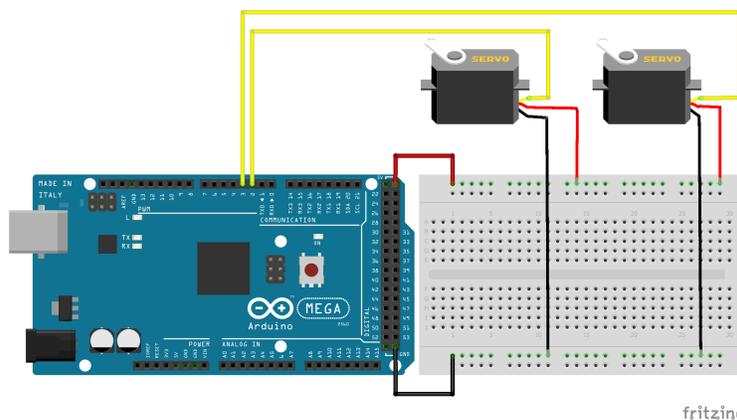


Figura 4.9: Diagrama de conexiones sistema ball and plate

4.5.1. Señal de Control

Los servomotores van a ser los encargados de aplicar el ángulo necesario a cada eje, en base a la señal eléctrica aplicada a cada uno; el Arduino Mega 2560 mandará la señal eléctrica por medio del PWM a cada motor, pero antes la laptop debe generar la señal de control a través del lazo cerrado y mediante la programación en el software de MATLAB, es decir, que la señal de control no se aplicara de forma directa a los servomotores y se debe hacer un ajuste en la programación para mandar a la señal de control y poder realizar los experimentos en el sistema real.

El software de MATLAB a través de sus bibliotecas permite controlar estos servomotores de una forma sencilla, al utilizar el Arduino como driver MATLAB ya cuenta con una biblioteca específica para controlar los servomotores con Arduino. Por medio de comandos se puede colocar el ángulo que se desee en un intervalo de 0 a 1, donde 0 es el valor mínimo del ángulo en el motor, es decir 0 grados, y el 1 es el ángulo máximo del motor, que varía en cada motor, pero en general es de 180 grados; de esta forma es posible colocar el ángulo necesario en cada motor, pero para este sistema de plato y bola y como la estructura mecánica no se tiene el intervalo completo de 0 a 1, se debe acondicionar la señal para este modelo.

A partir del intervalo de 0 a 1, se debe obtener otro intervalo que se ajuste al sistema de plato y bola, principalmente por que se colocaron los servomotores a modo de que puedan girar en un sentido o en otro, desde el punto de vista del motor los 90 grados serian 0 grados viendo el plato de forma lateral. Además de que por el propio diseño del plato es imposible tener los 0 grados o los 180 grados del servomotor en el plato, si esto fuera posible el plato estaría completamente vertical y la bola saldría completamente del plato.

Debido a esto se diseñó los acoplamientos en cada motor a modo de que el servomotor tenga un intervalo más corto y que permitiera mover el plato sin el riesgo de dañar el sistema o los servomotores pero que aún así aplique el ángulo en base a la señal de control implementada en el sistema real. De forma experimental se obtuvo un intervalo del ángulo aplicado al plato con el intervalo de 0 a 1 del servomotor, que fue de -14 a 14 grados. Si se ve como 0 grados el plato de forma completamente horizontal, para el servomotor serían 90 grados aproximadamente; con estos intervalos se puede generar una aproximación polinomial, que para este sistema fue de grado 5 y obtener una ecuación que permita convertir cualquier señal de control en una señal que pueda colocar el ángulo deseado en cada motor, las gráficas de los valores obtenidos (azul) y la aproximación polinomial (rojo) se muestra en la Fig. 4.10.

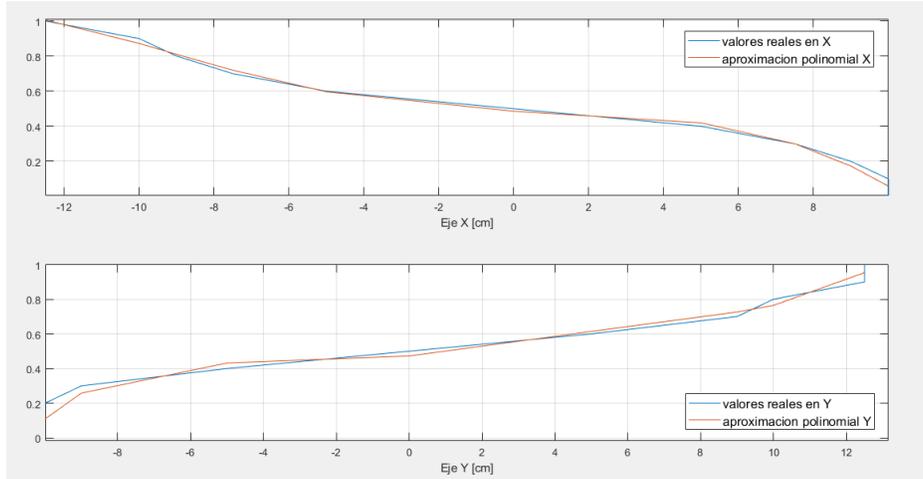


Figura 4.10: Acondicionamiento señal de control por aproximación polinomial

Con esta aproximación polinomial para cada eje se puede aplicar cualquier técnica de control, ya sea que la señal de control se exprese en radianes o en grados, que para este caso es en grados, se hace el ajuste con la aproximación polinomial y se tiene la seguridad de que es el ángulo necesario para controlar la posición de la bola en el plato, el siguiente paso es obtener la posición de la bola a través del sensor.

4.6. Sensor

En todo lazo de control realimentado se debe tener una forma de medir alguna de las variables de estado definidas en el modelo matemático; para poder hacer esto se requiere un sensor que, dependiendo de la variable a medir, sea capaz de obtener la variable física al momento de implementar cualquier técnica de control al sistema real. El sensor detecta la magnitud de la variable física a medir en el sistema y la procesa para generar una señal que comúnmente es una señal eléctrica y usarla para obtener información del sistema y completar el lazo de control. Para la elección de un sensor influyen muchos factores, pero principalmente la precisión y exactitud del sensor en base a la variable a medir, aunque también depende si la señal de salida del sensor requiere de un acondicionamiento adicional, el costo del sensor y si es resistente a condiciones de fuerza o temperatura, entre otros factores.

En el caso del sistema plato y bola se pueden utilizar diferentes tipos de sensores, al hacer una investigación se encontró que en otros experimentos se utiliza un touchscreen parecido al de cualquier smartphone, se coloca en la superficie del plato y cuando la bola se encuentra en algún punto genera cierta resistencia eléctrica y por medio de una relación matemática es posible definir la posición de la bola en el plato en todo momento, la desventaja de un touchscreen es que resulta ser muy costoso por el tamaño del plato, además de que no hay un touchscreen en el mercado del tamaño del plato y se tendría que juntar varios touchscreens para abarcar por completo la superficie del plato y también se debe manejar con cierto cuidado, ya que si se llega a romper en alguna parte cambia la relación de resistencia del mismo y no es posible medir la posición con la misma precisión que antes, por lo que se debe cambiar toda la pantalla si se llega a dañar, esto no resulta conveniente si se desea hacer un sistema más robusto y que además permita implementar diferentes técnicas de control.

Por todo esto para este sistema de plato y bola se optó por una webcam, la webcam es una cámara de video pequeña que se conecta a cualquier PC para grabar video según lo requiera el usuario. La elección de utilizar la webcam como sensor permite tener una gama más amplia para escoger el sensor, además que reduce el costo y es más fácil de conseguir en el mercado. Además de que la webcam permite medir la posición de manera independiente sin afectar el movimiento del plato y no hay forma se pueda dañar la webcam durante los experimentos, lo que si se debe considerar es la resolución de la webcam, mientras mayor sea la resolución, se tendrá de forma más precisa la posición de la bola en el plato. La webcam que se utilizó para este sistema fue la Logitech HD Webcam C270 (Fig.4.11).



Figura 4.11: Logitech HD Webcam C270

La webcam Logitech HD C270 permite capturar video e imagen en HD, aquí algunas de sus especificaciones:

- Video HD 1280x720 pixeles.
- Captura sobre 1280x720 pixeles.
- Fotos arriba de 3.0 megapíxeles.
- Alta velocidad USB 2.0

Esta webcam se utiliza principalmente para hacer conferencias en línea, pero resulta bastante útil que sea HD ya que tanto en video como imagen tiene mayor resolución y al utilizarla como sensor da una mejor precisión para obtener la posición de la bola en el plato. La desventaja de este modelo es que la webcam es una cámara de propósito general, es decir que solo captura fotos o video y no tiene un sistema que permita la capacidad de detectar objetos, por esta razón fue necesario que, por medio de programación, se diseñara un algoritmo que permita al sistema de control procesar la imagen para obtener la posición a través de la webcam.

4.6.1. Procesamiento de imagen para webcam

El proceso que se usó para que la webcam tuviera la capacidad de detectar la bola en el plato y obtener la posición de la bola en el plato para ambos ejes x , y , partió con base en las bibliotecas de MATLAB para la detección de objetos y de una biblioteca adicional para conectar la webcam con el software de MATLAB. Estas bibliotecas están disponibles en el mismo software de MATLAB, la biblioteca que se usó para que se conecte la webcam con MATLAB fue MATLAB Support Package for USB, esta biblioteca permite el enlace de una webcam tipo UVC (USB Video Class) y con el software de MATLAB, a partir de esta biblioteca es posible poner la cámara en vivo y utilizarla en cualquier archivo .m de MATLAB como se muestra en la Fig.4.12.

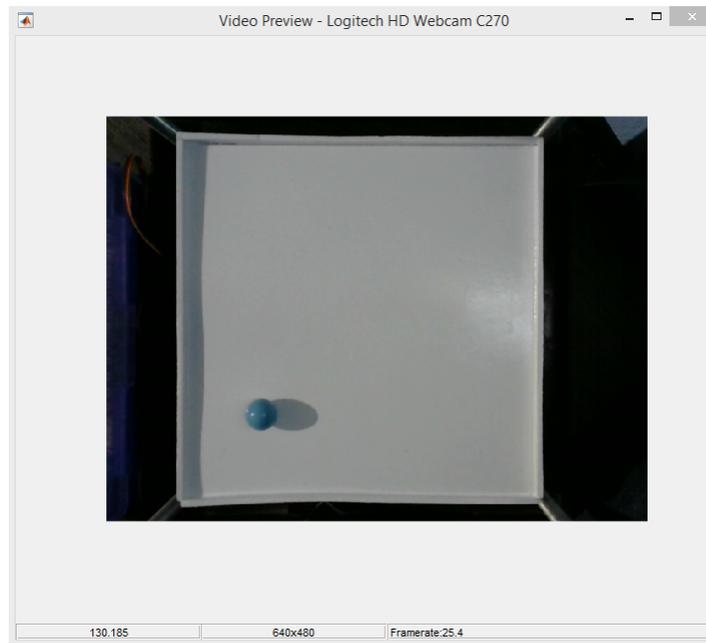


Figura 4.12: Enlace MatLAB-Webcam

Una vez obtenido este enlace, por medio de la programación de la biblioteca se puede obtener un frame o fotograma y de esta forma poder capturar la bola en un instante de tiempo. Utilizando la biblioteca de MATLAB, fue posible recortar la imagen al tamaño aproximado del plato para tener una relación directa entre el tamaño del frame y el tamaño del plato y después obtener la posición de la bola en cada eje. El frame recortado al tamaño del plato queda como se muestra como se muestra en la Fig [4.13](#)

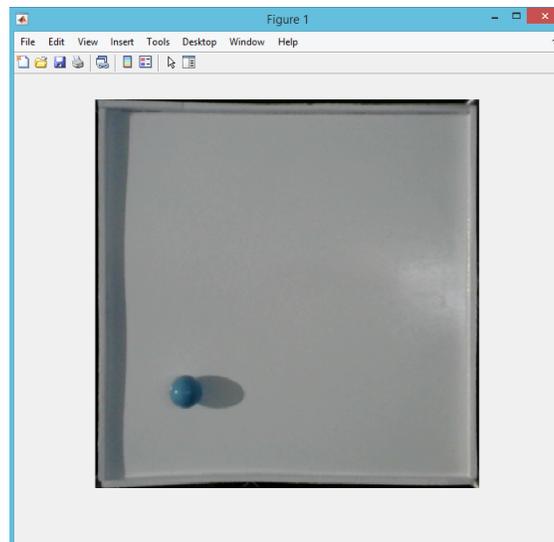


Figura 4.13: Frame recortado MatLAB-Webcam

El frame recortado no queda exactamente al mismo tamaño que el del plato, esto se hace para tener cierta consideración al momento de que el plato se mueva y permita obtener la mejor imagen posible durante los experimentos. Una vez obtenido el frame recortado al tamaño del plato, con ayuda de otra biblioteca de MATLAB se modifica el frame para poder generar otro frame, pero esta vez con una escala de grises, el nuevo frame resulta como se muestra en la Fig [4.14](#)

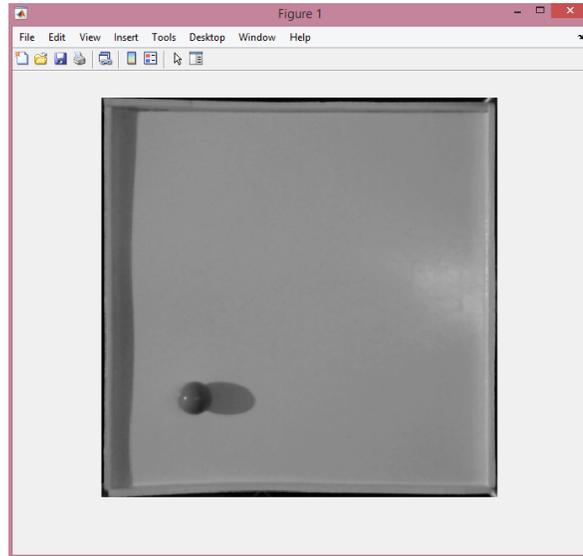


Figura 4.14: Frame en escala de grises MatLAB-Webcam

Teniendo los dos frames y por medio de la biblioteca de la webcam en MATLAB es posible sustraer el frame de escala de grises del primer frame recortado, similar a una resta, pero con la condición de que no extraiga un color definido. En la paleta de colores básica se definen tres colores rojo, amarillo y azul, se puede omitir cualquiera de estos tres colores en la resta anterior a modo de que el frame resultante solo tenga el color seleccionado y si se pinta la bola de cualquier color primario, en este caso azul, es posible obtener en el frame solo el color de la bola. La imagen del frame solo con el color de la bola se muestra en la Fig 4.15.

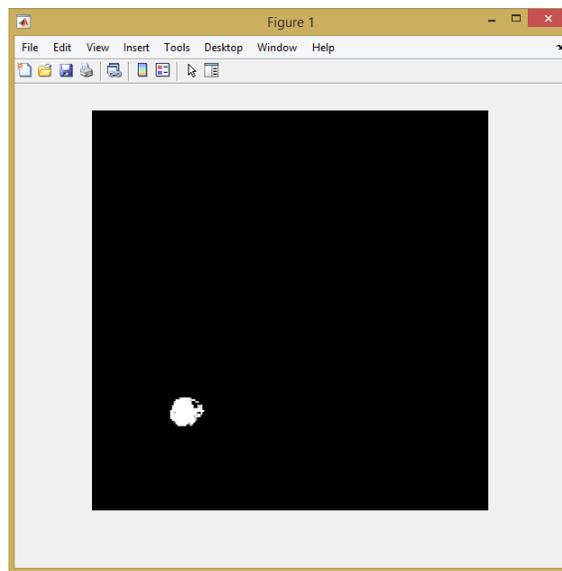


Figura 4.15: Frame de la bola MatLAB-Webcam

Un frame se puede visualizar como una matriz de píxeles, debido a esto es posible utilizar el concepto de centroide en un frame y como ya se conoce el tamaño exacto del frame recortado, además de que en el último frame solo se tiene los píxeles donde está la bola, se puede calcular el centroide para cada eje y de esta forma se tiene el máximo valor en cada eje ubicado en la matriz del frame.

Las ecuaciones para obtener el centroide en cada eje son las siguientes:

$$C_x = \frac{\sum(ix(i))}{\sum x(i)} \quad (4.1)$$

$$C_y = \frac{\sum(jy(j))}{\sum y(j)} \quad (4.2)$$

Al obtener el centroide se puede obtener la posición de la bola en el frame y escalarlo a su valor físico en cm para obtener una posición aproximada a su valor real en cada eje del plato. De esta forma al aplicarlo a cada eje se obtiene la posición en x y y respectivamente y así podemos definir la posición de la bola en cada instante de tiempo en función al tiempo de muestreo que permita la PC, con la ventaja de tener los valores de forma independiente y poder utilizarlos para implementar el control en cada eje como se tenía pensado en el modelo matemático.

En el software de MATLAB podemos obtener la posición como una variable tanto para el eje x como para el eje y del plato de forma independiente, aunque si es importante señalar que una vez que se implementen las técnicas de control al sistema se debe considerar el tiempo que le toma a MATLAB obtener el frame y modificarlo para obtener la posición, si esto no se considera puede fallar el controlador al momento de definir el tiempo de muestreo ya que al controlador no le daría tiempo en obtener la posición de la bola aplicar la señal de control, si vemos la interfaz de MATLAB nos muestra la posición como se muestra en la Fig. [4.16](#)

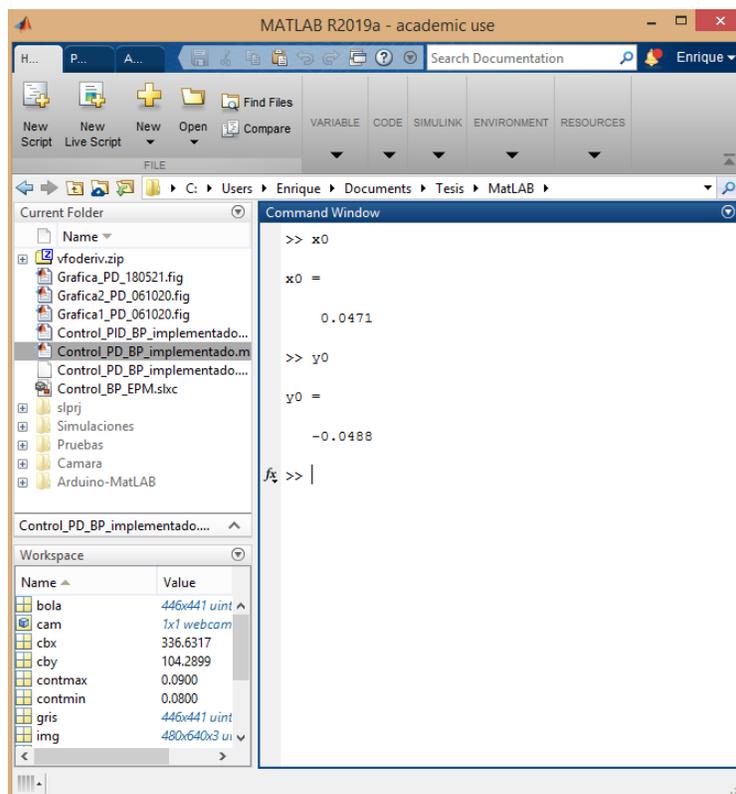


Figura 4.16: Obtención de posición de bola en plato

4.7. Planta Real

En esta sección se muestra el sistema de plato y bola real ya construido en donde se realizarán los diferentes experimentos de cada teoría de control, la finalidad es poder controlar la posición de la bola en el plato, ya sea en un punto fijo o siguiendo una trayectoria, de modo que es indispensable que el sistema real sea lo suficientemente robusto para implementar cualquier control, la base principal se muestra en la Fig. 4.17

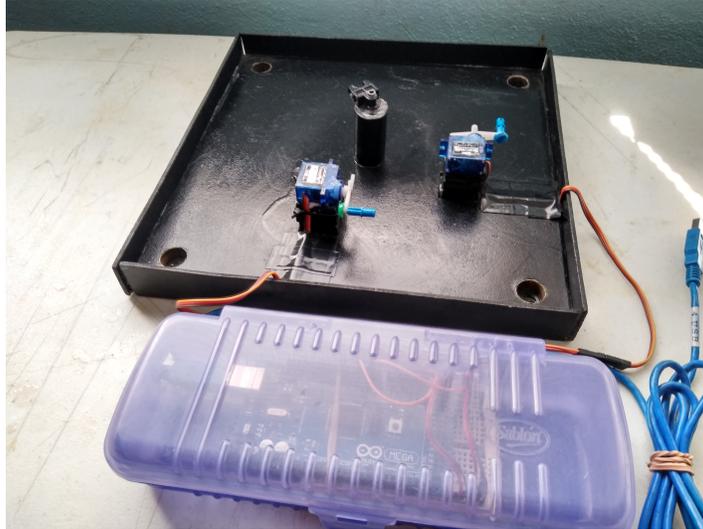


Figura 4.17: Base real del sistema Ball and Plate

En la base principal se acoplaron los servomotores para tener el control del ángulo en cada eje del plato, además de tener los huecos para colocar la base de la webcam. La tarjeta de desarrollo Arduino Mega 2560 se utilizó como driver para accionar cada servo-motor de forma independiente, aquí se muestra las conexiones del Arduino a los servo-motores en la Fig. 4.18.



Figura 4.18: Conexión de Arduino Mega 2560 - servo-motores

Debido a que el plato debe poder moverse en función al ángulo aplicado a cada servomotor en cada eje, se construyó dos piezas que conecten el movimiento de cada servomotor al plato como se muestra en la Fig. 4.19. Adicional a estas piezas se construyó otra pieza, para que el plato tenga un soporte y permita el movimiento en cada eje del plato sin forzar los servomotores.

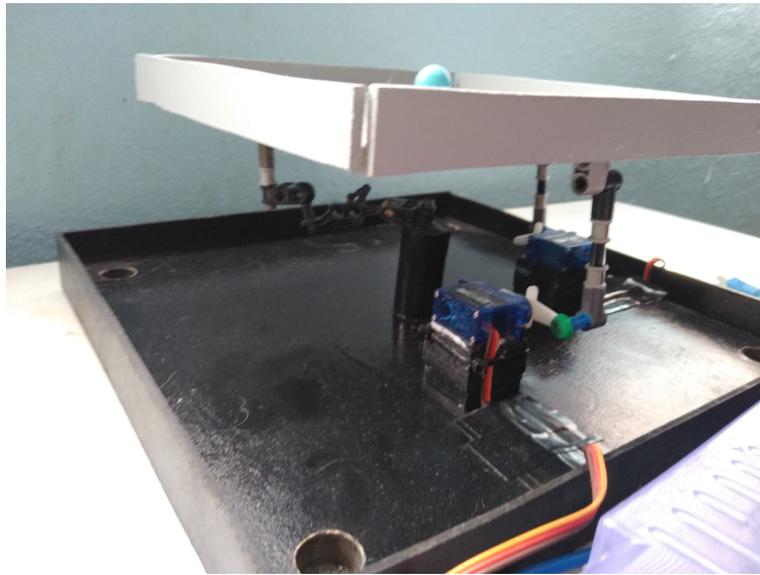


Figura 4.19: Primer acoplamiento del plato con la base y los servo-motores

A medida que se fueron realizando los experimentos se observó que el soporte del plato no era lo suficiente estable, no por el material, sino por la configuración de la estructura, por esta razón se modificó el soporte colocando una base del plato más robusta con un dado de los que se utilizan para las tuercas de los automóviles, la característica de este dado es que permite el movimiento curvo del plato es decir que gira similar al movimiento de la muñeca de la mano, con este ajuste mejor bastante el movimiento del plato durante los experimentos, el dado se colocó como se observa en la Fig. 4.20.

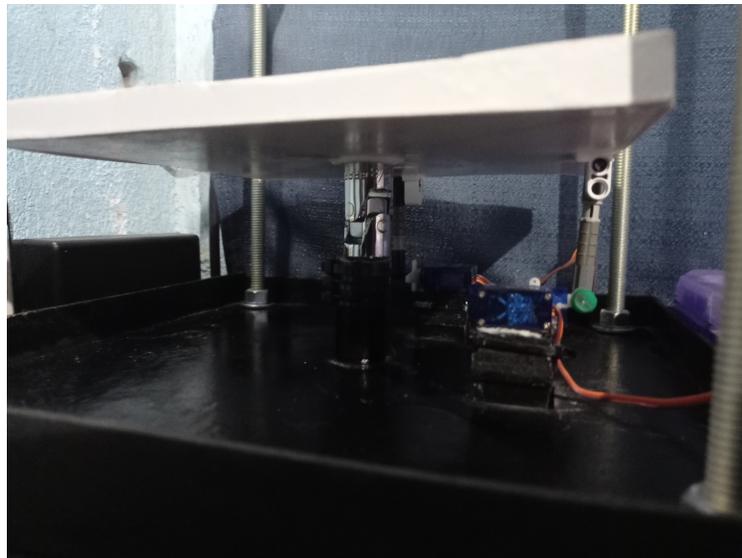


Figura 4.20: Segundo acoplamiento del plato con la base y los servo-motores

Como ya se mencionó la base de la webcam requirió ser modificada, aunque la primera versión del sistema completo de plato y bola, cumplía en cierta medida con el objetivo, no era lo suficientemente robusta para todos los experimentos, en la Fig. 4.21 se muestra la primera versión completa del sistema de plato y bola.

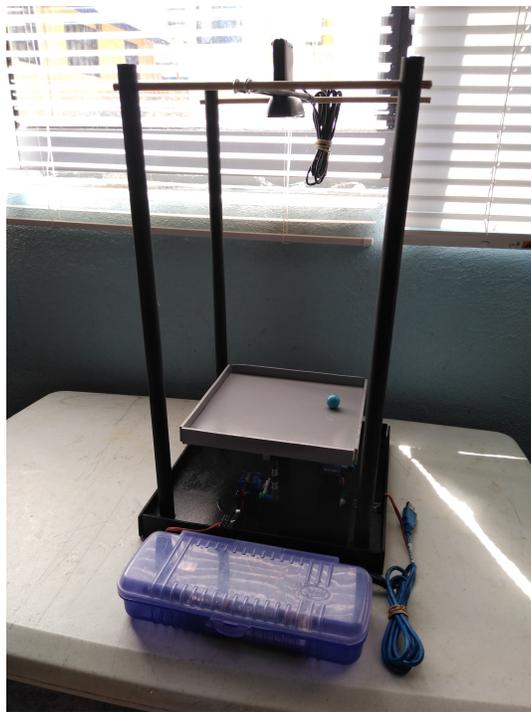


Figura 4.21: Sistema de plato y bola real

Las modificaciones fueron principalmente para la base de la webcam ya que la estructura es solo para el sensor, aunque se observó durante los experimentos más detalles que se fueron ajustando como el soporte del plato, que se modificó para tener mayor estabilidad, también se observó que se tenía un mejor frame de la bola si el plato era completamente blanco, todas estas modificaciones se muestra en la Fig [4.22](#)

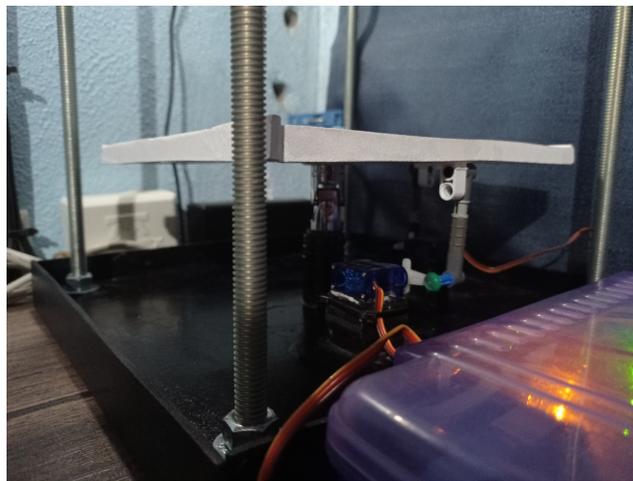


Figura 4.22: Acoplamiento mecánico de plato y bola

Así se ve la plataforma experimental del sistema de plato y bola final, como se muestra en la Fig. 4.23. Esta plataforma servirá para probar las diferentes técnicas de control. A medida que se realizaron diferentes experimentos, el sistema resultó ser bastante robusto y una precisión bastante buena al momento de aplicar las diferentes técnicas de control, la webcam respondió bastante bien y los motores no tuvieron problema al generar el movimiento que requería la señal de control, el paso siguiente es comenzar con el diseño de los controladores lineales.

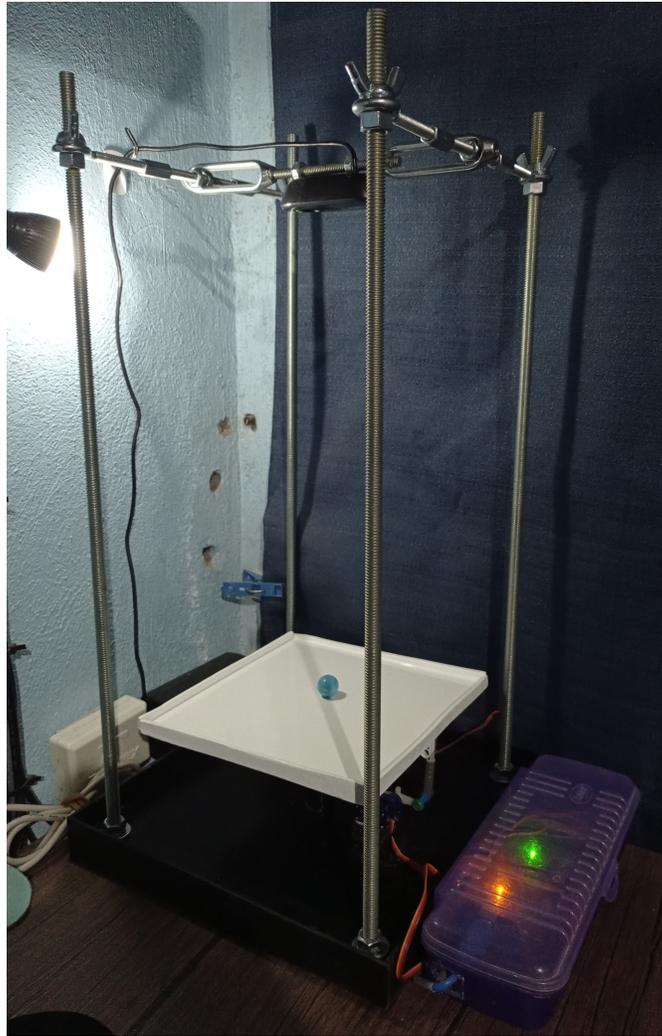


Figura 4.23: Sistema de plato y bola

Capítulo 5

Diseño de Controladores Lineales

Una vez que se tiene construida la planta real y ya se ha analizado toda la información obtenida del sistema a través de su modelo matemático, tanto en variables de estado como en función de transferencia, como son sus puntos de equilibrio y la estabilidad del propio sistema, ya se puede comenzar a implementar alguna técnica de control a dicho sistema, sin embargo, antes de diseñar los controladores se debe saber si el sistema en si permite ser controlado, a este concepto se le llama controlabilidad del sistema.

5.1. Controlabilidad

El concepto de controlabilidad surge a partir del modelo matemático en variables de estado, se define una metodología en la que se analiza, desde el punto de vista matemático, si un sistema dinámico puede o no ser controlado, cabe señalar que este método solo aplica a sistemas lineales o sistemas linealizados. Sea el sistema:

$$\dot{x} = Ax + Bu \quad (5.1)$$

El sistema dado por el par de matrices (A, B) en un intervalo de tiempo finito $[t_0, t_f]$, para una condición inicial $x(t_0) = x_0$ y cualquier condición final $x(t_f)$ y existe una señal de entrada $u(t)$ que lleva al sistema de $x(t_0)$ a $x(t_f)$, si se cumple cualquier condición siguiente:

- La matriz de controlabilidad $C_{(A,B)} = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$ Es de rango completo $rg(C_{(A,B)}) = n$
- La matriz $W_c(t) = \int_0^t e^{At} B B^T e^{-A^T t} dt$ es invertible $\forall t > 0$.
- El rango de la matriz $A - \lambda I B$ es completo $\forall \lambda = \lambda_i(A)$.

Se dice que el sistema es controlable. (Hendricks y cols., 2008).

5.1.1. Controlabilidad del Sistema de Plato y Bola

A través del modelo linealizado del sistema de plato y bola se puede conocer la controlabilidad del sistema. Sea el sistema:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \theta \end{bmatrix} \quad (5.2)$$

Utilizando el par de matrices (A, B) para generar la matriz de controlabilidad:

$$C_{(A,B)} = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (5.3)$$

$$C_{(A,B)} = \begin{bmatrix} 0 & \theta \\ \theta & 0 \end{bmatrix} \quad (5.4)$$

Al obtener el rango de la matriz de controlabilidad $C_{(A,B)}$ se obtiene $rg(C_{(A,B)}) = 2$, de modo que la matriz es de rango completo y por lo tanto el sistema linealizado de plato y bola es controlable.

Una vez definida la controlabilidad del sistema a partir de su modelo matemático linealizado, ya se puede definir qué tipo de control se puede diseñar para este sistema. En caso de que el sistema sea no controlable, no implica que no se pueda diseñar un controlador, más bien que se deben utilizar otras técnicas que tal vez puedan controlar al sistema. Si el sistema es controlable, como en este caso, podemos utilizar cualquier teoría de control lineal para este sistema, al menos en el modelo linealizado y dependiendo de la técnica de control a utilizar, el sistema real responderá de diferente manera. Una vez definida la controlabilidad de este sistema, el siguiente paso es establecer el objetivo de control, que es llevar una variable de un sistema dinámico a un valor deseado, para hacer esto es necesario definir un lazo de control.

5.2. Lazo de Control

La definición de controlar un sistema dinámico, en pocas palabras, es llevar una variable física del sistema, por medio de alguna técnica de control, a una referencia $r(t)$ o valor definido, utilizando un lazo de control. El lazo de control que más se utiliza es un lazo de control cerrado y para entender este concepto se hace un análisis en forma de bloques; se toma una señal de referencia $r(t)$ que es la que indica el valor de la variable física a controlar, y se compara con el valor actual de la misma variable, a la diferencia de estos valores se le llama error $e(t)$, por lo que mientras más grande sea la diferencia entre la referencia y el valor actual, mayor es el error. El error entra a un bloque de control, el cual puede estar diseñado con cualquier técnica de control lineal o no lineal, este bloque generara una salida de control, que comúnmente es definida como $u(t)$.

La señal de control entra al bloque del sistema dinámico a controlar, que se define como planta y a partir de esta señal, genera una reacción dinámica del sistema en sus variables de estado previamente definidas y ajusta la variable a controlar hasta que se obtiene una salida; la salida es recuperada comúnmente por un sensor y esta salida se compara nuevamente con la referencia hasta que el error sea cero o lo más cercano a cero, cuando el error sea cero o casi cero, la referencia y el valor actual son casi iguales, si esto sucede el objetivo de control se cumple en llevar la variable del sistema a un valor deseado. A este lazo cerrado de control también se le llama como lazo de control realimentado o retroalimentado y el diagrama en bloques se muestra en la Fig. 5.1.

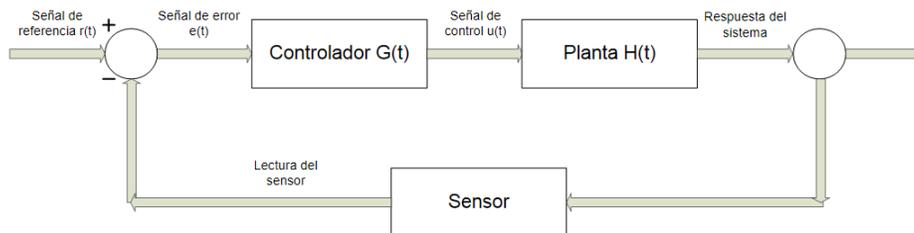


Figura 5.1: Diagrama de bloques de un control en lazo cerrado

Existe otro lazo de control muy común que es el control en lazo directo, donde la salida del sistema no se recupera a través de un sensor para compararla con la referencia y por lo tanto no influye en el lazo de control. También existen otros lazos donde agregan más detalles como perturbaciones al sistema o el ruido de medición, sin embargo, esto dependerá si se conocen estos parámetros de forma precisa y sobre todo si para el diseño del control se requiere esta información; aunque esto resulta complicado de obtener para algunos sistemas por lo que en muchos casos no es utilizado para el diseño del lazo control, sobre todo si se considera un control que soporte ciertas perturbaciones al sistema o si el ruido de medición es muy pequeño en comparación con la salida y entrada del sistema a controlar.

Para solucionar los problemas que genere el ruido de medición o perturbaciones al sistema se debe considerar un control robusto, que sea capaz de controlar al sistema aun desconociendo esta información y una de las técnicas, tal vez la que más se utiliza para sistemas dinámicos y además ser capaz de controlar a un sistema teniendo poca información es el control PID. (Ogata y Yang, 2002).

5.3. Control PID

El diseño de un control PID resulta bastante simple en comparación con otras técnicas de control, utilizando el lazo cerrado de control se toma la señal de error $e(t)$ que entra al bloque de control y se le aplica una ganancia proporcional K_p , también se integra el error de forma directa con una ganancia K_i y se deriva con una ganancia K_d , todo esto se suma y se obtiene una salida de control $u(t)$, dependiendo de las ganancias K_p , K_i y K_d , se define que tanto se integra y deriva la señal de error, siempre con una ganancia proporcional, a modo de poder llevar el sistema a el valor de referencia que se desea y así cumplir el objetivo de control hasta que el error sea cero o casi cero, en función a las ganancias este control tiene diferentes formas en función a los bloque proporcional, integral o derivativo, si alguna ganancia vale cero, ese bloque del PID no tendrá efecto en la salida de control, de modo que el control puede ser solo P, PD, PI o PID según el bloque que se ocupe o lo requiera el sistema. (Åström y Hägglund, 1995).

Analizando la salida de control de forma matemática en función al error se expresa de la siguiente forma:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (5.5)$$

De esta forma se puede aplicar un control PID a cualquier sistema, sin embargo, al utilizar esta teoría de control en el dominio del tiempo resulta muy complicado, tanto en las simulaciones como al momento de implementar el control al sistema real, para solucionar esto se pasa el modelo al dominio de complejo de s y esto se puede hacer a través de la transformada de Laplace. (Ogata y Yang, 2002).

Aplicando la transformada de Laplace a $u(t)$.

$$L[u(t)] = L[K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}] = u(s) \quad (5.6)$$

$$u(s) = K_p e(s) + K_i \frac{1}{s} e(s) + K_d s e(s) \quad (5.7)$$

Y despejando a $e(t)$ resulta:

$$\frac{u(s)}{e(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (5.8)$$

Donde $T_i = \frac{K_p}{K_i}$ y $T_d = \frac{K_d}{K_p}$

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (5.9)$$

Donde $G(s)$ resulta la función de transferencia del control PID. Al colocarlo en el diagrama de bloques del lazo de control se ve como se muestra en la Fig 5.2

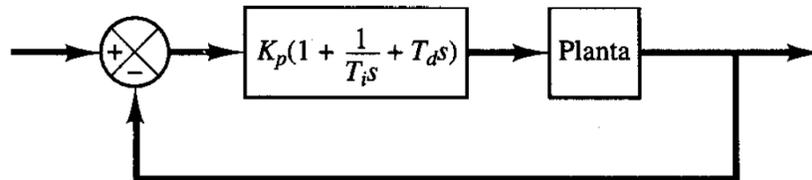


Figura 5.2: Diagrama de Bloques con un control PID

Resulta muy sencillo implementar un control PID a casi cualquier sistema dinámico, por esta razón es muy utilizada esta técnica para controladores industriales, el detalle en esta técnica de control, recae en sintonizar el PID, que es definir las ganancias K_p , K_i y K_d , ya que casi todos los controles PID se sintonizan una vez que se tiene integrado el control en el sistema. Hay una gran cantidad de literatura en donde se definen algunas reglas de sintonización, dependiendo del sistema a controlar se puede realizar una sintonización más fina y en algunos casos poseen una sintonización automática, pero como tal, no hay una regla general para sintonizar cualquier control PID. (Ogata y Yang, 2002).

Muchas de las reglas que se establecen para la sintonización de los controles PID parten a partir del modelo matemático del sistema a controlar o planta, si no es fácil obtener su modelo matemático no es posible hacer un análisis para sintonizar el PID y en estos casos se sintoniza de forma experimental, en caso de que si se cuenta con el modelo matemático se pueden aplicar algunas teorías para seleccionar las ganancias del controlador PID, una de las teorías más conocidas es la sintonización a través de las reglas de Ziegler y Nichols que, con base en la respuesta a una señal de entrada escalón y fijando un valor K_p se produce una estabilidad marginal de forma experimental y después de ver la respuesta se ajusta K_i y K_d definiendo un sobrepasso máximo de un 25 por ciento respecto a la respuesta escalón, para conseguir esto es necesario experimentar diferentes ganancias a la planta y ver su respuesta a la señal escalón hasta obtener un control que cumpla con el máximo de sobrepasso. (Ogata y Yang, 2002).

El método de sintonización de Ziegler y Nichols depende del modelo matemático pero al igual que en otras teorías es necesario hacer pruebas en el sistema y calibrar K_i y K_p ya una vez implementado el control, una mejor opción para la sintonización de las ganancias del PID es a través del análisis del lugar geométrico de las raíces, a partir de la función de transferencia se puede simular el comportamiento dinámico en lazo cerrado del sistema y ubicar los polos y ceros a modo de controlar la salida en base a la respuesta que esperamos tener y de esta forma se puede comenzar a implementar un control PID o cualquier otra técnica de control a cualquier sistema.

Un detalle importante del control PID es que cada bloque del PID es linealmente independiente entres si, de modo que se puede aplicar un control diferente según los bloques que se utilicen, es decir, que puede ser un control solamente proporcional (P), proporcional derivativo (PD) o proporcional integral (PI) y aunque matemáticamente es posible colocar un controlador puramente integral o derivativo, en la práctica no se usa ya que hace más inestable el sistema. Para algunos sistemas puede darse la situación de que con un control proporcional (P) sea más que suficiente para controlar la planta, si esto sucede no es necesario implementar el control PID a menos que se requiera un control más preciso o con las características que lo requiera, al diseñar un control PID para dicha planta puede hacer que la respuesta sea más rápida o más lenta según lo requiera, por ejemplo. Para el caso del sistema de plato y bola se hará el diseño de un control PD para ver si es posible controlar este sistema con este control y modular la velocidad de respuesta en función a las ganancias K_p y K_d .

5.3.1. Control PD para el Sistema de Plato y Bola

Partiendo de la ecuación del control PID:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (5.10)$$

Si $K_i = 0$ resulta:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (5.11)$$

Al aplicar la transformada de Laplace.

$$u(s) = K_p e(s) + K_d s e(s) \quad (5.12)$$

Por lo que la función de transferencia para el control PD resulta:

$$G(s) = K_p (1 + T_d s) \quad (5.13)$$

Donde $T_d = \frac{K_d}{K_p}$.

Al observar la función de transferencia del control PD se observa un cero en $s = -\frac{K_p}{K_d}$, este cero nos ayuda a generar una nueva función de transferencia, esta nueva función de transferencia dependerá $G(s)$ y $H(s)$, donde $G(s)$ es la función de transferencia del controlador y $H(s)$ la función de transferencia de la planta, en este caso la función de transferencia del plato y bola, se puede generar la nueva función de transferencia de la siguiente forma según las leyes del algebra de bloques. (Ogata y Yang, 2002).

$$H_2(s) = G(s) * H(s) \quad (5.14)$$

Agregando un cero a la función de transferencia es posible modificar la respuesta o salida de la planta hasta llevarla al valor deseado a través del lazo de control realimentado, el cual es el objetivo de control. Si aplicamos esta teoría a la función de transferencia del sistema de plato y bola resulta de la siguiente manera:

$$G(s) = K_p(1 + T_d s), H(s) = \frac{\theta}{s^2} \quad (5.15)$$

Al generar $H_{lc}(s) = G(s) * H(s)$ resulta:

$$H_{lc}(s) = \frac{\theta K_p(1 + T_d s)}{s^2} \quad (5.16)$$

Al analizar esta nueva función de transferencia con el lugar geométrico de las raíces y considerando que el valor de θ es negativo, además de que es un lazo de control cerrado se observa en el plano complejo como se muestra en la Fig [5.3](#).

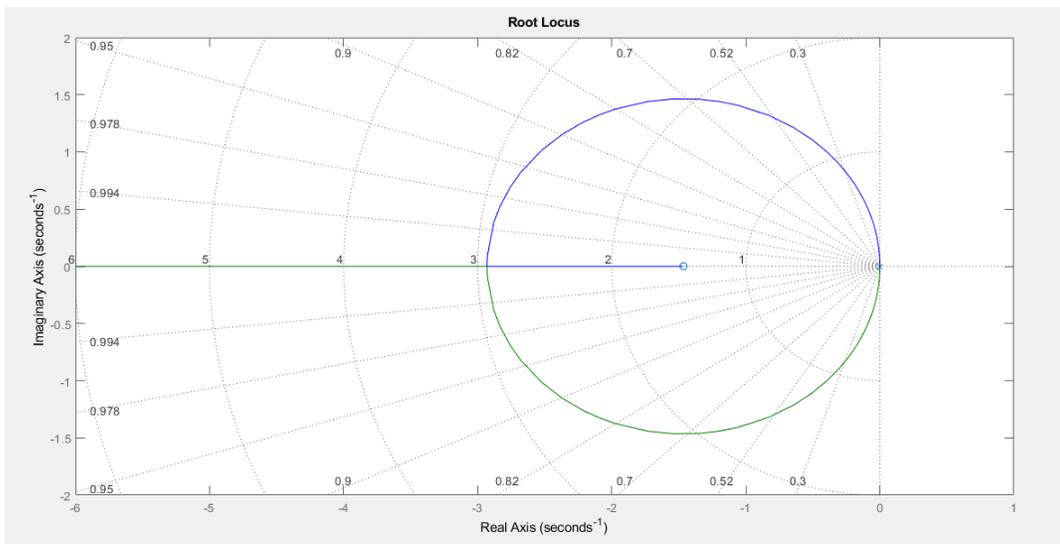


Figura 5.3: Lugar geométrico de $H_{lc}(s)$ con PD

La nueva función de transferencia ya pasa a ser un sistema con salida acotada ya que solo tiene un cero en infinito y el otro cero que se añadió con el control PD ya es un valor constante en $s = -\frac{K_p}{K_d}$, esto provoca que los polos nunca pasen al semiplano derecho, recordando que si llega a pasar al semiplano derecho ocasionaría que la salida tienda a infinito.

Para sintonizar el control PD podemos definir las ganancias que queremos en K_p y K_d y definir el modelo matemático en lazo cerrado a través de un simulador donde se pueda ver la respuesta del sistema en lazo cerrado. Un simulador muy utilizado es Simulink, que es una herramienta de MATLAB donde podemos simular el comportamiento del sistema de plato y bola a través de su modelo en función de transferencia y con el control PD.

Podemos generar los bloques del lazo cerrado y ajustar las ganancias para ver la respuesta del sistema en lazo cerrado a una señal escalón o a cualquier señal que queramos simular, dependiendo de la respuesta se ira modificando las ganancias K_p y K_d según la respuesta que se desee, para una primera simulación, se quiso que el sistema se mantuviera estable en el valor de la señal escalón, en este caso 0.5 a los 3 segundos después de mandar la señal de entrada y con condición inicial en el origen $(0, 0)$, es decir en el centro del plato y ver su comportamiento para cada eje, la salida que resulta para el eje x es la que se muestra en la Fig [5.4](#).

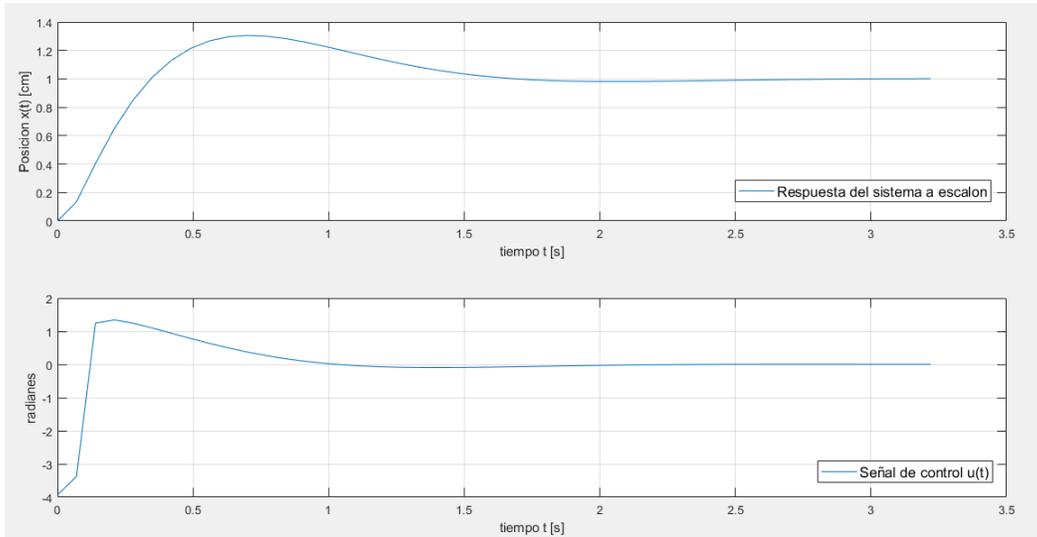


Figura 5.4: Respuesta a señal escalón de $H_2(s)$ en eje x

Esta simulación aplica para ambos ejes ya que se tiene la misma función de transferencia. Al observar la respuesta del sistema a una señal escalón vemos que responde bastante bien, visto de forma real sería llevar la bola de $(0,0)$ a $(1,1)$ en el plano del plato y aunque tiene un pequeño sobrepaso llega al valor deseado. Con esto se puede decir que el controlador PD, al menos de manera simulada cumple con el objetivo de control.

Dependiendo de la señal $r(t)$ será un problema de control de regulación o de seguimiento, un sistema de control que se le aplica una señal $r(t)$ con un valor constante y se desea que se mantenga en dicho valor por un largo tiempo se le conoce como regulación; a un sistema de control que se le aplica una señal $r(t)$ que no es constante como una trayectoria o algún tipo de señal, el control de seguimiento tratará de seguir la trayectoria de $r(t)$ y dependiendo del control implementado, la respuesta del sistema será lo más cercana a la señal de entrada deseada. Para el sistema de plato y bola también se puede realizar un control de seguimiento, generando una $r(t)$ que no sea constante se puede probar si el control PD es lo suficientemente robusto para este tipo de señales. Aprovechando que cada eje está actuado si aplicamos una señal al eje x como $r(t)_x = A \sin(\alpha t)$ y al eje y una señal $r(t)_y = A \cos(\alpha t)$ podemos probar en ambos ejes la respuesta del control PD a ambas señales, la respuesta para cada eje se muestra en la Fig 5.5.

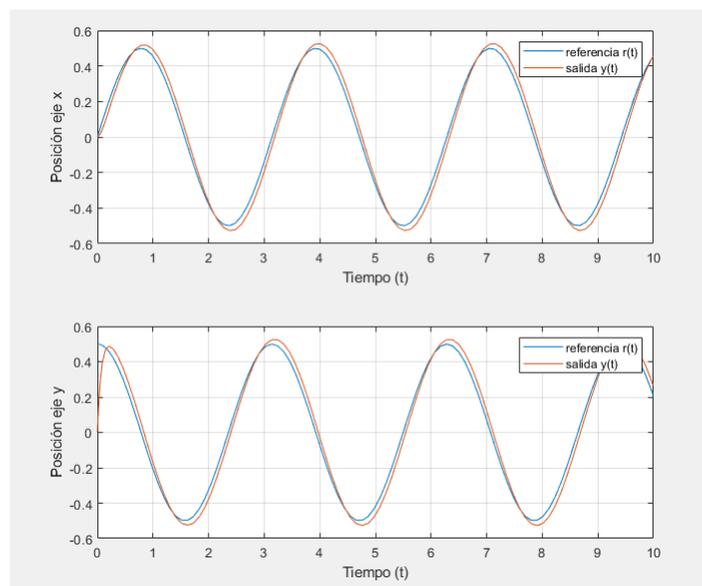


Figura 5.5: Respuesta una trayectoria de $H_2(s)$ en cada eje x, y con PD

Si vemos el plano x, y con estas dos señales en cada eje se genera un círculo en el plato de modo que la bola seguiría la trayectoria de un círculo dentro del plato, al hacer las simulaciones con las ganancias sin modificación la respuesta del sistema se muestra en la Fig 5.6

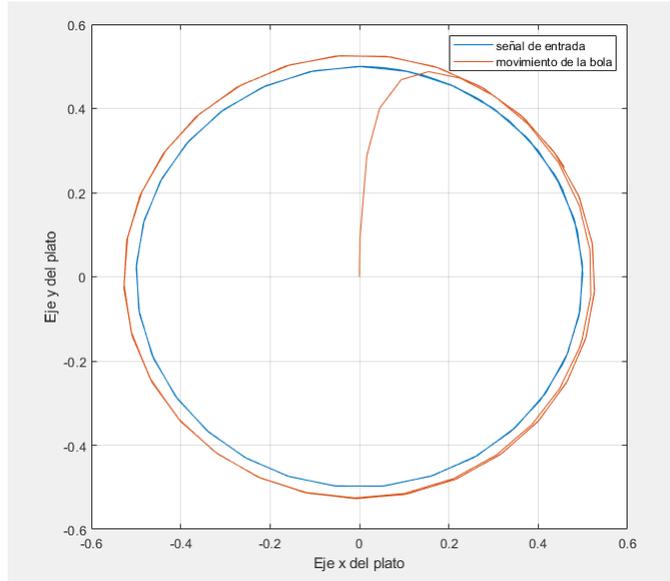


Figura 5.6: Respuesta una trayectoria de $H_2(s)$ en el plano x, y con PD

Como se observa en la simulación, la bola sí genera un círculo en el plano x, y con condiciones iniciales en $(0, 0)$, es decir en el centro del plato, sin embargo, no lo hace con la misma magnitud de la señal $r(t)$ aplicada al sistema, un control más robusto idealmente debería sobreponer las dos señales aunque se puede considerar un buen control de seguimiento en este caso. Habrá que probar con el control implementado en el sistema real para saber si cumple con el objetivo de control tanto en regulación como seguimiento, pero por lo pronto, se cumple el objetivo de control de manera simulada tanto en regulación como en seguimiento, el siguiente paso es probar con un control PID.

5.3.2. Control PID para el Sistema de Plato y Bola

Partiendo de la ecuación del control PID.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (5.17)$$

Al aplicar la transformada de Laplace resulta

$$u(s) = K_p e(s) + K_i \frac{1}{s} e(s) + K_d s e(s) \quad (5.18)$$

Después se obtiene la función de transferencia para el control PID y resulta

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (5.19)$$

Donde $T_i = \frac{K_p}{K_i}$ y $T_d = \frac{K_d}{K_p}$.

Vemos que para este caso la función de transferencia tiene un polo y un cero, si se junta esta función con la función de transferencia del plato y bola por el lazo cerrado resulta de la siguiente forma.

$$H_2(s) = G(s) * H(s) \quad (5.20)$$

Donde

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), H(s) = \frac{\theta}{s^2} \quad (5.21)$$

Resulta

$$H_2(s) = \frac{\theta(K_d s^2 + K_p s + K_i)}{s^3} \quad (5.22)$$

Al analizar esta nueva función de transferencia con el lugar geométrico de las raíces se observa en el plano complejo como se muestra en la Fig 5.7

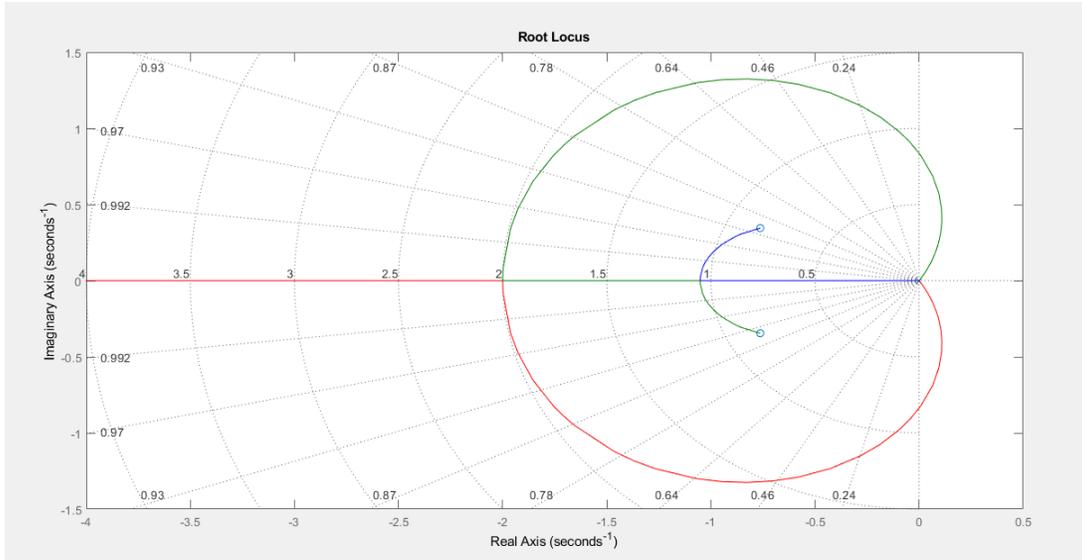


Figura 5.7: Lugar geométrico de $H_2(s)$ con PID

Con esta nueva función de transferencia y dependiendo de los valores de K_p , K_i y K_d harán que el sistema tenga una respuesta acotada en una región del plano ya que hay valores que hacen que la salida tienda a infinito; podemos realizar la misma prueba al control PID que es aplicar una señal escalón al sistema y ver su respuesta al menos de manera simulada con la herramienta de MatLAB Simulink, definiendo las ganancias de K_p , K_i y K_d para tener una respuesta acotada a los 2 segundos, haciendo que el sistema responda más rápido que con el control PD, vemos que para cada eje la salida del sistema en la posición de la bola resulta como se muestra en la Fig 5.8

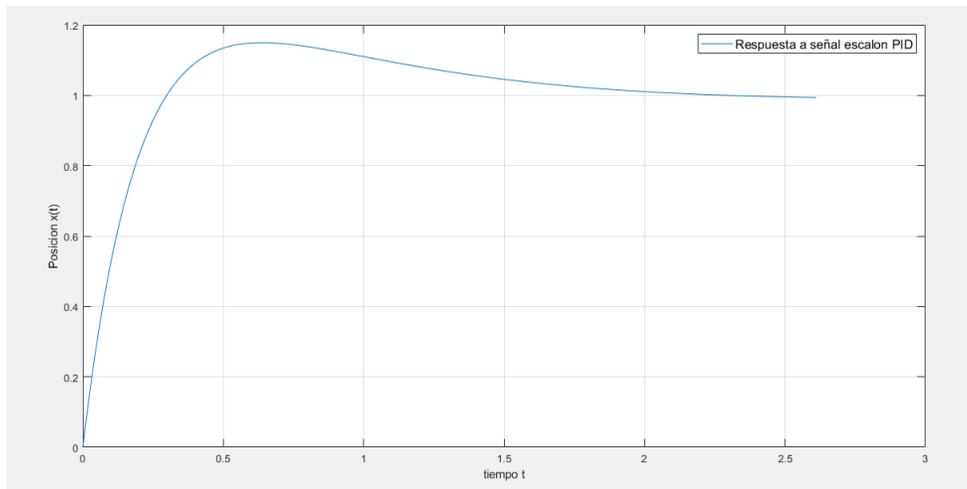


Figura 5.8: Respuesta a escalón en ambos ejes de $H_2(s)$ con PID

Cabe señalar que estas ganancias se obtuvieron a través de ir probando en el modelo simulado diferentes ganancias de K_p , K_i y K_d , se escogieron las ganancias una vez que la respuesta era un poco más cercana a la señal de entrada y con un sobrepaso menor. La respuesta al sistema de manera simulada parece responder muy rápido, esto no necesariamente pasa en el sistema real, dependerá de otros parámetros que se deben considerar al momento de implementar un control para que responda tan rápido como en la simulación, pero podemos decir que de manera simulada un control PID soluciona el problema de regulación. Si probamos este control PID para una señal que no es constante al igual que con el control PD podemos visualizar de manera simulada que tal responde un control PID a este sistema en un problema de control de seguimiento. Al igual que con el control PD las señales que se aplicaron al sistema fueron $r(t)_x = A \sin(\alpha t)$ y $r(t)_y = A \cos(\alpha t)$ y la respuesta del sistema plato y bola de manera simulada se muestra en la Fig. 5.9.

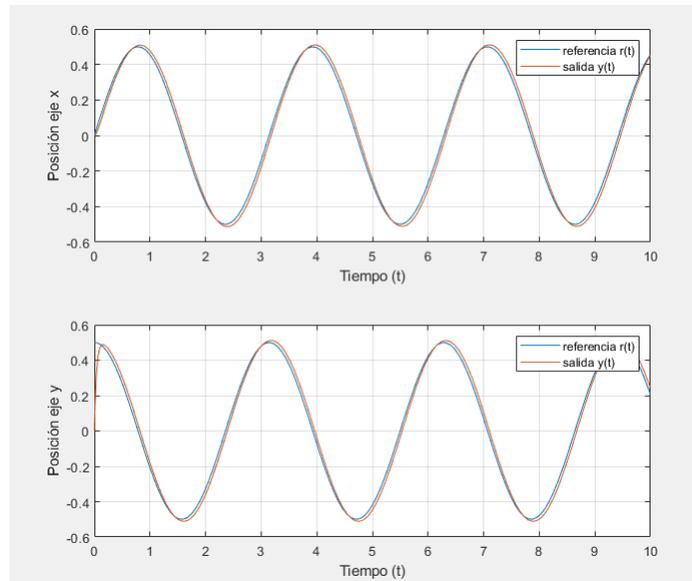


Figura 5.9: Respuesta una trayectoria de $H_2(s)$ en cada eje x, y con PID

Al igual que con el control PD si vemos el plano x, y con estas dos señales en cada eje se genera un círculo en el plato de modo que la bola tratara de seguir la trayectoria del círculo dentro del plato, al hacer las simulaciones con las ganancias sin modificación la respuesta del sistema se muestra en la Fig. 5.10.

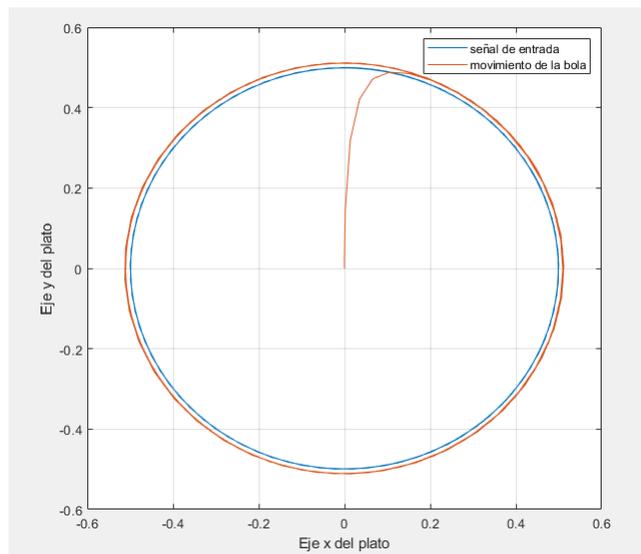


Figura 5.10: Respuesta una trayectoria de $H_2(s)$ en el plano x, y con PID

Como se observa la respuesta del sistema de plato y bola es más cercana a la señal de entrada, con esto podemos decir que al menos de manera simulada, se puede controlar el sistema con un control PID, el control resuelve de manera simulada el problema de control de seguimiento y de regulación, aunque para ambos casos la respuesta del control es muy rápida y esto no necesariamente se va a cumplir en el sistema real ya que influyen otros parámetros que hacen más lento la señal de control, además de que el controlador es una PC se debe cambiar un poco el control al momento de implementarlo.

Antes de implementar los controladores PD y PID, se debe analizar el lazo cerrado con otra técnica de control; partiendo de la teoría del control PID se puede diseñar un control PID de orden fraccionario, aunque para poder hacer esto es fundamental entender de forma general lo que es el cálculo fraccionario.

5.4. Cálculo Fraccionario

El Cálculo fraccionario surge en 1695, a partir de una carta a L'Hôpital de Leibniz, donde le pregunta que si puede generalizarse el significado de las derivadas con orden entero y no entero, esto hizo que L'Hôpital sintiera cierta curiosidad y le respondiera con la siguiente pregunta: "¿Qué pasaría si el orden fuera 1/2?", Leibniz en una carta del 30 de septiembre de 1695 le respondió: "Llevará a una paradoja, de la que un día se obtendrán consecuencias útiles", esta pregunta planteada por Leibniz fue un tema de discusión por más de 300 años y hoy en día se conoce como cálculo fraccionario como una generalización del cálculo ordinario tanto diferenciación como integración con orden arbitrario no entero. (Monje y cols., 2010).

Debido a toda la historia de las teorías del cálculo fraccionario hay muchas definiciones del cálculo fraccionario, sin embargo, para la implementación del cálculo fraccionario en sistema de control destacan tres, la definición de Grunwald-Letnikov, la definición de Riemann-Liouville y la definición de Caputo. (Shah y Agashe, 2016).

5.4.1. Definición de Grunwald-Letnikov

La definición de Grunwald-Letnikov se expresa como se muestra a continuación:

$${}_a D_t^\alpha = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^r \binom{n}{r} f(t - rh) \quad (5.23)$$

El valor del coeficiente binomial esta dado por:

$$\binom{n}{r} = \frac{\Gamma(n+1)}{\Gamma(r+1)\Gamma(n-r+1)} \quad (5.24)$$

Donde $\Gamma()$ se refiere a la función gamma definida por la siguiente ecuación:

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (5.25)$$

para $R(Z) > 0$.

La definición de Grunwald-Letnikov es usada para evaluaciones numéricas, esta definición es de mucha utilidad para obtener un valor numérico como solución de una ecuación diferencial fraccionaria. (Shah y Agashe, 2016).

5.4.2. Definición de Riemann-Liouville

Liouville definió la derivada de orden arbitrario como una serie infinita con la condición de que el orden debe limitarse a solo aquellos valores para los cuales la serie converge, de esta forma fue el primero en intentar resolver ecuaciones diferenciales con operadores fraccionarios. Por otra parte, Riemann uso la generalización de una serie de Taylor para derivar una formula relacionada con la integración de un número arbitrario, ambos trabajos se resumen en una sola formula que se define como se muestra a continuación:

$${}_a D_t^\alpha = D^n J^{n-\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt} \right)^n \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (5.26)$$

Donde n es un número entero que satisface la condición $n - 1 < \alpha < n$, α es un número real, J es el operador integral y a y t son los límites de integración, esta definición tanto para la integral como la derivada se utiliza mucho en la solución analítica de funciones simples como e^t o $\cos(t)$. (Shah y Agashe, 2016).

5.4.3. Definición de Caputo

Aunque hay muchas definiciones de integrales con orden fraccionario, la definición de Caputo es muy popular para aplicaciones de ingeniería, ya que esta definición permite condiciones iniciales en $y(0)$ y $y^{0.5}(0)$, además que la derivación fraccionaria es acotada y está definida como se muestra a continuación:

$${}_a D_t^\alpha = \frac{1}{\Gamma(n - \alpha)} \int_a^t \frac{f^n(\tau)}{(t - \tau)^{\alpha - n + 1}} d\tau \quad (5.27)$$

Donde n es un entero que satisface la condición $n - 1 < \alpha < n$, α es un número real y a y t son los límites de integración, esta definición es similar a la de Riemann-Liouville, aunque la definición de Riemann-Liouville sirve para una notación general de la derivada y la de Caputo es una notación general de la derivada de manera clásica. (Shah y Agashe, 2016).

5.5. Control PID Fraccionario

Las teorías del controlador PID fraccionario amplían la teoría del PID tradicional, no como una nueva teoría, más bien como complemento de esta teoría, aunque de forma matemática, el cálculo fraccionario tiene muchas definiciones que se pueden demostrar, pero como tal no tienen un sentido al momento de aplicar el cálculo fraccionario de forma real, por ejemplo, en el diseño de controladores. Un PID tradicional, como ya se mencionó, consta de 3 partes, una proporcional (P), una integral (I) y una derivativa (D), de forma física cada bloque funciona para compensar diferentes cosas al momento de implementar un controlador PID, la parte proporcional aumenta la velocidad de respuesta y disminuye el error en estado estacionario, la parte integral se encarga de eliminar el error en estado estacionario y la parte derivativa se encarga de la sensibilidad al ruido, de modo que si hay una respuesta física para cada bloque del controlador. (Monje y cols., 2010).

Dependiendo de las ganancias K_p , K_i y K_d los bloques del PID afectaran al control respectivamente, de esta forma tenemos 3 grados de libertad para ajustar el control según lo requiera la planta, el control PID fraccionario consiste en generar una derivada y una integral fraccionaria para tener 5 grados de libertad, es decir, las ganancias K_p , K_i y K_d y los valores fraccionarios para la derivada y la integral respectivamente, esto genera un control más adecuado al sistema a controlar ya que con 5 grados de libertad se tendría un control con mejor rendimiento a la planta que con un control de 3 grados de libertad, en teoría, la dificultad o el problema que genera este control es que en algunos casos es complicado sintonizar el controlador PID con 3 grados, por lo que con 5 grados la sintonización es todavía más complicada, pero la idea es adaptar lo mejor posible el control al sistema a la planta para obtener una mejor respuesta. (Shah y Agashe, 2016). Si vemos el control PID fraccionario en el lazo de control resulta como se muestra en la Fig. 5.11.

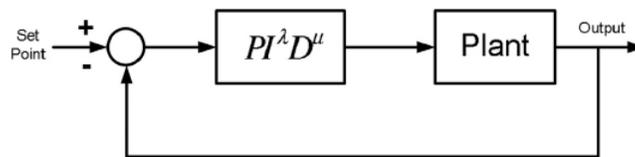


Figura 5.11: Lazo de control con PID fraccionario

Al igual que con el PID tradicional al PID fraccionario también se le puede aplicar la transformada de Laplace para obtener su función de transferencia, aplicando la transformada pasamos al dominio de s generando que las ecuaciones se conviertan en un problema algebraico más fácil de analizar, desde el punto de vista matemático, debido a que los valores fraccionarios del PID se convierten en valores exponenciales de s , de modo que la función de transferencia del control PID fraccionario resulta como:

$$G(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (5.28)$$

Donde K_p , K_i y K_d son las ganancias para cada bloque de control y λ y μ son el orden fraccionario para la integral y la derivada respectivamente. Es importante aclarar que λ y μ solo pueden estar en el rango de $(0 \leq \lambda \leq 1)$ y $(0 \leq \mu \leq 1)$, esto es por la interpretación del control PID ya que si los valores de λ y μ son iguales a 1 sería un control PID tradicional, si λ es igual a 0 y μ igual a 1 sería un control PD tradicional, de forma inversa si λ vale 1 y μ es igual a 0 sería un control PI tradicional, si λ y μ son iguales a cero sería un controlador P solamente. (Shah y Agashe, 2016). De forma gráfica se puede hacer un análisis de la región donde un PID fraccionario entra y se muestra en la Fig. 5.12.

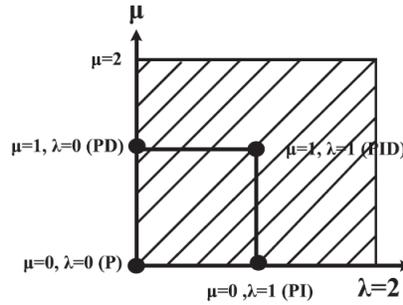


Figura 5.12: Región de operación de PID fraccionario

Al igual que con el PID tradicional podemos simular con esta función de transferencia para obtener una respuesta del sistema de plato y bola, sin embargo para poder utilizar funciones de transferencia con orden fraccionario, es necesario utilizar un Toolbox especial en MatLAB creado por Tepljakov Aleksei, Petlnkov Eduard, Belikov Juri, es una herramienta especial donde permite en MatLAB generar funciones de transferencia de orden fraccionario y además generar PID de orden fraccionario con la sintonización que requiera según la planta donde este implementado el control, la herramienta se llama FOMCON ("Fractional Order Modeling and Control") y se encuentra en los Toolbox que se pueden descargar en MatLAB, de modo que se utilizara esta herramienta para simular y sintonizar el controlador PID fraccionario junto con el sistema de plato y bola. (Aleksei, Eduard, y Juri, 2012).

5.5.1. Control PD Fraccionario para el Sistema de Plato y Bola

De la misma forma que con el PID normal primero se va a simular un control PD fraccionario, recordando la función de transferencia del control PID fraccionario:

$$G(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (5.29)$$

Si tenemos que $K_i = 0$ tenemos un PD fraccionario y a igual que con el PD normal podemos simular el lazo cerrado con el sistema de plato y bola, recordando la función de transferencia del plato y bola:

$$H(s) = \frac{\theta}{s^2} \quad (5.30)$$

Asignamos valores a K_p , K_d y para el caso de μ se puede simular con todos los valores en el intervalo de 0 a 1 de 0.1 a 0.9 y ver en que rango opera mejor de μ para después sintonizar K_p y K_d lo mejor posible. Si obtenemos la función de transferencia en cada valor de 0.1 a 1 de μ para visualizar mejor el comportamiento y obtenemos su diagrama de bode de cada uno vemos todos los diagramas de bode como se muestra en la Fig. 5.13.

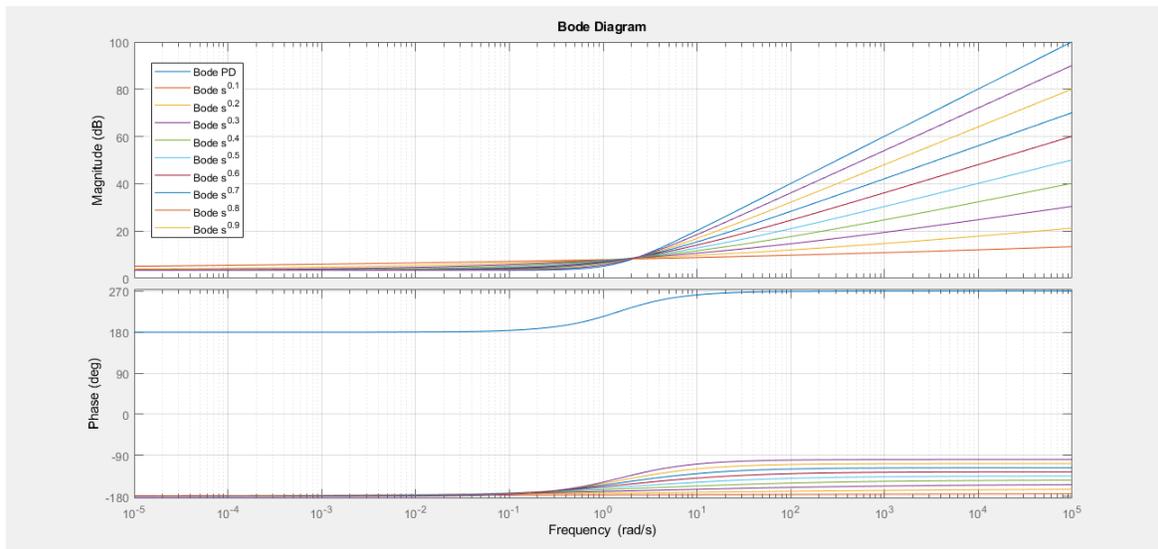


Figura 5.13: Diagramas de Bode de control PD fraccionario de 0.1 a 1

Como vemos, la magnitud a medida que se acerca al PD tradicional tiene el mismo comportamiento y al disminuir se acerca a un control proporcional, por otra parte, la fase resulta tener casi el mismo comportamiento con la diferencia de tener un atraso importante respecto al PD tradicional. Una vez que se tiene el lazo cerrado se simula la respuesta a una señal escalón, si juntamos las respuestas para comparar todas la graficas resulta como se muestra en la Fig. 5.14

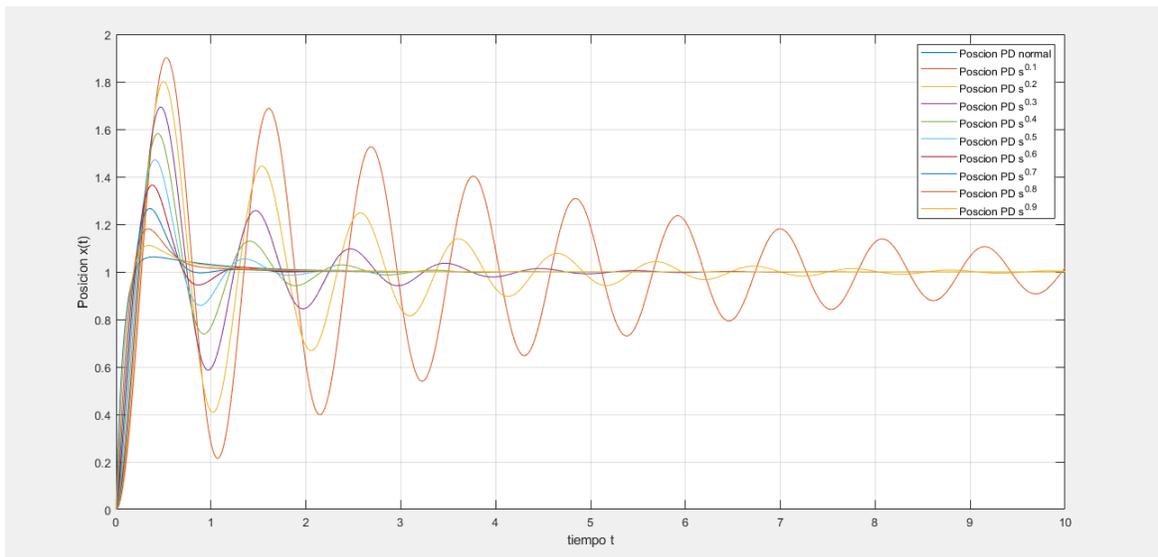


Figura 5.14: Respuestas a señal escalón de control PD fraccionario de 0.1 a 1

Como se puede observar el sistema oscila demasiado a medida que disminuye el valor fraccionario, si solo escogemos valores cercanos a 1 con valor mínimo de 0.7 como orden fraccionario se observa que son los mejores casos donde la respuesta no oscila tanto y aunque si hay sobrepaso, los valores del PD con el orden fraccionario de 0.7 a 1 resultan como se muestra en la Fig. 5.15

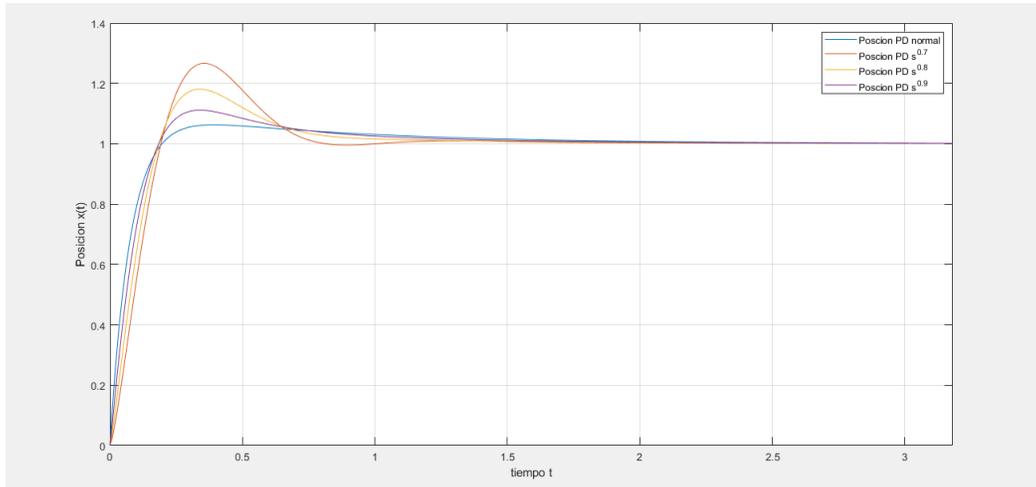


Figura 5.15: Respuestas a señal escalón de control PD fraccionario de 0.7 a 1

Como se observa en la gráfica a medida que se acerca el valor fraccionario a 1, es decir a un control PD normal, mejora el control en cuestión del sobrepaso, sin embargo, el PD fraccionario de 0.7 a pesar de tener mayor sobre paso estabiliza el sistema más rápido, se debe aclarar que todo esto es simulado, una vez que se implemente se verificara el comportamiento real de control fraccionario, aunque primero se debe simular el comportamiento con un control PID fraccionario.

5.5.2. Control PID Fraccionario para Sistema de Plato y Bola

De la misma forma que con el PID tradicional se obtuvo la función de transferencia del control PID, recordando que para el PID fraccionario resulta de la siguiente forma:

$$G(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (5.31)$$

Debido a que ya se simulo el PD fraccionario y se encontró que el comportamiento del sistema respecto al controlador PD fraccionario se estabiliza mejor cuando el valor de μ se acerca a 1, para obtener el PID fraccionario se dejó fijo un valor de $\mu=0.8$, esto se hizo para observa si cambia considerablemente el comportamiento al poner la parte integral al controlador, y se hace lo mismo que con el PD que es asignar valores de λ de 0.1 a 1, donde 1 sería un PID con derivada fraccionaria en $\mu=0.8$, si hacemos el mismo análisis de los diagramas de bode para cada valor de λ resulta como se muestra en la Fig. 5.16.

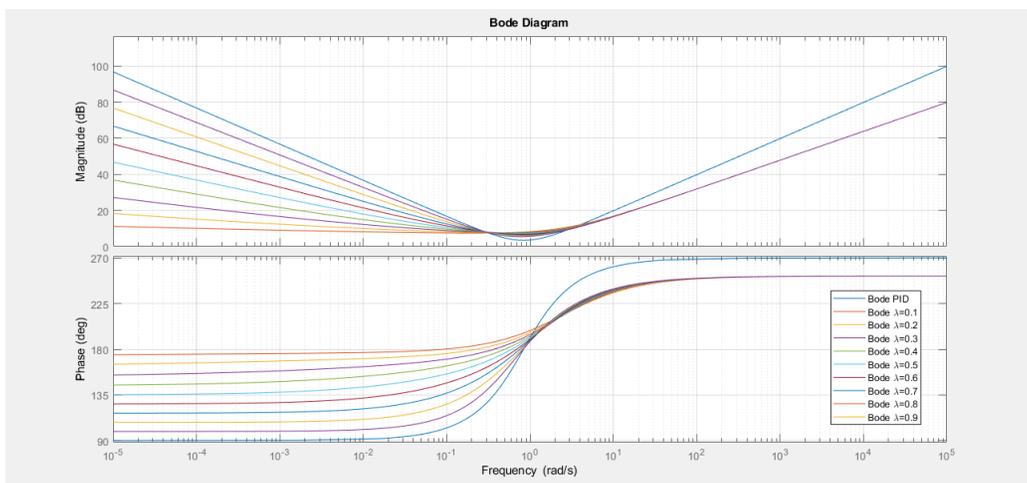


Figura 5.16: Diagramas de Bode de control PID fraccionario de 0.1 a 1

Después se simula la respuesta a la señal escalón, la respuesta de cada controlador resulta como se muestra en la Fig. 5.17

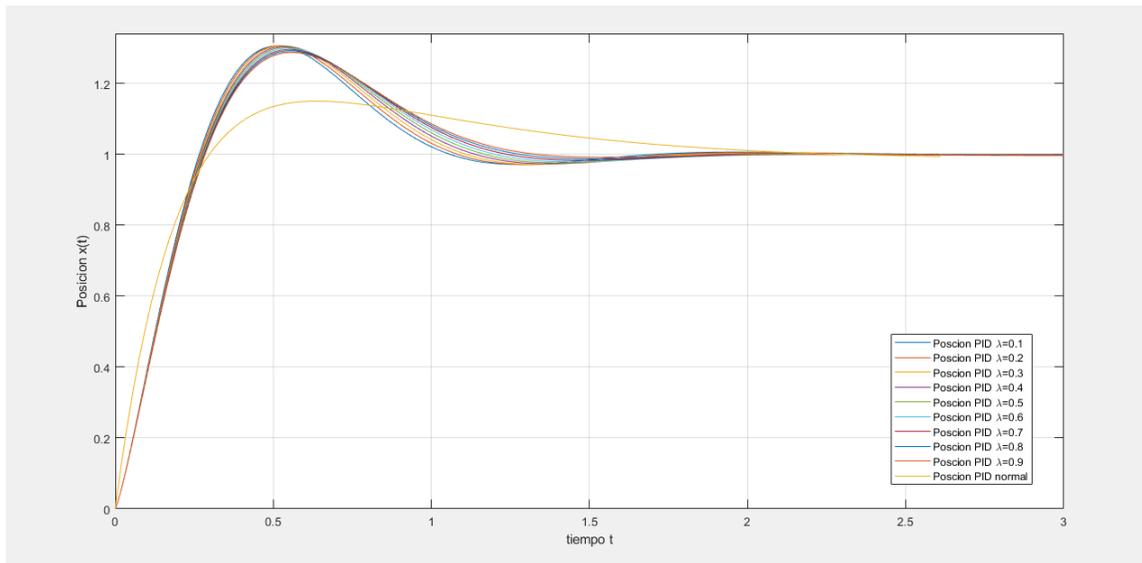


Figura 5.17: Respuestas a señal escalón de control PID fraccionario de 0.1 a 1

Como se observa el sobre paso sigue siendo mayor en comparación con el control PID tradicional, sin embargo todos los controladores PID responden más rápido que el tradicional, a pesar de esto no hay un cambio considerable en el intervalo de λ es decir que no importa mucho el valor que se le asigne ya que el comportamiento es similar en todos los valores de λ , con esto se llega a la conclusión que un PD fraccionario funcionaria mejor para este sistema que un PID fraccionario, al menos de manera simulada, habrá que comprobarlo cuando se implementen los controladores. Por el momento podemos decir que de manera simulada los controladores PD y PID tradicionales responden mejor para este sistema que los PD y PID fraccionarios, aunque los controladores fraccionarios parecen responder más rápido pero con mayor sobre-paso, una vez que implementen se llegara a una conclusión más certera.

Para poder implementar el control al sistema real hay que tomar en cuenta que el sistema donde se va a programar las diferentes técnicas de control, además de que va obtener la posición de la bola en el plato y mandar la señal a cada servomotor es una PC se debe hacer un cambio a los controladores por implementar, ya que como tal no podemos implementar el control en tiempo continuo para cualquier microprocesador o microcontrolador, se debe cambiar los controladores en tiempo discreto.

Capítulo 6

Control en Tiempo Discreto

El uso de controladores digitales nos ayuda a alcanzar el desempeño óptimo al momento de implementar cualquier técnica de control en una computadora o microprocesador, todas las computadoras trabajan con señales digitales; de modo que es necesario realizar el cambio en las señales para que la computadora pueda interpretar y aplicar al sistema real, por esta razón existe el control en tiempo discreto, la diferencia entre este control y el de tiempo continuo es principalmente es por las señales que se utilizan. (Ogata, 1996).

Existen dos tipos de señales, las señales en tiempo continuo son aquellas que se definen sobre un intervalo continuo de tiempo t y la amplitud puede tener un intervalo continuo de valores o solamente un número finito de valores distintos; a las señales en tiempo continuo también se les llama señales analógicas, por otra parte, una señal en tiempo discreto es una señal definida solo en valores discretos, estos son aquellos valores en los que la variable independiente t está cuantificada a través de un muestreo en un intervalo finito de tiempo t , a una señal en tiempo discreto también se le conoce como señal digital, aunque también puede referirse a una señal de datos muestreados. (Ogata, 1996). A continuación, se muestra en la Fig. 6.1 un ejemplo de ambas señales a través del tiempo t .

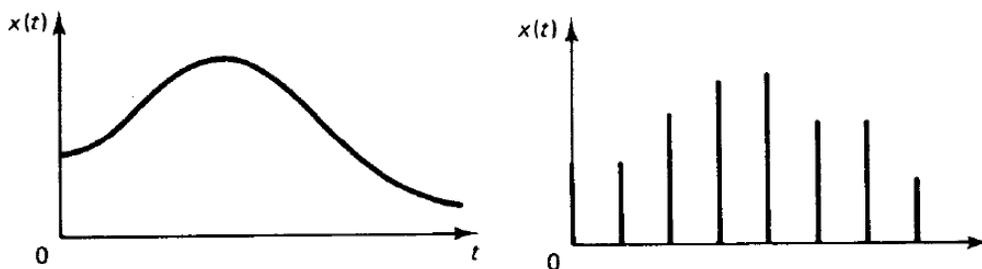


Figura 6.1: Muestreo en tiempo continuo y tiempo discreto t

Los sistemas de control en tiempo discreto son aquellos sistemas en los cuales una o más de las variables pueden cambiar solo en valores discretos, estos instantes se denotan como kT donde $k = 0, 1, 2, \dots$ y T es el tiempo entre cada muestra de la señal, siempre que se implemente un controlador en una computadora para controlar un sistema real se debe hacer un análisis en tiempo discreto y al igual que las señales en tiempo continuo que se pueden describir mediante ecuaciones diferenciales, los sistemas en tiempo discreto también se pueden describir mediante ecuaciones en diferencias una vez que se hace la discretización de las señales en tiempo continuo, para hacer esta discretización de sistemas es necesario el concepto de transformada z . (Ogata, 1996).

6.1. Transformada Z

Una herramienta matemática muy utilizada para el análisis de sistemas de control en tiempo discreto es la transformada z , el papel de la transformada z en sistemas de tiempo discreto es similar al de la transformada de Laplace en sistemas de tiempo continuo, se define un sistema de control en tiempo discreto como una ecuación en diferencias lineal, que caracteriza la dinámica del sistema y para determinar la respuesta del sistema a una entrada, se debe resolver la ecuación en diferencias, por esta razón se hace el cambio con la transformada Z en el dominio de la variable z . La solución a la ecuación en diferencias se convierte en un problema algebraico, similar a la transformada de Laplace, lo cual nos ayuda a hacer un mejor análisis de la dinámica del sistema, solo que en este caso en tiempo discreto. (Ogata, 1996).

Sea una señal en tiempo discreto $x(0), x(T), x(2T), \dots$, donde T es el periodo de muestreo se define como $x(kT)$, se le aplica la transformada Z como se muestra a continuación:

$$X(z) = Z[x(t)] = Z[x(kT)] = \sum_{k=0}^{\infty} x(kT)z^{-k} \quad (6.1)$$

A esta transformada Z se le conoce como transformada Z unilateral, supone que $x(t) = 0$ para $t < 0$ o $x(kT) = 0$ para $k < 0$, adicional a que z es una variable compleja. Para la mayoría de las aplicaciones en ingeniería, la transformada z unilateral tendrá solución en forma cerrada a su región de convergencia, cuando $X(z)$ es una serie infinita en z^{-1} , converge fuera del círculo $|z| = R$, donde R se conoce como radio de convergencia absoluta, sin embargo, al utilizar el método de la transformada z para resolver problemas en tiempo discreto, no es necesario especificar los valores para los cuales $X(z)$ converge. (Ogata, 1996).

Al igual que con la transformada de Laplace es necesario el concepto de transformada Z inversa, donde una función $X(Z)$ pasa a $x(kT)$. En la transformada inversa de z solo se obtiene la secuencia de tiempo en los instantes de muestreo, es decir que la transformada inversa de $X(z)$ da como resultado una ecuación única $x(kT)$ pero no da una ecuación única de $x(t)$; por esta razón existen diferentes métodos para obtener la transformada z inversa, entre muchos métodos los cuatro principales son los siguientes:

- Método de la división directa
- Método computacional
- Método de expansión en fracciones parciales
- Método de la integral de inversión

Para cualquiera de estos métodos es necesario tener en cuenta los polos y ceros en el plano z , ya que en aplicaciones de ingeniería se tiene la transformada z de la siguiente forma:

$$X(z) = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_m}{z^n + a_1 z^{n-1} + \dots + a_n} \quad (6.2)$$

Donde $m \leq n$, también se representa como:

$$X(z) = \frac{b_0(z - z_1)(z - z_2)\dots(z - z_m)}{(z - p_1)(z - p_2)\dots(z - p_n)} \quad (6.3)$$

Donde los $p_i (i = 1, 2, \dots, n)$ son los polos de $X(z)$ y $z_j (j = 1, 2, \dots, m)$ son los ceros de $X(z)$, en base a la ubicación de los polos y ceros de $X(z)$ se determinara las características de $x(k)$. Este cambio nos ayudara mucho después al momento de implementar el control en una computadora para después aplicarlo al sistema real, otra herramienta que se utiliza mucho al implementar un control en tiempo discreto es la correspondencia entre el plano complejo s y z . (Ogata, 1996).

6.2. Correspondencia entre el Plano s y z

En el diseño de un sistema de control en tiempo continuo la localización de polos y ceros en el plano s es de gran importancia para predecir el comportamiento dinámico del sistema, en el caso del diseño de sistemas de control en tiempo discreto es igual de importante la ubicación de polos y ceros en el plano z por esta razón hay una ecuación que relaciona ambas variables y es la siguiente:

$$z = e^{Ts} \quad (6.4)$$

Esto significa que un polo en el plano s se puede ubicar en el plano z mediante esta transformación, si tenemos que $s = \sigma + j\omega$ entonces:

$$z = e^{T\sigma + jT\omega} = e^{T\sigma} e^{j(T\omega + 2\pi k)} \quad (6.5)$$

De esta última ecuación vemos que los polos y ceros en el plano s , donde las frecuencias difieran de múltiplos enteros de la frecuencia $\frac{2\pi}{T}$, corresponden a las mismas localizaciones en el plano z . Esto significa que por cada valor de z existirá un número infinito de valores de s . Dado que σ es negativo en el semiplano izquierdo del plano s , corresponde a z de la siguiente forma:

$$|z| = e^{T\sigma} < 1 \quad (6.6)$$

El eje $j\omega$ en el plano s corresponde a $|z| = 1$, es decir el eje imaginario corresponde al círculo unitario y el interior del círculo unitario corresponde al semiplano izquierdo del plano s , debido a esta relación de correspondencia los polos y ceros de cualquier sistema también se pueden ubicar en el plano de z y al igual que en el análisis de la función de transferencia, también se puede hacer un análisis en tiempo discreto. (Ogata 1996)

6.3. Lugar Geométrico de las Raíces en Tiempo Discreto

Al igual que con la transformada de Laplace si se analiza el lugar geométrico desde el punto de vista en tiempo discreto se puede obtener información desde el punto de vista de z , la información principal es que al diseñar el control en tiempo discreto podemos simular de manera discreta tanto el controlador como el sistema similar a las simulaciones en tiempo continuo y predecir en gran medida la respuesta al control que se implemente en tiempo discreto. Es de vital importancia entender e implementar el control en tiempo discreto ya que es la forma en que implementa un controlador real a cualquier sistema que se controle a través de una computadora o microcontrolador y en teorías de control como el PID, nos ayuda a obtener las ganancias que estabilizan el sistema por medio de simulaciones en tiempo discreto con el modelo ya discretizado, esta simulación será lo más aproximado al control real por lo tanto las ganancias en teoría deberán ser similares para el control real y lo simulado en tiempo discreto. Un método que se utiliza mucho para obtener el lugar geométrico de las raíces en tiempo discreto de cualquier sistema es con el retén de orden cero.

6.3.1. Retén de Orden Cero

Para el caso de sistemas lineales y en tiempo continuo se puede hacer una discretización del sistema para pasar el sistema a tiempo discreto a través de muchos métodos, en muchos casos se considera que se puede obtener una mejor discretización del sistema con ayuda de un retén de orden cero, el retén de orden cero dependerá del periodo T entre muestra y muestra para que funcione mejor en cualquier señal continua ya que si se quiere reconstruir la señal original en tiempo continuo, debe existir una frecuencia mínima en la operación de muestreo, esto se define a través del teorema de muestreo y se expresa con la siguiente ecuación:

$$\omega_s > 2\omega_1 \quad (6.7)$$

El teorema implica que si $\omega_s > 2\omega_1$ entonces, a partir del conocimiento de la señal muestreada es teóricamente posible reconstruir con exactitud la señal en tiempo continuo original. (Ogata, 1996).

De forma matemática la función de transferencia de un retén de orden cero es:

$$G_{ho}(s) = \frac{1 - e^{-Ts}}{s} \quad (6.8)$$

Si obtenemos su respuesta en frecuencia mediante la sustitución de s por $j\omega$ la magnitud del retén de orden cero resulta:

$$|G_{ho}(j\omega)| = T \left| \frac{\text{sen}(\omega T/2)}{\omega T/2} \right| \quad (6.9)$$

De modo que la magnitud se hace cero en la frecuencia igual a la frecuencia de muestreo y en múltiplos enteros de la frecuencia de muestreo. Además de eso si se observa la fase del retén de orden cero se tiene que $\text{sen}(\omega T/2)$ adopta valores positivos y negativos a medida que ω se incrementa de modo que la curva de fase es discontinua en $\omega = k\omega_s = 2\pi k/T$, a esta discontinuidad se le puede considerar como un corrimiento de fase de ± 180 grados.

Con esta discontinuidad podemos ver que el espectro en frecuencia de la salida del retén de orden cero incluye las componentes complementarias, debido a que las características muestran que la magnitud es diferente de cero para $|\omega| > 1/2\omega_s$, esto en palabras simples se genera un filtro paso-bajas a través del retén de orden cero para obtener la señal real en tiempo continuo a partir de la señal muestreada, este cambio nos ayuda a obtener con ayuda del retén de orden cero la función de transferencia en tiempo discreto que será única para un periodo T y podemos asegurar que si pasamos la función en tiempo discreto será la función de transferencia en tiempo continuo que se obtuvo al principio. (Ogata 1996).

6.3.2. Lugar Geométrico de las Raíces en Tiempo Discreto del Sistema de Plato y Bola

Utilizando el retén de orden cero podemos obtener tanto los polos como los ceros del sistema al igual que en tiempo continuo, esta función de transferencia respecto a z se puede obtener también con el software de MATLAB, recordando que la función de transferencia en tiempo continuo es:

$$G(s) = \frac{\theta}{s^2} \quad (6.10)$$

Y recordando que $\theta = \frac{-mg}{m + \frac{J}{r^2}}$ junto con que ya tenemos el sistema de plato y bola construido, podemos asignar los parámetros a θ en función a la bola del sistema, los parámetros que se tomaron fueron de un balón que será la bola en el plato, los parámetros son:

- la masa $m = 0.017$ [kg].
- el radio de la bola $r = 0.0085$ [m].
- la aceleración gravitacional de $g = 9.81$ [$\frac{m}{s^2}$]
- la inercia rotacional de una esfera solida $J = \frac{2}{5}mr^2$.

resulta que para este sistema de plato y bola $\theta = -7.0071$.

Para utilizar la herramienta de MATLAB y obtener la función de transferencia en tiempo discreto además de definir parámetros, es necesario definir el tiempo de muestreo T , este parámetro es fundamental para la función de transferencia en tiempo discreto, para este caso se usó como $T = 0.1$ segundos y como ya se obtuvo el valor de θ y con la relación entre la variable s y z , además de la herramienta del retén de orden cero obtenemos la función en tiempo discreto para el sistema de plato y bola y resulta como se muestra a continuación:

$$G(z) = \frac{-0.03504z - 0.03504}{z^2 - 2z + 1} \quad (6.11)$$

Con esta función de transferencia obtenemos el lugar geométrico de este sistema de plato y bola en tiempo discreto como se muestra en la Fig 6.2.

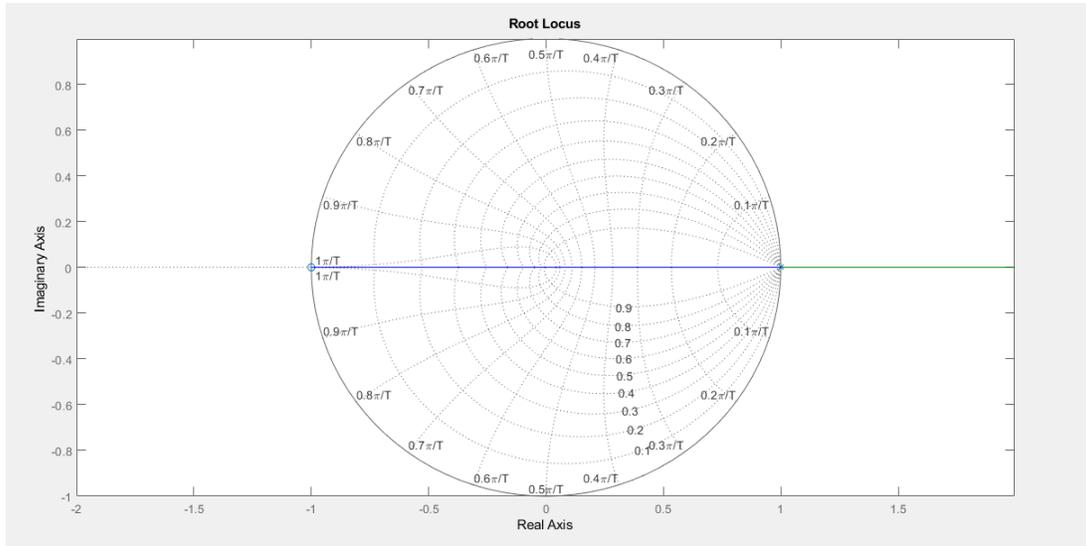


Figura 6.2: Lugar geométrico de las raíces en tiempo discreto de este sistema de plato y bola

Como vemos tiene un polo y un cero justo en el borde del círculo unitario, parecido al modelo en tiempo continuo que estaban ambos polos en el origen esto nos indica que el comportamiento del sistema será similar en tiempo continuo como en tiempo discreto, al menos para esta discretización.

Una vez que tenemos el lugar geométrico de las raíces del sistema en tiempo discreto, podemos empezar a diseñar el controlador para la planta en tiempo discreto por medio de esta nueva función de transferencia y simular para ver el comportamiento del sistema. Como hay una relación entre las variables s y z se puede realizar un cambio de variable, sin embargo, para poder hacer este cambio al tiempo discreto en z no es muy recomendable usar la relación entre s y z vista anteriormente, lo ideal es hacer una aproximación lineal en donde por medio de técnicas matemáticas se puede pasar del espacio de s al de z de manera más sencilla, las dos aproximaciones más comunes son la aproximación rectangular o de Euler y la aproximación trapezoidal o de Tustin.

6.4. Aproximación Rectangular o de Euler

La aproximación rectangular o de Euler parte de la forma en que se realiza el muestreo de una señal continua y con valor kT donde $k = 0, 1, 2, \dots$ que se refiere al tiempo entre cada muestra, genera una señal discreta aproximándose a la señal original, esta aproximación hace que un valor de la señal $y(t)$ se mantenga en un tiempo kT y pasando ese tiempo tome otra muestra de la señal en $(k + 1)T$ y mantenga otra vez ese valor hasta el siguiente tiempo de muestreo, al tomar varias muestras se observa la figura de rectángulos que de ahí el nombre de la aproximación, si se ve en un plano grafico resulta como se muestra en la Fig. 6.3

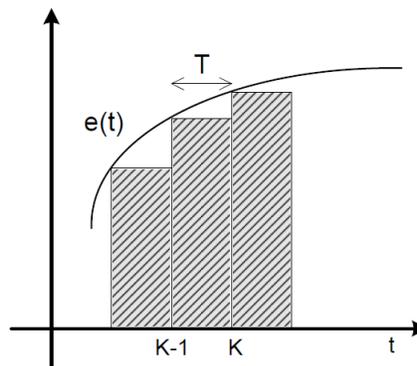


Figura 6.3: representación cartesiana de la aproximación rectangular

Visto de forma matemática se genera una ecuación en diferencias y se analiza en tiempo discreto, sin embargo, es más sencillo trabajar en el dominio de s y pasar al tiempo discreto a partir de la función de transferencia o de cualquier señal en el dominio de s . Esto es posible ya que esta aproximación nos permite hacer un cambio respecto a la relación entre la variable s y z , la relación matemática se expresa de la siguiente forma:

$$s \approx \frac{1 - z^{-1}}{T} \quad (6.12)$$

Donde T es el periodo de muestreo. (Ogata, 1996).

Esta aproximación nos ayuda a poder generar una ecuación en el dominio de z más sencilla de utilizar, desde el punto de vista matemático, y poder generar cualquier ecuación en tiempo discreto partiendo de la ecuación en tiempo continuo. Esta aproximación se utiliza mucho para generar controladores en tiempo discreto partiendo de la función de transferencia en tiempo continuo, por ejemplo, un PD o un PID como se muestra a continuación.

6.4.1. Control PD con Aproximación Rectangular

Como ya se obtuvo la función de transferencia del control PD y realizaron las simulaciones correspondientes en el sistema de plato y bola se puede implementar el control PD utilizando la aproximación rectangular partiendo de la función de transferencia del control PD:

$$G(s) = K_p(1 + T_d s) \quad (6.13)$$

Donde $T_d = \frac{K_d}{K_p}$.

Si $G(s) = G(\frac{1-z^{-1}}{T})$, donde T es el periodo y si se expresa a la función de transferencia en tiempo discreto de forma general de la siguiente forma:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.14)$$

Sustituimos $s = \frac{1-z^{-1}}{T}$ y obtenemos la función de transferencia en función de z de la siguiente forma:

$$H(z) = K_p(1 + T_d(\frac{1 - z^{-1}}{T})) \quad (6.15)$$

$$H(z) = K_p(1 + \frac{T_d}{T}) - (\frac{K_d}{T})z^{-1} \quad (6.16)$$

Si comparamos con la ecuación general $H(z)$ obtenemos que:

$$a_1 = 0, a_2 = 0 \quad (6.17)$$

$$b_0 = K_p(1 + \frac{T_d}{T}) \quad (6.18)$$

$$b_1 = \frac{-K_d}{T} \quad (6.19)$$

Esta ecuación $H(z)$ es la función de transferencia del control PD con la aproximación rectangular, si obtenemos la señal de control $u(kT)$ a partir de la señal de error $e(kT)$, ambas señales en tiempo discreto, resulta $u(kT)$ como:

$$u(kT) = -a_1 u((k-1)T) - a_2 u((k-2)T) + b_0 e(kT) + b_1 e((k-1)T) + b_2 e((k-2)T) \quad (6.20)$$

donde kT es el tiempo entre cada valor muestreado, u es la señal de control en el tiempo kT , $(k-1)T$ y $(k-2)T$ y $e(kT)$ es el error igual muestreado en los tiempos kT , $(k-1)T$ y $(k-2)T$, respectivamente, y por último sustituimos los valores de a_1 , b_0 y b_1 obtenemos la siguiente ecuación:

$$u = K_p(1 + \frac{T_d}{T})e(kT) + \frac{-K_d}{T}e((k-1)T) \quad (6.21)$$

Esta ecuación será la salida del control PD en tiempo discreto con la aproximación rectangular para cualquier sistema, el siguiente paso es simularlo para el sistema de plato y bola. (Salvá, Sánchez, Ortega, y Altamirano, s.f.).

6.4.2. Control PD con Aproximación Rectangular para el Sistema de Plato y Bola

Una vez que se obtiene un control PD en tiempo discreto, el siguiente paso es implementarlo al sistema y ver si soluciona el problema de control, tanto en regulación como en seguimiento, sin embargo con ayuda del software MatLAB podemos antes de implementar, simular el lazo cerrado del sistema, en este caso el sistema de plato y bola y sintonizar las ganancias del controlador, estas ganancias se pueden utilizar en la implementación real del control ya que en ambos casos se utilizara el control en tiempo discreto y como el controlador es la PC y todos los controladores que utilicen un procesador trabajaran en tiempo discreto de modo que esta simulación será lo más cercano al valor real, a diferencia de la simulación en tiempo continuo, cabe señalar que aunque la función de transferencia se obtuvo a partir del modelo en tiempo continuo, es igual de importante obtener las simulaciones en tiempo continuo y en tiempo discreto para tener una mejor idea del comportamiento del sistema.

Utilizando la función de transferencia del sistema de plato y bola:

$$G(s) = \frac{\theta}{s^2} \quad (6.22)$$

y como ya se obtuvo la función de transferencia en el dominio de z con un periodo $T = 0.1[s]$ y asignando los parámetros del modelo real la función de transferencia en tiempo discreto resulta:

$$G(z) = \frac{-0.03504z - 0.03504}{z^2 - 2z + 1} \quad (6.23)$$

Y ya obtenida la función de transferencia del controlador PD, podemos asignar ganancias para K_p y K_d , además de utilizar el mismo periodo T que del sistema, de modo que obtenemos la función del controlador PD en tiempo discreto como se muestra a continuación:

$$H(z) = \frac{-5.67z + 4.8}{z} \quad (6.24)$$

Obtenemos el lugar geométrico de las raíces del controlador PD en tiempo discreto y resulta la gráfica como se muestra en la Fig. 6.4

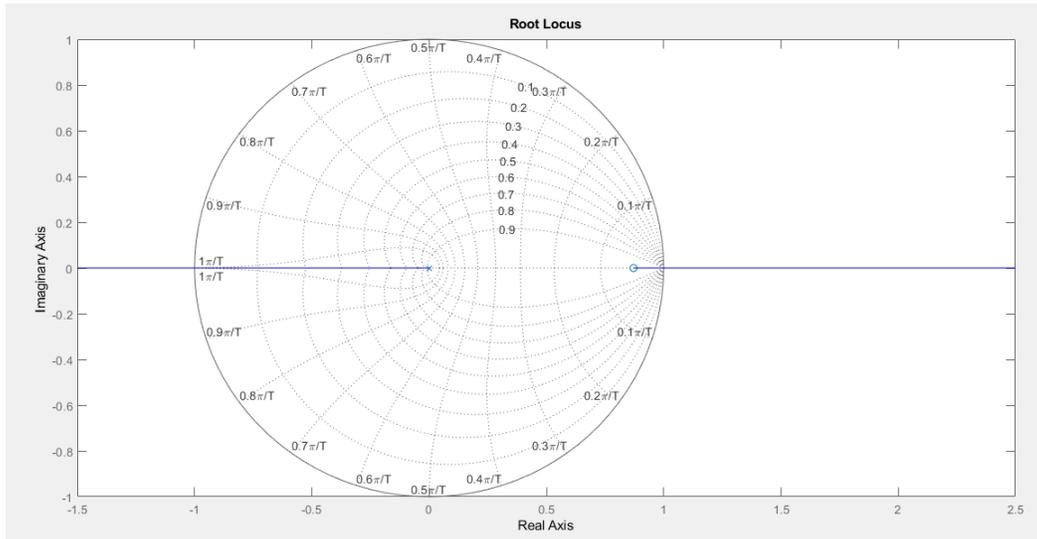


Figura 6.4: Lugar geométrico de las raíces de control PD en tiempo discreto

Implementamos el control PD en lazo cerrado a la función de transferencia y obtenemos la función de transferencia en tiempo discreto del lazo cerrado con el control PD que resulta ser:

$$H_{lc}(z) = \frac{0.1987z^2 + 0.03048z - 0.1682}{z^3 - 2z^2 + z} \quad (6.25)$$

Si obtenemos el lugar geométrico de las raíces de la función en lazo cerrado en tiempo discreto la gráfica resulta como se muestra en la Fig. 6.5

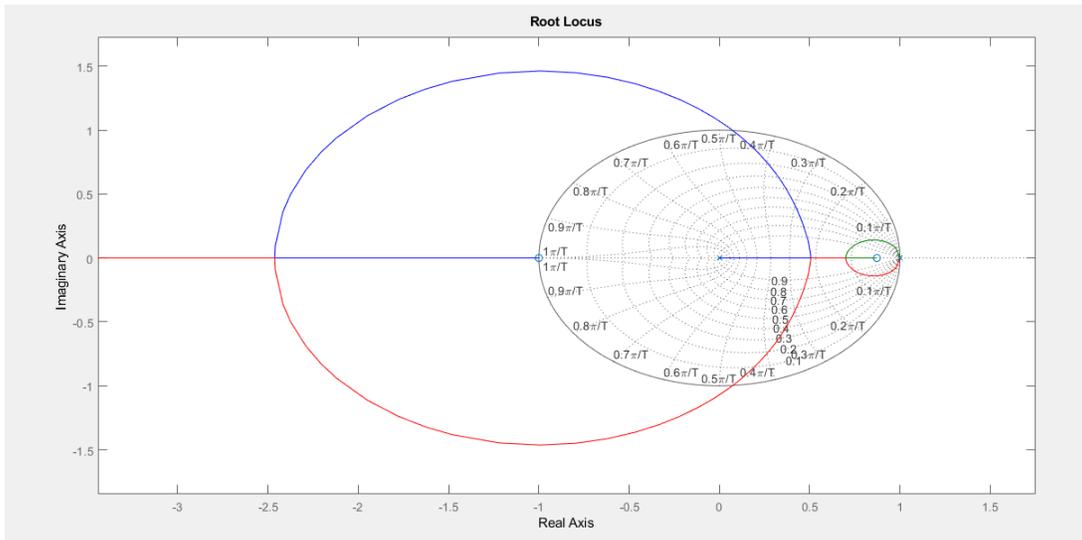


Figura 6.5: Lugar geométrico de las raíces del sistema en lazo cerrado

Por último, simulamos la respuesta a una señal escalón en función a la posición de la bola en el eje x, cabe aclarar que funciona igual para el caso del eje y ya que es la misma función de transferencia. A partir de esta simulación obtenemos la señal de control para ver como responderá los motores con esta aproximación, al simular con estos valores la respuesta del sistema y la señal de control resultan como se muestra en la Fig. 6.6.

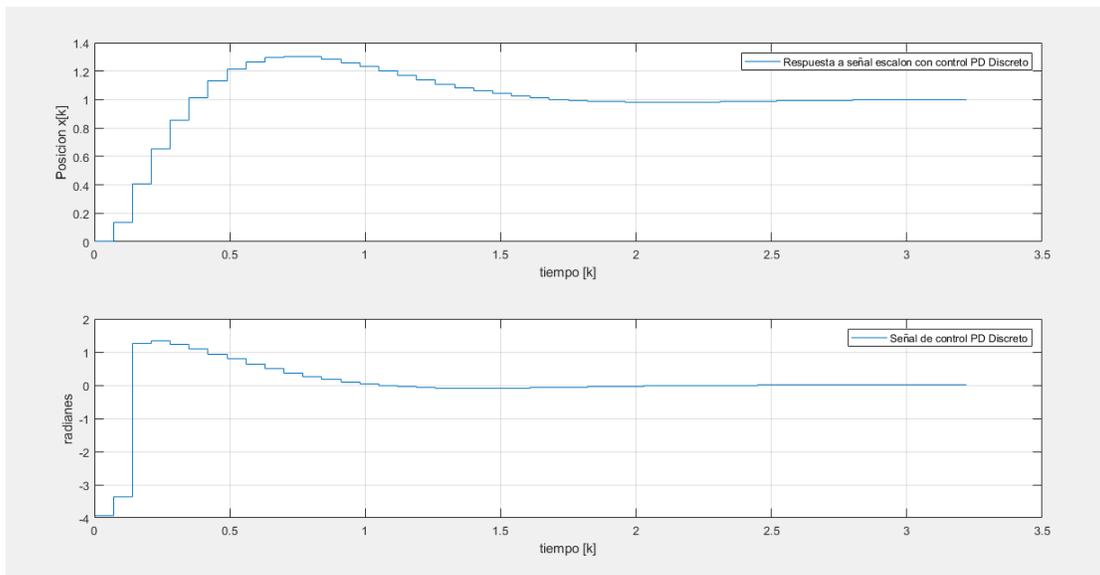


Figura 6.6: Respuesta a escalón y salida de control de sistema de plato y bola con control PD

Como vemos la señal de control es mucho más suave respecto a la simulación en tiempo continuo, al igual que la respuesta del sistema a la señal escalón, aproximadamente según esta simulación, el sistema estabilizará la bola 2 segundo después de que se le aplique la señal escalón, una vez que el control se implemente podremos saber en el sistema real el comportamiento que tendrá pero es una buena aproximación del comportamiento con este controlador, además de que se puede sintonizar una vez que este es implementado, aunque puede que mejore el control si se implementa un control PID en tiempo discreto.

6.4.3. Control PID con Aproximación Rectangular

Al igual que con el control PD ya se obtuvo la función de transferencia del control PID que es la siguiente:

$$H(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (6.26)$$

Donde $T_i = \frac{K_p}{K_i}$ y $T_d = \frac{K_d}{K_p}$.

Si sustituimos el valor de s con la aproximación rectangular que es $s \approx \frac{1-z^{-1}}{T}$ y sustituimos en $H(s)$ como $H(s) = H(\frac{1-z^{-1}}{T})$, donde T es el periodo y teniendo en cuenta la forma general de la función de transferencia en tiempo discreto que se expresa como se muestra a continuación:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.27)$$

Sustituimos $s = \frac{1-z^{-1}}{T}$ Y obtenemos la función de transferencia en función de z de la siguiente forma:

$$H(z) = K_p(1 + \frac{1}{T_i \frac{1-z^{-1}}{T}} + T_d(\frac{1-z^{-1}}{T})) \quad (6.28)$$

Si comparamos con la forma general de $H(z)$ obtenemos los siguientes valores:

$$a_1 = -1, a_2 = 0 \quad (6.29)$$

$$b_0 = K_p(1 + \frac{T}{T_i} + \frac{T_d}{T}) \quad (6.30)$$

$$b_1 = -K_p(1 + 2\frac{T_d}{T}) \quad (6.31)$$

$$b_2 = -K_p \frac{T_d}{T} \quad (6.32)$$

La ecuación $H(z)$ es la función de transferencia en tiempo discreto del control PID si se utiliza a partir del error en tiempo discreto $e(kT)$ obtenemos la señal de control $u(kT)$ como se muestra a continuación:

$$u(kT) = -a_1 u((k-1)T) - a_2 u((k-2)T) + b_0 e(kT) + b_1 e((k-1)T) + b_2 e((k-2)T) \quad (6.33)$$

Con esta ecuación se puede implementar el control PID discreto en el sistema de plato y bola pero antes se debe simular para una mejor sintonización del PID y obtener las ganancias para K_p , K_i y K_d . (Salvá y cols., s.f.).

6.4.4. Control PID con Aproximación Rectangular para el Sistema de Plato y Bola

Con la ecuación obtenida del control PID en tiempo discreto podemos simular al igual que con el control PD en el sistema de plato y bola y ver el comportamiento, al menos de manera simulada, como ya se mencionó, esta será una buena aproximación respecto al sistema real ya que el control a implementar en el sistema real es el mismo, ya que se hace con ayuda de un microcontrolador y todos funcionan en tiempo discreto. Partiendo de la función de transferencia en tiempo discreto del sistema de plato y bola que es la siguiente:

$$G(z) = \frac{-0.03504z - 0.03504}{z^2 - 2z + 1} \quad (6.34)$$

Después asignamos valores a las ganancias de K_p , K_i y K_d , además de utilizar el mismo periodo de muestreo $T = 0.1$, obtenemos la función de transferencia del PID y resulta:

$$H(z) = \frac{-8.153z^2 + 14.88z - 6.8}{z^2 - z} \quad (6.35)$$

Si obtenemos el lugar geométrico de las raíces del control PID en tiempo discreto se ve como se muestra en la Fig. 6.7

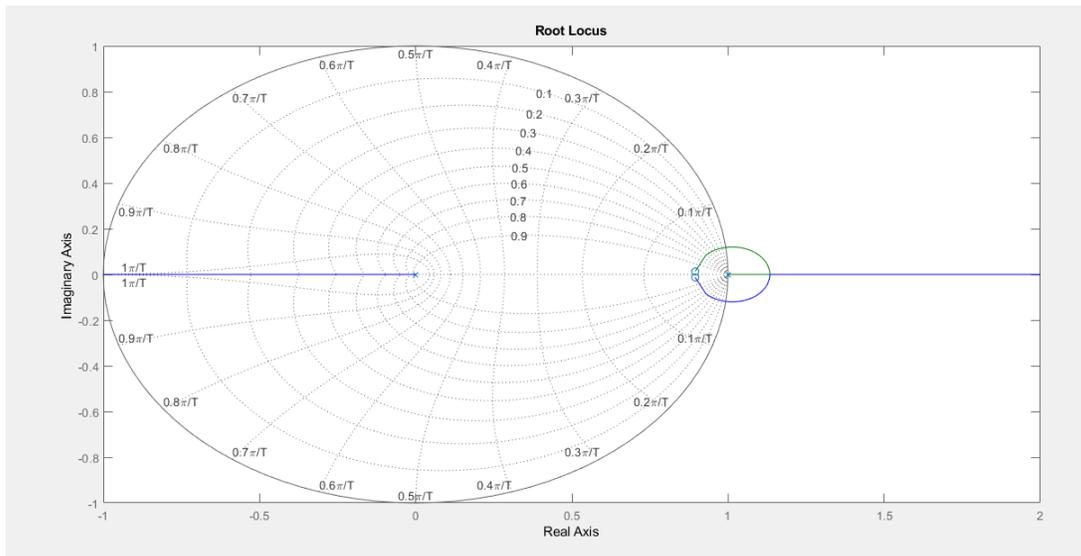


Figura 6.7: Lugar geométrico de las raíces de control PID en tiempo discreto

Como se observa, tiene dos polos y dos ceros, todos dentro del círculo unitario, si aplicamos el control PID discreto en lazo cerrado y obtenemos la gráfica resulta como se muestra en la Fig. 6.8

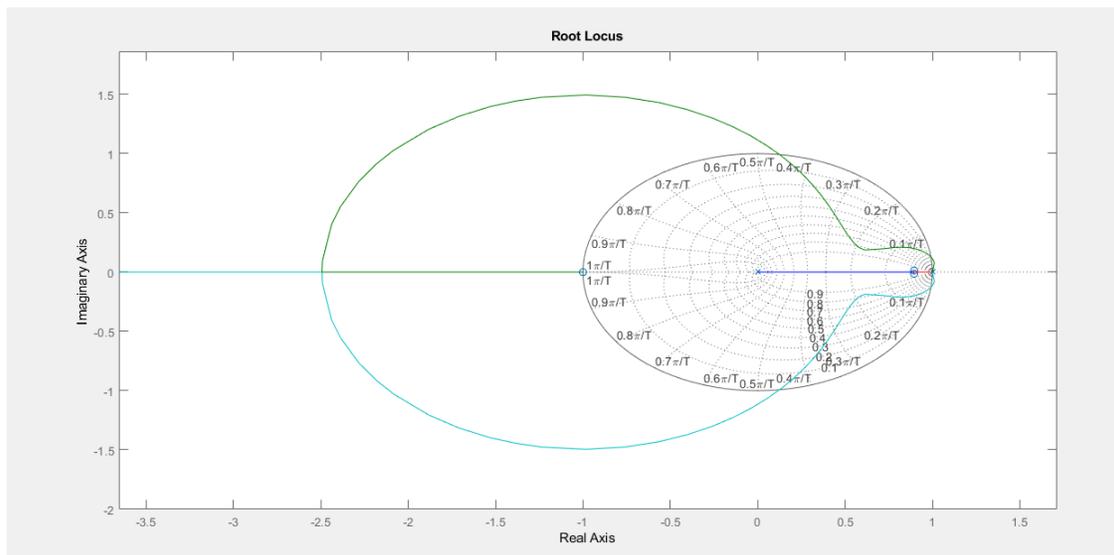


Figura 6.8: Lugar geométrico de las raíces del sistema en lazo cerrado

Aplicando una señal escalón obtenemos tanto la salida del sistema $y(t)$ como la posición de la bola en el plato en el eje x del plano del plato y obtenemos la señal de control $u(t)$ como se muestra en la Fig. 6.9.

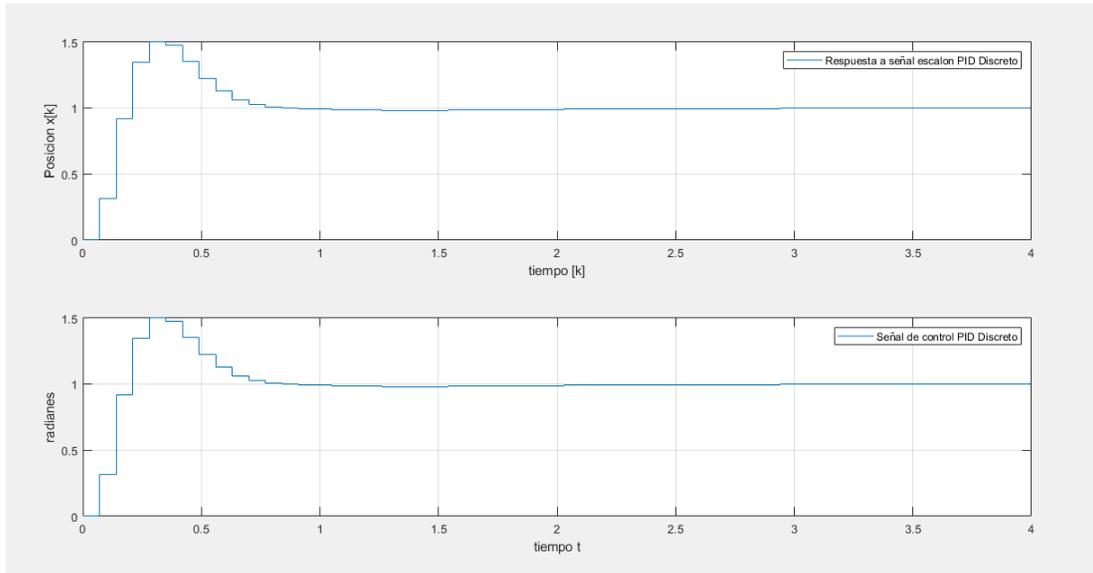


Figura 6.9: Respuesta a escalón y salida de control de sistema de plato y bola con control PID

A diferencia del control PD vemos una salida más rápida, esto puede ayudar cuando se implemente el control en el sistema real ya que es el mismo control implementado en tiempo discreto de modo que con estas ganancias se espera que el sistema se estabilice en 1 segundo, además de tener una salida de control relativamente suave. Con esta aproximación podemos ya generar el control real del sistema y comenzar a realizar los experimentos en la planta con diferentes señales de entrada, pero antes, se puede diseñar el control PID con otra aproximación que es la aproximación trapezoidal o de Tustin y ver si el control mejora tanto en estabilizar el sistema como en la señal de control.

6.5. Aproximación Trapezoidal o de Tustin

Al igual que con la aproximación rectangular la aproximación Trapezoidal o de Tustin no ayuda a pasar una señal en tiempo continuo a tiempo discreto de una forma más sencilla, matemáticamente hablando, con la relación entre la variable s y z , la diferencia entre una aproximación y otra es en gran mayoría por la forma en que se obtiene el muestreo de la señal a discretizar, para el caso de la rectangular mantiene el valor durante todo el periodo de muestreo k hasta el siguiente valor de muestreo $k+1$, en el caso de la aproximación trapezoidal obtiene un valor muestreado en k y hasta obtener el siguiente valor de muestreo $k+1$ genera una pendiente entre estos dos valores, visto de forma gráfica resulta como se muestra en la Fig. 6.10.

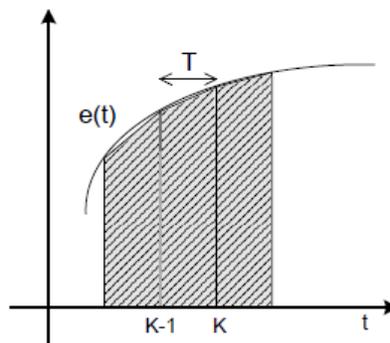


Figura 6.10: Aproximación trapezoidal o de Tustin de forma gráfica

Esta aproximación se acerca más a la función en tiempo continuo, esto dependiendo de que tanto varíe entre cada tiempo de muestreo, para la señal de la imagen se ve como si fuera la misma señal, sin embargo, se observa la pendiente entre los primeros valores de k y $k + 1$ y si vemos el área bajo la curva en ese intervalo se aprecia la figura de un trapecio de ahí el nombre de aproximación trapezoidal. Con este muestreo se obtiene una ecuación en diferencias en función de z y al igual que con la aproximación rectangular la relación entre la variable s y z cambia, de modo que se genera una ecuación entre estas dos variables y es la que se muestra a continuación:

$$s \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (6.36)$$

Con esta ecuación podemos sustituir el valor de s para obtener una ecuación en el dominio de z como en la aproximación rectangular, principalmente en la función de transferencia de los controladores PD y PID para después simular el lazo cerrado y observar cómo se comporta el sistema del plato y bola con esta aproximación y ver cual nos conviene utilizar en el sistema real. (Salvá y cols., s.f.).

6.5.1. Control PD con Aproximación de Tustin

A partir de la función de transferencia del control PD en tiempo continuo, con la que previamente se realizaron las simulaciones correspondientes en el sistema de plato y bola se puede implementar el control PD utilizando la aproximación de Tustin muy parecido a la aproximación rectangular, sea la función de transferencia del control PD:

$$G(s) = K_p(1 + T_d s) \quad (6.37)$$

Donde $T_d = \frac{K_d}{K_p}$.

Si $G(s) = G(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}})$, donde T es el periodo y recordando la forma de la función de transferencia en tiempo discreto de forma general que es:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.38)$$

sustituimos $s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$ y obtenemos la función de transferencia en función de z de la siguiente forma:

$$H(z) = K_p(1 + T_d(\frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}})) \quad (6.39)$$

Si comparamos con la ecuación general $H(z)$ obtenemos que:

$$a_1 = 1, a_2 = 0 \quad (6.40)$$

$$b_0 = K_p(1 + \frac{2T_d}{T}) \quad (6.41)$$

$$b_1 = K_p(1 - \frac{2T_d}{T}) \quad (6.42)$$

Esta ecuación $H(z)$ es la función de transferencia del control PD con la aproximación de Tustin, si obtenemos la señal de control $u(kT)$ a partir de la señal de error $e(kT)$, ambas señales en tiempo discreto, resulta $u(kT)$ como:

$$u(kT) = -a_1 u((k-1)T) - a_2 u((k-2)T) + b_0 e(kT) + b_1 e((k-1)T) + b_2 e((k-2)T) \quad (6.43)$$

donde k es el valor muestreado, u es la señal de control en el tiempo kT , $(k-1)T$ y $(k-2)T$ y $e(kT)$ es el error igual muestreado en los tiempos kT , $(k-1)T$ y $(k-2)T$, respectivamente, si sustituimos los valores de a_1 , b_0 y b_1 obtenemos la siguiente ecuación:

$$u = -u((k-1)T) + K_p(1 + \frac{2T_d}{T})e(kT) + K_p(1 - \frac{2T_d}{T})e((k-1)T) \quad (6.44)$$

Esta ecuación será la salida del control PD en tiempo discreto con la aproximación de Tustin para cualquier sistema, el siguiente paso es simularlo para el sistema de plato y bola. (Salvá y cols., s.f.).

6.5.2. Control PD con Aproximación de Tustin para el Sistema de Plato y Bola

Utilizando la función de transferencia del sistema de plato y bola:

$$G(s) = \frac{\theta}{s^2} \quad (6.45)$$

y como ya se obtuvo la función de transferencia en el dominio de z con un periodo $T = 0.1[s]$ y asignando los parámetros del modelo real la función de transferencia del sistema de plato y bola en tiempo discreto resulta:

$$G(z) = \frac{-0.03504z - 0.03504}{z^2 - 2z + 1} \quad (6.46)$$

Y ya obtenida la función de transferencia del controlador PD, podemos asignar ganancias para K_p y K_d , además de utilizar el mismo periodo $T = 0.1[s]$, de modo que obtenemos la función del controlador PD en tiempo discreto con la aproximación de Tustin como se muestra a continuación:

$$H(z) = \frac{-10.47z + 8.73}{z + 1} \quad (6.47)$$

Obtenemos el lugar geométrico de las raíces del controlador PD en tiempo discreto y resulta la gráfica como se muestra en la Fig. [6.11](#).

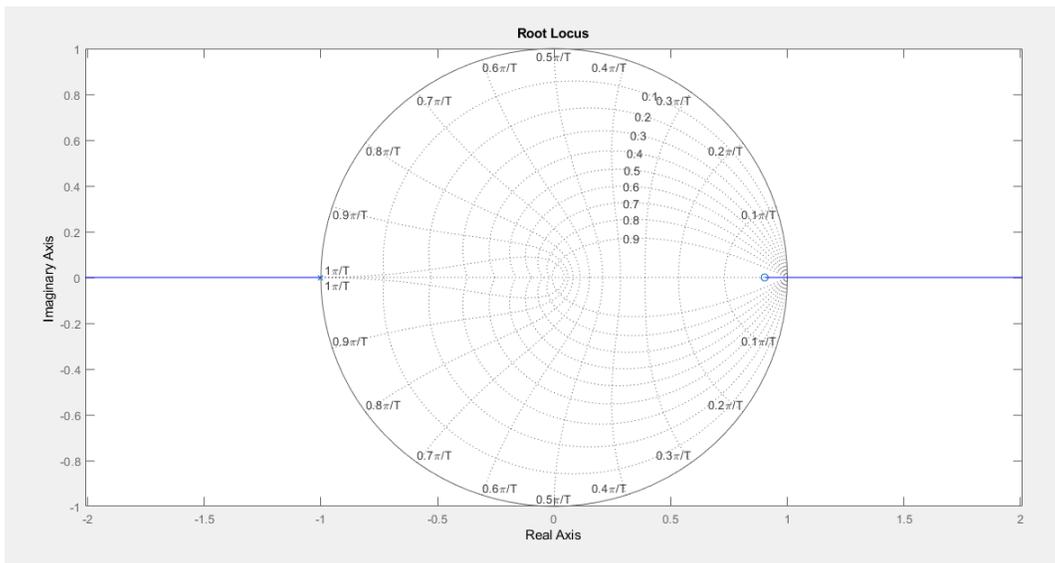


Figura 6.11: Lugar geométrico de las raíces de control PD en tiempo discreto con aproximación de Tustin

Implementamos el control PD discreto con la aproximación de Tustin en lazo cerrado y obtenemos la función de transferencia en tiempo discreto del sistema completo con el control PD con la aproximación de Tustin y resulta:

$$H_{lc}(z) = \frac{0.3668z^2 + 0.06096z - 0.3059}{z^3 - z^2 - z + 1} \quad (6.48)$$

Si obtenemos el lugar geométrico de las raíces de la función en lazo cerrado en tiempo discreto la gráfica resulta como se muestra en la Fig. [6.12](#).

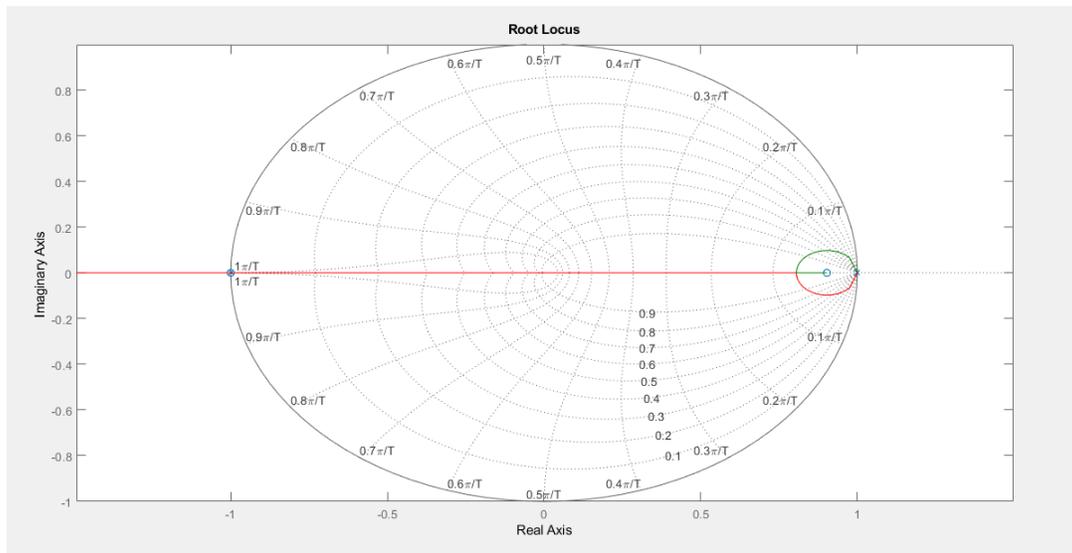


Figura 6.12: Lugar geométrico de las raíces del sistema en lazo cerrado

Por último, simulamos la respuesta a una señal escalón en función a la posición de la bola en el eje x, a partir de esta simulación obtenemos la señal de control para ver como responderá los motores con esta aproximación, al simular con estos valores la respuesta del sistema y la señal de control resultan como se muestra en la Fig [6.13](#)

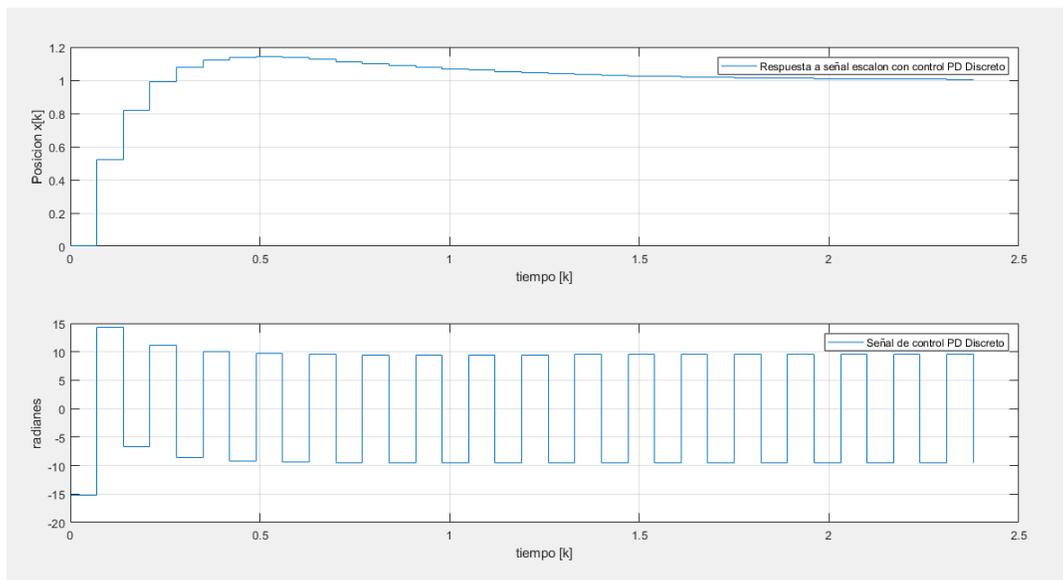


Figura 6.13: Respuesta a escalón y salida de control con control PD por aproximación de Tustin

Para esta aproximación resulta interesante ver la señal de control ya que visto de forma real parece que el motor sube y baja muy rápido para estabilizar el sistema, esto se deberá corroborar con el sistema real una vez que se implemente el control, podemos probar con un PID con la aproximación de Tustin y ver cómo responde el sistema de plato y bola, pero al menos de manera simulada llega a estabilizar el sistema aunque habrá que ver si la señal de control no genera más perturbación al sistema.

6.5.3. Control PID con Aproximación de Tustin

Al igual que con el control PD ya se obtuvo la función de transferencia del control PID que es la siguiente:

$$H(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (6.49)$$

Donde $T_i = \frac{K_p}{K_i}$ y $T_d = \frac{K_d}{K_p}$.

Si sustituimos el valor de s con la aproximación de Tustin que es $s \approx \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$ y sustituimos en $H(s)$ como $H(s) = H(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}})$, donde T es el periodo y teniendo en cuenta la forma general de la función de transferencia en tiempo discreto como se muestra a continuación:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.50)$$

Sustituimos $s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$ obtenemos la función de transferencia en función de z de la siguiente forma:

$$H(z) = K_p \left(1 + \frac{1}{T_i} \left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}\right) + Td \left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}\right)^2\right) \quad (6.51)$$

Si comparamos con la forma general de $H(z)$ obtenemos los siguientes valores:

$$a_1 = -1, a_2 = 0 \quad (6.52)$$

$$b_0 = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \quad (6.53)$$

$$b_1 = -K_p \left(1 + 2 \frac{T_d}{T}\right) \quad (6.54)$$

$$b_2 = -K_p \frac{T_d}{T} \quad (6.55)$$

La ecuación $H(z)$ es la función de transferencia en tiempo discreto del control PID, si se utiliza a partir del error en tiempo discreto $e(kT)$ obtenemos la señal de control $u(kT)$ como se muestra a continuación:

$$u(kT) = -a_1 u((k-1)T) - a_2 u((k-2)T) + b_0 e(kT) + b_1 e((k-1)T) + b_2 e((k-2)T) \quad (6.56)$$

Con esta ecuación se puede implementar el control PID discreto en el sistema de plato y bola pero antes se debe simular para una mejor sintonización del PID y obtener las ganancias para K_p , K_i y K_d . (Salvá y cols., s.f.).

6.5.4. Control PID con Aproximación de Tustin para el Sistema de Plato y Bola

Con la ecuación obtenida del control PID en tiempo discreto podemos simular al igual que con el control PD en el sistema de plato y bola y ver el comportamiento, al menos de manera simulada, como ya se mencionó, esta será una buena aproximación respecto al sistema real ya que el control a implementar en el sistema real es el mismo, ya que se hace con ayuda de un microcontrolador y todos funcionan en tiempo discreto. Partiendo de la función de transferencia en tiempo discreto del sistema de plato y bola que es la siguiente:

$$G(z) = \frac{-0.03504z - 0.03504}{z^2 - 2z + 1} \quad (6.57)$$

Después asignamos valores a las ganancias de K_p , K_i y K_d , además de utilizar el mismo periodo de muestreo $T = 0.1$, obtenemos la función de transferencia del PID y resulta:

$$H(z) = \frac{-20.91z^2 + 38.73z - 17.95}{z^2 - 1} \quad (6.58)$$

Si obtenemos el lugar geométrico de las raíces del control PID en tiempo discreto se ve como se muestra en la Fig. 6.14

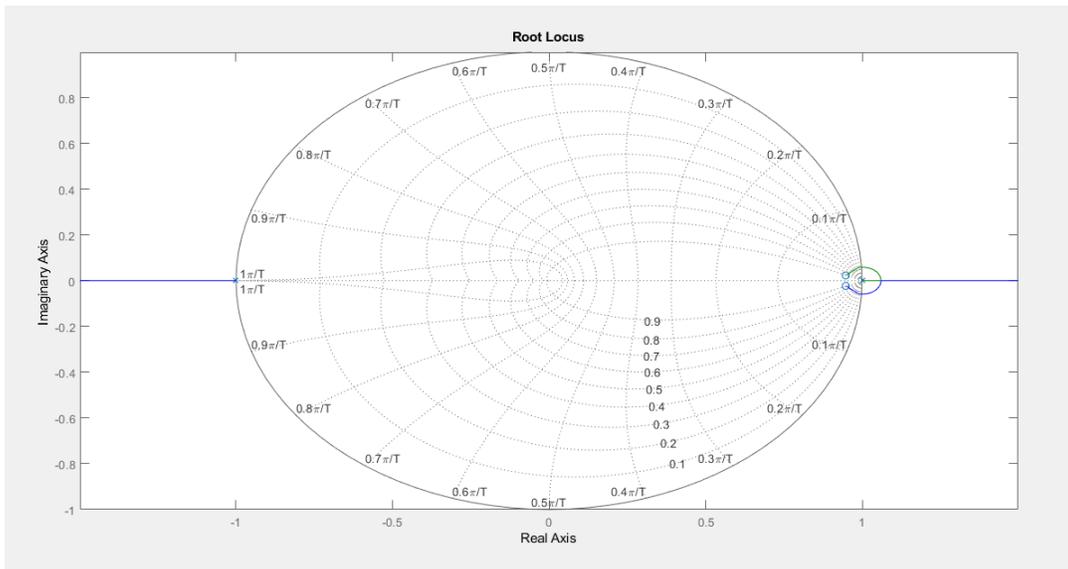


Figura 6.14: Lugar geométrico de las raíces de control PID en tiempo discreto

Como se observa, tiene dos polos y dos ceros, todos dentro del círculo unitario, si aplicamos el control PID discreto en lazo cerrado y obtenemos la gráfica resulta como se muestra en la Fig. 6.15

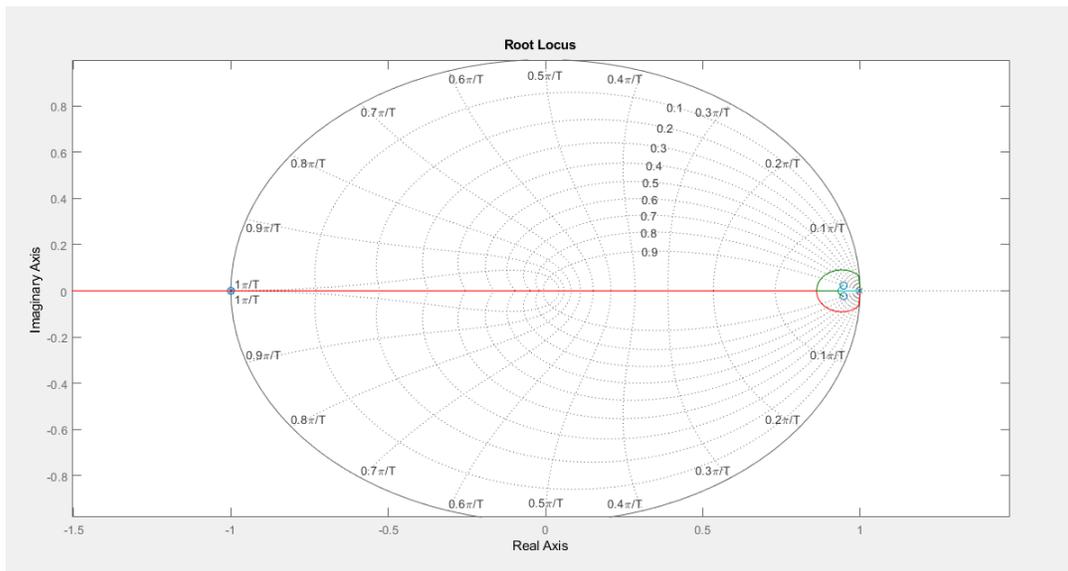


Figura 6.15: Lugar geométrico de las raíces del sistema en lazo cerrado

Aplicando una señal escalón obtenemos tanto la salida del sistema $y(t)$ como la posición de la bola en el plato en el eje x del plano del plato y obtenemos la señal de control $u(t)$ como se muestra en la Fig. 6.16

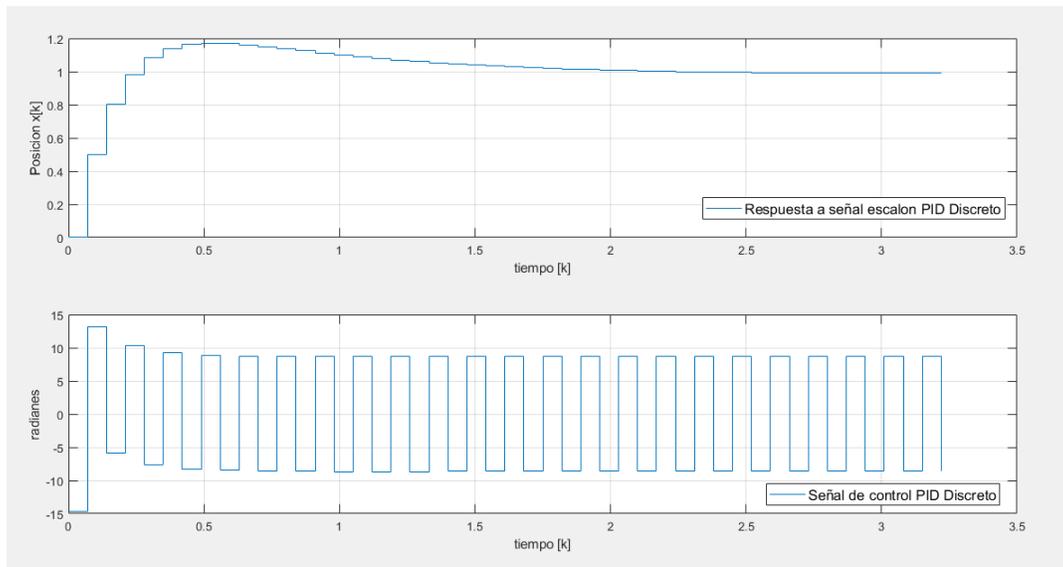


Figura 6.16: Respuesta a escalón y salida de control de sistema de plato y bola con control PID

Si se analiza desde el punto de vista del sistema real la señal de control genera que ambos motores suban y bajen de forma muy rápida, esto se debe validar con el sistema real, pero al menos de manera simulada y a pesar de lo brusco de la señal de control con esta aproximación se cumpliría el objetivo de control.

Al igual que en tiempo continuo el siguiente paso es aplicar al controlador PID fraccionario una aproximación en tiempo discreto que nos permita aplicarlo al sistema real.

6.6. Aproximación de Controladores de Orden Fraccionario

Teniendo en cuante que el cálculo fraccionario en tiempo continuo hay diferentes teorías válidas para generar una derivada e integral fraccionaria, en el caso del tiempo discreto también hay diferentes teorías donde tratan de aproximar una derivada o integral fraccionaria por medio de diferentes métodos matemáticos, sin embargo, para este caso hay algunas teorías que se utiliza mucho en el diseño e implementación de controladores PID fraccionarios en tiempo discreto, una es la teoría de Alain Oustaloup, quien a finales de los noventa desarrollo un método basado en la discretización de la ecuación diferencial fraccionaria usando la definición de Grunwald-Letnikov y una estimación de sus coeficientes por mínimos cuadrados. (Malti y Víctor, 2015).

Con ayuda de Trigessou en 1999 y Cois en 2001, Malti en 2013 y muchos otros, se ha ido desarrollando más y más esta teoría de A. Oustaloup, incluso hoy en día hay muchas herramientas o Toolbox de MatLAB para obtener controladores PID de orden fraccionario con una aproximación discreta como es el caso de CRONE desarrollado por Rachid Malti y Stephane Víctor donde se genera una aproximación de un PID de orden fraccionario para poder utilizarlo en tiempo discreto. (Malti y Víctor, 2015). El Toolbox de Tepljakov Aleksei, Petlnkov Eduard, Belikov Juri que se utilizó en la simulación de tiempo continuo FOMCON también utiliza la teoría de A. Oustaloup. (Aleksei y cols, 2012).

La idea de A. Oustaloup consiste en sintetizar los operadores fraccionarios en el dominio de la frecuencia usando una distribución recursiva de polos y ceros, de modo que el comportamiento de la frecuencia $s^{-\alpha}$ en un rango de frecuencias de $[\omega_A, \omega_B]$ a través de la siguiente expresión matemática:

$$S_{[\omega_A, \omega_B]}^{-\alpha} = G_{\alpha} \prod_{k=1}^{N_c} \frac{1 + s/\omega'_k}{1 + s/\omega_k} \quad (6.59)$$

Donde N_c es el número de celdas dependiendo de la calidad de la aproximación, G se fija de modo que $s^{-\alpha}$ tenga la misma ganancia que $s_{[\omega_A, \omega_B]}^{-\alpha}$ en el intervalos de frecuencias $[\omega_A, \omega_B]$, ω'_k y ω_k son los polos y ceros distribuidos en el rango de frecuencia $[\omega_b, \omega_h] = [\sigma - 1\omega_A, \sigma\omega_B]$ donde σ generalmente se establece en 10 para minimizar los efectos de borde y se definen mediante las siguientes expresiones:

$$\omega'_k = \gamma\omega_k \quad (6.60)$$

$$\omega_{k+1} = \nu\omega'_k \quad (6.61)$$

$$\alpha = 1 - \frac{\log \gamma}{\log \gamma\nu} \quad (6.62)$$

En resumen lo que hace A. Oustaloup es generar una función de transferencia con los polos N que quiera en un intervalo de frecuencias de $[\omega_A, \omega_B]$ donde $\omega_A > 0$ donde esta nueva función tiene términos no fraccionarios de s en base a la cantidad de polos, después utilizar los métodos de discretización comunes como Tustin para generar una ecuación aproximada en tiempo discreto y utilizarla para implementar controladores fraccionarios. (Malti y Victor, 2015).

6.6.1. Control PD Fraccionario con Aproximación de Oustaloup

Recordando la función de transferencia del control PD fraccionario:

$$G(s) = K_p + K_d s^\mu \quad (6.63)$$

Donde μ es el valor fraccionario de la derivada y en base a las simulaciones en tiempo continuo se escogió un valor de $\mu=0.8$, podemos aproximar esta ecuación a una con términos no fraccionarios por medio de la teoría de Oustaloup, con ayuda de la herramienta del Toolbox de MatLAB, FOMCON (Aleksy y cols., 2012), asignamos las ganancias que ya obtuvimos de K_p y K_d y obtenemos la siguiente expresión con la aproximación de Oustaloup.

$$G(s) = \frac{-41.689(s + 13.65)(s + 2.342)(s + 0.6168)}{(s + 79.43)(s + 7.943)(s + 0.7943)} \quad (6.64)$$

Esta aproximación ya no tiene términos de s^μ si comparamos los diagramas de bode de la aproximación junto con el control PD normal y con el PD fraccionario sin aproximación resulta como se muestra en la Fig.6.17.

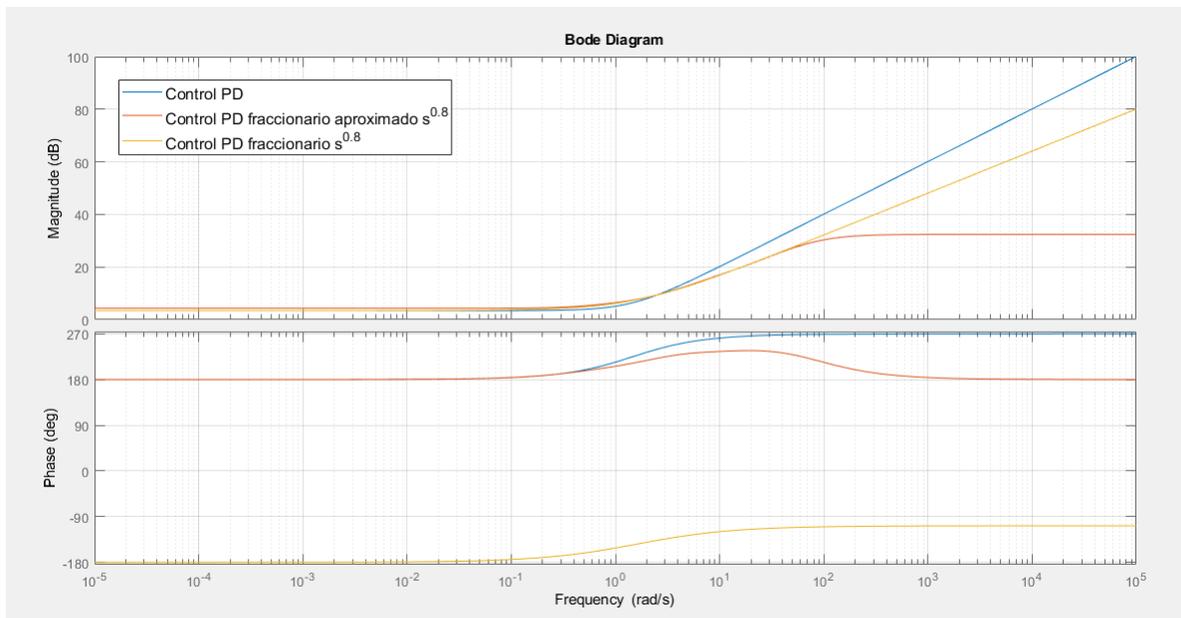


Figura 6.17: Diagramas de Bode de control PD, control PD fraccionario aproximado y control PD fraccionario

Como se observa el PD fraccionario con la aproximación de Oustaloup asemeja el comportamiento tanto en magnitud como fase del control PD tradicional y como ya se había visto en las simulaciones de tiempo continuo el PD fraccionario tiene un atraso de fase respecto al control PD. Una vez que tenemos la aproximación de Oustaloup podemos discretizar la función de transferencia con cualquier aproximación, aunque muchos artículos sugieren que se utilice la aproximación de Tustin, si aplicamos esta aproximación obtenemos la función de transferencia en tiempo discreto como se muestra a continuación:

$$G(z) = \frac{-13.714(z - 0.3534)(z - 0.8485)(z - 0.9577)}{(z + 0.4709)(z + 0.5649)(z - 0.9459)} \quad (6.65)$$

Si analizamos el diagrama de bode de esta discretización junto con el control PD normal con la aproximación rectangular resulta como se muestra en la Fig. 6.18

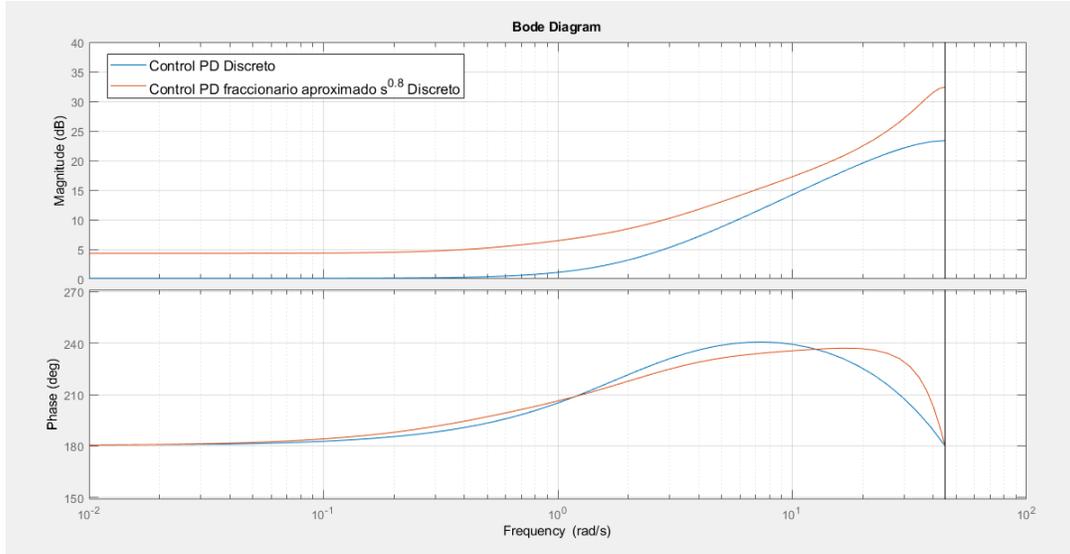


Figura 6.18: Diagramas de Bode de control PD discreto y control PD fraccionario aproximado discretizado

Como vemos el comportamiento es similar tanto el PD discreto con la aproximación rectangular que con la aproximación del PD fraccionario por Tustin el siguiente paso es simularlo con la planta del plato y bola y ver su respuesta en tiempo discreto.

6.6.2. Control PD Fraccionario Discretizado para el Sistema de Plato y Bola

Una vez que tenemos el control PD fraccionario con la aproximación de Oustaloup y además con la aproximación en tiempo discreto por Tustin expresado de la siguiente forma:

$$G(z) = \frac{-13.714(z - 0.3534)(z - 0.8485)(z - 0.9577)}{(z + 0.4709)(z + 0.5649)(z - 0.9459)} \quad (6.66)$$

Ya podemos simular la respuesta del sistema en tiempo discreto para el control PD fraccionario, recordando la ecuación en tiempo discreto del sistema de plato y bola:

$$H(z) = \frac{-5.67z + 4.8}{z} \quad (6.67)$$

Aplicamos el lazo cerrado con la función de transferencia del sistema de plato y bola y el control PD fraccionario discretizado, el lugar geométrico del sistema en lazo cerrado resulta como se muestra en la Fig. 6.19.

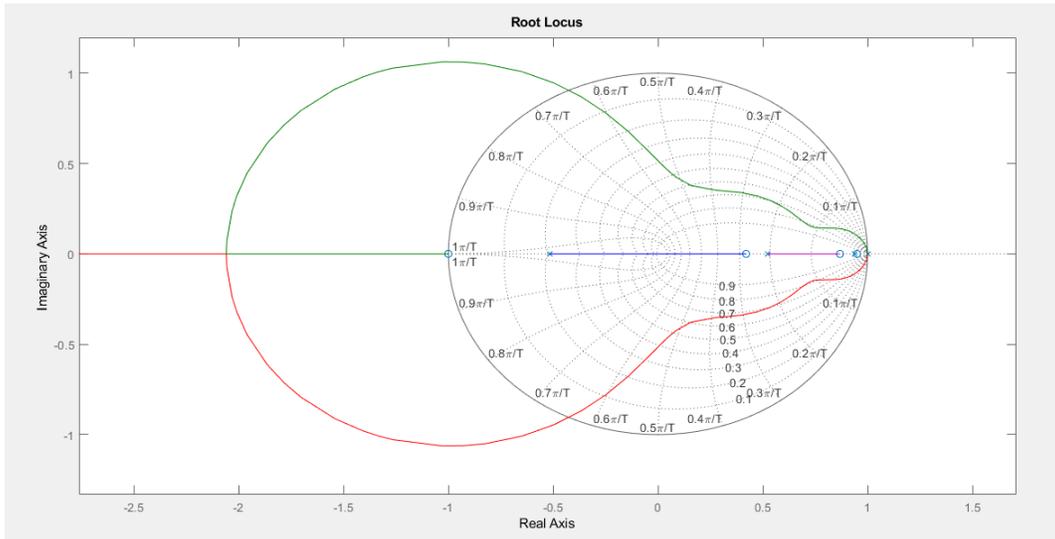


Figura 6.19: Lugar geométrico de las raíces en lazo cerrado con control PD fraccionario discretizado

Para visualizar si mejora el comportamiento de un control PD tradicional por uno fraccionario discretizado es si comparamos ambos en lazo cerrado, visto de forma gráfica la simulación resulta como se muestra en la Fig. 6.20.

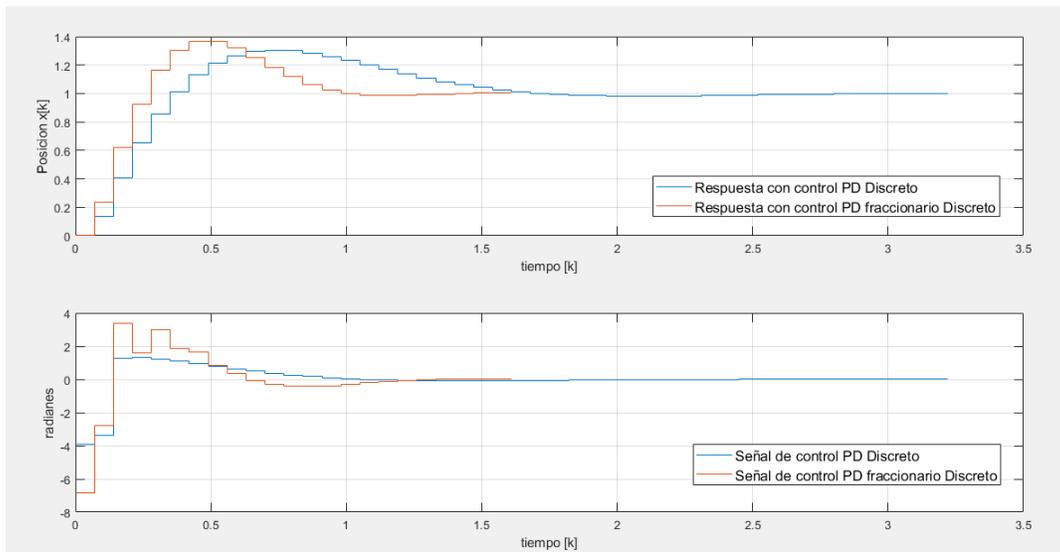


Figura 6.20: Respuesta a escalón y salida de control de sistema de plato y bola con control PD fraccionario

Al igual que con las simulaciones en tiempo continuo, el control PD fraccionario tiene un sobrepaso mayor pero estabiliza el sistema más rápido en comparación con el control PD tradicional, de esta forma podemos concluir que se esperarí un mejor rendimiento del control PD fraccionario que el control PD tradicional tanto en tiempo continuo como en tiempo discreto aunque, una vez que se implemente habrá que ver si el sobrepaso es considerable en el sistema real, el siguiente paso es simular el control PID fraccionario de manera discretizada.

6.6.3. Control PID Fraccionario con Aproximación de Oustaloup

Recordando la función de transferencia del control PID fraccionario:

$$G(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (6.68)$$

Al igual que con el control PD fraccionario podemos obtener una aproximación mediante Oustaloup para tener una función de transferencia sin términos de orden fraccionario, utilizamos las mismas ganancias de K_p , K_i y K_d que en tiempo continuo y para el caso de μ mantenemos el valor de 0.8, a λ le asignamos un valor arbitrario de 0.2 y generamos la aproximación de Oustaloup con la herramienta de MatLAB de FOMCON, la ecuación resulta como se muestra a continuación:

$$G(s) = \frac{-0.97(s + 2.342)(s + 0.3981)(s + 0.2828)}{(s + 0.3981)(s + 0.2512)} \quad (6.69)$$

Al igual que con la aproximación del PD fraccionario, en esta aproximación ya no hay términos ni de λ ni μ , si analizamos sus diagramas de Bode y los comparamos con el controlador PID de la simulación en tiempo continuo y el PID con $\lambda=0.2$ como se muestra en la Fig. [6.21](#)

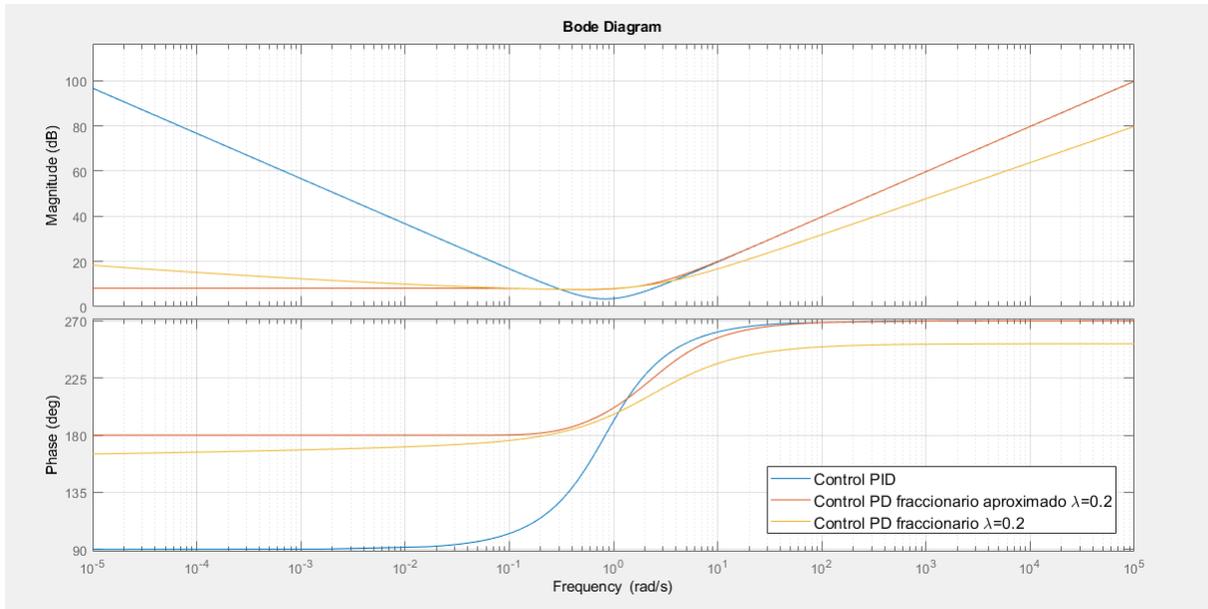


Figura 6.21: Diagramas de Bode de PID, PID fraccionario aproximado y PID fraccionario

Si vemos la magnitud del PID y la aproximación se parecen mucho hasta el punto de sobreponerse en la magnitud, y el control PID se aleja un poco, pero tiene el mismo comportamiento que los dos anteriores. De igual que la aproximación del PD fraccionario por Oustaloup, discretizamos el control PID aproximado por medio de Tustin y obtenemos su ecuación en tiempo discreto.

$$G(z) = \frac{-30.019(z - 0.8485)(z - 0.9725)(z - 0.9804)}{(z - 0.9725)(z - 0.9826)(z + 1)} \quad (6.70)$$

También hacemos el análisis del controlador comparando el PID tradicional con el PID aproximado, ambos en tiempo discreto, los diagramas resultan como que muestra en la Fig. [6.22](#)

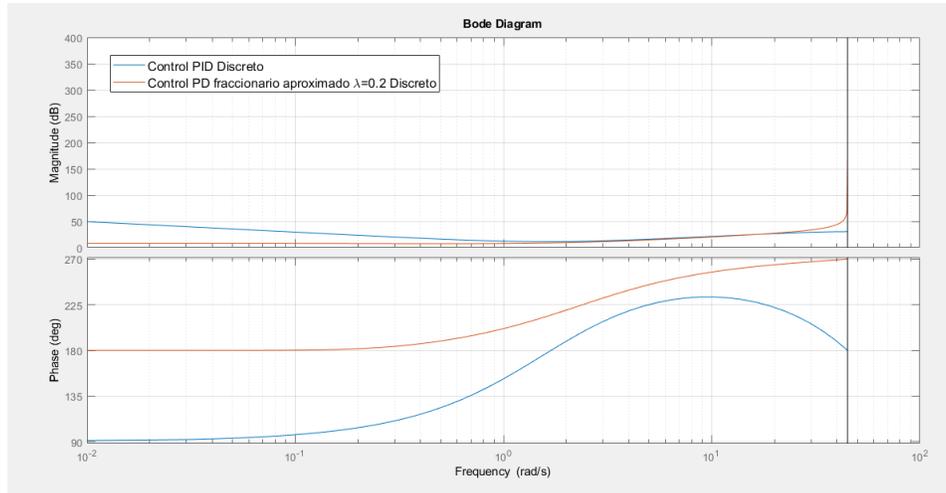


Figura 6.22: Diagramas de Bode de PID discreto y PID fraccionario aproximado discreto

Como se observa el control PID discreto y el control PID fraccionario aproximado y discreto son similares en su comportamiento tanto en magnitud como fase, el siguiente paso es simular este controlador con la planta del plato y bola y ver cómo responde el sistema.

6.6.4. Control PID Fraccionario Discretizado para el Sistema de Plato y Bola

Como ya tenemos el control PID fraccionario con la aproximación de Oustaloup y además con su aproximación en tiempo discreto por Tustin expresado de la siguiente forma:

$$G(z) = \frac{-30.019(z - 0.8485)(z - 0.9725)(z - 0.9804)}{(z - 0.9725)(z - 0.9826)(z + 1)} \quad (6.71)$$

Ya podemos simular la respuesta del sistema en tiempo discreto para el control PID fraccionario, recordando la ecuación en tiempo discreto del sistema de plato y bola

$$H(z) = \frac{-5.67z + 4.8}{z} \quad (6.72)$$

Aplicamos el lazo cerrado con la función de transferencia del sistema de plato y bola y el control PID fraccionario discretizado, el lugar geométrico del sistema en lazo cerrado resulta como se muestra en la Fig. 6.23.

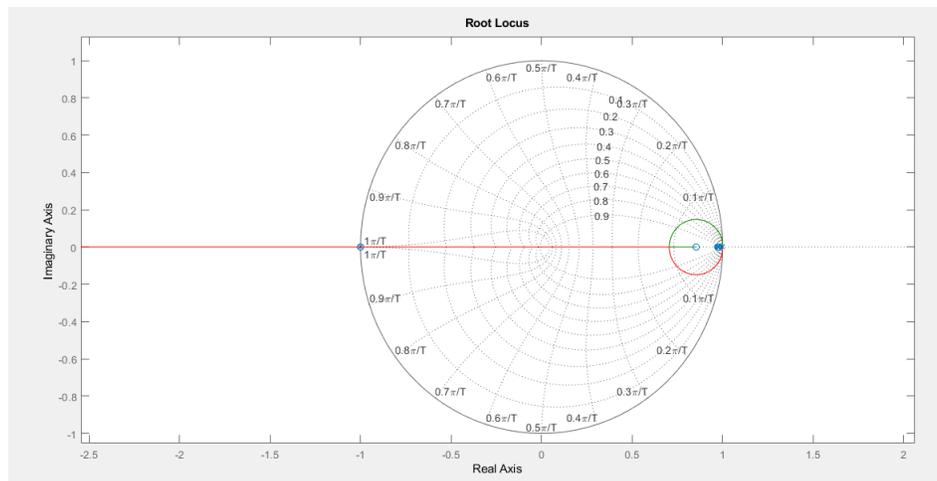


Figura 6.23: Lugar geométrico de las raíces en lazo cerrado con control PID fraccionario discretizado

Se genera de manera simulada la respuesta a una señal escalón. Para un mejor análisis y ver si podemos mejorar el comportamiento de un control PID tradicional por uno fraccionario comparamos ambos en lazo cerrado en tiempo discreto, visto de forma gráfica la simulación resulta como se muestra en la Fig. 6.24

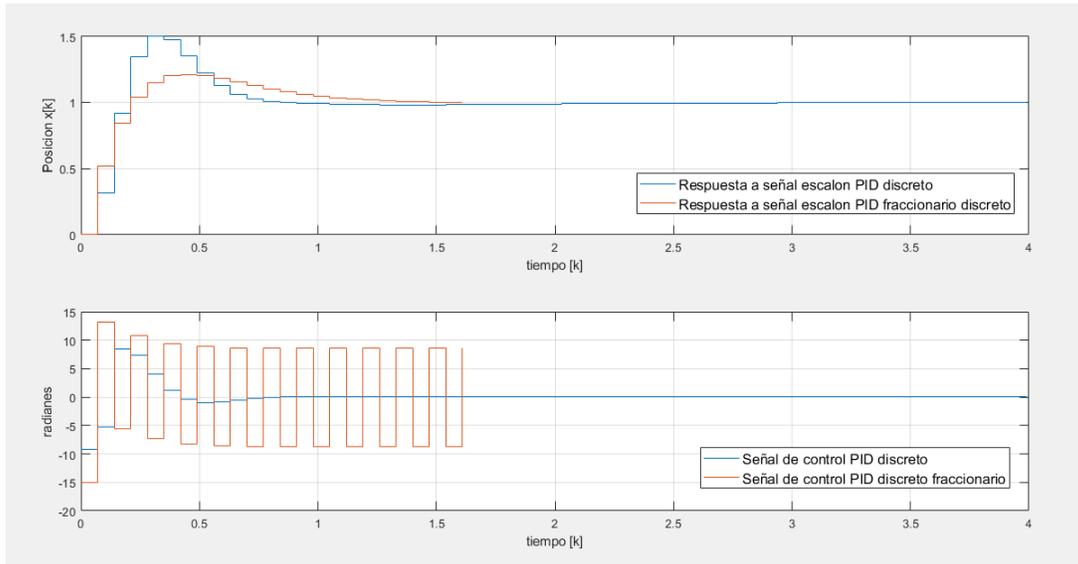


Figura 6.24: Respuesta a escalón y salida de control de sistema de plato y bola con control PD fraccionario

De esta forma se muestra que tanto el control PD fraccionario como el PID fraccionario sí estabiliza el sistema, aunque al igual que con el controlador PD y PID discretizados por Tustin la señal de control oscila mucho y puede generar más perturbación al sistema, esto puede ser por que la aproximación del PID fraccionario es discretizado también por Tustin. El único que al menos de manera simulada tiene una señal de control más suave con la aproximación de Tustin es el PD fraccionario aproximado, habrá que confirmar estas señales cuando se implementen en el sistema real y como ya se tiene todas las simulaciones de todos los controladores a implementar ya se puede implementar en el sistema real.

Capítulo 7

Implementación de Controladores al Sistema de Plato y Bola

Después de tener todo el análisis previo del sistema de plato y bola, con sus modelos matemáticos tanto en variables de estado como en función de transferencia y ya construido el sistema real con todas las pruebas necesarias para que soporte cualquier experimento, además de haber generado y simulado todos los controladores a implementar tanto en tiempo continuo como en tiempo discreto, el paso siguiente es probar los controladores en el sistema real del plato y bola. Para esto es conveniente realizar un proceso secuencial en el controlador y como ya menciono, el controlador será una PC con el software de MatLAB, el sensor a utilizar es una webcam y los actuadores que aplicaran la señal de control al sistema real son los servomotores instalados en el sistema de plato y bola, los servomotores se controlan a través de un driver utilizando la tarjeta de desarrollo Arduino de modo que los pasos a seguir al momento de implementar cualquier controlador en este sistema son los siguientes:

- Establecer la conexión de MATLAB con Arduino, en este caso un Arduino Mega2560.
- Desde MATLAB, cargar las librerías necesarias en el Arduino para controlar los servos.
- Establecer comunicación de MATLAB con la webcam.
- Calibrar el frame acorde a la posición de la webcam.
- Realizar el proceso para obtener la posición de la bola tanto en el eje x como en y a través de MATLAB.
- Establecer el valor deseado de posición o la señal de referencia, tanto para el control de regulación como control de seguimiento.
- Generar la aproximación polinomial de la señal de control para adecuar la señal a los servomotores.
- Calcular el error inicial.
- Generar una secuencia de recursividad donde aplique la señal de control y modifique en tiempo real la señal de control en base a la posición actual de la bola y con la señal acondicionada.
- Obtener de forma gráfica los valores de posición tanto en x como en y , así como la señal de control en ambos ejes.

Todos estos son los pasos a seguir al momento de implementar los diferentes controladores, con la diferencia de la señal de control a implementar ya que esta varía en función al controlador implementado. Si estos pasos a seguir se ven como un diagrama de flujo, resulta como se muestra en la Fig. [7.1](#).

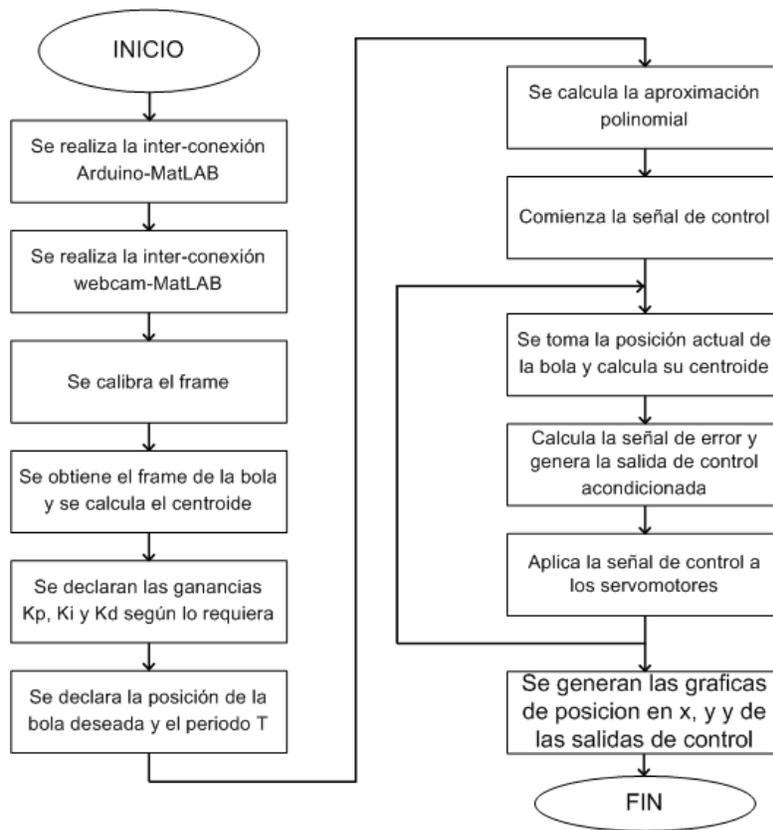


Figura 7.1: Diagrama de flujo de secuencia para implementar controladores en el plato y bola

Todo esto se hace a través de MatLAB y aunque en este software hay dos formas de implementar el sistema ya sea a través de la herramienta de Simulink o como en este caso, a través de la programación con código escrito directamente en MatLAB, se tomó esta decisión principalmente por que la Webcam no tiene una librería donde se pueda utilizar en Simulink y la única librería es a través del código escrito de MatLAB.

7.1. Código en MatLAB

Dependiendo del controlador a implementar en el código se modifican algunos parámetros, pero algunos bloques de programación se mantienen en todos los controladores. Siguiendo la secuencia del diagrama de flujo vemos que el primer paso es hacer la conexión de la tarjeta de desarrollo Arduino como el driver entre la PC y los servomotores, de modo que el bloque de programación de esta parte se muestra en la Fig. 7.2.

```

%% Control de Sistema de Plato y Bola Mediante Técnicas de Control Lineal
% Implementacion de controlador PD
% Aproximacion Rectangular

% Enrique Pérez Martínez

%% Inicializacion de Arduino

arduinoM = arduino('COM3','Mega2560','Libraries','Servo');
servox = servo(arduinoM,'D2');
servoy = servo(arduinoM,'D3');
writePosition(servox,0.5);
writePosition(servoy,0.5);

```

Figura 7.2: Bloque de programación integración de Arduino

El siguiente paso es integrar la webcam a MatLAB que como ya se menciona va a ser el sensor por el que se obtiene la posición de la bola en el plato y antes del experimento, es necesario calibrar el frame a modo de ajustarlo lo mejor posible al plato, la calibración va a depender de cómo se colocó físicamente la webcam ya que el soporte de la webcam permite todo el espacio de arriba del plato sin afectar el movimiento del mismo, el bloque de programación se ve como se muestra en la Fig. 7.3.

```

%% Inicializacion de Webcam con MatLAB

cam=webcam(2);           %Se crea la variable cam = webcam del sistema
cam.Brightness=128;     %Definimos el valor maximo de brillo
preview(cam)           %Muestra la Webcam en vivo

%% Calibracion del frame

img=snapshot(cam);      %Se obtiene el frame
rect=[75 10 460 460];  %se define un rectangulo aproximado al plato
imgc=imcrop(img,rect); %se recorta solo el plato del frame
gris=rgb2gray(imgc);    %se crea una imagen en escala de grises
imshow(imgc)           %muestra la imagen final

```

Figura 7.3: Bloque de programación integración webcam y calibración de frame

Una vez que se tiene el frame se debe obtener la posición de la bola en ambos ejes con el proceso de la obtención de la bola que ya se detalló anteriormente, el bloque de programación resulta como se ve en la Fig. 7.4

```

%% Obtencion de la bola

contmin=0.06;
contmax=0.07;
bola=imsubtract(imgc(:,:,3),gris); %se extrae la bola de la imagen
bola=imadjust(bola,[contmin contmax]); %se ajusta el contraste
imshow(bola) %muestra la imagen de la bola

%% Centroide

%Eje Y
sumy=sum(bola,1);
cby=sum((1:size(bola,2)).*sumy)/sum(sumy);
y0=(cby*(19/size(bola,2))-9.5);
y0=y0*(0.01);
%Eje X
sumx=sum(bola,2);
sumx=sumx';
cbx=sum((1:size(bola,1)).*sumx)/sum(sumx);
x0=(cbx*(19/size(bola,1))-9.5);
x0=x0*(0.01);

```

Figura 7.4: Obtención de la posición de la bola en el plato

Ya obtenido el valor de la posición de la bola se establece como posición inicial como x_0 y como y_0 para cada eje. Después se establecen los valores de ganancia de K_p , K_i y K_d , según se utilicen en el controlador implementado; con estos valores se definen los valores para la aproximación en tiempo discreto y también se coloca la posición deseada, cabe mencionar que en esta parte puede programarse un punto en específico y que se mantenga en esa posición o también se puede programar una señal diferente como por ejemplo la señal senoidal en ambos ejes para formar un círculo con el movimiento de la bola en el plato todo este bloque de programación resulta como se muestra en la Fig. 7.5.

```

%% Control PD

%Parametros del control PD

Kp=-1.72;
Kd=-1.48;

Td=Kd/Kp;

T=0.075;

b0=Kp*(1+(Td/T));
b1=-Kp*(Td/T);
a1=0;

%% Valores deseados de posicion

% Señal escalon
xd=0;
yd=0;

```

Figura 7.5: Aproximación polinomial para la señal de control

Una vez que se establece la señal de referencia se debe calcular la aproximación polinomial que acondicionara la señal de control para que se aplique a los servomotores, los valores obtenidos para cada motor y al aproximación polinomial para cada eje en la programación de MATLAB se muestra en la Fig. 7.6. También se definen los valores del polinomio con la aproximación definida anteriormente y se colocan las gráficas que obtendrán los datos de posición y la señal de control para cada eje, durante los experimentos de cada controlador; estos bloques de programación resultan como se muestra en la Fig. 7.7.

```

datay=[
-10 0;
-10 0.1;
-10 0.2;
-9 0.3;
-5 0.4;
0 0.5;
5 0.6;
9 0.7;
10 0.8;
12.5 0.9;
12.5 1];

datax(:,3)=datax(:,1)*(pi/180);
radx = datax(:,3);
pwmX=datax(:,2);
polx=[ones(size(radx)) radx radx.^2 radx.^3 radx.^4 radx.^5];
parsx=polx\pwmX;
fx=@(x,parsx) parsx(1)+parsx(2)*x+parsx(3)*x.^2+parsx(4)*x.^3+parsx(5)*x.^4;

datay(:,3)=datay(:,1)*(pi/180);
rady = datay(:,3);
pwmY=datay(:,2);
poly=[ones(size(rady)) rady rady.^2 rady.^3 rady.^4 rady.^5];
parsy=poly\pwmY;
fy=@(y,parsy) parsy(1)+parsy(2)*y+parsy(3)*y.^2+parsy(4)*y.^3+parsy(5)*y.^4;

```

Figura 7.6: Aproximación polinomial para la señal de control

```

ax0=parsx(1);
ax1=parsx(2);
ax2=parsx(3);
ax3=parsx(4);
ax4=parsx(5);
ax5=parsx(6);

ay0=parsy(1);
ay1=parsy(2);
ay2=parsy(3);
ay3=parsy(4);
ay4=parsy(5);
ay5=parsy(6);

%% Graficas

posx=zeros(size((tini:Ts:tfin')));
posy=zeros(size((tini:Ts:tfin')));

contx=zeros(size((tini:Ts:tfin')));
conty=zeros(size((tini:Ts:tfin')));

tiempo=zeros(size((tini:Ts:tfin')));
i=1;

```

Figura 7.7: Aproximación polinomial para la señal de control

Después de todo esto ya se puede generar la señal de control y para que sea recurrente a modo de que modifique la posición durante el experimento y para hacer esto se utiliza una secuencia de programación muy común que es un ciclo for. El ciclo for dependiendo del periodo de muestreo T generara el lazo de control cerrado calculando el error actual y aplicando la señal de control que requiere el sistema para llevar la bola a la posición deseada, después obtener nuevamente la posición de la bola y volver a generar la señal de control, además de eso, cada que aplique la señal de control guardara los valores de posición en ambos ejes así como la señal de control para ambos servomotores hasta que acabe el tiempo definido en que dura el experimento. Una vez que termine el ciclo for colocara el plato completamente horizontal como señal de que el experimento termino, si se ve en forma de programación resulta como se muestra en la Fig. 7.8, la Fig. 7.9 y la Fig. 7.10

```

%% Salida de control
for t=tini:Ts:tfin

img=snapshot(cam);
imgc=imcrop(img,rect);
gris=rgb2gray(imgc);
bola=imsubtract(imgc(:,:,3),gris);
bola=imadjust(bola,[contmin contmax]);

sumy=sum(bola,1);
cby=sum((1:size(bola,2)).*sumy)/sum(sumy);
ya=(cby*(19/size(bola,2))-9.5);
ya=ya*(0.01);

sumx=sum(bola,2);
sumx=sumx';
cbx=sum((1:size(bola,1)).*sumx)/sum(sumx);
xa=(cbx*(19/size(bola,1))-9.5);
xa=xa*(0.01);

posx(i,1)=xa;
posy(i,1)=ya;

ex=xd-xa;
ey=yd-ya;

```

Figura 7.8: Programación salida de control 1

```

ux=- (a1*uax) + (b0*ex) + (b1*eax);
uy=- (a1*uay) + (b0*ey) + (b1*eay);

uxr=ax5*(ux^5)+ax4*(ux^4)+ax3*(ux^3)+ax2*(ux^2)+ax1*(ux)+ax0;
uyr=ay5*(uy^5)+ay4*(uy^4)+ay3*(uy^3)+ay2*(uy^2)+ay1*(uy)+ay0;

if(uxr>angmax)
    uxr=angmax;
end
if(uxr<angmin)
    uxr=angmin;
end
if(uyr>angmax)
    uyr=angmax;
end
if(uyr<angmin)
    uyr=angmin;
end

writePosition(servox,uxr)
writePosition(servoy,uyr)

contx(i,1)=ux;
conty(i,1)=uy;
tiempo(i,1)=t;

uanx=ux;
uany=uy;

```

Figura 7.9: Programación salida de control 2

```

writePosition(servox,uxr)
writePosition(servoy,uyr)

contx(i,1)=ux;
conty(i,1)=uy;
tiempo(i,1)=t;

uanx=ux;
uany=uy;

eax=ex;
eay=ey;

i=i+1;

end

writePosition(servox,0.5);
writePosition(servoy,0.5);

```

Figura 7.10: Programación salida de control 3

Esta señal de control varía en función al control implementado, sin embargo, como todos los controladores que se implementaron fueron en tiempo discreto el proceso es el mismo y solo cambia la señal de control, pero es la misma que la que se obtuvo en las simulaciones en tiempo discreto de cada controlador. El último paso en la programación consiste en graficar todos los valores obtenidos a modo de tener la posición de la bola y la señal de control de para cada eje y mostrarlo gráficamente, haciendo la comparación entre cada eje para mostrar el movimiento de la bola en el plato durante el experimento; en otra gráfica se muestra la señal de control respecto al tiempo y en otra la posición de la bola en cada eje respecto al tiempo t .

El primer controlador que se implementó fue el control PD, que como ya se explicó puede tener dos aproximaciones y dependiendo de la aproximación varía la señal de control de modo que primero se implementó el control PD con la aproximación rectangular.

7.2. Control PD con Aproximación Rectangular Implementado

Partiendo de la señal de control en tiempo discreto $u(kT)$ que se obtuvo con la aproximación rectangular del control PD discreto:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.1)$$

Se puede implementar exactamente la misma función en el controlador real debido a que funciona en tiempo discreto al igual que cualquier microprocesador. Para este controlador el ajuste en la señal de control cambia al modificar los valores de a_1 , a_2 , b_1 , b_2 y b_3 de la señal de control; para el caso de la aproximación rectangular estos son los valores:

$$a_1 = 0, a_2 = 0 \quad (7.2)$$

$$b_0 = K_p \left(1 + \frac{T_d}{T}\right) \quad (7.3)$$

$$b_1 = \frac{-K_d}{T} \quad (7.4)$$

Estos valores se sustituyen en la señal de control $u(kT)$ y se obtiene la señal de control PD con aproximación rectangular. (Salvá y cols., s.f.). Como a través de las simulaciones con el modelo en tiempo discreto del sistema de plato y bola ya se tienen unas ganancias para K_p y K_d se implementó el control al sistema real con las mismas ganancias, el resultado del experimento con las ganancias obtenidas en las simulaciones se muestran en la Fig. 7.11

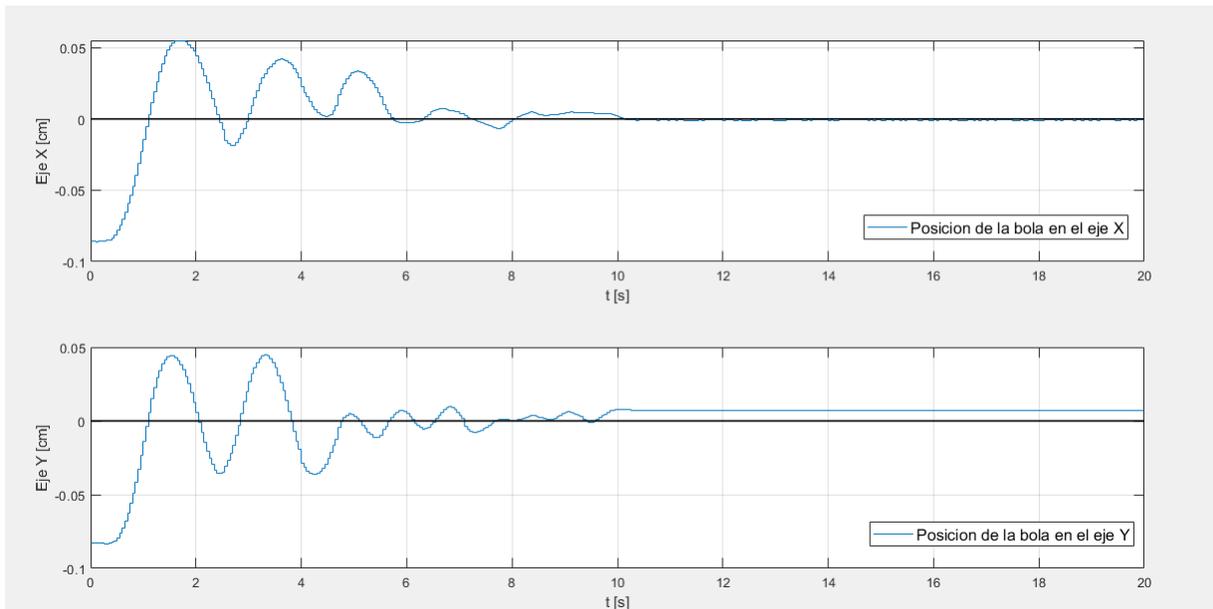


Figura 7.11: Experimento 1 con control PD por aproximación rectangular

Y la señal de control para cada eje resulta como se muestra en la Fig. [7.12](#).

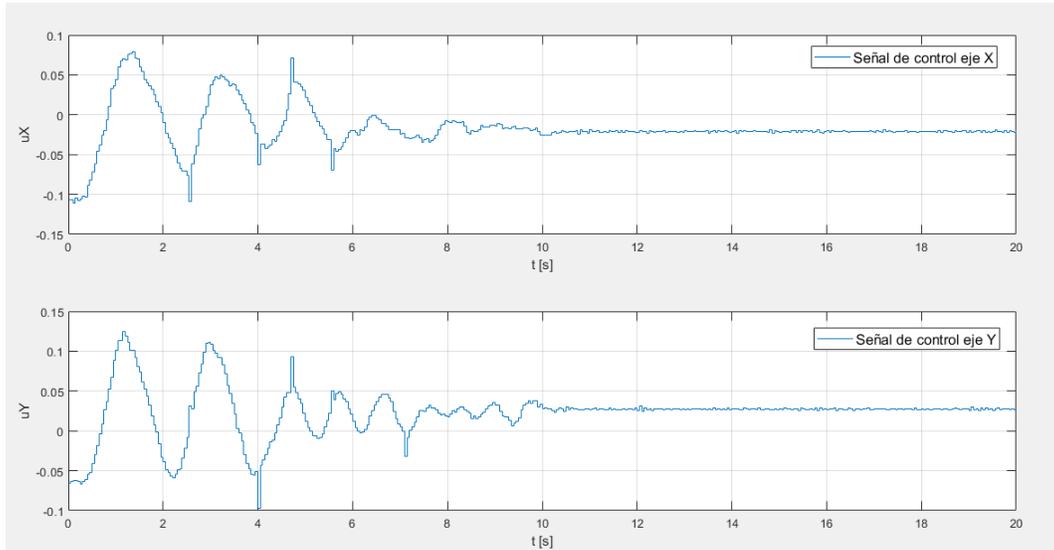


Figura 7.12: Señal de control de experimento 1 con control PD por aproximación rectangular

Como se observa, el control trata de ajustar la posición de la bola en el origen (0, 0) aunque oscila bastante en comparación con las simulaciones. A pesar de esto el sistema si lleva la bola al centro del plato y la oscilación se debe a que en el sistema real afecta bastante la inercia rotacional. Si ponemos en comparación el eje x, y para ver cómo el movimiento de la bola durante el experimento resulta como se muestra en la Fig. [7.13](#).

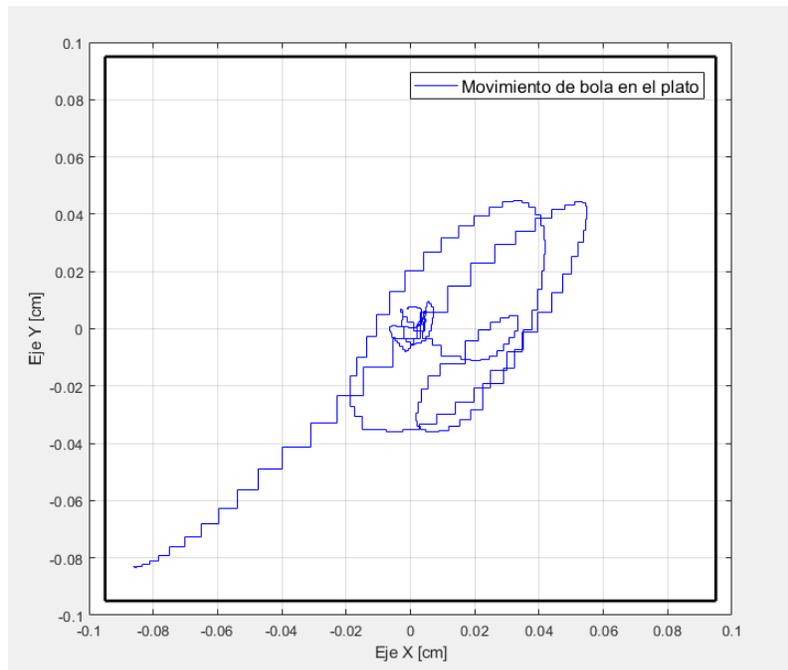


Figura 7.13: Movimiento de la bola durante el experimento 1 con control PD por aproximación rectangular

Al observar el movimiento de la bola en el plato vemos que empieza de un lado a otro pero a medida que pasa el tiempo va reduciendo la velocidad y la oscilación hasta colocar la bola lo más cerca al centro del plato.

A medida que se realizaron diferentes experimentos con esta aproximación se fue tratando de sintonizar a modo de ver que tan rápido respondía el sistema en base a sus límites de diseño, también se modificó el periodo T ya que la respuesta de la PC permitía llegar a un valor T más pequeño, con este valor de T se modificó el modelo en tiempo discreto del sistema de plato y bola y se realizaron las simulaciones correspondientes, para los valores de K_p y K_d se fueron sintonizando tratando de obtener una mejor respuesta del sistema real; la mejor respuesta obtenida con el control PD con aproximación rectangular se muestra en la Fig. 7.14

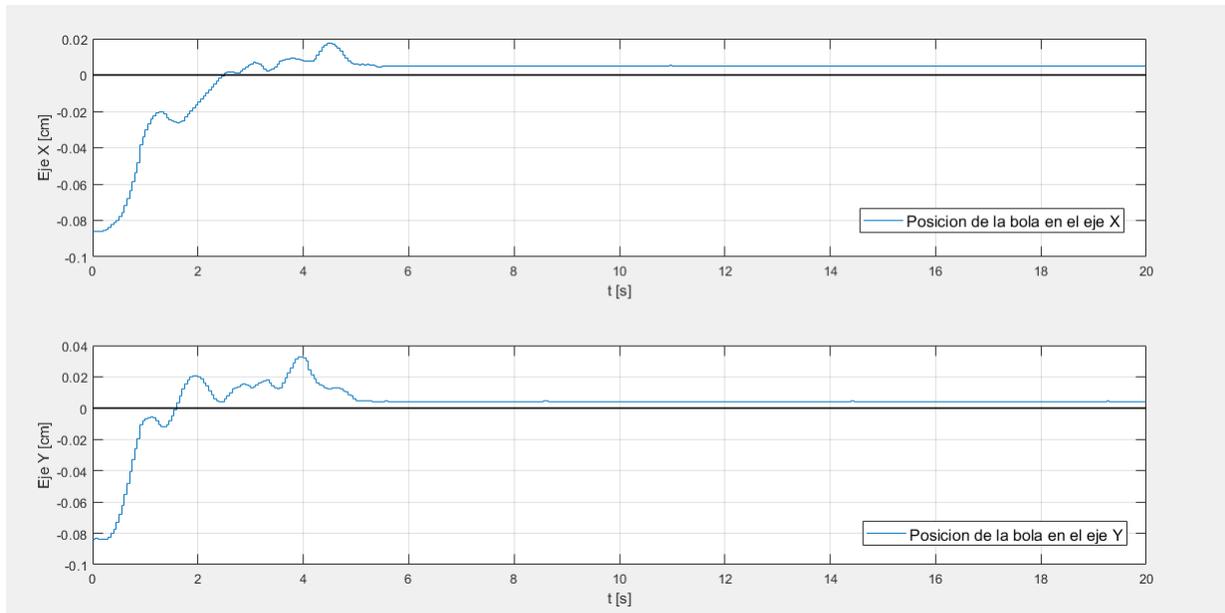


Figura 7.14: Experimento 2 con control PD por aproximación rectangular

Y la señal de control para cada eje resulto como se muestra en la Fig. 7.15

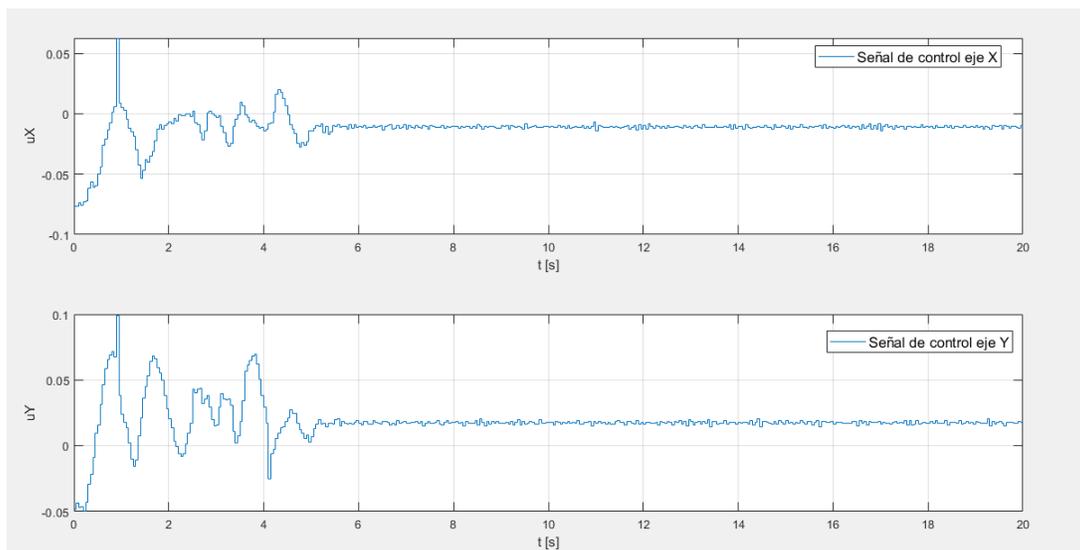


Figura 7.15: Señal de control de experimento 2 con control PD por aproximación rectangular

Al analizar vemos que se acerca más al valor deseado en menos tiempo, no se descarta que aún se puede seguir sintonizando con alguna técnica y tal vez obtener una mejor respuesta, pero por el momento cumple con el objetivo de control y en este controlador se reduce bastante el efecto de la inercia rotacional de la bola. Si comparamos nuevamente la posición en cada eje para ver el movimiento de la bola durante el experimento resulta como se muestra en la Fig. [7.16](#).

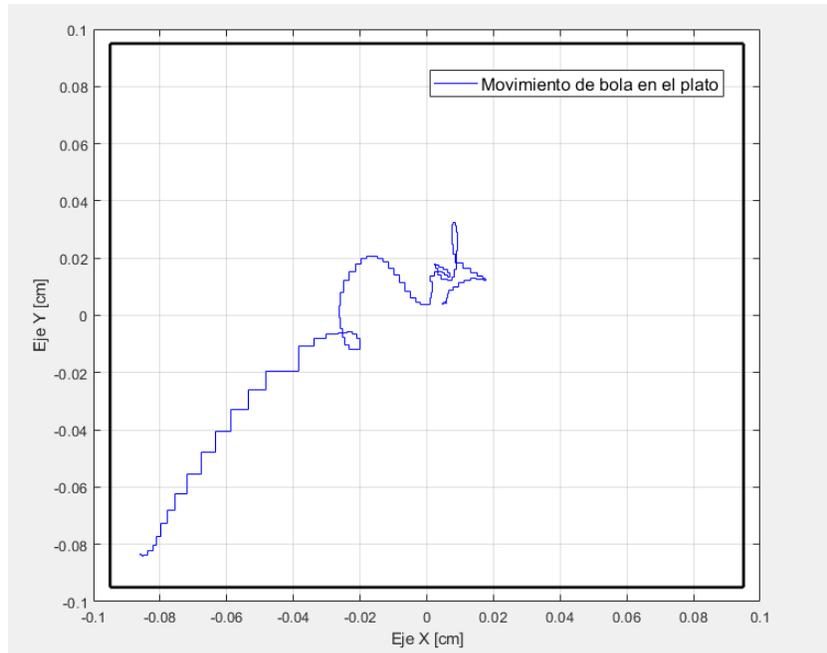


Figura 7.16: Movimiento de la bola durante el experimento 2 con control PD por aproximación rectangular

Una forma de probar el controlador es que tanto soporta las perturbaciones al sistema, las perturbaciones para los experimentos de cada controlador consistieron en mover con un lápiz la bola una vez que se estabiliza en el centro del plato y ver que tan rápido responde el controlador para llevar de nuevo la bola al centro del plato, el experimento con perturbaciones resulta como se muestra en la Fig. [7.17](#).

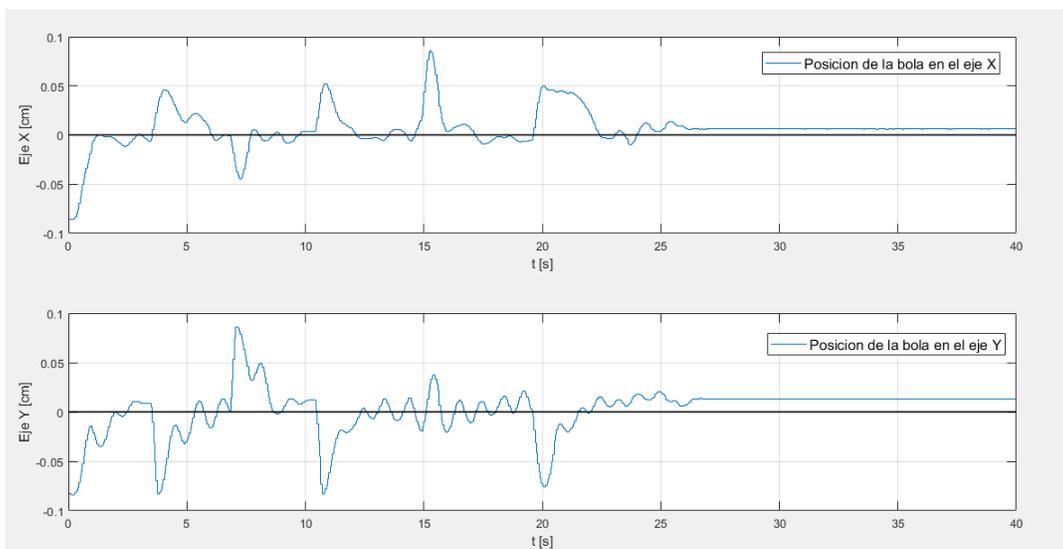


Figura 7.17: Experimento 3 con control PD por aproximación rectangular

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.18.

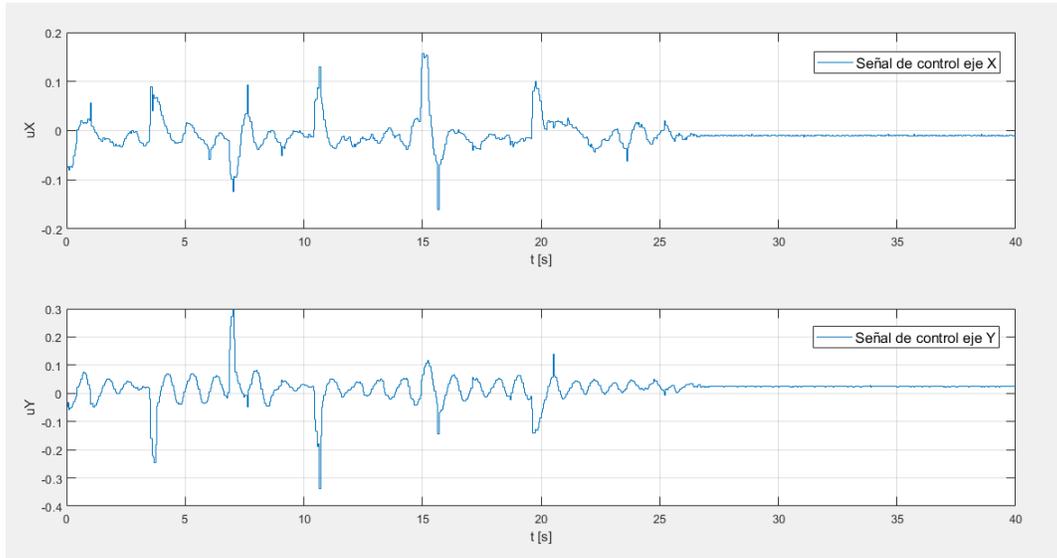


Figura 7.18: Señal de control de experimento 3 con control PD por aproximación rectangular

El controlador PD con aproximación rectangular soporta bastante bien las perturbaciones al sistema y trata de estabilizar la bola siempre en el centro del plato, si vemos el movimiento de la bola en el plato con estas perturbaciones, resulta como se muestra en la Fig. 7.19.

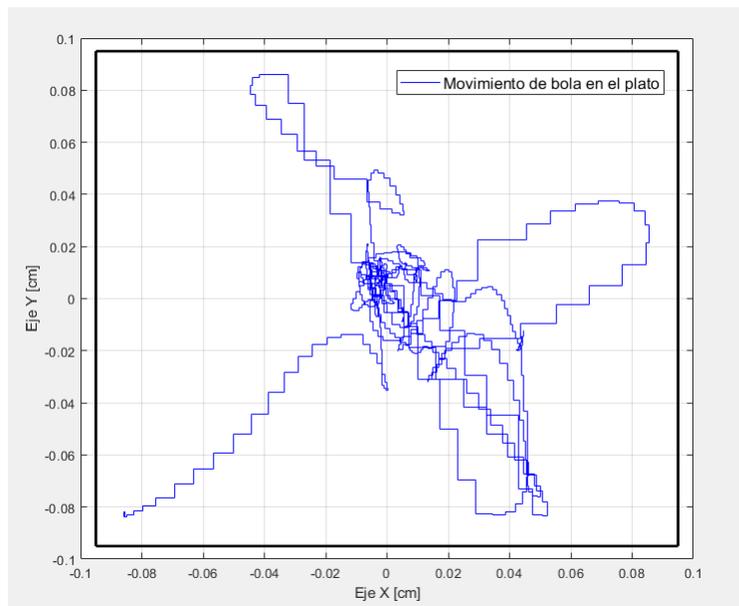


Figura 7.19: Movimiento de la bola durante el experimento 3 con control PD por aproximación rectangular

Al igual que con las simulaciones, el siguiente experimento consiste en integrar al control una trayectoria y ver cómo responde al control por seguimiento, en todos los experimentos se utilizó la figura de un círculo y se generó la señal para cada eje a modo de tratar de hacer esta figura. Durante los experimentos para la trayectoria del círculo se modificaron las ganancias de K_p y K_d que mejor respondían, de modo que el experimento con la trayectoria del círculo resultó como se muestra Fig. 7.20.

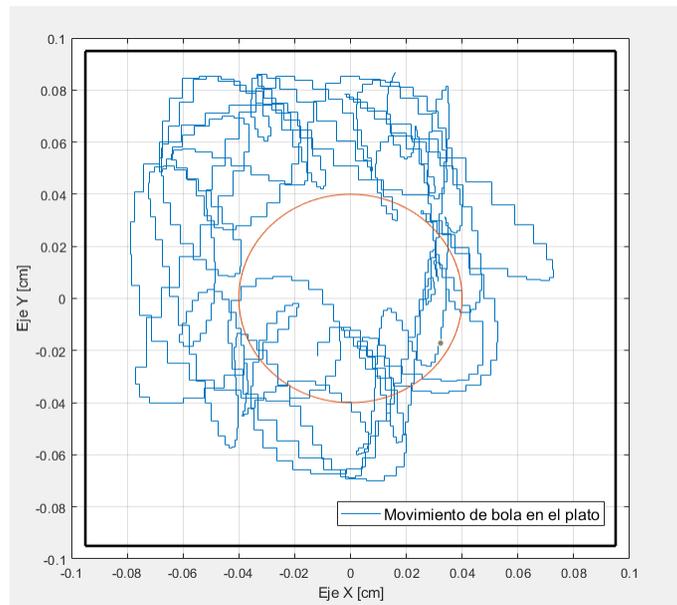


Figura 7.20: Experimento de círculo con control PD por aproximación rectangular

Como se observa la trayectoria que se genera no es un círculo, sin embargo se asemeja a una trayectoria circular, haciendo un análisis de comportamiento del sistema real en los diferentes experimento se llegó a la conclusión que cuando se trata de seguir una trayectoria afecta más la inercia rotacional de la bola y aunque no iguala la trayectoria del círculo, si parece responder el control por seguimiento, esto se ve mejor con la posición de cada eje como se muestra en la Fig. [7.21](#).

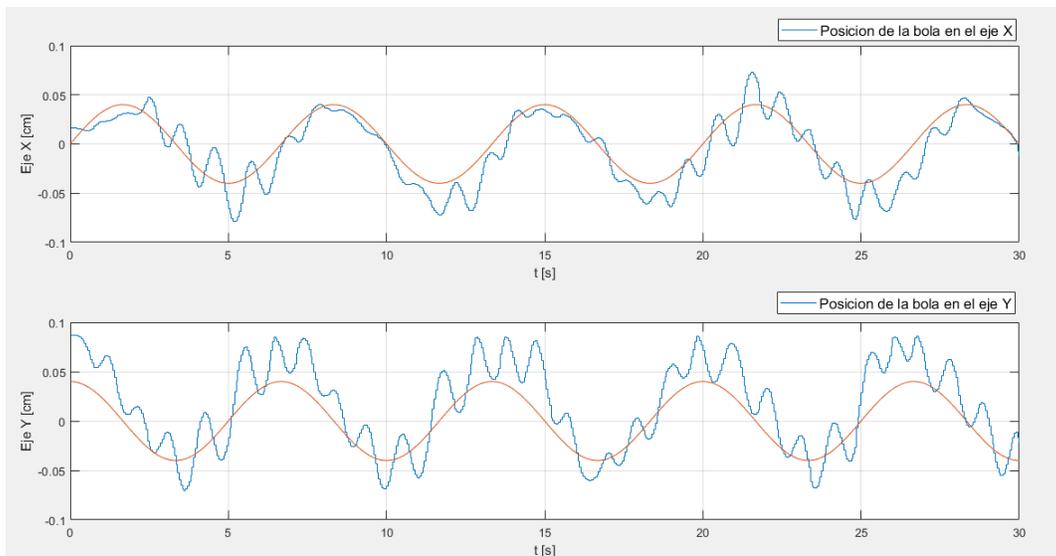


Figura 7.21: Experimento de círculo con control PD por aproximación rectangular en cada eje

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.22.

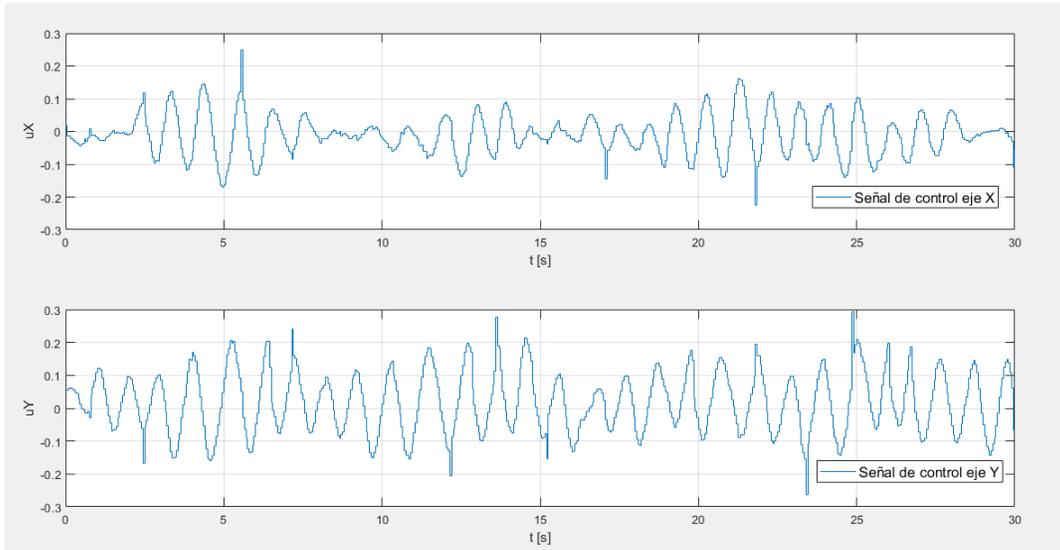


Figura 7.22: Señal de control de experimento de círculo con control PD por aproximación rectangular

Con estos experimentos se puede afirmar que el control PD cumple con el objetivo de control por regulación y parcialmente el control por seguimiento, para el caso del control por seguimiento resulta un problema la inercia rotacional de la bola y afecta bastante, aunque no se descarta que sintonizando con alguna técnica pueda mejorar el desempeño del controlador. El siguiente paso es probar un control PID con la aproximación rectangular y ver si mejora o no la respuesta del sistema real.

7.3. Control PID con Aproximación Rectangular Implementado

Utilizando la misma función de control en tiempo discreto que el PD con aproximación rectangular que es:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.5)$$

Con los valores para a_1 , a_2 , b_0 , b_1 y b_2 ya obtenidos para la aproximación rectangular:

$$a_1 = -1, a_2 = 0 \quad (7.6)$$

$$a_2 = 0 \quad (7.7)$$

$$b_0 = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \quad (7.8)$$

$$b_1 = -K_p \left(1 + 2\frac{T_d}{T}\right) \quad (7.9)$$

$$b_2 = -K_p \frac{T_d}{T} \quad (7.10)$$

Se utiliza esta misma ecuación en el sistema real y se ajusta el bloque de programación con estos parámetros. Al igual que con el control PD primero se realizaron los experimentos con las ganancias obtenidas en las simulaciones del modelo en tiempo discreto. (Salvá y cols., s.f.). De modo que el primer experimento con el control PID con aproximación rectangular resultó como se muestra en la Fig. 7.23

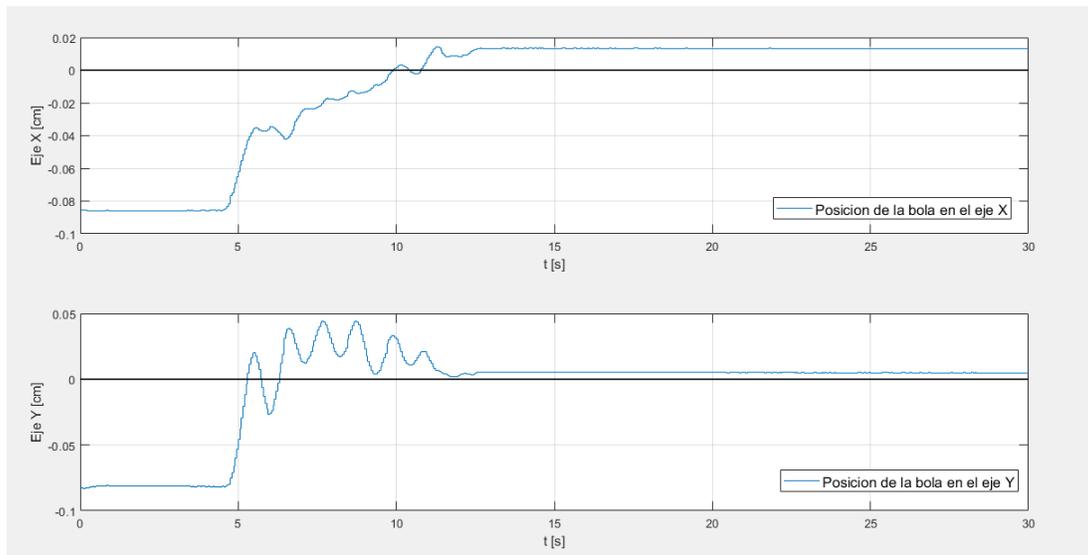


Figura 7.23: Experimento 1 con control PID por aproximación rectangular

Y la señal de control para cada eje resulto como se muestra en la Fig. [7.24](#)

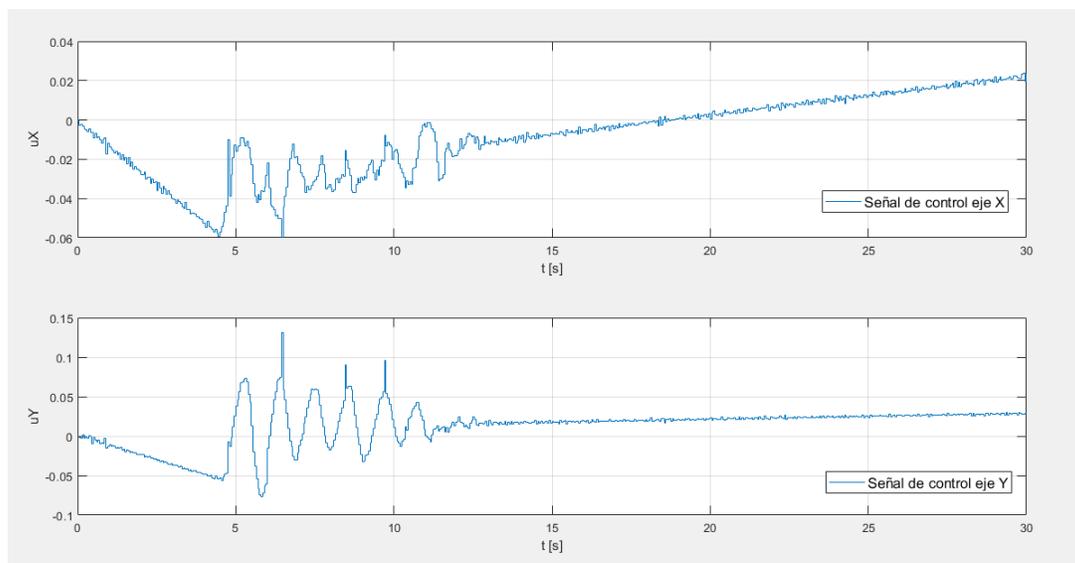


Figura 7.24: Señal de control de experimento 1 con control PID por aproximación rectangular

Al analizar el comportamiento del sistema vemos que si trata de estabilizar el sistema pero lo hace muy lento, esto se debe principalmente a que en la función en tiempo discreto la ecuación requiere la señal de control $(k - 2T)$ y $(k - 1T)$ y tarda ese tiempo hasta obtener estos valores y comenzar a llevar la bola al centro del plato. Adicional a esto a medida que se realizaron los experimentos se observó que la oscilación de la bola no disminuía, esto se debe a que el sistema en si se comporta como un doble integrador por la función de transferencia $\frac{1}{s^2}$, de modo que en lazo cerrado hay un triple integrador con el PID.

Si comparamos la posición del eje x contra el eje y se observa el movimiento de la bola durante el primer experimento del controlador PID como se muestra en la Fig. 7.25.

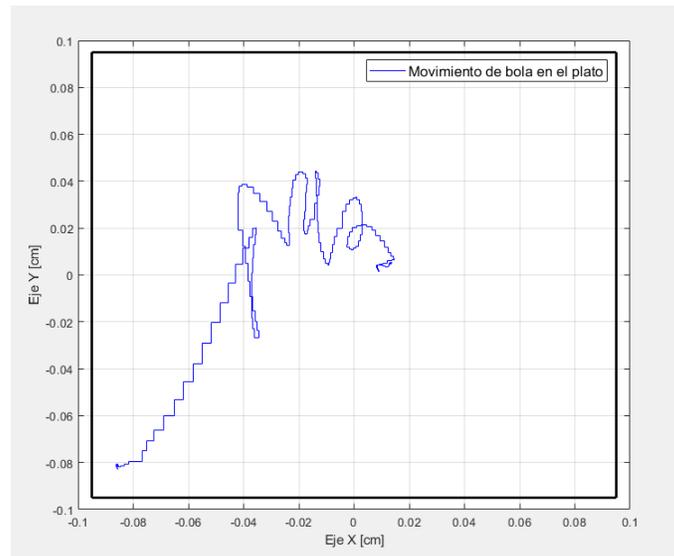


Figura 7.25: Movimiento de la bola durante el experimento 1 con control PD por aproximación rectangular

A pesar de esto se trató de sintonizar lo mejor posible en base a lo permitido por el sistema y al igual que con el control PD se observó que el periodo T puede hacerse más corto, con este valor T se modificaron las simulaciones para obtener otras ganancias de K_p , K_i y K_d , se trató de hacer una respuesta más rápida del control para estabilizar mejor el sistema, después de muchos experimentos la mejor respuesta que se pudo obtener se muestra en la Fig. 7.26.

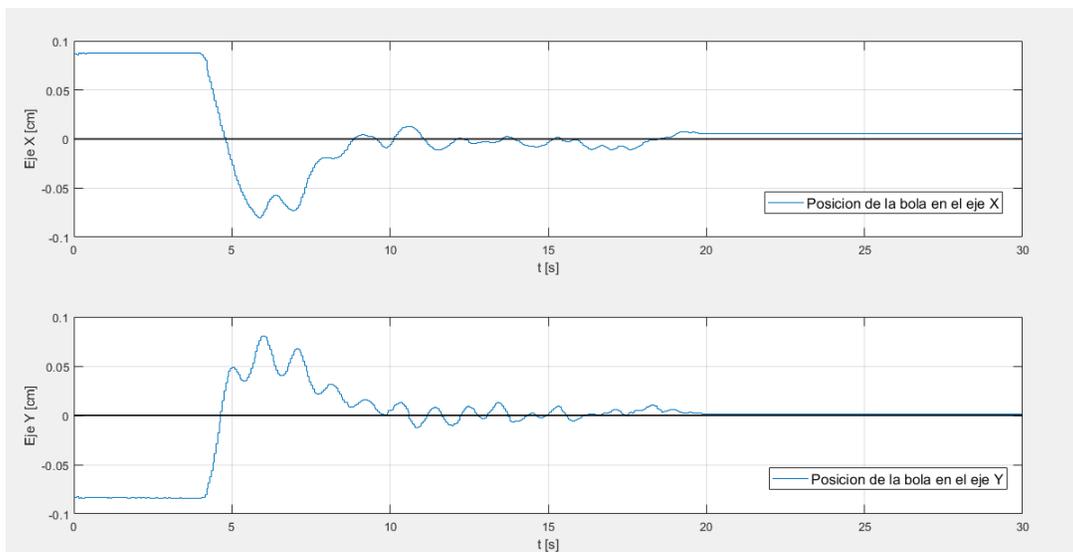


Figura 7.26: Experimento 2 con control PID por aproximación rectangular

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.27.

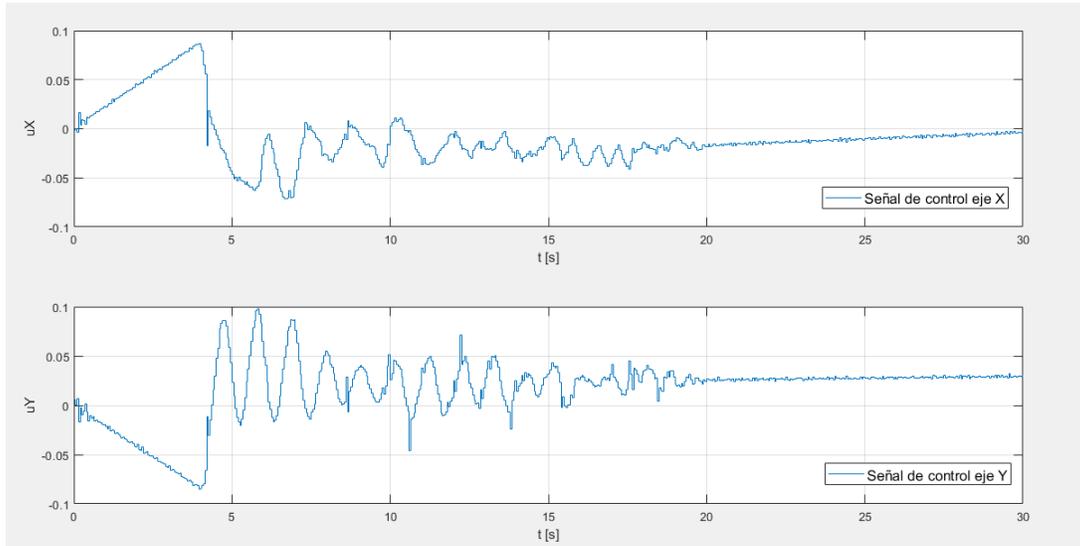


Figura 7.27: Señal de control de experimento 2 con control PID por aproximación rectangular

El desempeño del controlador PID mejoró bastante una vez que se sintoniza en el sistema real, aunque se tiene un sobrepaso al principio del experimento y también compensa un poco la oscilación de la bola; con estos experimentos se puede definir que el controlador PID cumple con el objetivo de control con un sobrepaso inicial considerable. Si comparamos la posición del eje x contra el eje y se observa el movimiento de la bola durante el experimento del controlador PID como se muestra en la Fig. 7.28.

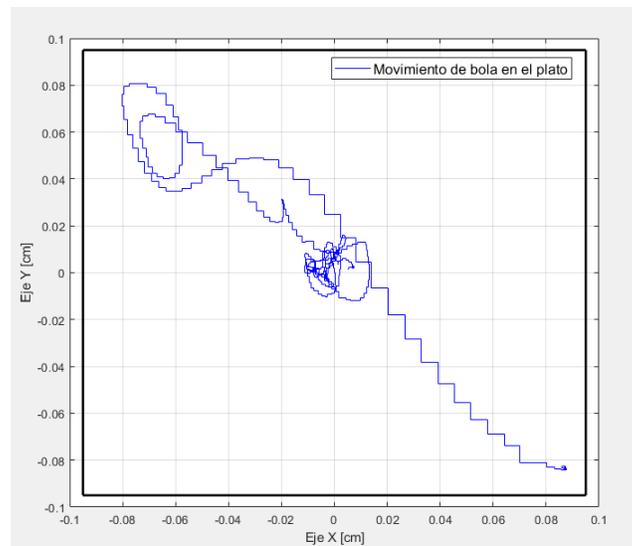


Figura 7.28: Movimiento de la bola durante el experimento 2 con control PID por aproximación rectangular

La siguiente prueba para el controlador PID con aproximación rectangular es ver que tanto puede soportar perturbaciones al sistema de modo que el experimento con perturbaciones al sistema resulte como se muestra en la Fig. 7.29.

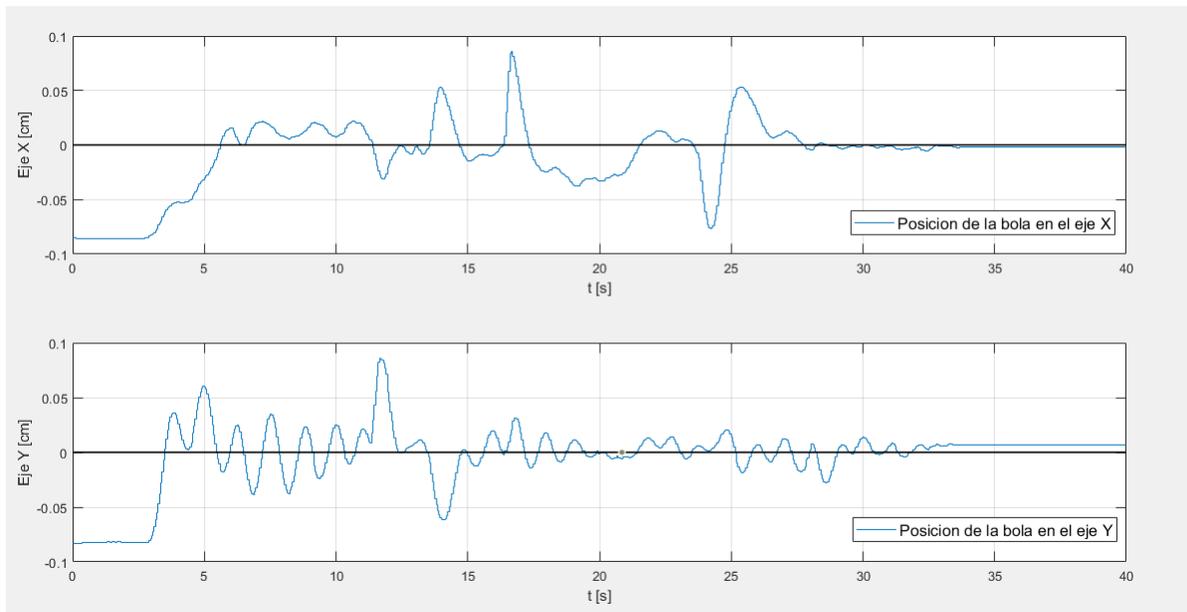


Figura 7.29: Experimento 3 con control PID por aproximación rectangular

Y la señal de control para cada eje resulta como se muestra en la Fig. [7.30](#)

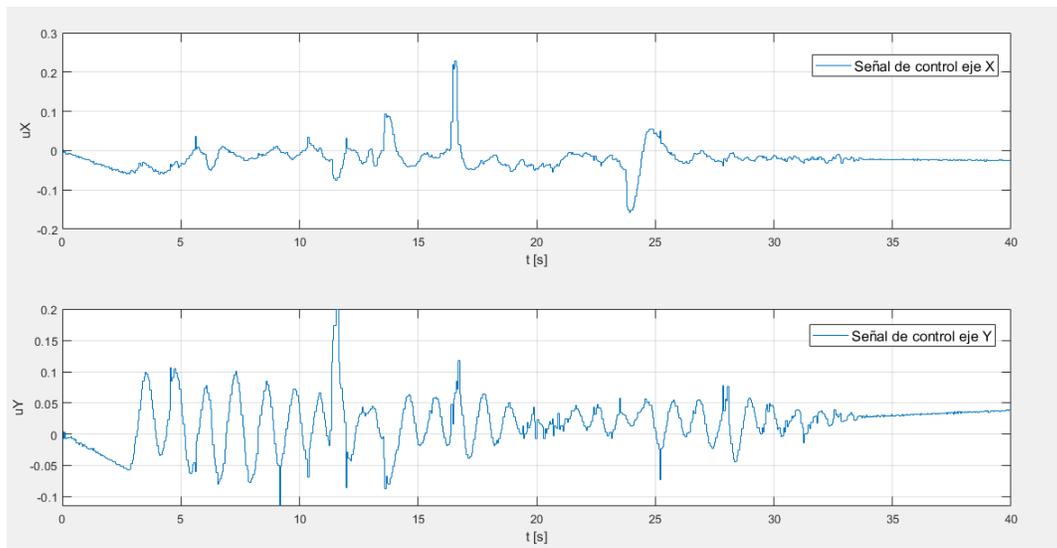


Figura 7.30: Señal de control de experimento 3 con control PID por aproximación rectangular

Se observa que corrige la posición de la bola bastante bien, similar al controlador PD, aunque tarda un poco más en llevar la bola de nuevo al centro, pero se puede afirmar que el controlador PID rectangular soporta bastante bien las perturbaciones al sistema. Si comparamos la posición del eje x contra el eje y se observa el movimiento de la bola durante el experimento del controlador PID como se muestra en la Fig. [7.31](#)

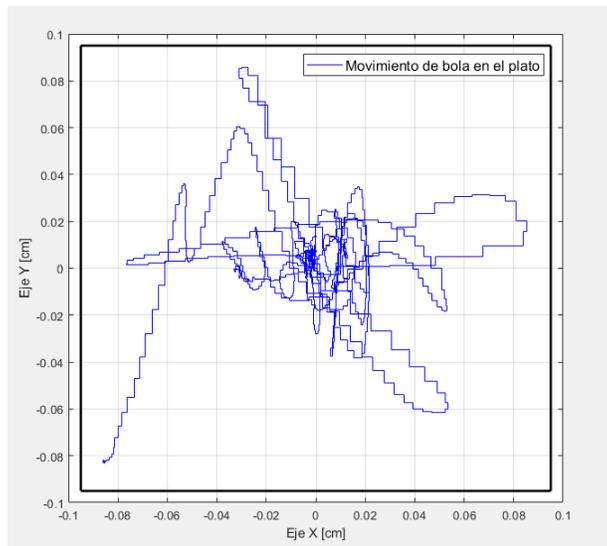


Figura 7.31: Movimiento de la bola durante el experimento 3 con control PID por aproximación rectangular

La siguiente prueba para el controlador es aplicar el control por seguimiento a través de la trayectoria para generar el círculo, a medida que se realizaron los experimentos se sintonizó de nuevo para que el control PID respondiera más rápido, los resultados del experimento para el control por seguimiento se muestra en la Fig. [7.32](#)

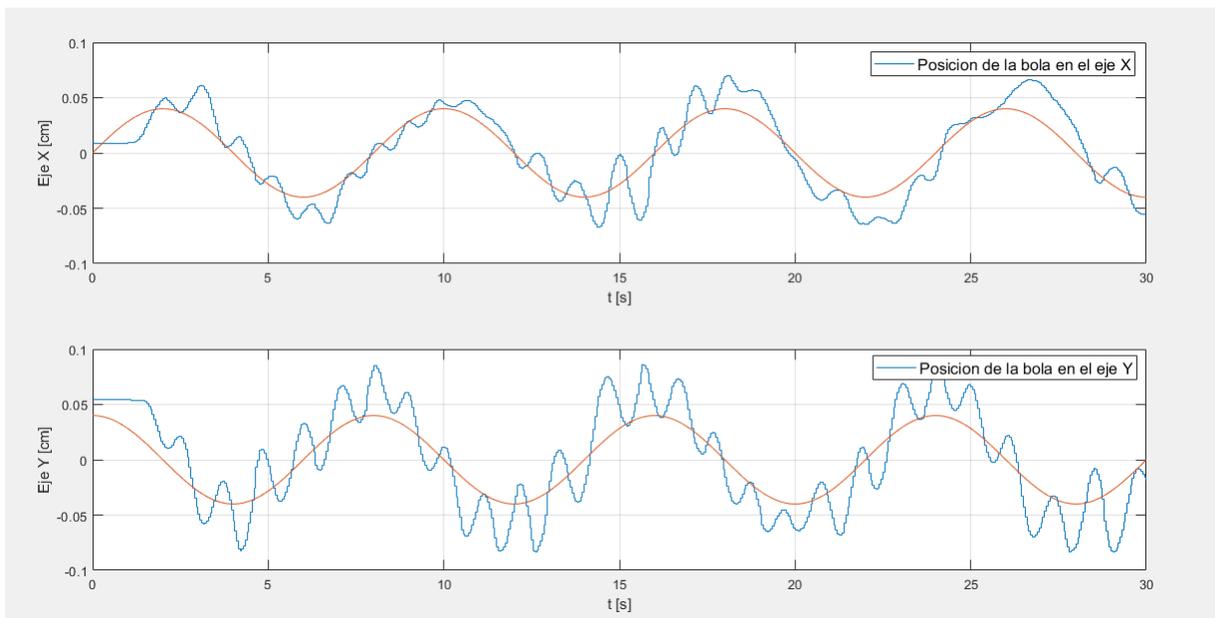


Figura 7.32: Experimento de círculo con control PID por aproximación rectangular en cada eje

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.33.

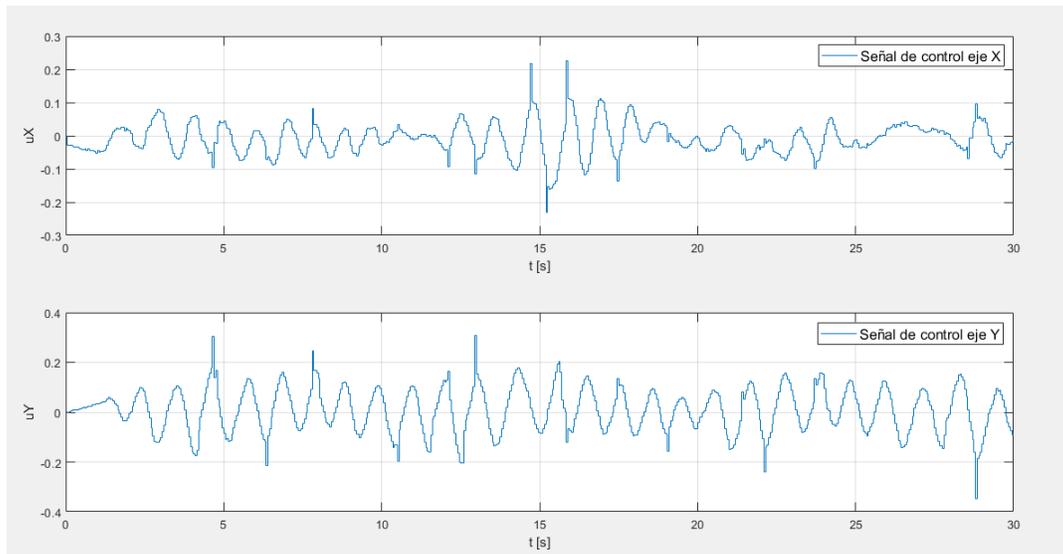


Figura 7.33: Señal de control de experimento de círculo con control PD por aproximación rectangular

Por último comparamos la posición del eje x contra el eje y se observa el movimiento de la bola durante el experimento del controlador PID como se muestra en la Fig. 7.34.

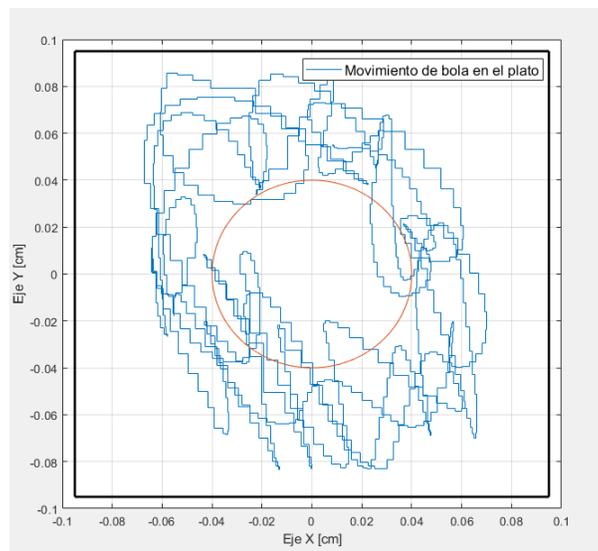


Figura 7.34: Experimento de círculo con control PD por aproximación rectangular

Al igual que con el controlador PD, la trayectoria que se genera no es un círculo, aunque en este caso la oscilación aumenta por lo que ya se explicó del triple integrador y por lo tanto la inercia rotacional de bola que genera que la figura no llegue a generar el círculo en ningún momento. No se descarta que el control PID pueda mejorar su desempeño si se sintoniza con alguna técnica de sintonización, pero en comparación con el control PD, el control PID tiene un mejor desempeño que el control PID, ambos en la aproximación rectangular, además de que el control PID cumple parcialmente el control por regulación y por seguimiento. el siguiente paso es probar el control PD y PID con la aproximación de Tustin y ver su comportamiento con el sistema real y comparar con los controladores previamente integrados.

7.4. Control PD con Aproximación de Tustin Implementado

La diferencia para implementar estos controladores en comparación con los controladores anteriores es la aproximación en tiempo discreto, para este caso se utilizó la aproximación de Tustin ya que fue la otra aproximación con la que también se realizaron simulaciones. Recordando la señal de control:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.11)$$

Se aplican los valores para la aproximación de Tustin que son:

$$a_1 = 1 \quad (7.12)$$

$$a_2 = 0 \quad (7.13)$$

$$b_0 = K_p \left(1 + \frac{2T_d}{T}\right) \quad (7.14)$$

$$b_1 = K_p \left(1 - \frac{2T_d}{T}\right) \quad (7.15)$$

Esta ecuación de control se cambia en el bloque de programación para implementarlo en el sistema real. (Salvá y cols., s.f.) y como en los otros controladores, se realizaron los experimentos empezando con el control PD; la respuesta del sistema resulto como se muestra en la Fig. 7.35

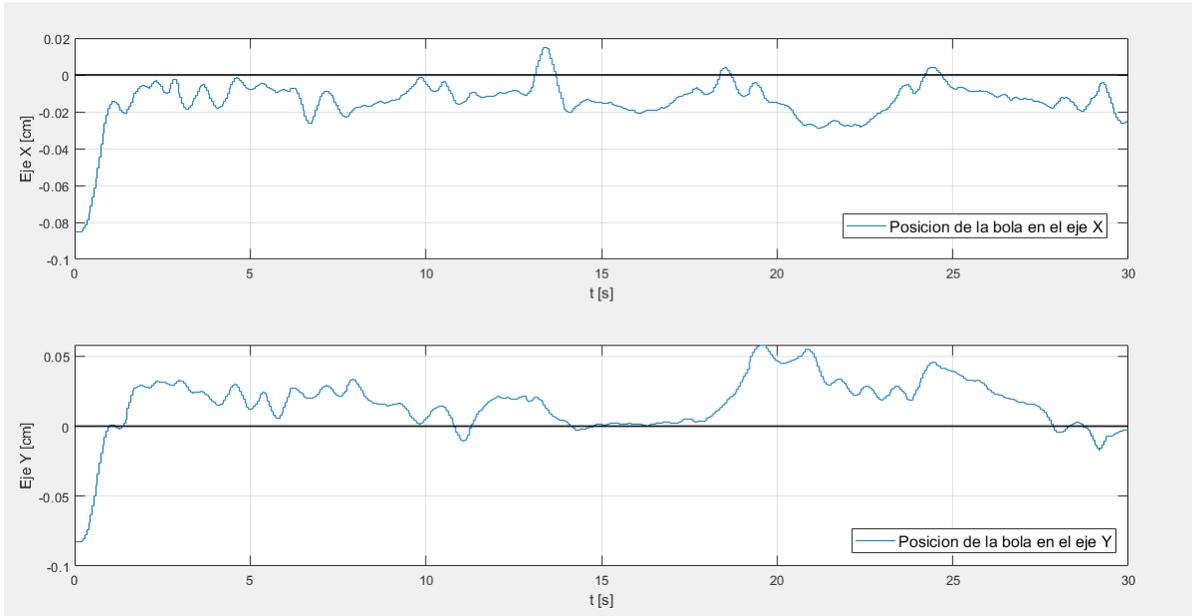


Figura 7.35: Experimento con control PD por aproximación de Tustin

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.36.

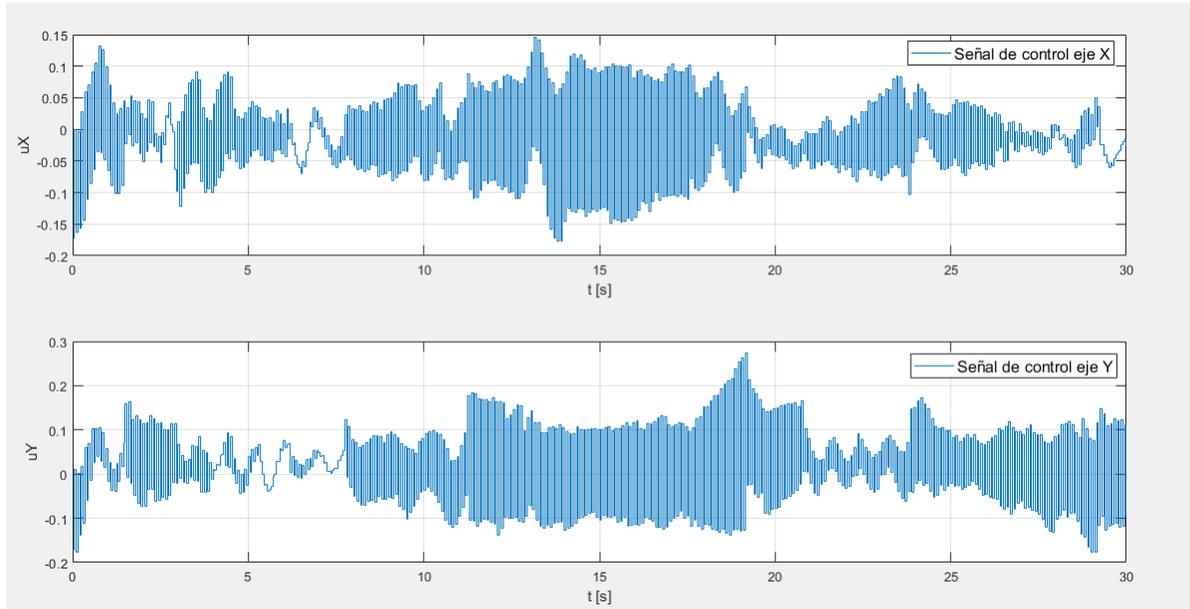


Figura 7.36: Señal de control de experimento con control PD por aproximación de Tustin

Si comparamos la posición del eje x contra el eje y se observa el movimiento de la bola durante el experimento del controlador PD como se muestra en la Fig. 7.37.

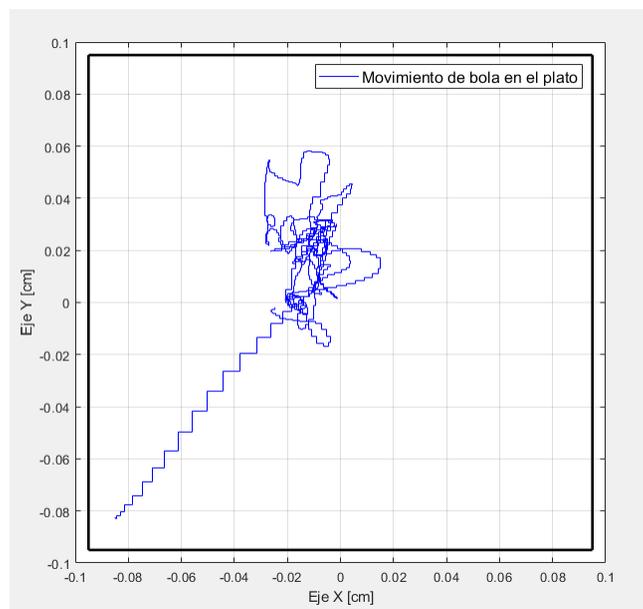


Figura 7.37: Movimiento de la bola durante el experimento con control PD por aproximación de Tustin

A pesar de que el control trata de llevar la bola al centro del plato no mantiene la posición, esto es debido a que la señal de control es muy rápida en comparación con los otros controladores y esto ocasiona al sistema real que suba y baje muy rápido los servomotores generando más perturbación al sistema, adicional a la inercia rotacional, lo que ocasiona que la bola nunca se mantenga en un punto fijo. A pesar de esto se trató de sintonizar un poco más, pero los resultados fueron los mismos, además de que en algunos experimentos no llegaba al valor deseado y generaba trayectorias en el plato completamente diferentes por tanta oscilación.

Este comportamiento ya se esperaba por las simulaciones en tiempo discreto y al menos se demostró que en el sistema real como simulado el comportamiento fue el mismo. Para este controlador como no cumple el objetivo de control por regulación no se realizaron pruebas con perturbación al sistema ya que sin perturbación no lleva la bola al centro del plato y la mantiene en un punto; aunque si se probó el control por seguimiento y los resultados tampoco fueron muy satisfactorios ya que el control en ningún momento trata de seguir la trayectoria y una vez que comienza el experimento el controlador genera una trayectoria completamente diferente al círculo, por esta razón se puede afirmar que el controlador PD con aproximación de Tustin no cumple con el objetivo de control ni por regulación ni por seguimiento; lo siguiente es probar el control PID con aproximación de Tustin y ver si cumple con el objetivo de control.

7.5. Control PID con Aproximación de Tustin Implementado

Recordando la señal de control:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.16)$$

Y sustituyendo los valores para al aproximacion de Tustin del control PID que son:

$$a_1 = -1 \quad (7.17)$$

$$a_2 = 0 \quad (7.18)$$

$$b_0 = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \quad (7.19)$$

$$b_1 = -K_p \left(1 + 2\frac{T_d}{T}\right) \quad (7.20)$$

$$b_2 = -K_p \frac{T_d}{T} \quad (7.21)$$

Se cambia en el bloque de programación para implementarlo en el sistema real. (Salvá y cols. s.f.) y se realizan los experimentos para el control PID como con los otros controladores. Particularmente en este controlador a medida que se realizaron los experimentos para que el plato llevara la bola al centro del plato se observó que en ningún momento el plato llevaba la bola al centro del plato y la oscilación es tal que genera trayectorias completamente diferentes entre los experimentos.

Este comportamiento ya esperaba por las simulaciones en tiempo discreto y por como respondió el control PD con aproximación de Tustin, aunque en el controlador PID la oscilación es mucho mayor en el control con aproximación de Tustin, adicional a la oscilación de la propia señal de control más la inercia rotacional y agregando un integrador más, teniendo un triple integrador. Por esta razón la bola nunca llega al centro del plato. A pesar de esto, se trató de sintonizar lo mejor posible, pero los experimentos mostraron que no sigue la señal de control ni en regulación ni en seguimiento y solo mueve la bola de un lado a otro en todos los experimentos, por esta razón se llegó a la conclusión que el control PID con la aproximación de Tustin no funciona para este sistema y se optó por probar otros controladores como el controlador PD de orden fraccionario.

7.6. Control PD Fraccionario Implementado

Al igual que con los otros controladores, para el caso de los controladores de orden fraccionario con la aproximación de Oustaloup se puede utilizar la misma ecuación de la señal de control $u(kT)$ que es:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.22)$$

Esta señal se puede implementar al controlador real, que sigue siendo la PC con el software de MATLAB; a partir de la función de transferencia del PD fraccionario aproximado y discretizado y modificándola para que tenga la misma forma que la señal de control $u(kT)$.

A través de MATLAB es posible tener la función de transferencia del PD fraccionario ya con las ganancias K_p y K_d obtenidas en las simulaciones, además de definir a μ con un valor entre 0.1 y 0.9; las simulaciones indicaron que un valor cercano a uno el controlador PD fraccionario responde bastante bien de modo que se escogió como $\mu=0.9$ para implementarlo en el sistema real, con este valor obtenemos la función de transferencia de la siguiente forma:

$$G_{PD\mu=0.8} = \frac{-13.71z^3 + 29.62z^2 - 19.9z + 3.939}{z^3 - 1.04z^2 - 0.1771z + 0.2516} \quad (7.23)$$

Esta ecuación se ajusta para obtener los valores para a_0, a_1, a_2 y a_3 al igual que b_0, b_1, b_2 y b_3 en la ecuación $u(kT)$ y obtenemos la señal de control del PD de orden fraccionario para el sistema del plato y bola. Se ajusta los parámetros para obtener los valores tanto del error como del control en tiempo $k-2$ y $k-3$ y obtenemos la respuesta de cada eje como se muestra en la Fig. [7.38](#).

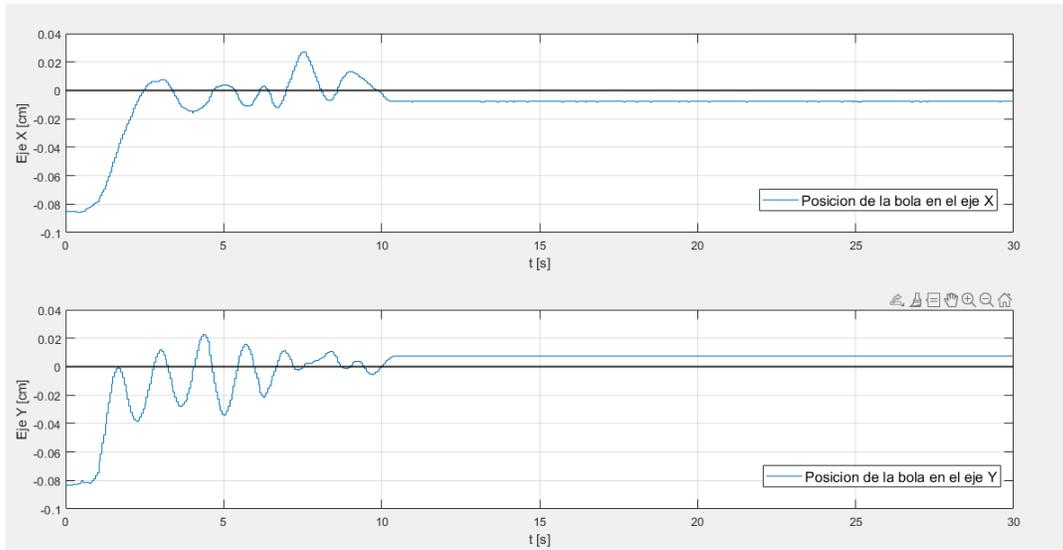


Figura 7.38: Experimento 1 con control PD fraccionario implementado

Y la señal de control para cada eje resulto como se muestra en la Fig. [7.39](#).

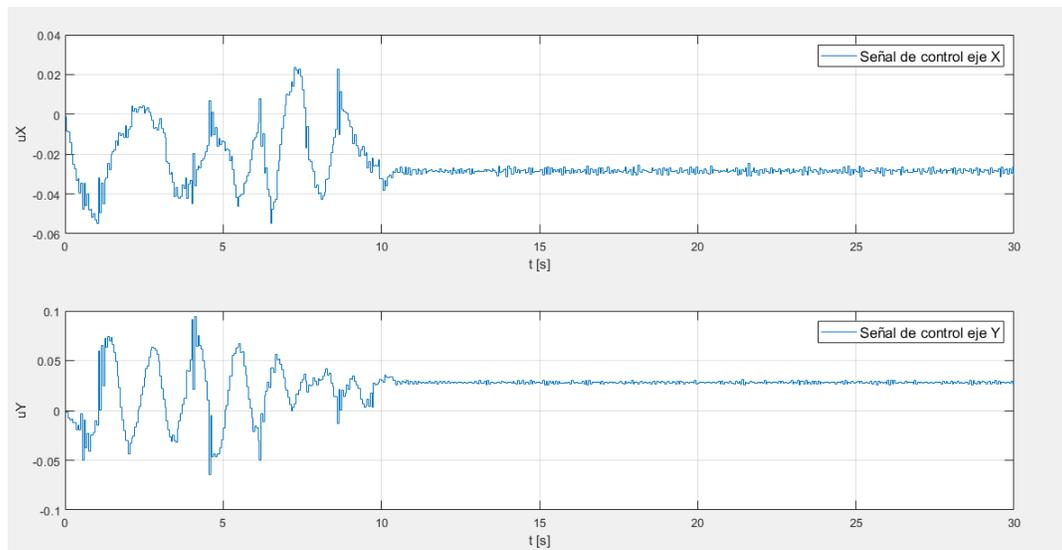


Figura 7.39: Señal de control de experimento 1 con control PD fraccionario implementado

Con este controlador en particular la oscilación se redujo bastante y aunque sigue afectando la oscilación de la bola en el centro del plato el menor; si comparamos la posición en ambos ejes tenemos el movimiento de la bola durante el experimento y resulta como se muestra en la Fig. 7.40.

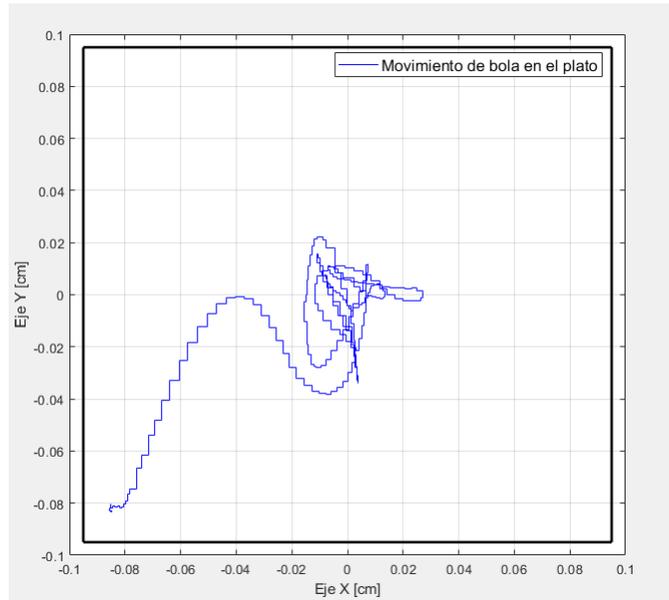


Figura 7.40: Movimiento de la bola durante el experimento 1 con control PD fraccionario implementado

A medida que se realizaron los experimento se trató de sintonizar mejor las ganancias de K_p y K_d y con la ventaja de tener otro grado de libertad con μ también se realizaron experimentos acercando el valor de $\mu = 0.99$, los resultados mejoraron a medida que se ajustaron estas 3 variables de modo que la mejor respuesta que se obtuvo resultado como se muestra en la Fig. 7.41.

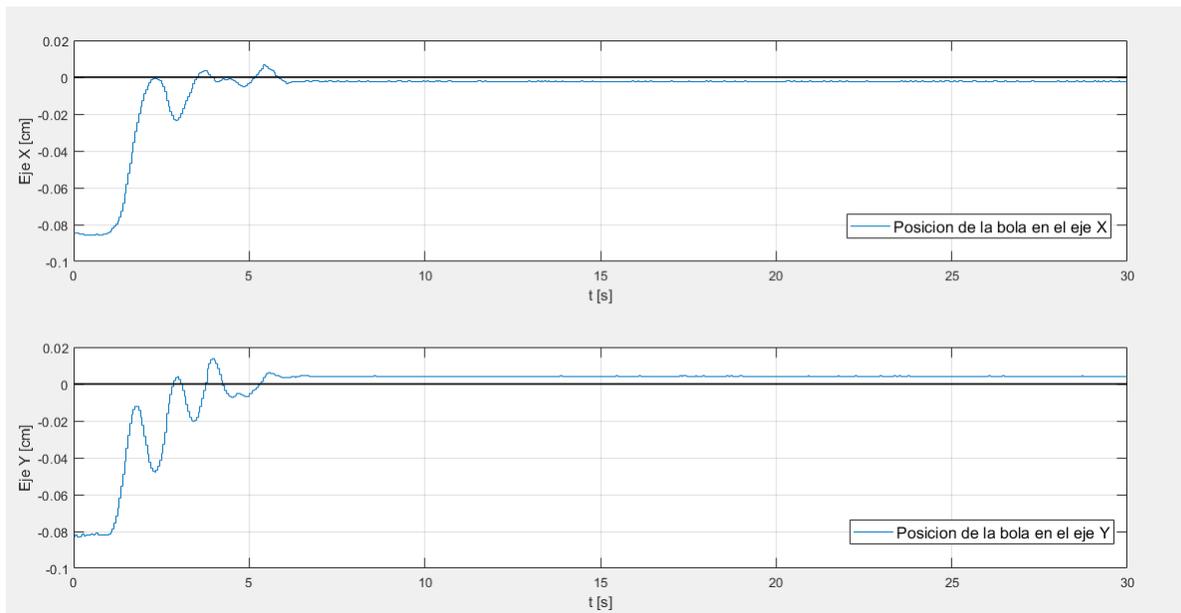


Figura 7.41: Experimento 2 con control PD fraccionario implementado

Y la señal de control para cada eje resulta como se muestra en la Fig. 7.42.

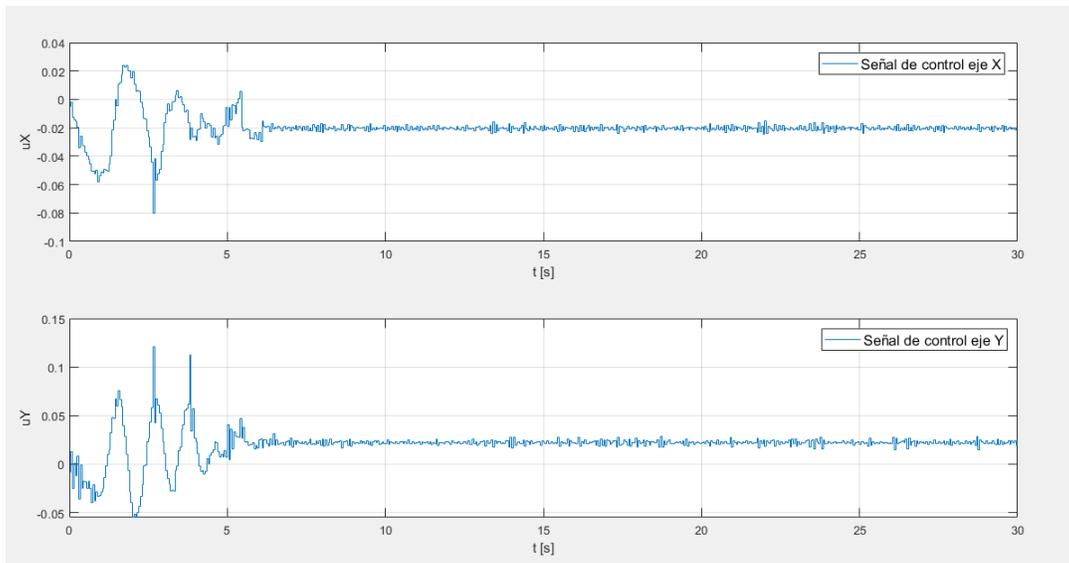


Figura 7.42: Señal de control de experimento 2 con control PD fraccionario implementado

Una vez sintonizado el control PD de orden fraccionario se redujo casi por completo la oscilación de la bola en el plato, el hecho de que el controlador de orden fraccionario genere señales de control más suaves hace que la oscilación se reduzca bastante, si comparamos la posición en ambos ejes tenemos el movimiento de la bola durante el experimento y resulta como se muestra en la Fig. 7.43.

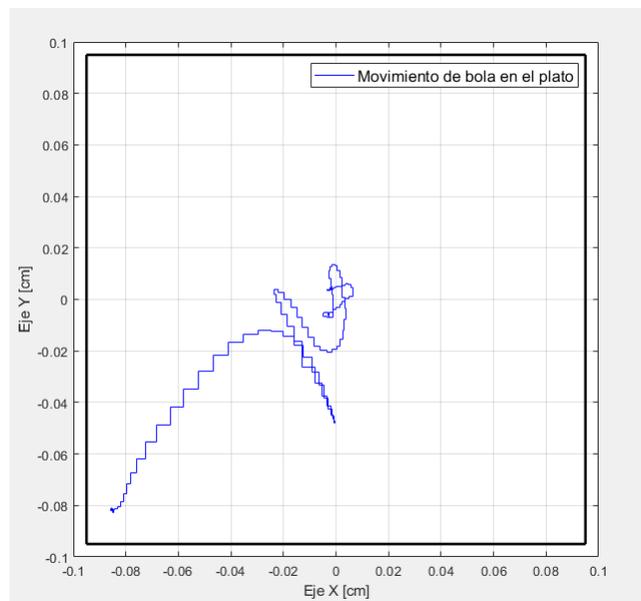


Figura 7.43: Movimiento de la bola durante el experimento 2 con control PD fraccionario implementado

Al igual que con los otros controladores, se realizaron pruebas al controlado PD fraccionario para ver cómo responde a las perturbaciones al sistema, las perturbaciones consisten en mover la bola con un lápiz una vez que llega al centro del plato y que durante todo el experimento el controlador lleve de nuevo la bola al centro del plato, los resultados de los experimentos con perturbación al sistema se muestran en la Fig. 7.44.

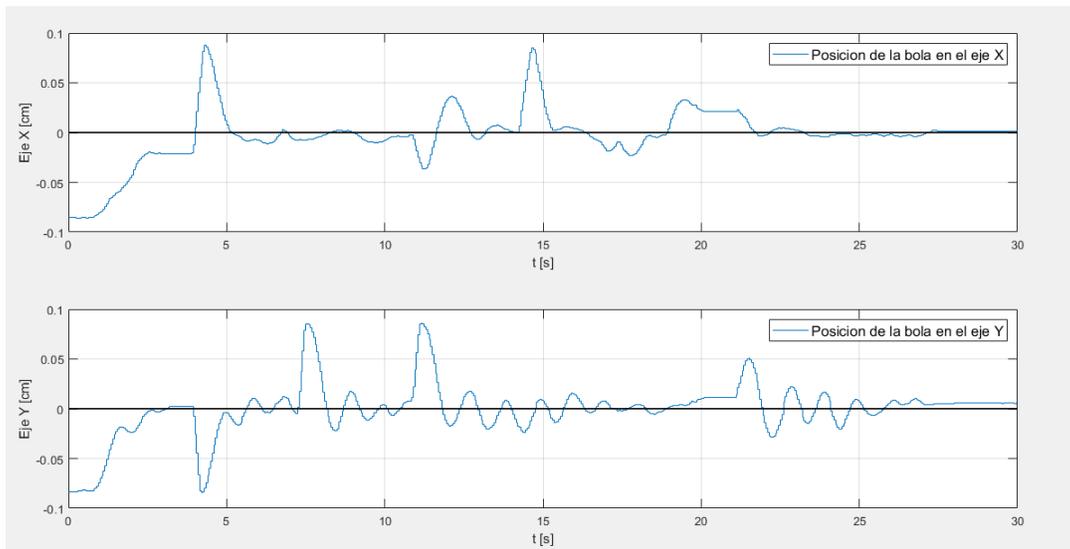


Figura 7.44: Experimento 3 con control PD fraccionario implementado

Y la señal de control para cada eje resulta como se muestra en la Fig. [7.45](#).

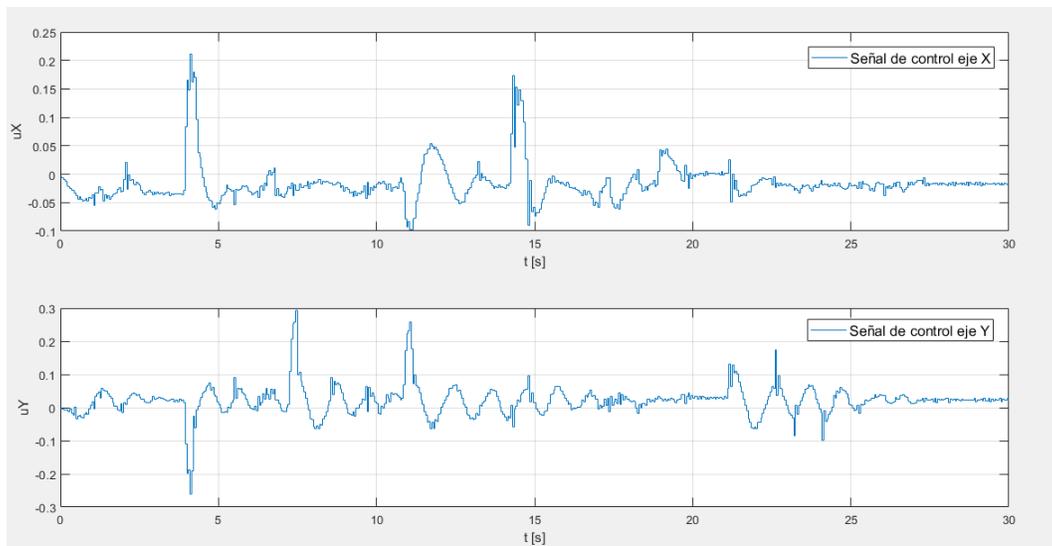


Figura 7.45: Señal de control de experimento 3 con control PD fraccionario implementado

El control PD fraccionario soporta las perturbaciones al sistema igual que los otros controladores con la ventaja de que se reduce la oscilación del sistema, ocasionando que el control lleva un poco más rápido la bola de nuevo al centro una vez que se mueve; si comparamos la posición en ambos ejes tenemos el movimiento de la bola durante el experimento y resulta como se muestra en la Fig. [7.46](#).

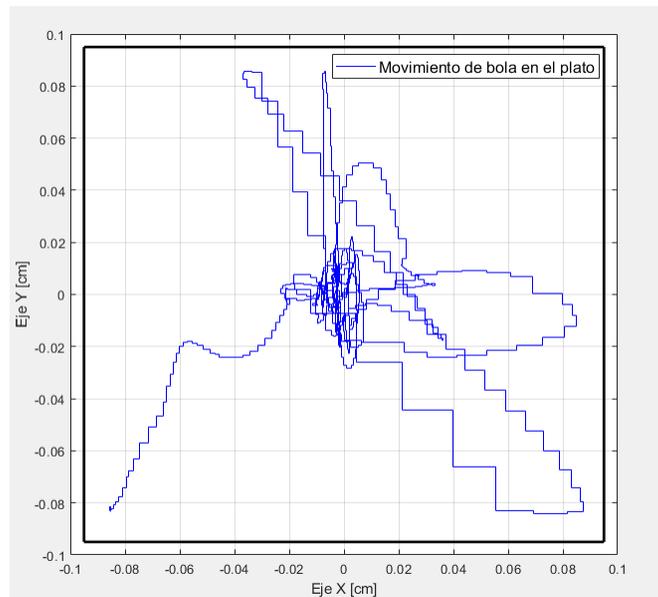


Figura 7.46: Movimiento de la bola durante el experimento 3 con control PD fraccionario implementado

La última prueba para este controlador es aplicar el control por seguimiento a través de la trayectoria para generar el círculo, a medida que se realizaron los experimentos se sintonizó de nuevo para que el control PD de orden fraccionario respondiera más rápido y tratar de seguir lo mejor posible la trayectoria, los resultados del experimento para el control por seguimiento se muestra en la Fig [7.47](#)

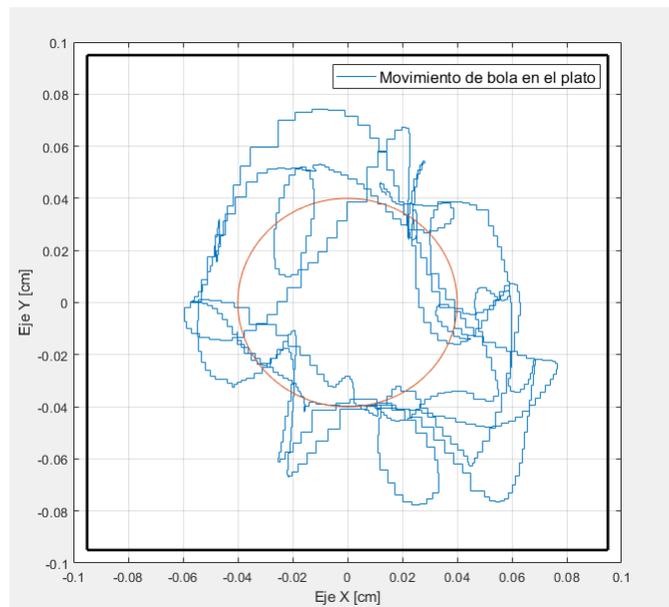


Figura 7.47: Experimento de círculo con control PD fraccionario

La trayectoria que genera no es un círculo y aunque trata de seguir la trayectoria en algunos momentos, la inercia rotacional que se genera es más rápida que la respuesta del controlador y como en todas las pruebas de control por seguimiento oscila demasiado que nunca genera el círculo. Sin embargo, se observó que para cada eje el valor se aproximaba en ciertos momentos del experimento, el resultado de los experimentos de seguimiento para cada eje se muestra en la Fig [7.48](#)

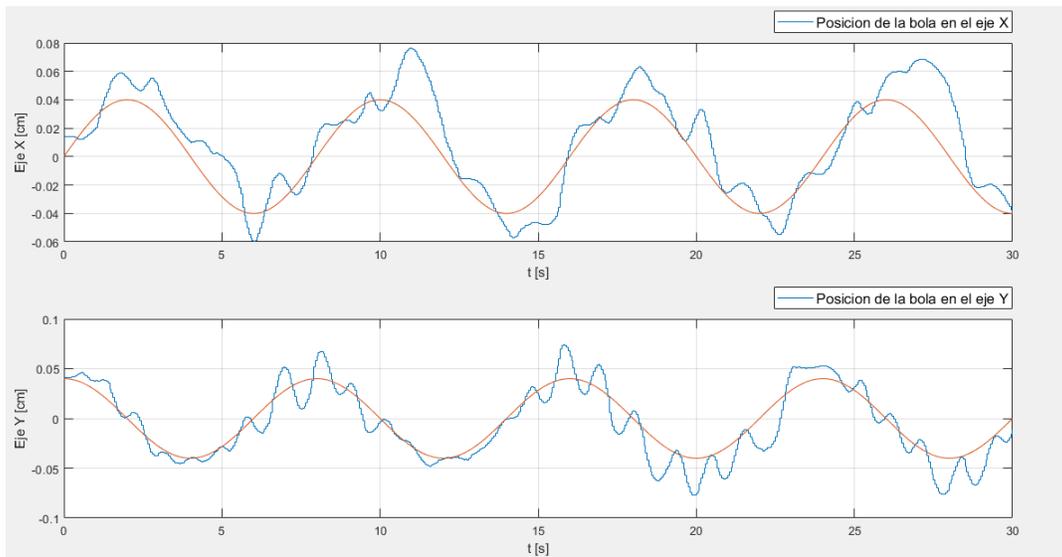


Figura 7.48: Experimento de círculo con control PD fraccionario

Y la señal de control para cada eje resulta como se muestra en la Fig. [7.49](#)

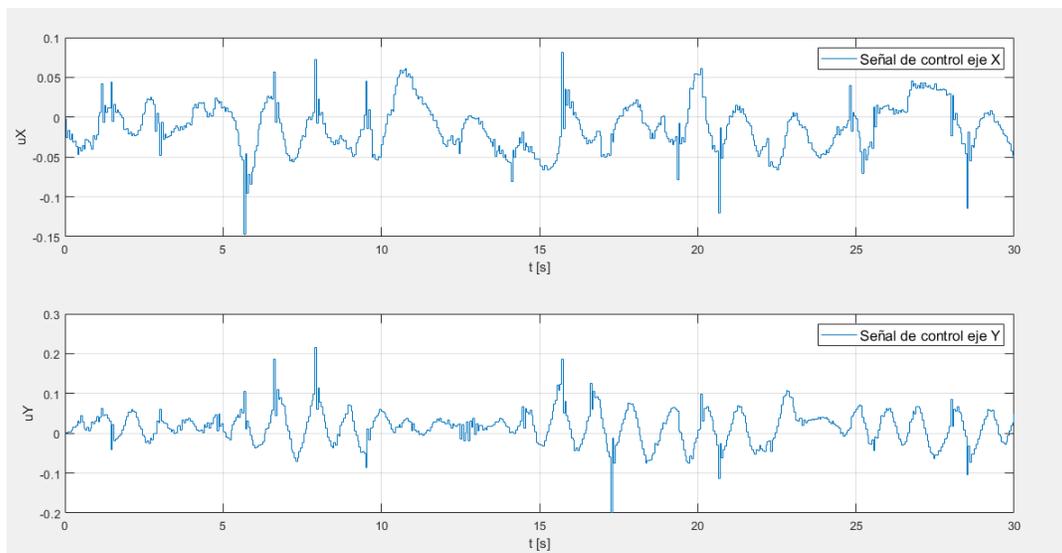


Figura 7.49: Señal de control de experimento 3 con control PD fraccionario implementado

El control PD fraccionario, a pesar de no seguir la trayectoria trata de seguirla de modo que se puede decir que cumple parcialmente con el control por seguimiento y cumple por completo con el control por regulación aun con perturbaciones, al hacer la comparación con los otros controladores el control PD de orden fraccionario ha sido el que ha tenido mejor desempeño para este sistema y no se descarta que se puede mejorar la sintonización, sin embargo, el siguiente paso es probar el ultimo controlador en el sistema de plato y bola y es el control PID fraccionario.

7.7. Control PID Fraccionario Implementado

Igual que con los otros controladores se utiliza la señal de control $u(kT)$:

$$u(kT) = -a_1u((k-1)T) - a_2u((k-2)T) + b_0e(kT) + b_1e((k-1)T) + b_2e((k-2)T) \quad (7.24)$$

Y al igual que con el controlador PD de orden fraccionario se utiliza la funcion de transferencia con la aproximacion de Oustaloup, aunque en el caso del controlador PID de orden fraccionario se tienen que definir las ganancias K_p , K_i y K_d obtenidas en las simulaciones, además de definir a $\mu = 0.9$ que fue el valor con el mejor rendimiento para el PD fraccionario, también se selecciona el valor de λ que por las simulaciones indicaron que el valor debe ser cercano a cero, de modo que se seleccionó como $\lambda=0.2$, con estos parámetros obtenemos la función de transferencia de la siguiente forma:

$$G_{PID\mu=0.9,\lambda=0.2} = \frac{-29.95z^4 + 113.2z^3 + 160.4z^2 - 100.9z + 23.76}{z^4 - 1.931z^3 - 0.06787z^2 + 1.931z - 0.9321} \quad (7.25)$$

Con esta ecuación asignamos los valores para a_0 , a_1 , a_2 , a_3 y a_4 al igual que b_0 , b_1 , b_2 , b_3 y b_4 , obtenemos la señal de control para el sistema del plato y bola, se ajustan los parámetros para obtener los valores tanto del error como del control en tiempo $k-2$, $k-3$ y $k-4$ y obtenemos la señal de control para cada eje.

A pesar de que se trató de sintonizar, los experimentos demostraron que para el caso del controlador PID fraccionario discretizado por la aproximación de Oustaloup y Tustin no lleva la bola al centro del plato en ningún momento y solo sube y baja los motores generando que la bola genere trayectorias sin que se acerque al centro del plato. Este comportamiento ya se tenía previsto por las simulaciones ya que la señal de control generaba que los motores se comportaran así, pero en el sistema real afecta la propia señal de control a la bola y genera más inercia rotacional y más perturbación, por esta razón se puede decir que el control PID de orden fraccionario no cumple el objetivo de control por regulación y mucho menos por seguimiento.

En este caso lo ideal sería tratar de utilizar otra aproximación como en el caso de los controladores PID tradicionales, sin embargo hacer el cambio con los controladores de orden fraccionario es más complicado, por esta razón se optó por descartar el controlador PID de orden fraccionario y solo tomar en cuenta los controladores que tuvieron mejor desempeño que fueron los controles PD y PD de orden fraccionario, para hacer un mejor análisis de estos se puede hacer una experimento adicional comparando estos controladores.

7.8. Comparación de controladores PD y PD fraccionario

Para tener un mejor análisis respecto al desempeño de los controladores implementados en el sistema de plato y bola se optó por comparar los dos mejores controladores que en base a los experimentos tuvieron una mejor respuesta en el sistema real, estos controladores son el control PD con la aproximación rectangular y el PD fraccionario con la aproximación de Oustaloup y Tustin.

Para mostrar ambas graficas en este caso se modificó la programación de MATLAB a modo de poder realizar el primer experimento con el control PD con la aproximación rectangular, una vez finalizado el experimento se guardarían los datos obtenidos de la misma forma que en los otros experimentos y para hacer la comparación con el controlador PD fraccionario fue necesario colocar la bola en el mismo punto inicial del primer experimento para repetir el experimento pero con el otro controlador y guardar los datos en otras variables similares a las del primer experimento para después mostrarlo en la gráfica, en la primera gráfica se muestra el movimiento de la bola a través del plato de los dos experimentos como se muestra en la Fig. [7.50](#)

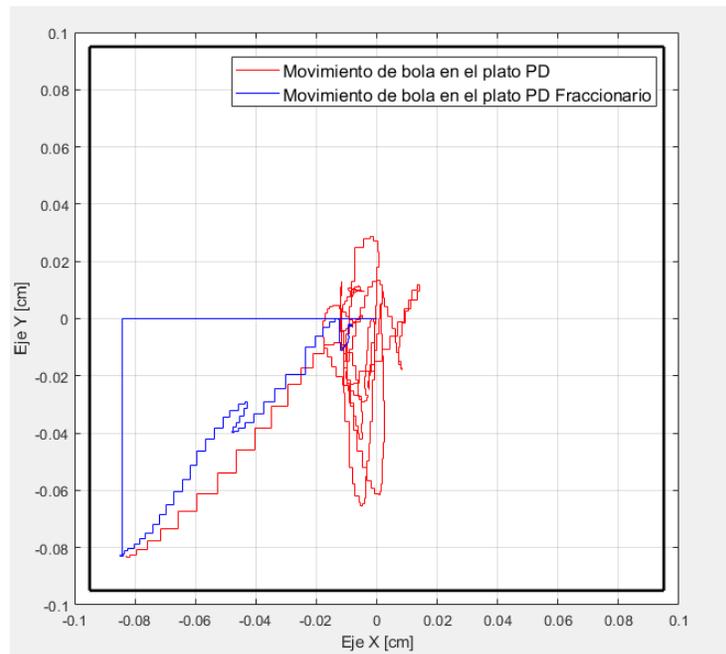


Figura 7.50: Movimiento de la bola en la comparación entre el controlador PD y PD fraccionario

En este experimento se estableció el valor de $\mu = 0.99$, que fue el valor que mostró el mejor desempeño en el controlador PD de orden fraccionario, el comportamiento como era de esperarse resulta ser muy similar a un controlador PD normal por el valor de μ cercano a 1, si se hace el análisis comparando las señales de control de ambos controladores se ve como se muestra en la Fig. 7.51.

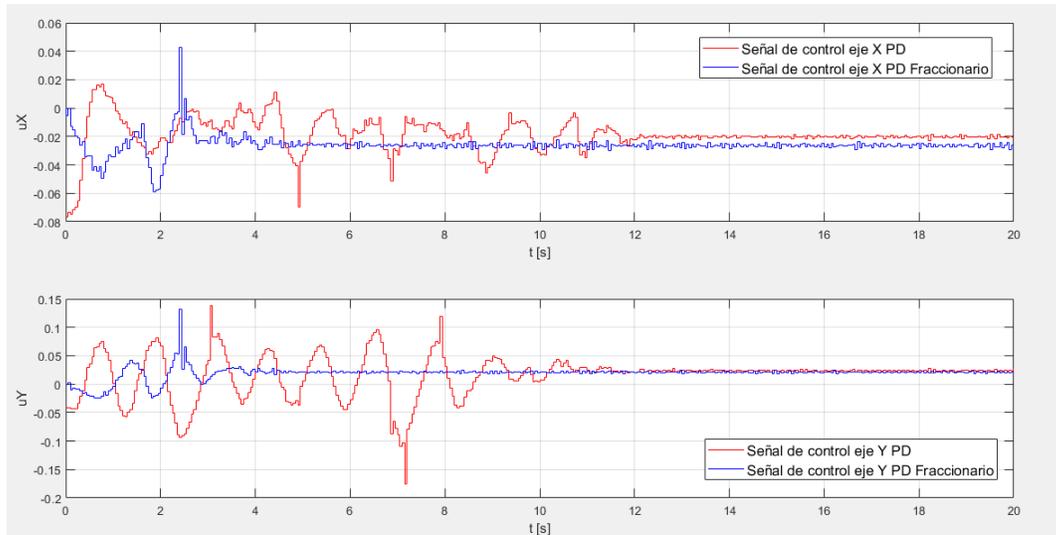


Figura 7.51: Señal de control de la comparación entre el controlador PD y PD fraccionario

Por ultimo comparamos la posición de la bola para cada eje colocando todos los datos en una sola gráfica y tener más claro cual controlador tiene mejor desempeño en llevar del punto inicial al valor deseado, que en este caso es el centro del plato, los resultados se muestran en la Fig. 7.52.

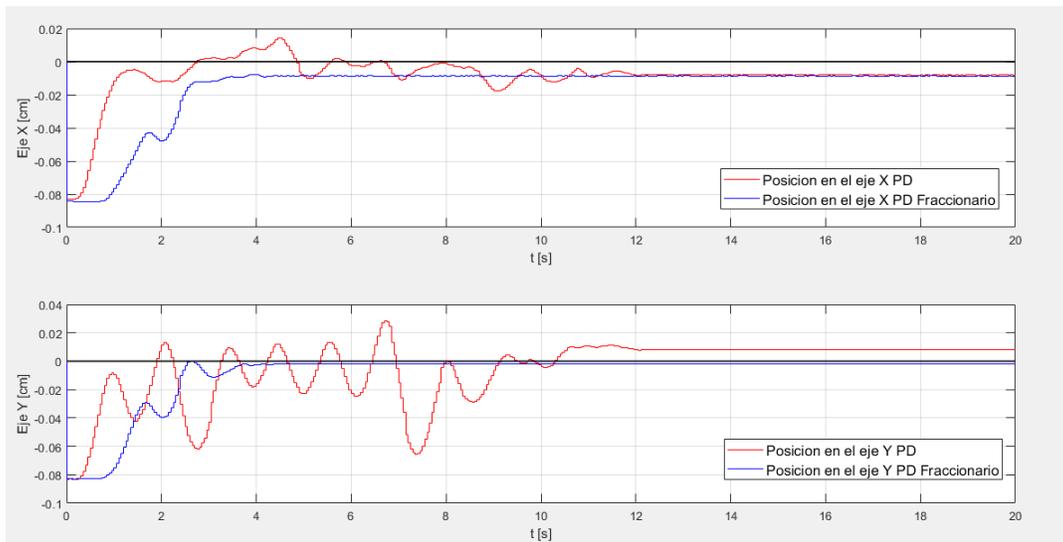


Figura 7.52: Posición de la bola en cada eje de la comparación entre el controlador PD y PD fraccionario

Como se observa, ambos controladores llevan la bola al centro del plato, pero en el caso del controlador PD de orden fraccionario la señal de control es un poco más suave y hace que se reduzca la inercia rotacional, esto hace que la bola se mueva más lento y oscile menos que con el control PD tradicional y aunque ambos cumplen con el objetivo de control y su comportamiento es similar por el valor de μ tan cercano a 1 el control PD de orden fraccionario tiene un mejor desempeño que el controlador PD tradicional. A pesar de esto, no se descarta que para ambos controladores tal vez se pueda mejorar su desempeño con una mejor sintonización, pero por los experimentos realizados se puede afirmar que, para este sistema de plato y bola, el mejor controlador de los que se probaron es el control PD de orden fraccionario, esta conclusión se analizará más a detalle en las conclusiones que se muestran en el siguiente capítulo.

Capítulo 8

Conclusiones y Bibliografía

8.1. Conclusiones

De forma general se puede concluir como primer punto, que si se pudo implementar todos los controladores en el sistema real de plato y bola, en particular que si se pudo implementar los controladores de orden fraccionario y sobre todo que a partir de los modelos matemáticos tanto del sistema como de cada controlador se pudo aplicar al sistema real y aunque algunos controladores resultaron mejor que otros, de forma general la mayoría pudo llevar la bola al centro del plato, cumpliendo de forma total o parcial el objetivo de control por regulación. En el caso de los controladores que no pudieron llevar la bola al centro del plato, que fueron los controladores PD y PID con la aproximación de Tustin, ya se esperaba que la señal de control moviera los motores muy rápido desde las simulaciones en los modelos matemáticos y al implementarlo en el sistema real se pudo confirmar físicamente, de modo que la teoría fue acorde a la realidad.

El factor que más afectó al momento de implementar los controladores al sistema real fue la inercia rotacional, dependiendo del controlador implementado afectaba en mayor o menor medida pero siempre estaba presente, este factor podía definir si el control era óptimo para el sistema y aun con el mejor controlador que se implementó en este sistema de plato y bola, que fue el PD de orden fraccionario, aun depende de que la inercia rotacional se reduzca lo menor posible para que la bola no oscile tanto y llevar la bola al centro del plato. Lo que sí es un hecho es que en base a los experimentos mostrados la inercia rotacional afectó en todos los controladores en el control por seguimiento y ninguno pudo seguir al cien por ciento la trayectoria para tratar de generar el movimiento un círculo con la bola en el plato. Aunque todos se trataron de aproximar a la trayectoria en cada eje, algunos mejor que otros, solo los controladores PD y PD fraccionario pudieron cumplir parcialmente con el objetivo de control por regulación, para el control de seguimiento, sí se debe considerar implementar otro controlador con una teoría diferente que permita seguir la trayectoria, pero reduciendo lo mejor posible la inercia rotacional.

Otro factor que definió el controlador más óptimo fue el modelo matemático del sistema en sí, como ya se mencionó el modelo matemático del sistema de plato y bola resulta ser un integrador doble, con dos polos en el origen y por esta razón tanto el controlador PD como el PD fraccionario, compensan en gran medida el doble integrador pero al probar los controladores PID y PID fraccionario, el lazo cerrado de control genera un triple integrador generando una oscilación mayor que se suma a la propia inercia rotacional y en muchos experimentos no se llegó al objetivo de control por regulación y mucho menos por seguimiento en estos controladores, no se descarta que con una mejor sintonización mejoren el desempeño de los controladores PID pero aun así, sería menor al desempeño de los controladores PD y PD fraccionario.

Adicional a la implementación de los controladores, se pudo desarrollar una metodología que pudiera utilizar los modelos matemáticos a través de las funciones de transferencia en tiempo discreto y aplicarlas al controlador real, todos los pasos a seguir en la programación permiten no solo implementar controladores de orden fraccionario si no que permite integrar cualquier controlador que tenga una función de transferencia discretizada, ya dependerá del sistema de plato y bola si responde mejor que los implementados en esta tesis pero la idea también era generar esta metodología en un sistema robusto, capaz de someterse a pruebas con un controlador con cualquier teoría de control.

El sistema de plato y bola se construyó para aplicar cualquier señal de control sin afectar la integridad del sistema real y con una base para colocar el sensor lo más lejos para no afectar el movimiento del plato en ningún momento tanto así, que se tuvo que modificar muchas veces hasta que se llegó a un sistema más robusto. A pesar de esto, a medida que se realizaron los experimentos se observó que para el caso del sensor pudiera no ser tan robusto como se esperaba. Utilizar la webcam como sensor fue la mejor opción desde el punto de vista de costos y de fácil integración al controlador y a pesar de que se tuvo de desarrollar toda la programación desde MATLAB para que pudiera obtener la posición en todo momento de la bola durante los experimentos, sigue siendo una buena opción como sensor, aunque tiene sus desventajas. El factor que más afecta a la webcam como sensor es la luz de donde se realicen los experimentos, se puede calibrar para obtener lo mejor posible la imagen de la bola, pero llegaba un punto donde no se podía y aunque esto se compenso colocando una lampara cerca para tener un valor constante de luz si es un factor a considerar.

Otro factor a considerar utilizando la webcam como sensor es el tiempo de muestreo y al reducir el tiempo de muestreo se debe considerar las características del controlador, en este caso una laptop con el software de MATLAB, si el tiempo de muestreo es muy grande la inercia rotacional le gana a la muestra tomada por la webcam y hace que oscile más, aunque para reducir el tiempo de muestreo afectan factores propios de la PC, como la tarjeta gráfica que ayuda al procesamiento de la imagen de la webcam, la memoria RAM y el disco duro, en este caso se pudo mejorar la PC donde se integran los controladores y los resultados de los experimentos fueron mejores al reducir el tiempo de muestreo. Esto no quiere decir que se requiere una super computadora para implementar los controladores, pero si se debe tomar en cuenta al momento de definir el tiempo de muestreo y sobre todo sintonizar el controlador si se utilizan controladores PID.

Como conclusión final, se puede afirmar que se llegó a los objetivos propuestos en la tesis, cada parte desde el diseño del sistema, la construcción del mismo, la elección del sensor y la adecuación para obtener la mejor lectura posible, además de todos los modelos matemáticos, tanto del sistema como de cada controlador, de los que se obtuvo bastante información tanto en sus modelos en variables de estado como su función de transferencia, la simulación con cada controlador, el hecho de que se tenga que cambiar los controladores y el sistema a tiempo discreto para simular y después implementar todos los controladores al sistema real y como dependiendo de esta aproximación cambiaba la respuesta del sistema de plato y bola creo que aportan, en cierta medida a la teoría del PID y el PID de orden fraccionario con la esperanza de utilizar algún día estas teorías de control para cualquier sistema y por qué no en la rama industrial ya que sin problema se puede programar esta metodología a un PLC siempre que tenga la teoría de Oustaloup y las bases del cálculo fraccionario.

8.2. Bibliografía

Referencias

- Aleksei, T., Eduard, P., y Juri, B. (2012). A flexible matlab tool for optimal fractional-order pid controller design subject to specifications. En *Proceedings of the 31st chinese control conference* (pp. 4698–4703).
- Åström, K. J., y Hägglund, T. (1995). *Pid controllers: theory, design, and tuning* (Vol. 2). Instrument society of America Research Triangle Park, NC.
- Debono, D., y Bugeja, M. (2015). Application of sliding mode control to the ball and plate problem. En *2015 12th international conference on informatics in control, automation and robotics (icinco)* (Vol. 1, pp. 412–419).
- Hendricks, E., Jannerup, O., y Sørensen, P. H. (2008). *Linear systems control: deterministic and stochastic methods*. Springer.
- Malti, R., y Victor, S. (2015). Crone toolbox for system identification using fractional differentiation models. *IFAC-PapersOnLine*, 48(28), 769–774.
- Monje, C. A., Chen, Y., Vinagre, B. M., Xue, D., y Feliu-Batlle, V. (2010). *Fractional-order systems and controls: fundamentals and applications*. Springer Science & Business Media.
- Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Pearson educación.
- Ogata, K., y Yang, Y. (2002). *Modern control engineering* (Vol. 4). London.
- Salvá, A., Sánchez, V., Ortega, S., y Altamirano, L. (s.f.). Construcción de un controlador digital pid para enseñanza empleando un dispositivo chipbas8.
- Shah, P., y Agashe, S. (2016). Review of fractional pid controller. *Mechatronics*, 38, 29–41.
- Slotine, J.-J. E., Li, W., y cols. (1991). *Applied nonlinear control* (Vol. 199) (n.º 1). Prentice hall Englewood Cliffs, NJ.
- Zamojski, M. (2018). Implementation of fractional order pid controller based on recursive oustaloup’e filter. En *2018 international interdisciplinary phd workshop (iiphdw)* (pp. 414–417).