



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**TEMA: DESARROLLO DE ESTRATEGIAS DE CONTROL EN BASE A
MATRIZ DINÁMICA EN UN ROBOT PARALELO TIPO STEWART
PARA APLICACIÓN DE BOLA EN PLATAFORMA A TRAVÉS DE
VISIÓN ARTIFICIAL.**

AUTOR: MONTENEGRO RODRÍGUEZ, EDISON JAVIER

DIRECTOR: ING. AYALA TACO, JAIME PAÚL, PhD.

SANGOLQUÍ

2019



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, "*Desarrollo de estrategias de control en base a matriz dinámica en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial*" fue realizado por el señor *Montenegro Rodríguez, Edison Javier* el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 18 de noviembre de 2019

Firma:

Ing. Jaime Paúl Ayala Taco, PhD.

C.C. 1709434243



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, *Montenegro Rodríguez, Edison Javier* declaro que el contenido, ideas y criterios del trabajo de titulación: “*Desarrollo de estrategias de control en base a matriz dinámica en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial*” es de mi autoría y responsabilidad cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 18 de noviembre de 2019

Firma:

Edison Javier Montenegro Rodríguez

C.C. 0502795941



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, Montenegro Rodríguez, Edison Javier autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “Desarrollo de estrategias de control en base a matriz dinámica en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial” en el Repositorio Institucional cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 18 de noviembre de 2019

Firma:



Edison Javier Montenegro Rodríguez

C.C. 0502795941

DEDICATORIA

A mi familia, a mis amigos y a todos los que me acompañaron y creyeron en mí durante toda esta etapa.

AGRADECIMIENTO

Agradezco a Dios, por darme vida y permitirme llegar a este punto.

Afradezco a toda mi familia, quien ha estado conmigo, levantándome y forjándome como la persona que ellos esperan que sea. Anhele que sientan orgullo de mí. A mis padres, Daniel y Ana; a mi hermano, Gustavo; a mis tías, María, Silvia y Rosita; a mis abuelitos, Marujita y papá Jaime, que siempre me han acompañado, cada uno a su manera.

A los maestros, por su rol de facilitadores de conocimiento. De manera especial, a mi director, el Ing. Paúl Ayala, por la paciencia y orientación que me brindó a lo largo del desarrollo de este trabajo.

A Zasky, con quien coincidí en esta Universidad y se convirtió en parte importante de mi vida, marcando esta etapa.

A mis amigos, quienes además de compañeros se convirtieron en personas muy cercanas, y que además de apoyarme académicamente no faltaron en levantarme el ánimo en momentos de convalecencia personal.

Y, a todos quienes en mayor o menor medida fueron aportando para convertirme en la persona que soy actualmente.

Edison Javier Montenegro Rodríguez

Índice general

Certificación	i
Autoría de Responsabilidad	ii
Autorización	iii
Dedicatoria	iv
Agradecimiento	v
Índice General	vi
Índice de Figuras	xii
Índice de Tablas	xiii
Resumen	xiv
Abstract	xv
INTRODUCCIÓN	1
Antecedentes	1
Justificación e Importancia	4
Definición del Proyecto	5

Alcance	6
Objetivos	7
1. FUNDAMENTACIÓN TEÓRICA	9
1.1. Generalidades de Robótica	9
1.1.1. Morfología de robots	9
1.1.2. Robots paralelos	10
1.1.3. Matemática para localización espacial	12
1.2. La plataforma Stewart	13
1.2.1. Aplicaciones	14
1.3. Control Predictivo basado en Modelos	15
1.4. Control por Matriz Dinámica	17
1.4.1. Modelo de predicción DMC	17
1.4.2. Optimización y Ley de Control	20
1.5. Generalidades sobre procesamiento de imágenes	23
1.5.1. Espacio del color	23
1.6. Visión Artificial	25
1.6.1. OpenCV	26
1.6.2. Detección de contornos y bordes	27
1.7. Conclusiones del Capítulo	27
2. PLANTEAMIENTO DE LA SOLUCIÓN	29
2.1. Descripción del Sistema	29
2.1.1. Cinemática Inversa	31
2.1.2. Calibración de los motores	35
2.2. Componentes adicionales al sistema	36
2.2.1. Adquisición de datos	37

2.2.2. Control	39
2.2.3. Ensamble del sistema	40
2.3. Identificación del sistema	41
2.3.1. Señal de entrada	42
2.3.2. Identificación en lazo cerrado	43
2.3.3. Identificación	44
2.4. Desarrollo de la estrategia de control	45
2.4.1. Inicialización del DMC	46
2.4.2. Algoritmo DMC en línea	48
2.5. Conclusiones del Capítulo	51
3. IMPLEMENTACIÓN Y RESULTADOS	52
3.1. Simulación	52
3.1.1. Análisis de resultados simulados	53
3.2. Implementación	55
3.2.1. Detección de la pelota	55
3.2.2. Pruebas de funcionamiento	56
3.2.3. Análisis de resultados	57
3.3. Tiempos de ejecución	61
3.4. Conclusiones del Capítulo	62
CONCLUSIONES Y RECOMENDACIONES	64
REFERENCIAS	73
ANEXOS	A.1
Anexo A. Diagrama electrónico	A.1

Anexo B. Códigos de los programas	B.3
B.1. Programa de reconocimiento de pelota	B.3
B.2. Programa de control	B.8
B.3. Programas Matlab	B.31
Anexo C. Pruebas de funcionamiento	C.35

Índice de figuras

Figura 1.	Articulaciones de robots industriales con sus grados de libertad (GDL).	10
Figura 2.	Configuraciones típicas de robots industriales.	11
Figura 3.	Los robots paralelos más comunes	11
Figura 4.	Coordenadas espaciales cartesianas y cilíndricas	12
Figura 5.	Plataforma Stewart.	14
Figura 6.	Primer simulador de vuelo construido por Sikorsky Aircraft Division of United Technologies.	15
Figura 7.	Concepto básico de los controladores predictivos basados en modelos.	16
Figura 8.	Esquema general de un MPC	17
Figura 9.	Esquema de respuesta al escalón de un sistema	18
Figura 10.	Esquema del perfil de predicción de un controlador DMC.	21
Figura 11.	El proceso de adquisición de una imagen digital.	23
Figura 12.	Comparación de los diversas descomposiciones de los espacios del color.	25
Figura 13.	Ejemplo de detección de bordes.	28
Figura 14.	Plataforma Stewart de la Universidad de las Fuerzas Armadas ESPE.	29
Figura 15.	Sistema de la plataforma con cámara.	30
Figura 16.	Esquema de plataforma Stewart.	31

Figura 17. Esquema de un servomotor con su brazo y eslabón (izq.) y vista superior de la posición del servomotor en relación a la plataforma (der.)	32
Figura 18. Diagrama de flujo del proceso de captura de posición de la pelota en la plataforma.	38
Figura 19. Esquema completo de interacción de los componentes del sistema.	41
Figura 20. LFSR de 4 bits para generación de SBPA.	42
Figura 21. Diagrama para toma de datos para identificación.	44
Figura 22. Comparación de datos del eje x.	46
Figura 23. Comparación de datos del eje y.	47
Figura 24. Diagrama de flujo de control DMC.	48
Figura 25. Respuesta al escalón de los sistemas a controlar.	49
Figura 26. Trayectoria suave de setpoint.	50
Figura 27. Esquema de simulación.	53
Figura 28. Simulación del DMC para varios valores de λ	54
Figura 29. Simulación de perturbaciones.	55
Figura 30. Prueba de funcionamiento con desempeño esperado.	58
Figura 31. Prueba de funcionamiento con mejor tiempo de establecimiento.	59
Figura 32. Prueba de funcionamiento con mayor error estacionario.	60
Figura 33. Prueba de funcionamiento con perturbación en eje x.	60
Figura 34. Prueba de funcionamiento con perturbación en eje y.	61
Figura 35. Diagrama de tiempo general de los componentes de adquisición de datos y de control.	62
Figura 36. Diagrama de tiempo del detalle de tareas del microcontrolador.	62
Figura 37. Prueba 1.	C.35
Figura 38. Prueba 2.	C.36

Figura 39. Prueba 3.	C.36
Figura 40. Prueba 4.	C.37
Figura 41. Prueba 5.	C.38
Figura 42. Prueba 6.	C.39
Figura 43. Prueba 7.	C.39
Figura 44. Prueba 8.	C.40
Figura 45. Prueba 9.	C.40
Figura 46. Prueba 10.	C.41
Figura 47. Prueba 11.	C.41

Índice de tablas

Tabla 1	<i>Ventajas y desventajas de controladores por matriz dinámica.</i>	18
Tabla 2	<i>Leyes de control para funciones de costo con distintas penalizaciones.</i>	22
Tabla 3	<i>Rangos de giro de los motores de la plataforma.</i>	36
Tabla 4	<i>Características principales de la tarjeta STM32F4Discovery.</i>	40
Tabla 5	<i>Coefficientes para generar señales binarias pseudo aleatorias.</i>	43
Tabla 6	<i>Parámetros del controlador PD</i>	44
Tabla 7	<i>Parámetros para rango de detección de pixeles HSV</i>	56
Tabla 8	<i>Parámetros de los controladores DMC implementados</i>	56
Tabla 9	<i>Pruebas funcionales del controlador práctico</i>	57

RESUMEN

Este trabajo presenta el diseño e implementación de una estrategia de control por matriz dinámica o DMC, la cual pertenece al grupo de controladores predictivos basados en modelos, para el control de posición de una pelota en una plataforma Stewart de seis grados de libertad disponible en los laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE. La detección de la posición de la pelota se la realiza mediante procesamiento digital de imagen y visión artificial, tareas que son desempeñadas por una microcomputadora Raspberry Pi 3B; y para el controlador como tal se eligió una tarjeta STM32F4Discovery, que desempeña también los cálculos cinemáticos para la plataforma. En este documento se detallan las cualidades de este tipo de controlador y las variaciones existentes para el algoritmo de control, así como las restricciones que exige el sistema para implementación física. Se explican, además, las características del sistema a controlar, informando acerca de cada uno de sus componentes, funciones y medios de interacción. Finalmente, se realizan pruebas de funcionamiento con pelotas de tenis de mesa de varios colores y con esferas metálicas, y se analizan los resultados obtenidos tanto en simulación como en implementación, además de los tiempos de ejecución de las tareas de control.

PALABRAS CLAVE:

- **ROBOT PARALELO.**
- **PLATAFORMA STEWART.**
- **CONTROL PREDICTIVO BASADO EN MODELOS.**
- **CONTROL POR MATRIZ DINÁMICA.**

ABSTRACT

This work introduces the design and implementation of a Dynamic Matrix Control strategy or DMC, which belongs to the group of Model-based Predictive Controllers, in order to control the position of a ball on a Stewart platform with six degrees of freedom available in the laboratories of Electronics of the Armed Forces University ESPE. The detection of the position of the ball is deployed by digital image processing and computer vision, tasks that are performed by a Raspberry Pi 3B microcomputer; and for the controller itself, it was chosen the STM32 F4Discovery board, which also performs the kinematic calculations for the platform. This document details the attributes of this type of controller and the existing variations for the control algorithm, as well as the restrictions required by the system for physical implementation. The main features of the system to be controlled are also explained, reporting about each of its components, functions and connections. Finally, performance tests are carried out with table tennis balls of various colors and with metallic spheres, the results obtained in both simulations and implementation are analyzed, in addition to the execution times of the control tasks.

KEYWORDS:

- **PARALLEL ROBOT.**
- **STEWART PLATFORM.**
- **MODEL PREDICTIVE CONTROL.**
- **DYNAMIC MATRIX CONTROL.**

INTRODUCCIÓN

Antecedentes

Se han realizado varios trabajos respecto al control para la aplicación específica de “bola en plataforma”. Se consideran como puntos clave la estructura del sistema, la estrategia de control utilizada y como parámetros de desempeño, el error estacionario y el tiempo de establecimiento en que la bola se equilibra en la plataforma. Todos los trabajos expuestos tienen fines netamente investigativos en cuanto a métodos de diseño, implementación, puesta en marcha y comparación de las diferentes estrategias de control.

En el trabajo titulado “Diseño, construcción e implementación de una plataforma tipo Stewart para control de una aplicación bola en plataforma a través de visión artificial” realizado por Bueno Sánchez Cristian Alejandro y Maldonado Soria Alejandro Emmanuel bajo la dirección del Ing. David César Loza Matovelle en 2017, se desarrolló el diseño y la construcción de una plataforma Stewart, mediante el uso de software CAD/CAE, procesos de manufactura, corte láser e impresión 3D (Bueno & Maldonado, 2017). Con dicho sistema, Vela (2019) presentó el trabajo titulado “Desarrollo de estrategias de control en base a lógica difusa y redes neuronales en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial” bajo la dirección del Ing. Oswaldo Alexander Ibarra Jácome, en donde se mencionan ciertos cambios a nivel estructural de la planta y la implementación exitosa de estrategias de control basados en lógica difusa obteniendo un tiempo de establecimiento de 6.21 segundos y un error de 5.81 % en estado estacionario.

Ambos trabajos fueron realizados en la Universidad de las Fuerzas Armadas ESPE.

La aplicación de bola en plataforma ha sido ampliamente estudiada y se han utilizado varias técnicas de control para lograr este cometido. Un ejemplo se muestra en el trabajo titulado “Construcción y diseño del sistema de control de posición de una esfera sobre una plataforma móvil” en el cual se diseña, construye y modela una plataforma móvil de dos grados de libertad y además se usa un controlador PID para mantener la bola en la misma. Como resultado en la práctica se obtienen errores de aproximadamente 2 cm y tiempos de estabilización de 5 segundos (De Otaola, 2015).

El artículo de 2015 “Comparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform”, en el cual se caracteriza matemáticamente la planta y se aplican en simulación como en implementación controladores PID, LQR, de modo deslizante y Fuzzy. Con PID se obtuvo un tiempo de establecimiento de 0.82 segundos y un error de 1 mm en un solo eje; hubo un error de 6 mm en seguimiento de una trayectoria. Con LQR, en un eje se obtuvo un tiempo de 0.85 segundos y un error similar al PID tanto en un eje como en seguimiento de trayectoria. Con Fuzzy se obtuvo el mismo desempeño que con LQR. Los resultados de esta comparación apuntan a que los mejores resultados se obtuvieron con el control de modo deslizante argumentando la capacidad de este tipo de controlador para el seguimiento dinámico de sistemas no lineales, pues se tiene un error estacionario de 0.5 mm y un tiempo de establecimiento de 0.52 segundos (Kassem et al., 2015).

En 2016, del artículo titulado “Adaptive Embedded Control for a Ball and Plate System” se desarrolla un controlador PID adaptativo para la aplicación de bola en plataforma utilizando visión artificial y una tarjeta STM32F4, obteniendo un error menor a 2 cm en 4 segundos como resultado ante varios períodos de muestreo (Xiao & Buttazzo, 2016). Un trabajo similar se realizó en 2015 en el artículo “Development of a Ball and Plate System”, no obstante, en lugar de usar una cámara de video, se usó a modo de plataforma un panel táctil para determinar la posición de la pelota y como tarjeta de control un Arduino MEGA. En este caso se obtuvo un tiempo de estabilización de

aproximadamente 3.8 segundos (Ham & Taufiq, 2015).

En el artículo titulado “Ball and Plate Model For Robotic System” de 2018, se utilizan estrategias de control Lineal Cuadrático (LQ) polinomial, un regulador Lineal cuadrático (LQR) en espacio de estados y Proporcional Derivativo (PD) para la aplicación de la bola en plataforma. La notorio de este trabajo es que se usa un Robot Colaborativo YuMi de ABB, el cual es un robot humanoide, y solo uno de sus brazos sirve para el cometido de equilibrar la plataforma. De la comparación, se obtiene que el PD tiene un desempeño muy bueno en cuanto a seguimiento a referencia, no obstante, produce un sobre impulso grande, y es opacado por el LQ de 2 grados de libertad que responde mejor ante perturbaciones y que además usa menos energía que los demás, sin embargo, es más lento pues se tarda en seguir trayectorias. El LQR por su parte, en cuanto energía y seguimiento a referencia, está en un término medio entre el LQ y el PD (Spacek et al., 2018).

El uso de otros tipos de estrategias de control, concretamente predictivos, se puede apreciar en el artículo “Architecture of Predictive Control for a Stewart Platform Manipulato” de 2010, en el que se compara el clásico PID con el Control Predictivo Generalizado (GPC). Como resultado en GPC presenta menos tiempo de estabilización, siendo de 0.0158 segundos en comparación de 0.97 segundos del PID; el error también es menor en GPC siendo de 0.0264 mm control el error con PID de 1.5041 mm; no se llega a implementación física, pues los datos expuestos son resultados de implementación virtual (Lara-Molina & Dumur, 2010).

El artículo “Intelligent Positioning Plate Predictive Control and Concept of Diagnosis System Desig” del año 2017, propone controladores predictivos basados en modelos, tanto de manera simple, como combinando con diferentes formas de optimización de la función de costo e incluso con modelos regresivos ARX en lo que llaman Intelligent Positioning Plate (IPP) (Oravec & Jadlovská, 2017).

Justificación e Importancia

Por la naturaleza de este proyecto de investigación y desarrollo, de estrategias de control aplicadas a robótica, se promueve la transferencia tecnológica dentro de la universidad. La plataforma Stewart ofrece una amplia gama de campos que pueden ser desarrollados, entre ellos la obtención de modelos matemáticos, la robótica, y, sobre todo, se puede probar y comparar varias estrategias de control.

Acogiendo lo propuesto por el creador de esta plataforma, la misma puede contribuir en el desarrollo de sistemas de transporte moderno de pasajeros, en aire, tierra o mar, incluso en el espacio como medio de estabilización de las cabinas de los tripulantes (Stewart, 1965). La plataforma ha sido útil en simuladores de aviación, siendo de los primordial en los diseños tempranos de aviones Boeing 707; también se ha visto presente en la NASA, en aplicaciones de acoplamiento espacial (Official Organ of the Royal Aero Club, 1962). La plataforma Stewart ha sido ya aplicada en dispositivos para acoplamiento de naves de rescate para submarinos y ensambladores de robots en años recientes, entonces, la búsqueda e implementación de métodos de control en la plataforma Stewart es instructivo y significativo para aplicaciones prácticas (Bo et al., 2011).

El presente proyecto pretende aplicar una estrategia de control DMC o de matriz dinámica. El uso de DMC se justifica dado que es un tipo de control predictivo que presenta ventajas tanto para sistemas SISO como MIMO por sobre las estrategias de control clásico como PID; este tipo de control predice la dinámica del proceso en una ventana de tiempo y, además, compensa no linealidades relativamente altas, lo que le otorga la cualidad de ser bastante robusto (Åström & Hägglund, 2009) y (Acedo, 2003).

Siendo el equilibrio el eje principal de la aplicación objeto del presente trabajo y constituyéndose como uno de los problemas más desafiantes en el campo de control, pues el reto de balancear ha sido motivo de constante investigación y desarrollo para aplicaciones de robótica, transporte y simulación (Awtar & Craig, 2002), y por la no linealidad inherente que presenta este sistema (Bo

et al., 2011) se justifica la realización de este proyecto con fines investigativos, pues, Vela (2019) desarrolló también dos técnicas de control moderno en la misma planta, por lo que se lograría una comparación de desempeño real y bajo condiciones similares entre las estrategias propuestas.

Este trabajo se justifica también porque el control predictivo basado en modelos (MPC) como el generalizado (GPC) o concretamente por matriz dinámica (DMC) no ha sido objeto recurrente de estudio a nivel de grado en la universidad, como sí lo han sido las estrategias de control clásico como PID o modernas como Fuzzy. Existen trabajos recientes en los cuales se utilizan controladores predictivos aplicados en el sector químico (Ramdani & Grouni, 2017), a reactores de agitación continua y tanques de mezclado (Campoverde & Guayasamín, 2018), a control de temperatura (Rodríguez, 2018), en plantas de endulzamiento de gas (Arteaga & Contreras, 2003); existen además trabajos que se enfocan en desarrollar herramientas para control DMC (Oravec & Jadlovska, 2017), o mejorar la dependencia de software en la implementación de controladores DMC explícitos (Gawkowski et al., 2008); por lo tanto, es menester introducir, conocer y sobre todo difundir este tipo de estrategias de control, pues, controladores predictivos están siendo cada vez más tomados en cuenta a nivel industrial (Moriones, 2016).

Definición del Proyecto

La plataforma Stewart es un diseño mecánico concebido tanto para el control de posición como de movimiento (Stewart, 1965). Aplicaciones varias han surgido a partir del desarrollo de esta plataforma, especialmente en los campos de aeronáutica (Official Organ of the Royal Aero Club, 1962), automotriz, militar y recientemente en el área industrial (Moriones, 2016). En el artículo en el que Stewart presenta este modelo, describe el mecanismo como: “un vehículo para representar un cuerpo en el espacio sujeto a todas las fuerzas las cuales intervendrían; una plataforma que se debe mantener estable ante movimientos aleatorios, por ejemplo, al flotar en el mar; un simulador de un helicóptero”; y finalmente deja a criterio de quien vaya a usar la misma el uso que se le vaya

a dar (Stewart, 1965).

En el presente proyecto se pretende desarrollar una estrategia de control en base a matriz dinámica (DMC) que forma parte de los algoritmos de control predictivo basado en modelos, para la plataforma Stewart de 6 grados de libertad en una aplicación llamada “bola en plataforma”. Este sistema de equilibrio no lineal, debe posicionar y equilibrar la bola en un punto definido en el menor tiempo posible.

El sistema existe ya en la Universidad de las Fuerzas Armadas ESPE, construido por Bueno & Maldonado (2017), y modificado y puesto en marcha con varias estrategias de control por Vela (2019). El sistema cuenta con una plataforma de acrílico negro, 6 servomotores en la base conectados a un driver que los gobierna mediante pulsos; y para la toma de datos de la posición del objeto en la plataforma se tiene una cámara. Con los datos tomados de la misma entrará en acción el algoritmo de control a implementar.

Alcance

En el presente trabajo, se van a estudiar las estrategias de control predictivo basado en modelos, concretamente el control por matriz dinámica o DMC por sus siglas en inglés (Dynamic Matrix Control). DMC es una estrategia de control que forma parte de las estrategias de control predictivo basado en modelos o MPC por sus siglas en inglés. El control por matriz dinámica DMC se basa modelo de respuesta a salto en escalón finito. Fue propuesto por Cutter y Ramaker en los años 70 y desarrollado posteriormente por la firma Dynamic Matrix Control Corporation. Tiene el criterio de minimizar una función de costo objetivo (Åström & Hägglund, 2009) y (Acedo, 2003).

La planta objetivo del presente proyecto será la plataforma Stewart construida por Bueno & Maldonado (2017) quienes obtuvieron los modelos cinemáticos de la misma. La planta fue modificada posteriormente por Vela (2019), quien implementó estrategias de control Fuzzy y con Redes Neuronales y, además, propuso mejoras al algoritmo de visión artificial de esta planta para determi-

nar la posición de la bola mediante ajuste de valores en el modelo de color HSV o Matiz, Saturación y Brillo por sus siglas en inglés.

Debido a que la naturaleza de las estrategias de control Fuzzy propuestas por Vela (2019) no requerían un modelo matemático de la planta, previo al diseño del controlador, se realizará la identificación del modelo de la planta, y una vez se tenga una aproximación válida, se realizará el diseño de un controlador basado en estrategia DMC (Controlador de Matriz Dinámica) para equilibrar la bola en la plataforma en una posición deseada dentro de la misma en el menor tiempo posible. En cuanto a parámetros de desempeño, se tendrán como referencia los obtenidos por Vela (2019), debido a que el sistema de estudio será el mismo. Estos parámetros son, un tiempo de establecimiento de aproximadamente 7 segundos y un error menor al 6 %.

Cabe recalcar que el presente trabajo se basa en el diseño de una estrategia de control para la aplicación de bola en plataforma, y ya que se desea comparar con otras estrategias propuestas de control usadas anteriormente se intentará no modificar la estructura actual de la misma. Los resultados del trabajo se verán reflejados en simulaciones computacionales para evaluar el rendimiento del controlador diseñado y, posteriormente, se implementará el controlador en la plataforma existente mediante una tarjeta micro controlada o microcomputadora, la cual será elegida tras evaluar el cumplimiento de los parámetros de procesamiento requeridos del algoritmo de control. Posteriormente, se verificará la validez del mismo y se hará una comparación, tanto con los resultados simulados, como con los obtenidos en trabajos pasados.

Objetivos

Objetivo General

Desarrollar una estrategia de control en base a matriz dinámica en un robot paralelo tipo Stewart para aplicación de bola en plataforma a través de visión artificial.

Objetivos Específicos

- Investigar acerca de las estrategias de control predictivo basado en modelos, concretamente el control por matriz dinámica para su aplicación a la plataforma Stewart.
- Utilizar algoritmos de visión artificial para determinar la posición de la bola en la plataforma Stewart.
- Implementar una estrategia de control basado en matriz dinámica para equilibrar la bola en una posición deseada en la plataforma.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

1.1. Generalidades de Robótica

Un robot, según el Robot Institute of America (1978) es un manipulador reprogramable, multifuncional, diseñado para mover material, partes, herramientas o dispositivos especializados a través de movimientos variables programados para la realización de una variedad de tareas, el cual también puede adquirir información del entorno y moverse inteligentemente en respuesta. La ISO 8373:2012 define a un robot como un manipulador controlado automáticamente, reprogramable, mutipropósito, programable en tres o más ejes, los cuales pueden estar fijos o ser móviles para uso en aplicaciones de automatización industrial (International Organization for Standardization, 2012).

Los robots pueden dividirse fundamentalmente en manipuladores fijos y en robots móviles. No obstante, existen otros tipos de robots, entre los que destacan los humanoides, robots suaves y los robots paralelos (Williams, 2007).

1.1.1. Morfología de robots

Los robots o manipuladores industriales están definidos como cadenas cinemáticas abiertas en donde conviven dos elementos: eslabones y articulaciones. Las articulaciones permiten el movi-

miento relativo de cada eslabón sucesivo (Ollero Baturone, 2001).

Existen varios tipos de articulaciones de acuerdo al movimiento que pueden producir y al número de grados de libertad. La figura 1 muestra esta clasificación.

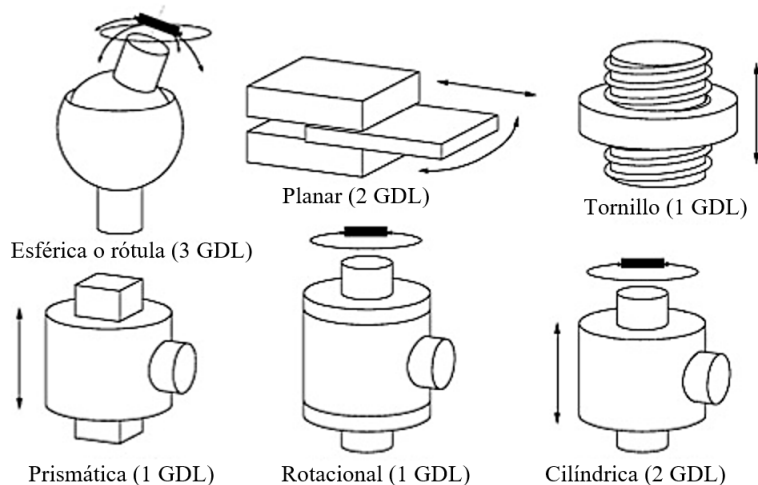


Figura 1. Articulaciones de robots industriales con sus grados de libertad (GDL).

Fuente: (Ollero Baturone, 2001)

La figura 2 muestra las principales configuraciones de manipuladores industriales existentes gracias a la combinación de varios tipos de articulaciones. Todas estas configuraciones siguen un patrón de cadena cinemática abierto, sin embargo, existen los robots con configuraciones cinemáticas cerradas, estos son los llamados robots paralelos (De Luca, 2016).

1.1.2. Robots paralelos

Son los robots formados por varias cadenas cinemáticas independientes cerradas. Al contrario de los robots con cadenas seriales, estos puede tener juntas, las cuales no contienen un actuador encargado de su movimiento, sino que la misma se mueve por efecto del movimiento de todo el sistema. Esto puede ocasionar que el análisis cinemático y dinámico de este tipo de robots sea más complejo que el de los manipuladores seriales (Angeles, 2003).

La figura 3 muestra ejemplos de robots paralelos. Destacan los robots Delta, los cuales pueden

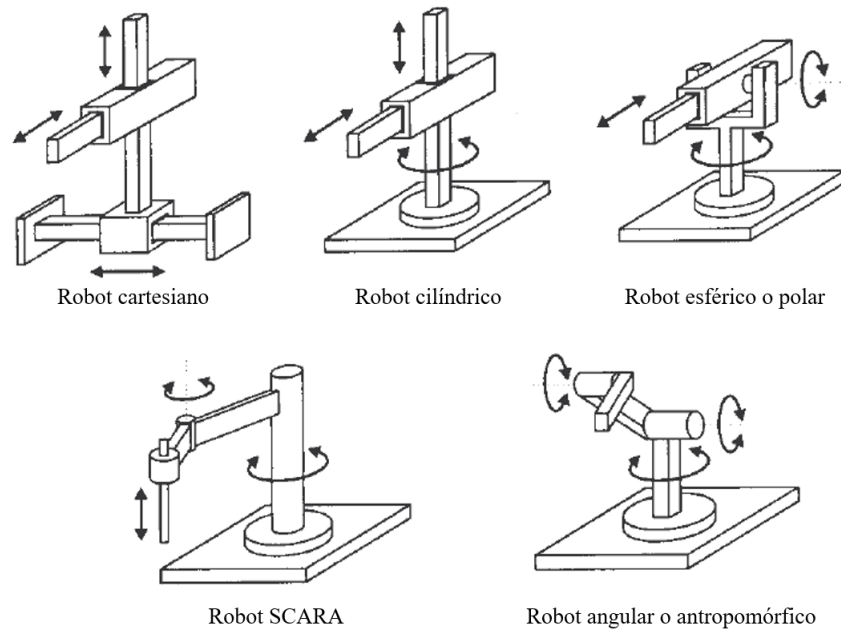


Figura 2. Configuraciones típicas de robots industriales.

Fuente: (Barrientos et al., 1997)

tener tres o cuatro brazos moviéndose y soportando carga de forma simultánea, y las plataformas Stewart, concebidas para aplicaciones de simuladores de vuelo (De Luca, 2016).



Figura 3. Los robots paralelos más comunes

Fuente: (Williams, 2007)

1.1.3. Matemática para localización espacial

En el ámbito de la robótica, es necesario tener una herramienta para definir cuánto mover cada actuador para lograr una posición u orientación deseada. La posición puede estar definida por cualquier tipo de coordenadas espaciales, usualmente cartesianas $\mathbf{P}(x, y, z)$ o cilíndricas $\mathbf{P}(r, \theta, z)$, como se muestra en la figura 4.

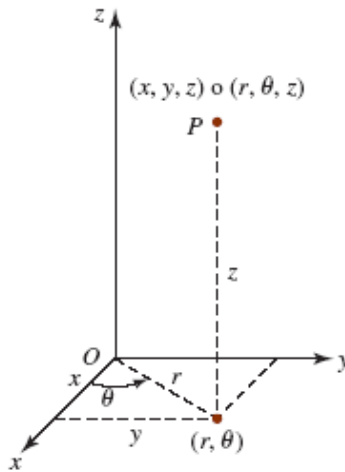


Figura 4. Coordenadas espaciales cartesianas y cilíndricas

La orientación en cambio, puede ser obtenida mediante la representación por matrices de rotación. Se tienen las matrices de rotación en x , y y z de la siguiente forma:

$$\mathbf{R}(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad \mathbf{R}(y, \phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix}, \quad \mathbf{R}(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.1.1)$$

Siendo α , ϕ y θ las rotaciones en los ejes x , y y z , respectivamente. Una rotación continua en los tres ejes puede ser representada por el producto de las tres matrices anteriores (abreviando las funciones \sin y \cos con las letras S y C para facilidad de escritura) se tiene, si se realizan las

rotaciones en x , luego en y y finalmente en z , que:

$$\mathbf{R}_T = \mathbf{R}(x, \alpha)\mathbf{R}(y, \phi)\mathbf{R}(z, \theta) = \begin{bmatrix} C\theta C\phi & -S\theta C\alpha + C\theta S\phi S\alpha & S\theta S\alpha + C\theta S\phi C\alpha \\ S\theta C\phi & C\theta C\alpha + S\theta S\phi S\alpha & -C\theta S\alpha + S\theta S\phi C\alpha \\ S\phi & C\phi S\alpha & C\phi C\alpha \end{bmatrix} \quad (1.1.2)$$

Es importante notar el orden de los productos matriciales, dado que no es conmutativo, la ecuación 1.1.2 sirve exclusivamente para el orden descrito de rotaciones (Barrientos et al., 1997).

Las *matrices de transformación homogénea* unen la posición y la rotación para poder representar la posición y orientación nueva de un sistema respecto a otro de referencia. Está compuesta por la matriz de orientación \mathbf{R}_{3x3} , el vector de posición \mathbf{P}_{3x1} , y dos vectores \mathbf{f}_{1x3} y \mathbf{w}_{1x1} , que representan la perspectiva y el escalado (Ollero Baturone, 2001). Para aplicaciones de robótica estos valores son típicamente 0 y 1 respectivamente (Barrientos et al., 1997). La matriz se expresa entonces como:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{P}_{3x1} \\ 0 & 1 \end{bmatrix} \quad (1.1.3)$$

Esta matriz ayuda a expresar matemáticamente la posición y orientación de un robot, y la aplicabilidad en el presente trabajo es en el cálculo de la cinemática inversa, que consiste, básicamente en encontrar cuánto se debe mover cada actuador del robot para alcanzar determinada posición y orientación (Angeles, 2003).

1.2. La plataforma Stewart

La plataforma Stewart es un robot paralelo de 6 grados de libertad capaz de moverse en tres direcciones lineales y tres direcciones angulares, de forma simple o combinada (Stewart, 1965).

La figura 5 muestra un esquema de construcción típico de este sistema. Está constituido por dos

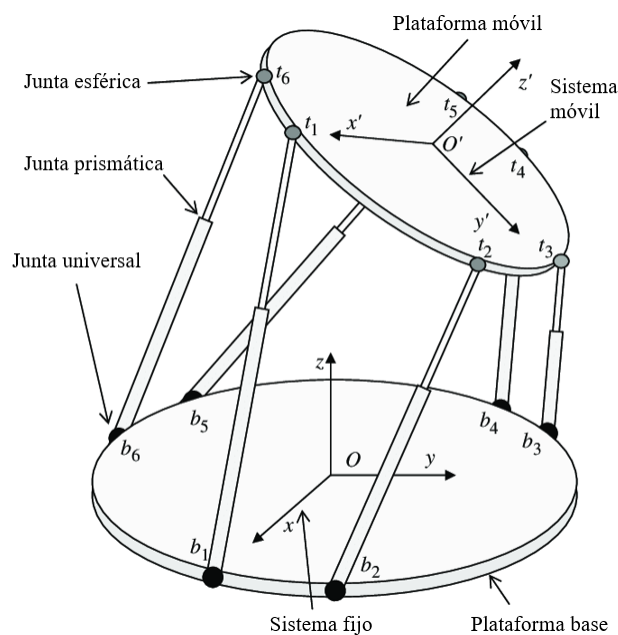


Figura 5. Plataforma Stewart.

Fuente: (He et al., 2013)

plataformas, una base y una móvil. La plataforma base cumple el rol de sistema de referencia fijo para la móvil. La plataforma tiene seis actuadores con articulaciones prismáticas, sin embargo, los mismos se conectan con la plataforma base mediante juntas universales, y con la plataforma móvil con juntas esféricas.

En la figura 5 se utilizan cilindros a modo de actuadores, no obstante, se puede alcanzar un funcionamiento similar utilizando actuadores rotacionales como servomotores, aunque, los cálculos cinemáticos se dificultan un poco (Eisele, 2019).

1.2.1. Aplicaciones

La plataforma Stewart nació con la idea de servir a simuladores de vuelo, y ha trascendido a aplicaciones espaciales, como de acoplamiento de naves, hoy en día implementado por la NASA (Official Organ of the Royal Aero Club, 1962). Stewart (1965) menciona su aplicabilidad a simulación de sistemas de transporte, o a un medio para estabilización de las cabinas de los mismos.

Históricamente, el primer simulador de vuelo construido por Sikorsky Aircraft Division of United Technologies en la década de los años 60 es un robot de este tipo como se muestra en la figura 6, y trasciende hasta ser de igual utilidad en simuladores actuales, como el del Boeing 737 o del Sukhoi Superjet 100 (Official Organ of the Royal Aero Club, 1962).

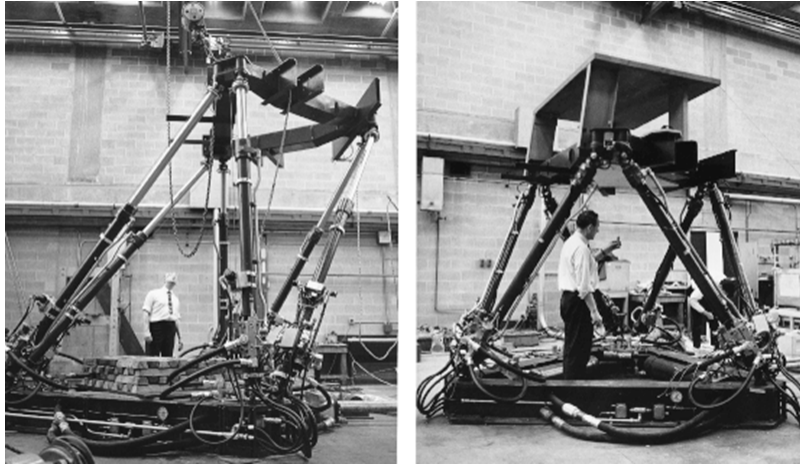


Figura 6. Primer simulador de vuelo construido por Sikorsky Aircraft Division of United Technologies.

Fuente: (Official Organ of the Royal Aero Club, 1962)

En la actualidad, la plataforma tiene muchísimas otras aplicaciones, en biomédica, en terapias de rehabilitación de extremidades inferiores (Boian et al., 2005) o incluso en la industria del entrenamiento, con simuladores de movimiento para videojuegos de carreras como el propuesto por Full Motion Dynamics (Kroymann & Dee, 2013), o en radiotelescopios como el AMiBA ubicado en Hawaii (Koch et al., 2009) o sistemas de paneles para seguimiento solar (Basim et al., 2018).

1.3. Control Predictivo basado en Modelos

El control predictivo basado en modelos o MPC (Model Predictive Control) fue inicialmente concebido para aplicaciones en procesos químicos con varias entradas, salidas y muchas restricciones (Ruchika, 2013). Este tipo de controladores utilizan un modelo explícito para predecir la salida

del proceso en instantes de tiempo futuros y calculan la secuencia de control con base en la optimización de una función de costo (Camacho & Bordons, 2007). La figura 7 muestra la representación gráfica de los MPC.

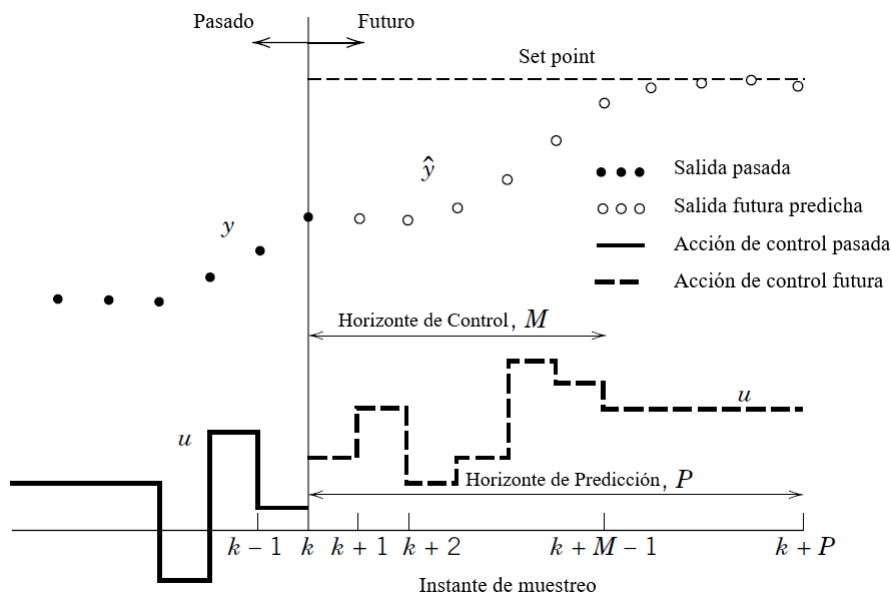


Figura 7. Concepto básico de los controladores predictivos basados en modelos.

Fuente: (Seborg et al., 2011)

Los controladores predictivos basados en modelos constituyen en realidad una gran cantidad de algoritmos. Existe un primer grupo de MPC que utilizan modelos de convolución, el DMC (Dynamic Matrix Control) que emplea la respuesta al escalón, y el MAC (Model Algorithmic Control) o más conocido como IDCOM, que emplea la respuesta al impulso; además se tiene el PFC (Predictive Functional Control) el cual utiliza funciones base para estimar las acciones de control futuras (Camacho & Bordons, 2007). Posteriormente aparecerían variaciones más complejas, que otorgan a los controladores predictivos más robustez y eficiencia en cuanto a tareas de optimización, identificación y predicción (Ruchika, 2013). Bajo todo lo establecido, el diagrama de bloques de un controlador MPC es el que se muestra en la figura 8.

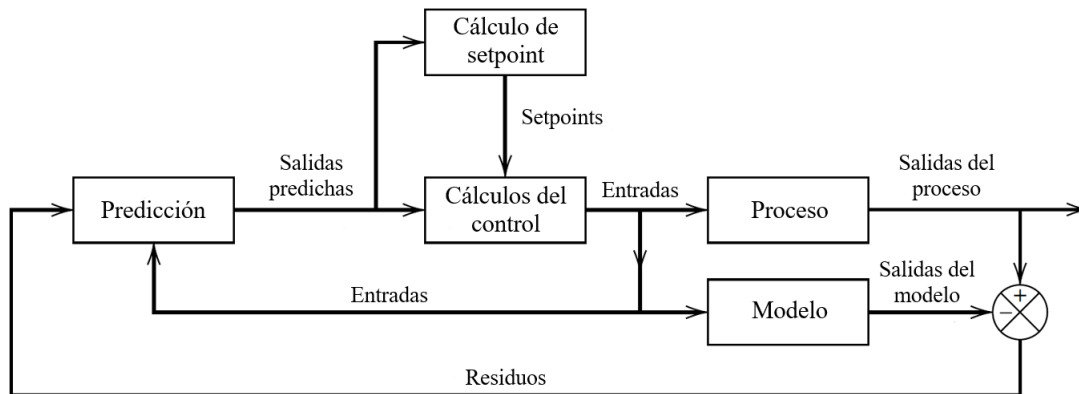


Figura 8. Esquema general de un MPC

Fuente: (Seborg et al., 2011)

1.4. Control por Matriz Dinámica

El control DMC, pertenece al grupo de controladores predictivos basados en modelos y representa la primera generación de tecnologías MPC (Ruchika, 2013). Propuesto en 1967 por Charlie Cutler, durante su estancia en Shell Oil Co.; DMC se volvería una compañía (The Dynamic Matrix Control Corp.) fundada por el mismo Cutler y por Chuck Johnston en 1984 (Cutler, 2003).

Como derivación del MPC, toma el mismo esquema de la figura 7. DMC es un controlador lineal que usa el modelo de respuesta al escalón de la planta a controlar, predice la dinámica del proceso en una ventana de tiempo, y estima los movimientos de la acción de control con base en la minimización de una función de costo. La tabla 1, detalla las ventajas y contras de los DMC

1.4.1. Modelo de predicción DMC

DMC utiliza el modelo de respuesta al escalón en un sistema discreto. Gráficamente, se puede representar esto como en la figura 9.

Como modelo de convolución, esta respuesta al escalón puede escribirse de la siguiente forma (Noerpamoengkas, 2009):

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i) \quad (1.4.1)$$

Tabla 1

Ventajas y desventajas de controladores por matriz dinámica.

Ventajas	Desventajas
Puede compensar altas no linealidades del proceso, lo que hace que sea muy robusto.	No adecuado para procesos inestables ante la entrada escalón.
Fácilmente adaptable a sistemas multivariable	Pese a ser uno de los controladores predictivos más simples, su implementación representa un gran costo de procesamiento.
La diferencia entre la salida real del sistema y la salida predicha puede ser modelada como una perturbación.	Rendimiento pobre en plantas con perturbaciones tipo rampa en la salida.

Fuente: (Lopez-Guede et al., 2013) y (Lundström et al., 1995)

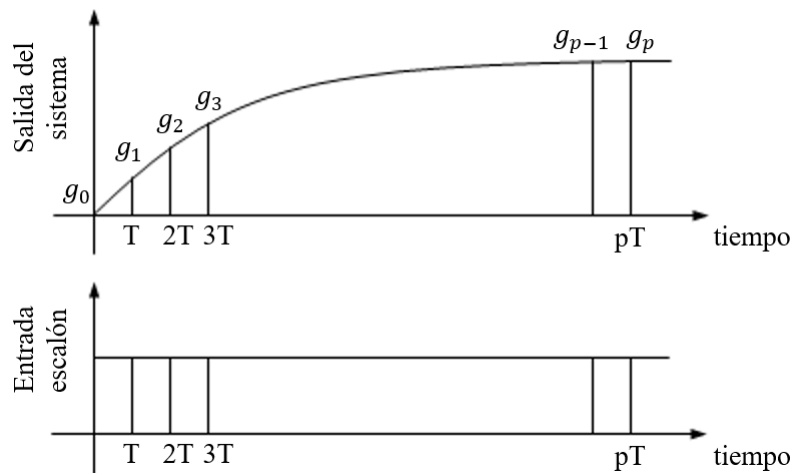


Figura 9. Esquema de respuesta al escalón de un sistema

Un modelo de predicción para $y(t+k|t)$ y añadiendo una perturbación $n(t+k|t)$ puede ser escrito como:

$$\hat{y}(t+k|t) = y_0 + \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) + \hat{n}(t+k|t) \quad (1.4.2)$$

La suma infinita puede ser separada en una parte que vaya hasta k que representa el valor de predicción subsecuente, y la parte que va hasta infinito. Si se considera lo anterior y que la perturbación es constante, es decir $\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t)$, entonces la predicción se

expresa como:

$$\hat{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i) \quad (1.4.3)$$

Se define entonces la respuesta libre del sistema $f(t+k)$ como:

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t-i) \quad (1.4.4)$$

Y la predicción, reemplazando 1.4.4 en 1.4.3 estaría dada por:

$$\hat{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + f(t+k) \quad (1.4.5)$$

Dado que DMC exige procesos estables ante la entrada escalón, el valor de $g_{k+i} - g_i$ se volverá 0 cuando el sistema alcance el estado estacionario (Camacho & Bordons, 2007). Si la planta alcanza este estado en N muestras, entonces la respuesta libre queda expresada como:

$$f(t+k) = y_m(t) + \sum_{i=1}^N (g_{k+i} - g_i) \Delta u(t-i) \quad (1.4.6)$$

Las predicciones se calculan para cada muestra futura a lo largo de lo que se llama el horizonte de predicción, un valor entero k que va desde 1 hasta p , considerando m acciones de control

(Castaño, 2015). Desglosando la ecuación 1.4.5, para cada valor que toma k , se tiene:

$$\hat{y}(t + 1|t) = g_1 \Delta u(t) + f(t + 1) \quad (1.4.7)$$

$$\hat{y}(t + 2|t) = g_2 \Delta u(t) + g_1 \Delta u(t + 1) + f(t + 2)$$

$$\hat{y}(t + 3|t) = g_3 \Delta u(t) + g_2 \Delta u(t + 1) + g_1 \Delta u(t + 2) + f(t + 3)$$

$$\vdots$$

$$\hat{y}(t + k|t) = \sum_{i=p-m+1}^p g_i \Delta u(t + p - i) + f(t + p)$$

Si se define la *matriz dinámica* \mathbf{G} del sistema como (Noerpamoengkas, 2009)

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix} \quad (1.4.8)$$

Se puede escribir matricialmente la expresión para la predicción como (Camacho & Bordons, 2007):

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f} \quad (1.4.9)$$

1.4.2. Optimización y Ley de Control

La principal característica de los controladores predictivos basados en modelos es la de optimizar una función de costo. En el caso de DMC, la función de costo involucrarán los movimientos de la acción de control y el error.

La idea, en primera instancia, será minimizar el error y penalizar la acción de control. El error

para cada instante en el horizonte de predicción es:

$$e_1 = y_{sp1} - \hat{y}(t + 1|t) \quad (1.4.10)$$

$$e_2 = y_{sp2} - \hat{y}(t + 2|t)$$

$$\vdots$$

$$e_p = y_{sp} - \hat{y}(t + p|t)$$

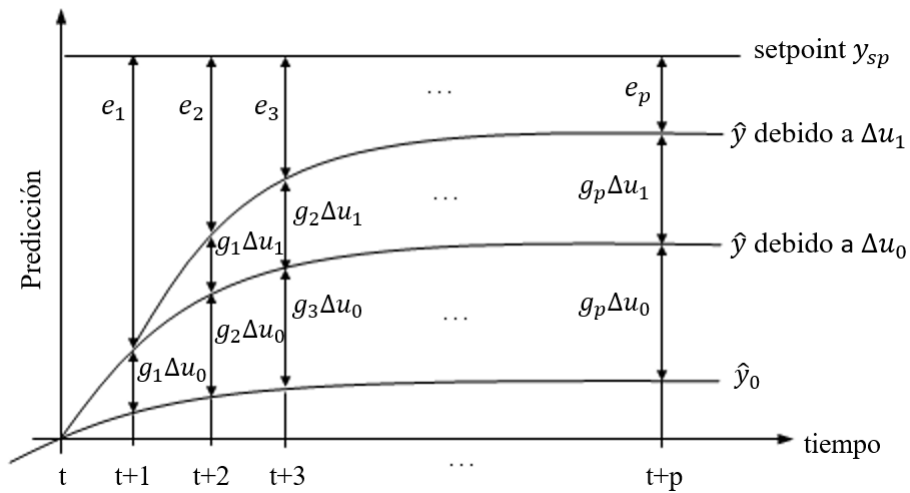


Figura 10. Esquema del perfil de predicción de un controlador DMC.

Fuente: (Noerpamoengkas, 2009)

Este error para cada instante de predicción es apreciable en la figura 10. Se tiene que minimizar la función de costo correspondiente a los mínimos cuadrados de los errores, es decir que la función de costo es:

$$J = e_1^2 + e_2^2 + \dots + e_p^2 = \sum_{i=1}^p e_i^2 \quad (1.4.11)$$

El objetivo sería entonces encontrar el mínimo de la función anterior. No obstante, ya que se

tiene la función así, se suele añadir la penalización en el esfuerzo de la acción de control como:

$$J = \sum_{i=1}^p e_i^2 + \sum_{j=1}^m \lambda [\Delta u(t+j-1)]^2 \quad (1.4.12)$$

Si se considera que no hay restricciones, se puede escribir matricialmente esta función de costo:

$$J = \mathbf{e}\mathbf{e}^T + \lambda \mathbf{u}\mathbf{u}^T \quad (1.4.13)$$

Optimizando (derivando e igualando a cero) la ecuación 1.4.13, se obtiene:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{y}_{sp} - \mathbf{f}) \quad (1.4.14)$$

Si bien es cierto, ya se tiene el vector \mathbf{u} de movimientos de la acción de control, no obstante, físicamente no se puede enviar todas estas variaciones, entonces se envía a la planta solo el primer elemento de este vector. Como se puede notar, en la función de costo en realidad se puede penalizar lo que se desee, por lo tanto, la tabla 2 muestra distintas leyes de control para diferentes funciones de costo.

Tabla 2

Leyes de control para funciones de costo con distintas penalizaciones.

Descripción	Función de costo	Ley de control
Peso en los movimientos de la acción de control	$J = \mathbf{e}^T \mathbf{e} + \mathbf{u}^T \lambda \mathbf{u}$	$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{y}_{sp} - \mathbf{f})$
Peso en los errores	$J = \mathbf{e}^T \gamma \mathbf{e} + \mathbf{u}^T \lambda \mathbf{u}$	$\mathbf{u} = (\mathbf{G}^T \gamma \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \gamma (\mathbf{y}_{sp} - \mathbf{f})$
Peso en ambas (acción de control y errores)	$J = \mathbf{e}^T \gamma \mathbf{e} + \mathbf{u}^T \lambda \mathbf{u}$	$\mathbf{u} = (\mathbf{G}^T \gamma \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \gamma (\mathbf{y}_{sp} - \mathbf{f})$

Fuente: (Noerpamoengkas, 2009)

La ley de control más utilizada es la expresada en la ecuación 1.4.14, no obstante, para ahorrar tiempo de procesamiento computacional se hace la optimización fuera de línea y se obtiene solo un vector resultado \mathbf{K} , formado por la primera fila de la matriz $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T$ que multiplicará a

la diferencia de los vectores set point y respuesta libre (Camacho & Bordons, 2007).

1.5. Generalidades sobre procesamiento de imágenes

Con la sofisticación de los métodos de captura de imágenes del mundo real, nacen varias maneras de utilizar estas imágenes captadas en varias aplicaciones. Todo nace del principio de que una imagen puede ser representada por una función $f(x, y)$, y si para cada valor finito de x e y existe valores finitos de f , se dice que es una imagen digital (Vernon, 1991), usualmente siendo los valores de f la intensidad de los grises en la unidad básica de la imagen llamada *pixel*. La figura 11 muestra el proceso de adquisición y digitalización de una imagen.

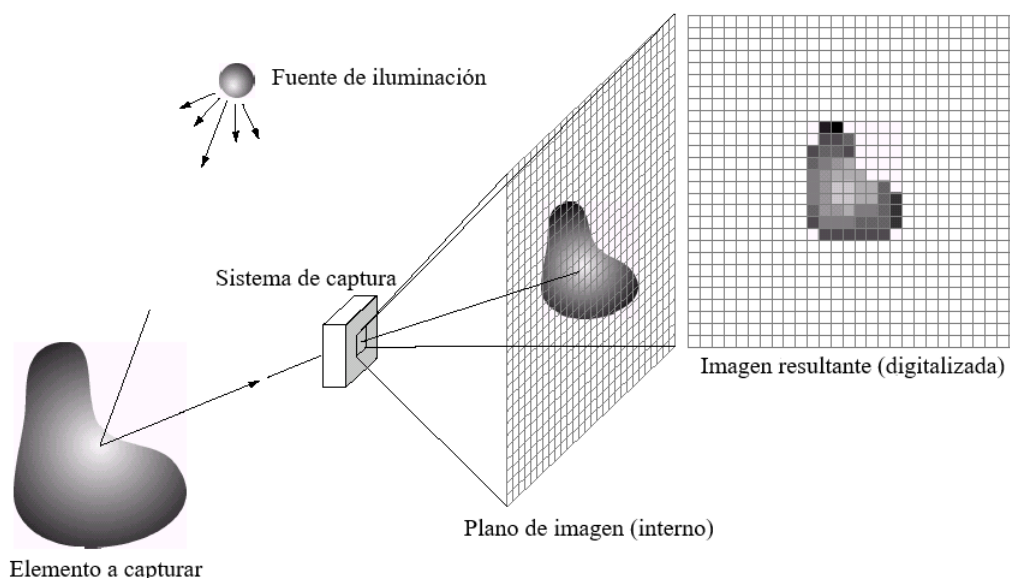


Figura 11. El proceso de adquisición de una imagen digital.

Fuente: (Gonzalez & Woods, 2008)

1.5.1. Espacio del color

Para imágenes a color, cada pixel es en realidad un vector $c_{n \times 1}$ que define la intensidad de varios componentes que definen el color. Existen varios sistemas en los que se puede representar un pixel,

los más importantes son:

RGB (Red, Green, Blue) Define un color como la combinación de los colores rojo, verde y azul en diferente intensidad.

HSV (Hue, Saturation, Value) Define un color como la combinación en diferente magnitud de matiz, saturación y valor o intensidad.

YIQ Representa un color como valores de luminancia, matiz y pureza.

CMY (Cyan, Magenta, Yellow) Representa un color como combinación de valores de los colores cian, magenta y amarillo.

CMYK (Cyan, Magenta, Yellow, Black) Representa un color como combinación de valores de los colores cian, magenta, amarillo y un valor de intensidad de grises.

Para todos los casos descritos, un pixel cuenta con varios valores y por tanto cada valor de este vector es una función de la posición (Gonzalez & Woods, 2008). Si, por ejemplo, se expresa en términos de RGB, un pixel estaría definido como:

$$\mathbf{c}(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \quad (1.5.1)$$

La figura 12 muestra la descomposición de una imagen en los sistemas de espacio de color descritos anteriormente.

Es posible transformar las expresiones del color de un espacio a otro, por ejemplo, si se tiene un pixel en RGB normalizado $[0, 1]$, y se definen $M = \max(R, G, B)$; $m = \min(R, G, B)$; y $C = M - m$; se pueden calcular los valores HSV mediante las siguientes ecuaciones (Smith,

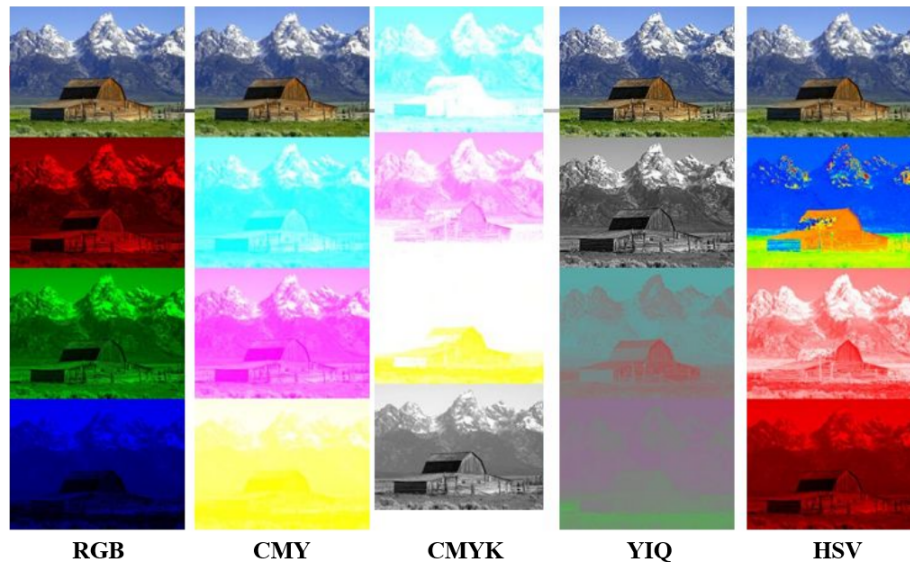


Figura 12. Comparación de los diversas descomposiciones de los espacios del color.

Fuente: (O'Neal, 2016)

1978):

$$H = \begin{cases} 0, & \text{si } C = 0 \\ \frac{G - B}{C} \bmod 6, & \text{si } M = R \\ \frac{B - R}{C} + 2, & \text{si } M = G \\ \frac{R - G}{C} + 4, & \text{si } M = B \end{cases} \quad S = \begin{cases} 0, & \text{si } M = 0 \\ 1 - \frac{m}{M}, & \text{en otros casos} \end{cases} \quad V = M \quad (1.5.2)$$

1.6. Visión Artificial

La visión artificial, comúnmente abreviada CV (Computer Vision), es un campo de estudio que desarrolla técnicas para que sistemas de cómputo puedan entender e interpretar imágenes digitales como fotografías o videos (Brownlee, 2019).

La visión artificial está especialmente relacionada con la inteligencia artificial y el Machine Learning (Brownlee, 2019), sobre todo porque el procesamiento de imágenes que conlleva y la interpretación de las mismas son tareas que el ser humano realiza casi de manera intuitiva y desde

muy tempranas edades, pero constituye un reto especialmente amplio el trasladar esta clase de procesos mentales a máquinas computacionales, y los mejores resultados se han obtenido con esta clase de algoritmos (Jähne & Haußecker, 2000).

Las aplicaciones de CV son muy variadas, casi trascendiendo a cualquier campo, desde reconocimiento óptico de caracteres u OCR (Breuel, 2008), reconocimiento de huellas digitales (Jain et al., 2010), inspección mecánica (Gadelmawla, 2011), técnicas de imagen para diagnóstico médico (Greenspan et al., 2016), captura de movimiento (Moeslund et al., 2006), seguridad, vigilancia e incluso monitoreo de tráfico (Coifman et al., 1998).

1.6.1. OpenCV

OpenCV es una librería de código abierto enfocada en suministrar algoritmos optimizados para Machine Learning y Visión Artificial, que tiene soporte para C++, Python, Java y MATLAB en los sistemas operativos más comunes como Windows, Linux, Mac OS y Android (The OpenCV Team, 2019).

Esta librería fue inicialmente desarrollada por Intel, pero actualmente mantenida por Willow Garage; opera bajo licencia BSD, lo que quiere decir que es libre tanto para usos académicos como comerciales (Solem, 2012). Las operaciones de procesamiento de imagen más básicas que incluye OpenCV son (Kapur, 2017):

- Operaciones morfológicas.
- Detección de contornos y bordes.
- Filtros.
- Comparación de plantillas.

1.6.2. Detección de contornos y bordes

La detección de bordes y contornos es una de las tareas más recurrentes y necesarias en visión artificial. Cuantitativamente, los bordes ocurren en los límites entre regiones de diferente color, intensidad o textura; entonces, matemáticamente este cambio puede ser expresado mediante su gradiente que detectaría un cambio rápido en la intensidad de los píxeles (Szeliski, 2010).

Aunque, es posible entrenar un sistema para optimizar el reconocimiento de bordes de cierto tipo de imágenes, los algoritmos más sencillos, que usa solo información local, tienen tres etapas principales (Shah, 1997):

- Etapa de filtrado.
- Etapa de diferenciación.
- Etapa de detección.

En la etapa de filtrado se utiliza un filtro Gaussian Blur que es un filtro pasa bajos gaussiano que reduce el ruido de la imagen y como efecto suaviza la misma (Krishna, 2017). En la etapa de diferenciación se aplican las derivadas parciales correspondientes al gradiente de la imagen, para finalmente, en la etapa de detección, habiendo definido ciertos umbrales, mostrar los píxeles seleccionados correspondientes a los resultantes del gradiente, que serían los contornos de los objetos (Shah, 1997).

La figura 13 muestra un ejemplo de la aplicación de las tres etapas descritas anteriormente para detección de bordes, se utiliza un filtro gaussiano con $\sigma = 1$ (Forsyth & Ponce, 2003).

1.7. Conclusiones del Capítulo

En el presente capítulo se han estudiado aspectos puntuales de los temas pertinentes al proyecto en desarrollo. Se ha analizado la naturaleza de la plataforma Stewart desde el punto de vista de la robótica. Se ha desglosado y explicado de manera detallada el algoritmo de control por matriz



Figura 13. Ejemplo de detección de bordes.

Fuente: (Forsyth & Ponce, 2003)

dinámica, y además se ha añadido temática puntual acerca de procesamiento digital de imágenes y visión artificial. Todos los temas tratados constituyen una parte para la solución que se planterá posteriormente, siendo la visión artificial el medio para la obtención de datos de la bola en la plataforma, y siendo DMC la estrategia de control a utilizar.

Capítulo 2

PLANTEAMIENTO DE LA SOLUCIÓN

2.1. Descripción del Sistema

El sistema que se encuentra en los laboratorios de Universidad de las Fuerzas Armadas ESPE es una plataforma Stewart de seis grados de libertad. La figura 14 muestra las partes principales de la plataforma como tal.

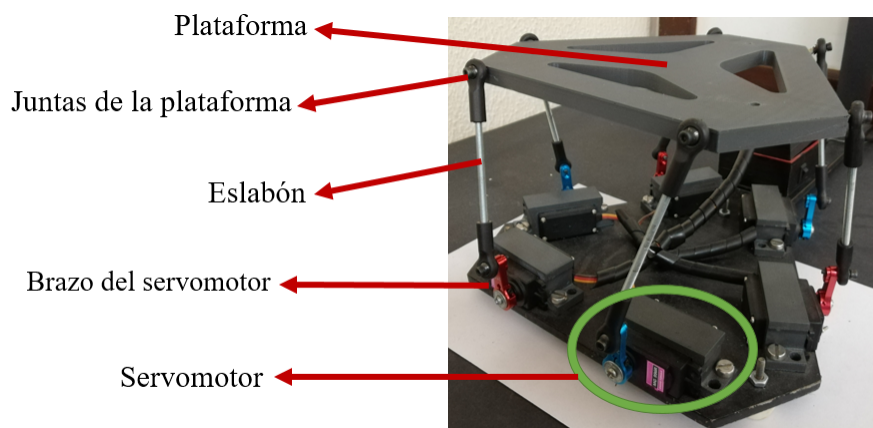


Figura 14. Plataforma Stewart de la Universidad de las Fuerzas Armadas ESPE.

El sistema completo, como se muestra en la figura 15 cuenta con los siguientes componentes:

Cámara: Modelo Logitech C170 USB; 720p a 30 fps de enfoque fijo.

Plataforma: De acrílico negro de 6mm de espesor; 35cm x 35cm.

Rieles: De aluminio para sostener la cámara por encima de la plataforma.

Actuadores: Servomotores Tower Pro MG996R de 9.4 kgf.cm a 4.8 V.

Brazos: De aluminio de 24 mm, conectan el eje de giro de los servomotores con los eslabones.

Eslabones: Extremos de plástico y cuerpo de aluminio con juntas esféricas.

Driver: USB Servo Controller Pololu Micro Maestro de 6 canales. 5 a 16 V. Control por USB o Serial. Resolución de pulso de $0.25 \mu s$ y rango de 816 a $2176 \mu s$.



Figura 15. Sistema de la plataforma con cámara.

Fuente: (Bueno & Maldonado, 2017)

Es importante remarcar que Vela (2019) fue el que modificó el sistema a como se presenta en esta sección, no obstante, para la implementación de las estrategias de control, utilizó una computadora personal, cosa que se pretende modificar en el presente trabajo.

2.1.1. Cinemática Inversa

La cinemática inversa determinará cuánto se debe mover cada actuador de la plataforma para alcanzar conjuntamente una posición y orientación deseada (Barrientos et al., 1997).

El procedimiento detallado a continuación se basa en el artículo de Eisele (2019) y el proyecto de grado de Bueno & Maldonado (2017). Tomando en cuenta el esquema de la figura 16, se tiene la plataforma base cuyo centro es O_b y la plataforma móvil con centro O_p . Además, P representa el punto de la junta del actuador con la plataforma móvil, y B representa el punto de la junta del actuador con la plataforma fija.

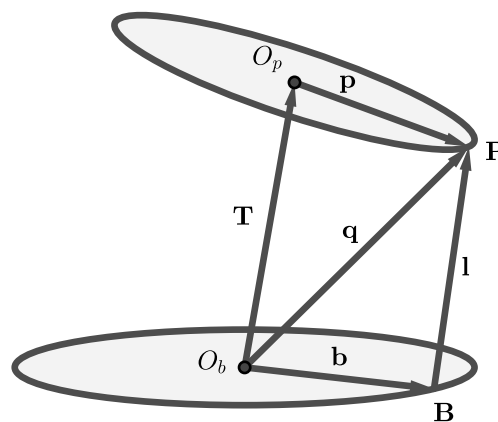


Figura 16. Esquema de plataforma Stewart.

Fuente: (Eisele, 2019)

En primer lugar, se va a obviar la construcción real de la plataforma y se va a asumir que en lugar de los servomotores, existen unas piernas que pueden extenderse dando como resultado una longitud l . Para ello, se puede decir que \mathbf{q} , el vector que une el origen de la base con la junta en la plataforma, es igual al vector de traslación \mathbf{T} sumado la rotación de la plataforma móvil y desplazado la distancia del origen de la misma a la junta. Si esta distancia es \mathbf{p} y la matriz de rotación está dada por \mathbf{R} , entonces:

$$\mathbf{q} = \mathbf{T} + \mathbf{R}\mathbf{p} \quad (2.1.1)$$

Vectorialmente, de la figura 16 se puede decir que:

$$\mathbf{l} = \mathbf{P} - \mathbf{B} = \mathbf{q} - \mathbf{b} \quad (2.1.2)$$

Reemplazando 2.1.1 en 2.1.2 se tiene entonces:

$$\mathbf{l} = \mathbf{T} + \mathbf{R}\mathbf{p} - \mathbf{b} \quad (2.1.3)$$

Con la ecuación 2.1.3 se obtiene el vector que representa la longitud de la pierna de la plataforma en ese punto en relación a una traslación y rotación deseada. Para obtener la longitud, simplemente se debe sacar el módulo de \mathbf{l} , es decir, $|\mathbf{l}| = \sqrt{l_x^2 + l_y^2 + l_z^2}$.

Si se tuvieran actuadores de movimiento lineal como cilindros, repitiendo el proceso anterior para los seis actuadores bastaría y se lograrían las posiciones y rotaciones deseadas, pero el sistema real está construido con servomotores, entonces, se realiza el análisis de lo que sucede en cada servomotor con base en los esquemas de la figura 17.

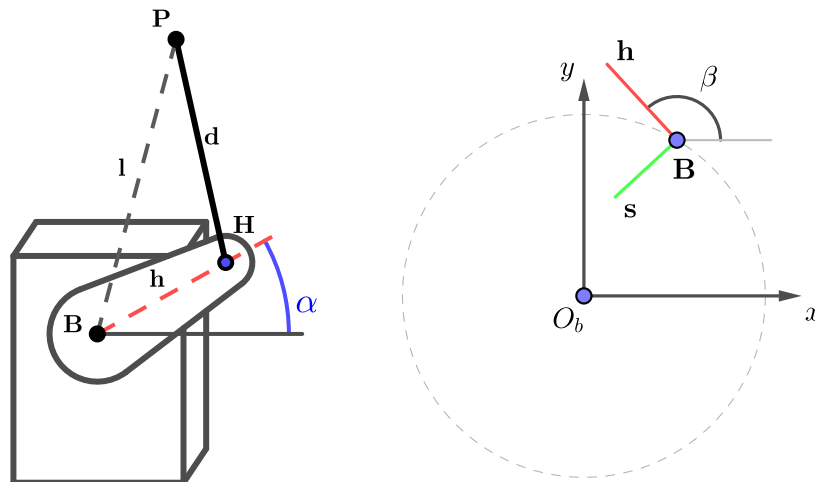


Figura 17. Esquema de un servomotor con su brazo y eslabón (izq.) y vista superior de la posición del servomotor en relación a la plataforma (der.)

Fuente: (Eisele, 2019)

Según el esquema izquierdo de la figura 17, ahora l es función de las longitudes del brazo del servomotor h y del eslabón que lo junta con la plataforma d . El servomotor puede rotar un ángulo α en el eje y local. Si se pone atención al esquema derecho de la figura 17, se puede notar que, como se tiene juntas esféricas, puede verse reflejado en un giro β en el eje z a lo largo del plano x - y que es ortogonal al eje de movimiento s del motor.

Con lo anterior, el brazo del servomotor estaría expresado desde O_b en primer lugar con una traslación B que sería la posición del motor en el plano x - y , y ya ubicado ahí, localmente, se da una traslación $|\mathbf{h}|$ en el eje x local, una rotación en y $\mathbf{R}_y(\alpha)$, seguido de una rotación en z $\mathbf{R}_z(\beta)$:

$$\begin{aligned} \mathbf{H} &= \mathbf{B} + \mathbf{R}_z(\beta)\mathbf{R}_y(\alpha) \begin{pmatrix} |\mathbf{h}| \\ 0 \\ 0 \end{pmatrix} \\ &= \mathbf{B} + |\mathbf{h}| \begin{pmatrix} \cos \alpha \cos \beta \\ \cos \alpha \sin \beta \\ \sin \alpha \end{pmatrix} \end{aligned} \quad (2.1.4)$$

Por otro lado, por teorema de Pitágoras, se puede llegar a:

$$|\mathbf{h}|^2 = (H_x - B_x)^2 + (H_y - B_y)^2 + (H_z - B_z)^2 \quad (2.1.5)$$

$$|\mathbf{d}|^2 = (P_x - H_x)^2 + (P_y - H_y)^2 + (P_z - H_z)^2 \quad (2.1.6)$$

$$|\mathbf{l}|^2 = (P_x - B_x)^2 + (P_y - B_y)^2 + (P_z - B_z)^2 \quad (2.1.7)$$

Para facilitar las operaciones, podemos expresar cuadrados anteriores como una multiplicación

de la matriz por su transpuesta, dando como equivalencia:

$$|\mathbf{h}|^2 = (\mathbf{H} - \mathbf{B})^T (\mathbf{H} - \mathbf{B}) \quad (2.1.8)$$

$$|\mathbf{d}|^2 = (\mathbf{P} - \mathbf{H})^T (\mathbf{P} - \mathbf{H}) \quad (2.1.9)$$

$$|\mathbf{l}|^2 = (\mathbf{P} - \mathbf{B})^T (\mathbf{P} - \mathbf{B}) \quad (2.1.10)$$

Si restamos las ecuaciones anteriores, se obtiene:

$$\begin{aligned} |\mathbf{l}|^2 - (|\mathbf{d}|^2 - |\mathbf{h}|^2) &= 2\mathbf{B}^T \mathbf{B} - 2\mathbf{B}^T \mathbf{H} - 2\mathbf{B}^T \mathbf{P} + 2\mathbf{H}^T \mathbf{P} \\ &= 2(\mathbf{H} - \mathbf{B})^T (\mathbf{P} - \mathbf{B}) \end{aligned} \quad (2.1.11)$$

Si reemplazamos 2.1.4 en 2.1.11 y sabiendo de la figura 17 que $\mathbf{l} = \mathbf{P} - \mathbf{B}$ se obtiene:

$$|\mathbf{l}|^2 - (|\mathbf{d}|^2 - |\mathbf{h}|^2) = 2|\mathbf{h}| \begin{pmatrix} \cos \alpha \cos \beta \\ \cos \alpha \sin \beta \\ \sin \alpha \end{pmatrix}^T \mathbf{l} \quad (2.1.12)$$

Expresando \mathbf{l} en sus componentes x , y y z , resulta en:

$$|\mathbf{l}|^2 - (|\mathbf{d}|^2 - |\mathbf{h}|^2) = 2|\mathbf{h}|l_z \sin \alpha + 2|\mathbf{h}| \cos \alpha (l_x \cos \beta + l_y \sin \beta) \quad (2.1.13)$$

Se definen las variables auxiliares:

$$e = 2l_z |\mathbf{h}| \quad (2.1.14)$$

$$f = 2|\mathbf{h}|(l_x \cos \beta + l_y \sin \beta) \quad (2.1.15)$$

$$g = |\mathbf{l}|^2 - (|\mathbf{d}|^2 - |\mathbf{h}|^2) \quad (2.1.16)$$

Se utiliza la identidad trigonométrica $e \sin \varphi + f \cos \varphi = \sqrt{e^2 + f^2} \sin(\varphi + \text{atan2}(f, e))$, resultando en:

$$\begin{aligned} g &= e \sin \alpha + f \cos \alpha \\ &= \sqrt{e^2 + f^2} \sin(\alpha + \text{atan2}(f, e)) \end{aligned} \quad (2.1.17)$$

La función atan2 o arcotangente de dos parámetros da como resultado el ángulo desde el eje positivo x hasta el vector de sus dos parámetros, y sirve para dar el valor real en transformaciones de coordenadas cartesianas a polares (Glisson, 2011). Con todo esto, se puede despejar α en 2.1.17:

$$\alpha = \sin^{-1} \left(\frac{g}{\sqrt{e^2 + f^2}} \right) - \text{atan2}(f, e) \quad (2.1.18)$$

Así, si se desea que la plataforma Stewart alcance determinada posición y orientación, se deberá utilizar la ecuación 2.1.18 para cada uno de los servomotores obteniendo el ángulo α que deberán rotar individualmente para que la plataforma logre su cometido.

2.1.2. Calibración de los motores

Aunque la plataforma dispone seis servomotores del mismo modelo, se ha notado que cada uno tiene peculiaridades en cuanto a los rangos mínimos y máximos de giro, por lo tanto no se puede hallar una sola expresión que generalice el comportamiento de los seis. Entonces, se han tomado datos individualmente, y dependiendo del ancho de pulso, se han caracterizado estos rangos en la tabla 3.

El driver de los servomotores, permite rangos en μs de ancho de pulsos de la señal de control, siendo de 816 μs para el valor inferior, y de 2176 μs para el valor superior con un período de 20 ms.

Conociendo los valores límite superiores e inferiores, y luego de haber comprobado que la rela-

Tabla 3

Rangos de giro de los motores de la plataforma.

Motor	$\alpha_{\text{mín}} [^\circ]$	$\alpha_{\text{máx}} [^\circ]$
0	-18.73	102.17
1	-23.21	93.74
2	-45.01	107.22
3	-40.22	94.47
4	-16.59	97.12
5	-35.17	90.23

ción de giro de los motores es proporcional (Ibrahim, 2019), se pueden obtener una aproximación del comportamiento de los motores mediante regresiones lineales de primer grado.

De esta manera, se obtienen las ecuaciones que describen ancho de pulso pw vs. ángulo de giro α para cada uno de los motores del sistema.

$$pw_0 = 1965,31 - 11,25\alpha_0$$

$$pw_1 = 1085,91 + 11,63\alpha_1$$

$$pw_2 = 1773,95 - 8,93\alpha_2$$

$$pw_3 = 1222,11 + 10,1\alpha_3$$

$$pw_4 = 1977,58 - 11,96\alpha_4$$

$$pw_5 = 1197,43 + 10,58\alpha_5 \tag{2.1.19}$$

2.2. Componentes adicionales al sistema

Hasta el momento, se ha caracterizado matemáticamente el sistema tal como se encuentra construido, no obstante, no se tienen aún elementos que sean los encargados de procesar los datos y posteriormente controlar el sistema. La presente sección detalla los componentes adicionales a implementar para la adquisición de los datos necesarios, que sería la posición de la bola en la

plataforma, y para el control de la misma.

2.2.1. Adquisición de datos

El sistema tiene una cámara web implementada, se utilizará la misma para la obtención de la posición de la bola en la plataforma. Dada la carga de procesamiento requerida, la disponibilidad de puerto USB compatibles con cámaras web, y la compatibilidad que presenta este sistema con librerías de procesamiento digital de imagen y visión artificial, se opta por utilizar una microcomputadora Raspberry Pi 3B.

El diagrama de flujo que se utilizará para la detección de la pelota se detalla en la figura 18. Este algoritmo fue trasladado a lenguaje Python, usando métodos de la librería OpenCV. A continuación se detalla cada una de las etapas del mismo.

Captura del frame: Se almacena en un buffer la información de un frame captado por la cámara.

Para ello se utiliza el método *read* de OpenCV. El espacio de color que utiliza para capturar el frame es BGR.

Filtro Gaussian Blur: Como ya fue descrito en el primer capítulo, es un filtro pasabajos con distribución gaussiana. Ayuda a suavizar la imagen y eliminar frecuencias altas que pueden afectar procesos de diferenciación.

Transformación a HSV: Transforma la imagen al espacio de color HSV. Se utiliza el método *cvtColor*

Detección de píxeles en rango de colores: Asigna magnitudes a los parámetros de matiz, saturación y valor que servirán como umbrales de detección de píxeles. Se utiliza para esto el método *inRange*.

Máscara AND: Resultado del proceso anterior, se tiene una máscara con los píxeles donde coinciden los umbrales del color; si se hace una operación AND de esta máscara con la imagen

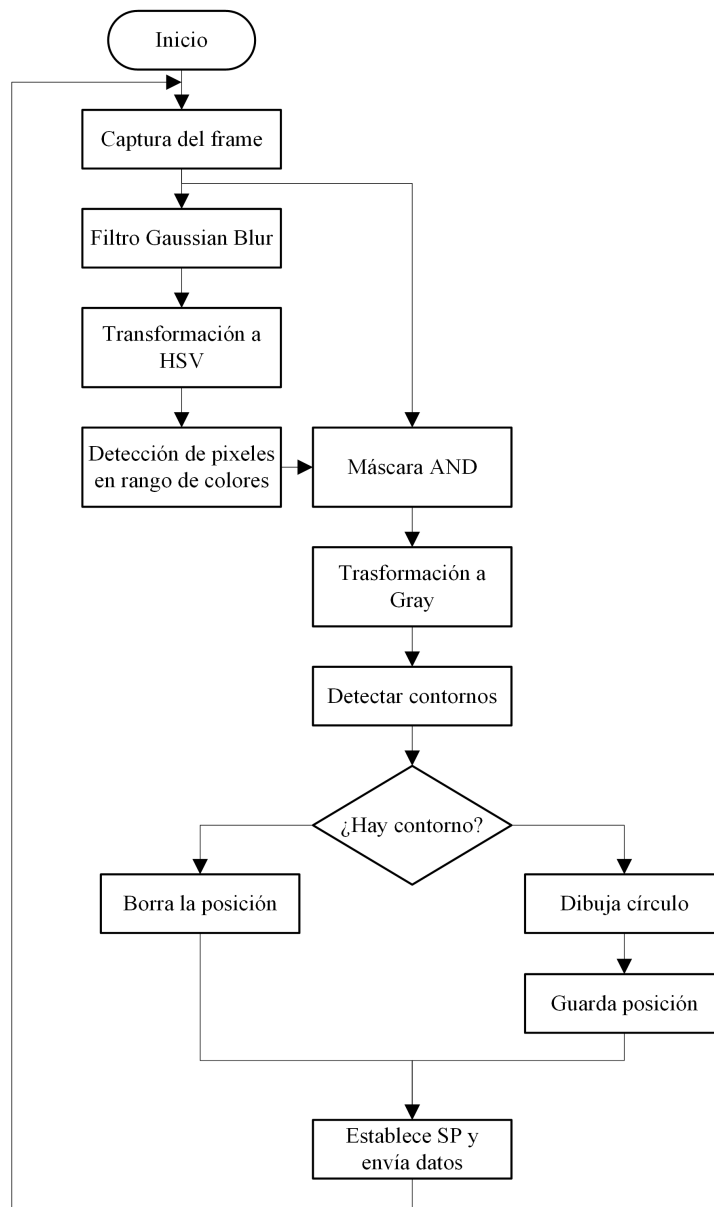


Figura 18. Diagrama de flujo del proceso de captura de posición de la pelota en la plataforma.

original, quedará específicamente la parte donde está ubicado el objeto.

Transformación a Gray: Cambia el espacio del color a escala de grises con el fin de que un pixel constituya un solo valor de intensidad de gris y se facilite su procesamiento.

Detectar contornos: Se lo realiza mediante el método *findContours* de OpenCV. Si existen con-

tornos, es decir, si se encuentra un objeto, se lo caracteriza en posición y radio mediante el método *minEnclosingCircle* y se dibuja el contorno del objeto como un círculo (ya que se espera una pelota) con el método *circle*. Si no se detectaron contornos se activa una bandera.

Envío de datos: Se envían los datos de forma serial por la interfaz miniUART. Se envía el setpoint en x e y , seguido de la posición encontrada de la pelota en x e y y finalmente la bandera de detección de objeto.

Todo este procesamiento, desde la captura de la imagen hasta el envío de los datos es llevado a cabo por la microcomputadora en 0.133 ± 0.00356 segundos. Es decir, tardando todos los demás componentes menos tiempo en procesamiento, se puede decir que este es el mínimo período de muestreo que se puede obtener en el sistema. Para la comunicación serial se utiliza una tasa de 9600 baudios, y se envían un total de 13 bytes.

2.2.2. Control

Como se pudo estudiar en el anterior capítulo, la implementación de un controlador por matriz dinámica es demandante a nivel computacional (Lundström et al., 1995) debido a la existencia de operaciones simples pero que se deben realizar recursivamente y por tanto consumen tiempo. Adicionalmente, el elemento de control es el que realizará a su vez los cálculos de la cinemática inversa para mover los actuadores de la plataforma, y estos involucran varias operaciones matriciales y de punto flotante. Debe tener varios puertos seriales, ya que recibirá datos desde la microcomputadora encargada de la adquisición de los datos de la posición de la bola, y debe enviar las secuencias seriales al driver de los servomotores.

Por estos motivos, la tarjeta que se ha elegido es una STM32F4Discovery. La tabla 4 muestra las principales características de esta tarjeta microcontrolada.

Se puede notar la presencia de un FPU o unidad de punto flotante, que actúa como un coprocesador matemático especializado en operaciones de punto flotante, además, estos coprocesadores

Tabla 4

Características principales de la tarjeta STM32F4Discovery.

Componente	Características
Microcontrolador	STM32F407VGT6 con microprocesador ARM [®] Cortex [®] -M4 y núcleo FPU.
Memorias	Flash de 1 MB; RAM de 192 KB.
Alimentación	5 VDC a través de USB; externa de 3 y 5 V.
Interfaces	USB ST-LINK; USB-OTG
Acelerómetro	LIS3DSH ST MEMS 3-ejes
Pulsadores	2 (User y Reset)
LEDs	8
Otros componentes	CS43L22 audio DAC; Driver clase D para parlante; sensor de audio MP45DT02 ST-MEMS

Fuente: (STMicroelectronics, 2016)

son compatibles con librerías especializadas en procesamiento digital de señales, que incluyen funciones para operaciones matriciales (Arm Ltd., 2019).

2.2.3. Ensamble del sistema

Con todos los componentes seleccionados, la comunicación entre ellos se la realizará mediante UART a 9600 baudios. El esquema general de como está estructurado el sistema completo se muestra en la figura 19.

La cámara es la encargada de capturar la imagen la cual será transmitida mediante USB a la microcomputadora Raspberry Pi donde será procesada para obtener la posición de la bola en la plataforma. Esta posición será enviada de manera serial a la tarjeta STM32F4Discovery junto con el setpoint y la misma ejecutará las estrategias de control pertinentes. Así mismo, la tarjeta ejecutará el algoritmo de la cinemática inversa y transmitirá serialmente al driver de los servomotores el ancho de pulso que deberá ser enviado a cada uno. Nótese la necesidad de incluir un convertor de nivel dado que el driver trabaja con niveles de tensión de 5 V mientras que la tarjeta trabaja con 3 V. Finalmente, el driver generará las señales PWM en cada motor haciendo que estos se muevan a la

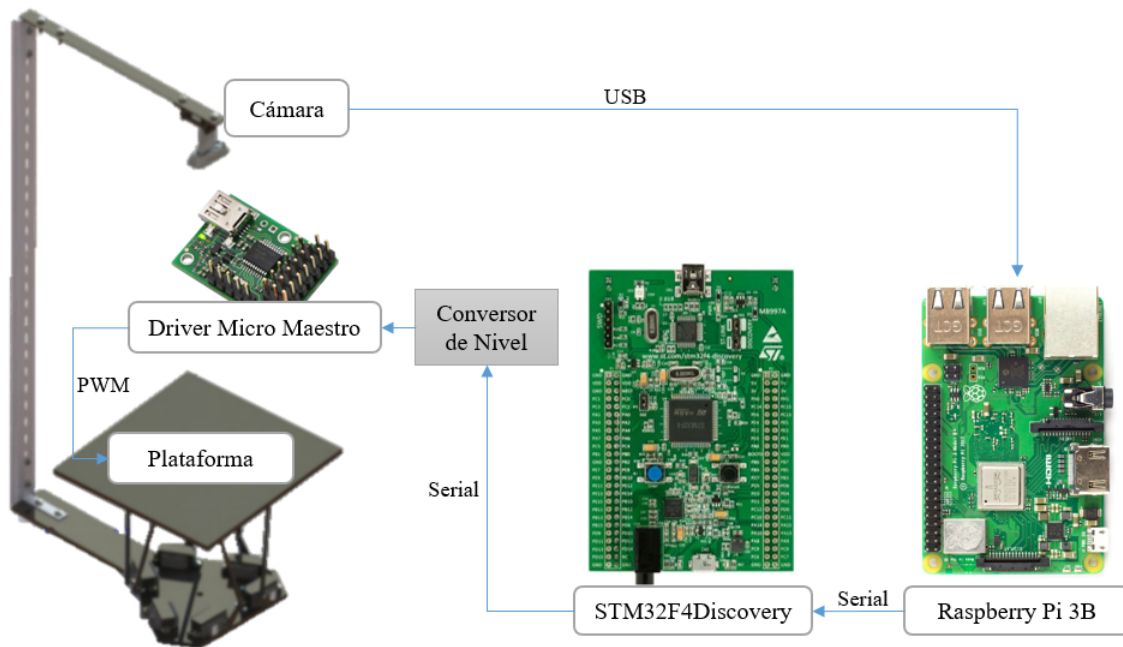


Figura 19. Esquema completo de interacción de los componentes del sistema.

posición requerida.

2.3. Identificación del sistema

La identificación de sistemas es el medio por el cual se intenta definir la relación entre la entrada y la salida de un sistema (Aguado Behar & Martinez Iranzo, 2003). La relación que se quiere encontrar para el sistema completo de la figura 19 es entonces la de la posición de la pelota en función de las señales de control. Por fines de simplicidad, se determinan las señales de control como los valores angulares de roll y pitch que deberá girar la plataforma puesto que son los que tienen el efecto más directo con los cambios de posiciones en x e y respectivamente (De Otaola, 2015).

2.3.1. Señal de entrada

Existen varios aspectos a tomar en cuenta cuando se debe identificar un sistema. Una de las consideraciones más importantes es el tipo de señal de entrada con la que se excita el sistema para obtener los datos de salida. Esta señal debe ser capaz de mostrar todas las dinámicas del sistema, por esto, se opta por utilizar una señal binaria pseudo aleatoria (SBPA) que es con la que se consigue esto (Aguado Behar & Martinez Iranzo, 2003).

La SBPA es una señal que puede adquirir dos valores que representarían el 1 y el 0 lógicos, es periódica y se genera mediante registros de desplazamiento; si se usan n registros, el período máximo está dado por (Ruiz, 2013):

$$M = 2^n - 1 \quad (2.3.1)$$

La señal es generada mediante registros de desplazamiento con retroalimentación lineal o LFSR, en donde los registros a_k se realimentan a la entrada tras ser operandos de una compuerta XOR (Pini, 2018). La figura 20 muestra el esquema de un LFSR de 4 bits; y la tabla 5 muestra el período máximo y los operandos para diferentes valores de n en una SBPA. Nótese que el tiempo más corto en que se activa la señal deberá ser por lo menos la mitad del tiempo del sistema, y el tiempo más largo deberá ser al menos el tiempo de establecimiento (Ayala-Taco et al., 2014).

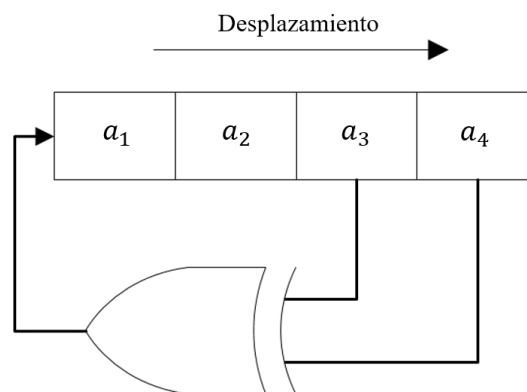


Figura 20. LFSR de 4 bits para generación de SBPA.

Fuente: (Ayala-Taco et al., 2014)

Tabla 5

Coeficientes para generar señales binarias pseudo aleatorias.

Orden n	$M = 2^n - 1$	$a_k \ k = 1, 2, \dots, n$
2	3	1, 2
3	7	2, 3
4	15	3, 4
5	31	3, 5
6	63	5, 6
7	127	6, 7
8	255	4, 5, 6, 8
9	511	5, 9
10	1023	7, 10
11	2047	9, 11

Fuente: (Aguado Behar & Martinez Iranzo, 2003)

2.3.2. Identificación en lazo cerrado

Existe una restricción más dada la naturaleza del sistema. Si se coloca la bola en la plataforma y se aplica la señal de control, sea roll o pitch, la plataforma se inclinará ocasionando que irremediablemente la bola caiga. Suponiendo que la plataforma fuese infinita, la bola seguiría rodando indefinidamente ante cualquier giro, esto causaría que no se pueda estimar ciertamente el tiempo del sistema ni el tiempo de establecimiento, para establecer y validar rangos prudentes de tiempo de muestreo.

Se recurre entonces, aunque se recomienda evitarlo de ser posible (Poulsen, 2019), a identificación del sistema en lazo cerrado. Mandloi & Shah (2015) establecen usar este método cuando se presenten plantas inestables, cuando el lazo abierto pudiere involucrar problemas en la producción, económicos o de seguridad, o cuando se desee evitar comportamientos no deseados del sistema, aunque se advierte que el lazo cerrado puede desencadenar inexactitud en el resultado final (Forsell, 1999).

El procedimiento es el mismo al que se planteó, se excitará al sistema con una señal binaria pseudo aleatoria, pero, en este caso el sistema ya estará provisto de un controlador que evita que

la bola caiga de la plataforma. Se diseñó, para ello, un controlador PD de forma experimental. La forma de tomar datos en el sistema es entonces la que se detalla en la figura 21.

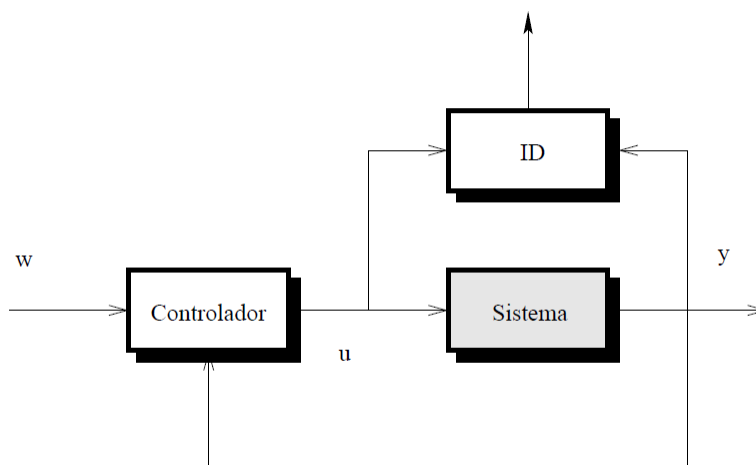


Figura 21. Diagrama para toma de datos para identificación.

Fuente: (Poulsen, 2019)

El controlador PD implementado tiene los parámetros que se describen en la tabla 6. Su presencia hace que la bola no caiga pero no llega a establecerse en el setpoint deseado.

Tabla 6

Parámetros del controlador PD

Parámetro	Eje x	Eje y
k_p	0.0075	0.0026
k_d	0.023	0.02

2.3.3. Identificación

Habiendo planteado que se realizará identificación en lazo cerrado, dado que se quiere evitar comportamientos no deseados (que se caiga la bola) y que se utilizará una señal binaria pseudo-aleatoria de orden 7 para tener un rango amplio entre los tiempos corto y largo ante la dinámica del sistema, se procede a tomar datos del mismo tal como se encuentra ensamblado de acuerdo a la

figura 19. Es importante aclarar que se hace la toma de datos con un tiempo de muestreo de 0.133 segundos, que es el mínimo posible en la configuración actual del sistema.

Primero se hace el ensayo de toma de datos en el eje x , es decir, la SBPA es el setpoint en x , la señal de control es el roll en grados y la salida del sistema es la posición en x . Se toman 2290 muestras, de las cuales se utilizaron 1700 para la identificación y 590 para verificación; como algoritmo de identificación se utiliza el método de mínimos cuadrados no lineal. Como está en lazo cerrado, posteriormente se despeja la acción del controlador obteniendo la función de transferencia del sistema. Se presenta la que obtuvo mejor porcentaje de coincidencia.

$$\frac{pos_x}{roll} = \frac{6,568s^2 + 335,7s - 2,1}{s^3 + 6,295s^2 + 1,827s + 0,542} \quad (2.3.2)$$

Para el eje y se hace lo mismo. En este caso la entrada es el setpoint en y , la acción de control es el pitch en grados y la salida del sistema es la posición en y de la bola. De las 525 muestras tomadas, 350 se utilizan para la identificación y 175 para verificación. Luego de despejar del controlador, la función de transferencia con mejor porcentaje de coincidencia es:

$$\frac{pos_y}{pitch} = \frac{5,056s^3 + 63,46s^2 + 3,492s + 0,4599}{s^3 + 0,3602s^2 + 0,1312s + 0,002924} \quad (2.3.3)$$

La comparación de los datos tomados y los identificados en el eje x se puede observar en la figura 22; y en el eje y se puede observar en la figura 23.

2.4. Desarrollo de la estrategia de control

Ya que se pretende implementar una estrategia de control DMC en el sistema real, se van a tomar varias consideraciones adicionales a las expuestas en el capítulo 1. El algoritmo de control consta de dos etapas: la etapa de inicialización que se puede realizar fuera de línea, y la parte que actuará en línea con el proceso. El diagrama de flujo para ambas partes se muestra en la figura 24 y

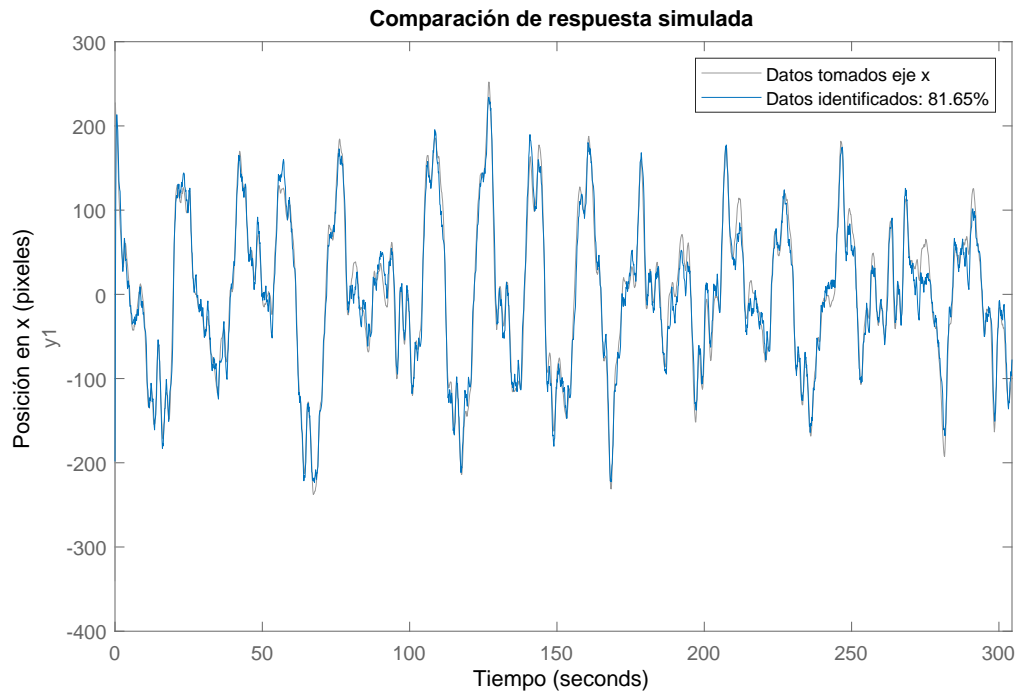


Figura 22. Comparación de datos del eje x.

se detalla posteriormente lo que realiza cada etapa del mismo.

2.4.1. Inicialización del DMC

Inicializar horizontes de control y predicción: El horizonte de predicción p es el número de muestras en las que se realizará la predicción a partir del instante actual y el horizonte de control m es el número de movimientos de control que influirá en la predicción (Camacho & Bordons, 2007).

Inicializar matriz dinámica: Se forma la matriz dinámica como se muestra en la ecuación 1.4.8 a partir de la respuesta normalizada al escalón del sistema. Como producto de la identificación de los sistemas, se tienen las respuestas al escalón que se muestran en la figura 25. Como se puede notar se ha forzado la estabilidad en los sistemas, dado que de no ser estables no sería

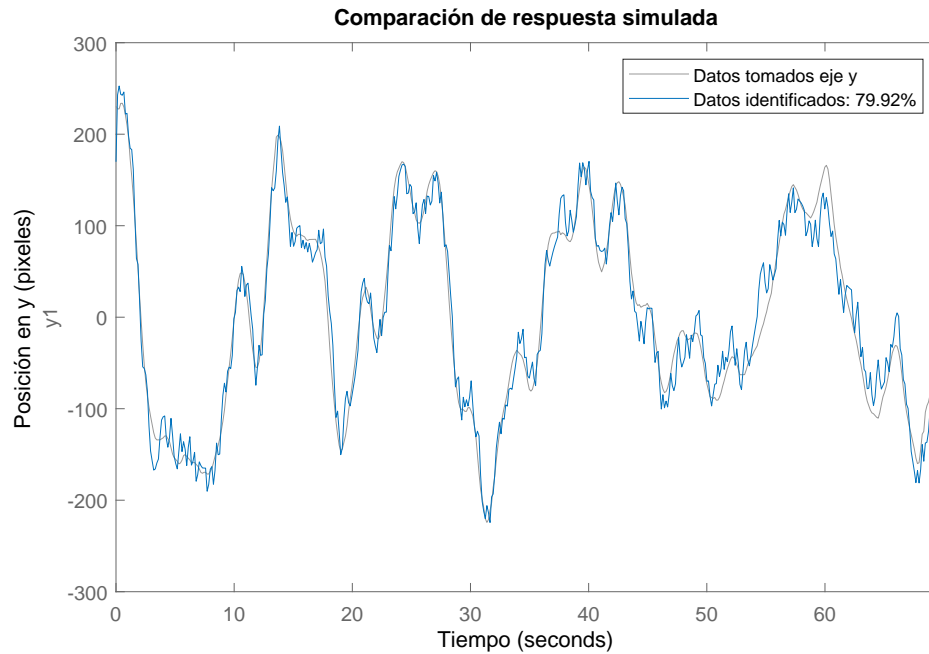


Figura 23. Comparación de datos del eje y.

factible de ser controlado por DMC.

Inicializar el vector de trayectoria y de movimientos de control: El vector de trayectoria y_{sp} de tamaño p constituye la referencia que debe alcanzar el sistema; el vector de movimientos de control contiene los movimientos de iteraciones anteriores; su tamaño N depende de la cantidad de muestras que le toma al sistema estabilizarse en lazo abierto (Castaño, 2015).

Determinar el peso de penalización: Este peso λ , determina cuánto se debe penalizar el valor al que está multiplicado durante el proceso de optimización de la función de costo (Camacho & Bordons, 2007).

Hallar el vector K óptimo Si la función de costo está determinada por la ecuación 1.4.13, su ley de control es la ecuación 1.4.14; no obstante el proceso de optimización solo se realiza en esta etapa de inicialización, por lo tanto, se puede obtener un vector K que es la primera fila

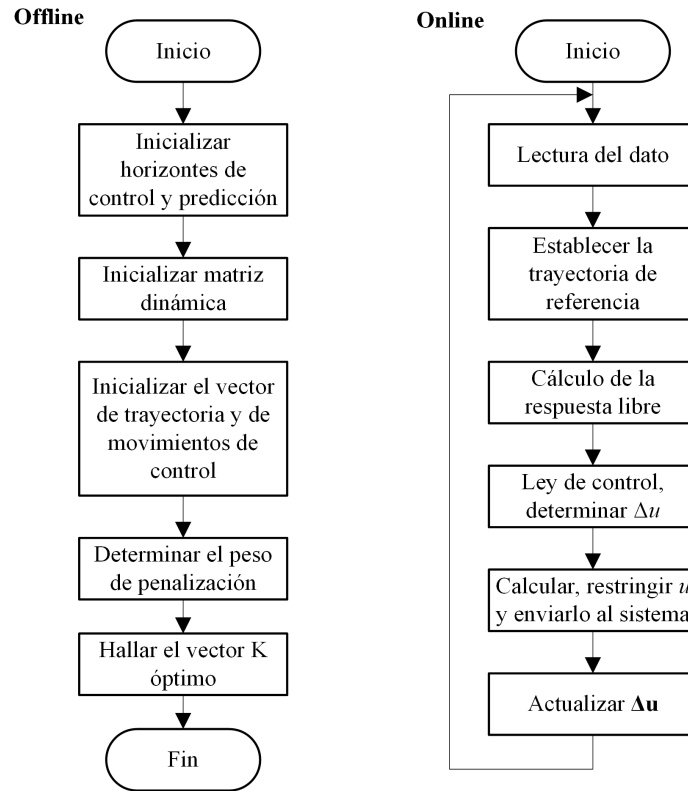


Figura 24. Diagrama de flujo de control DMC.

de la expresión \mathbf{O}_p que multiplica al factor $(\mathbf{y}_{sp} - \mathbf{f})$, es decir:

$$\mathbf{O}_p = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \quad (2.4.1)$$

$$\mathbf{K} = \mathbf{O}_{p_{1,*}} \quad (2.4.2)$$

2.4.2. Algoritmo DMC en línea

Habiendo inicializado todas las variables necesarias, se puede empezar el lazo de control, con las etapas descritas a continuación.

Lectura del dato: Se empieza tomando la muestra, es decir el dato actual $y_m = y(t)$. En el caso del sistema, se toma la posición de la bola en x y en y , ambas en píxeles.

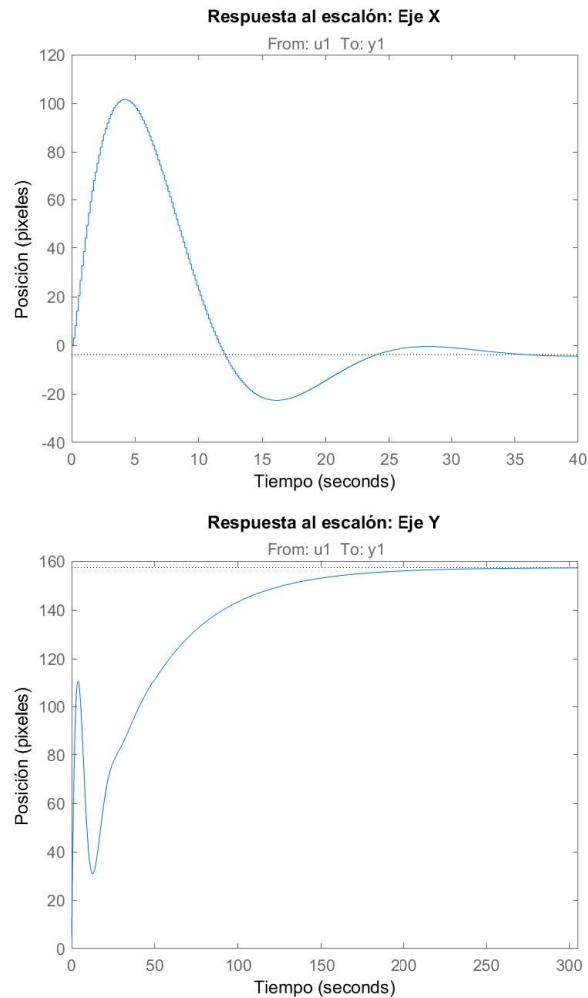


Figura 25. Respuesta al escalón de los sistemas a controlar.

Establecer la trayectoria de referencia: El vector y_{sp} puede ser simplemente el setpoint deseado y_{sp} ; pero si se quiere establecer una trayectoria más suave a lo largo del horizonte de predicción se puede usar la expresión:

$$y(t+j) = \alpha y(t+j-1) + (1-\alpha)y_{sp} \quad \text{para } j = 1, 2, \dots, p \quad (2.4.3)$$

La expresión anterior da como resultado una trayectoria suave, como la que se muestra en la figura 26, dependiendo del valor del parámetro α .

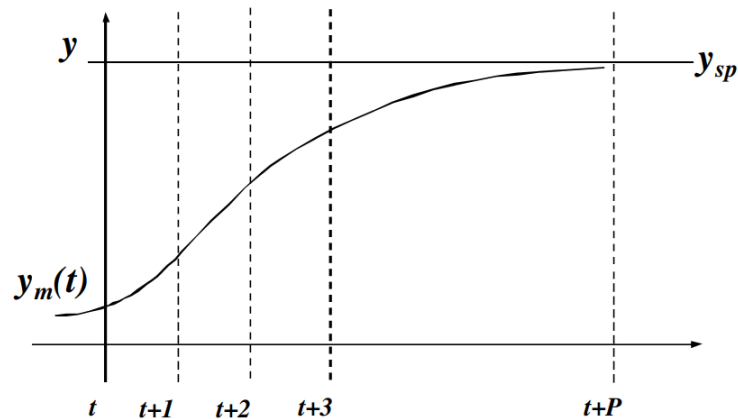


Figura 26. Trayectoria suave de setpoint.

Fuente: (Noerpamoengkas, 2009)

Cálculo de la respuesta libre: La repuesta libre puede ser calculada mediante la ecuación 1.4.6.

Ley de control, determinar Δu : El movimiento de control actual puede ser calculado mediante la expresión (Camacho & Bordons, 2007):

$$\Delta u(t) = \mathbf{K}(y_{sp} - f) \quad (2.4.4)$$

Calcular, restringir u y enviarlo al sistema: Se calculó el movimiento del control, para la señal de control que debe ser enviada se debe sumarla al valor anterior de la señal de control, esto es:

$$u(t) = u(t - 1) + \Delta u(t) \quad (2.4.5)$$

El sistema puede girar de -15° a 15° tanto en roll y pitch, por lo tanto habrá que limitar a estos valores cuando se tenga una señal de control mayor que esos rangos.

Actualizar Δu : Finalmente, es importante actualizar el vector de movimientos de control, es decir, $\Delta u^-(j) = u(j)$ para $j = 0, 1, \dots, N - 1$. Además, se debe actualizar el valor pasado de la acción de control, es decir, $u(t - 1) = u(t)$ y se procede a repetir todo este procedimiento

para la siguiente muestra.

Todo el algoritmo se escribe en lenguaje C compatible con la tarjeta STM32F4Discovery.

2.5. Conclusiones del Capítulo

En este capítulo se ha caracterizado todo el sistema, desde sus partes constitutivas, hasta los algoritmos que ejecutarán los componentes programables. Se determinó usar una computadora Raspberry Pi 3B para procesar las imágenes obtenidas de la cámara, y se usa una tarjeta STM32F4Discovery para los algoritmos de control y de cinemática inversa. Se mantiene el driver Micro Maestro de 6 canales para comandar los servomotores y estos componentes se comunicarán de manera serial a 9600 baudios.

Se identificó la planta en lazo cerrado, con una SBPA como señal de entrada, obteniendo porcentajes de coincidencia de 81.65 % y 79.92 % para los ejes x e y respectivamente. Del ensamble del sistema se obtiene el período de muestreo mínimo posible de 0.133 segundos que es el tiempo más lento correspondiente a las tareas de visión artificial.

Se definieron los lenguajes de programación a utilizar como Python para los procedimientos de visión artificial, y C para los algoritmos de control. Finalmente se particularizó la estrategia de control para el sistema, teniendo en cuenta las operaciones que realizará el microcontrolador en las etapas de inicialización y de control como tal, así como las restricciones a tener en cuenta en lo que se refiere a rangos de la señal de control.

Capítulo 3

IMPLEMENTACIÓN Y RESULTADOS

Se simula el comportamiento del controlador, se valida el funcionamiento del mismo y se lo implementa. Los parámetros de desempeño a tener en cuenta son:

- Tiempo de establecimiento t_s .
- Tiempo de subida t_r .
- Error en estado estacionario e_{ss} .
- Sobreimpulso M_p .

3.1. Simulación

El diagrama de simulación para los controladores es el que se muestra en la figura 27. Se expresa el sistema como las funciones de transferencia en tiempo continuo identificadas en el capítulo anterior. Además, dada la presencia de un controlador digital, están presentes los bloques ZOH o retenedores de orden cero, los cuales representan los conversores analógico digital y digital analógico que tienen los microcontroladores (Landau & Zito, 2006). Se incluye además una discontinuidad de saturación, la cual hará las funciones de restricción en las magnitudes límite de la señal de

control y una perturbación a manera de escalón a la entrada del sistema. La función representa el microcontrolador como tal y en ella se desarrolla el algoritmo de control.

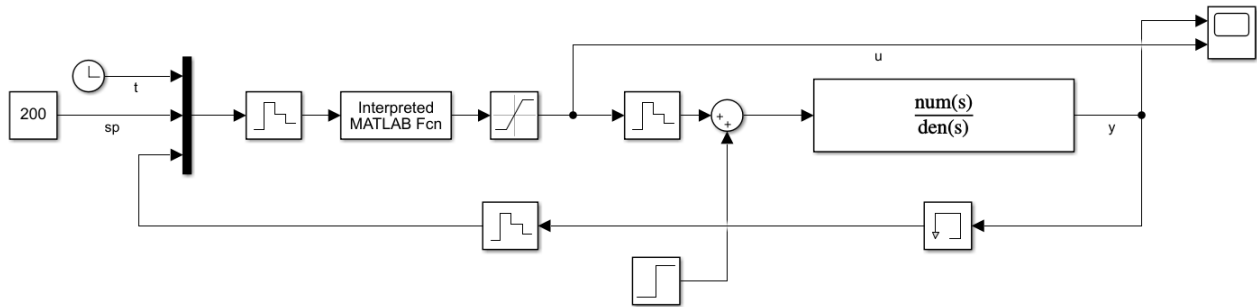


Figura 27. Esquema de simulación.

Se probaron varios valores para los parámetros λ correspondiente al peso de la acción de control en la función de costo, y se definieron valores de 10 y 5 para el horizonte de predicción y de control respectivamente, esto quiere decir que la predicción será hecha en cada instante a diez muestras en el futuro, es decir 1.33 segundos, tiempo en que aproximadamente, para el sistema en lazo abierto, la bola recorrería la mitad de distancia de la plataforma ante una inclinación de 10° .

Se presenta en la figura 28 los resultados para ciertos valores de λ para ambos ejes. La figura 29, muestra el desempeño del controlador ante perturbaciones tipo escalón de amplitud 0.25 con los valores de λ más altos propuestos anteriormente para cada eje. La perturbación se introduce a los 2 segundos.

3.1.1. Análisis de resultados simulados

El controlador funciona y se nota la influencia sobre todo del peso sobre la acción de control en la función de costo, el cual debe ser muy alto para lograr que no supere con mucha frecuencia los umbrales máximos alcanzables por el sistema real. De hecho, en los casos de $\lambda = 5$ para el eje x , se nota que la saturación actúa. En cuanto al eje y , se eligieron valores para el peso aun más altos, dado que, con valores más bajos el sistema tendía a oscilar por la agresividad impuesta en la señal de control, y siendo muy bajos se perdía el control sobre la misma.

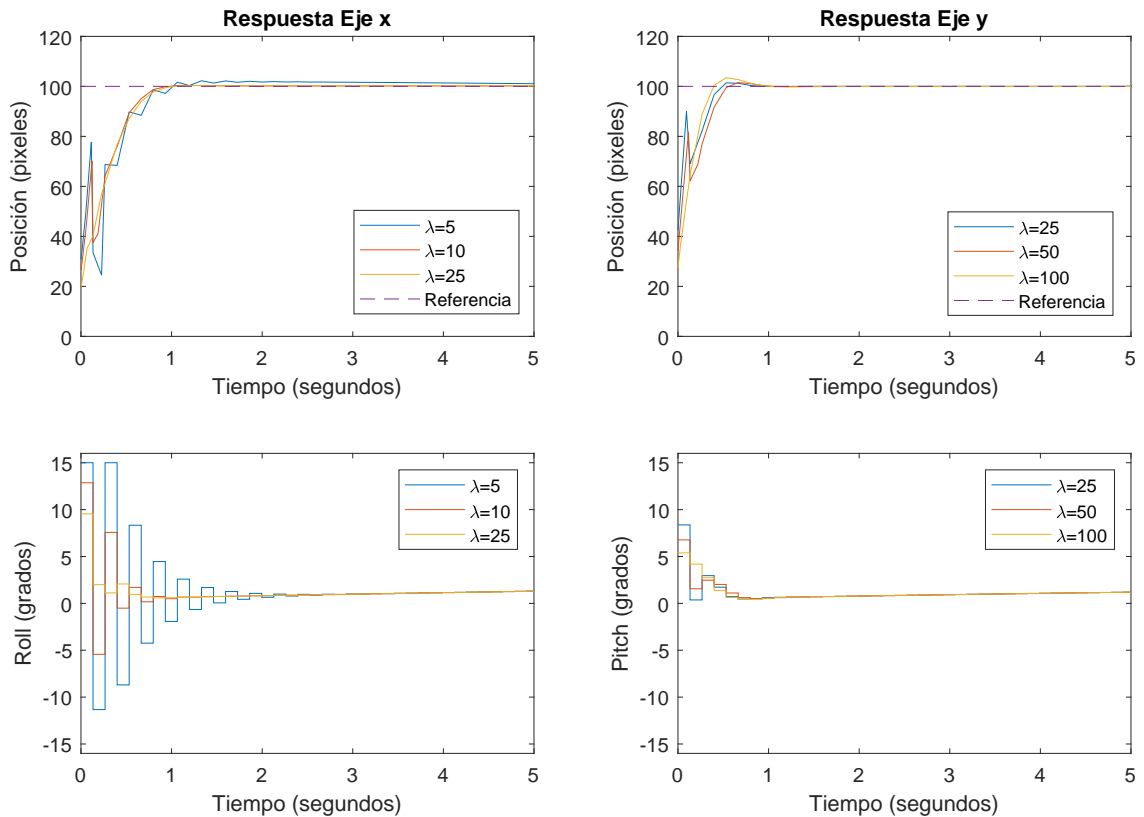


Figura 28. Simulación del DMC para varios valores de λ .

Respecto a los parámetros de desempeño, se nota, para todos los casos que el sobre impulso no supera el 5%; el tiempo de subida es de aproximadamente 0.6 segundos; no hay error en estado estacionario y el tiempo de establecimiento oscila entre 1.0 a 1.5 segundos.

Se usarán los valores más altos de λ como referencia para la implementación física, debido a que, sobre todo en la señal de control se aprecia que es menos variante en cuanto a valores positivos y negativos. Físicamente, esto implicaría giros muy pronunciados de la plataforma, además, el desempeño simulado para valores más altos genera una respuesta más suave.

Respecto a perturbaciones, en la simulación se introdujo una señal escalón. El controlador reacciona, aunque dado el peso alto sobre la señal de control casi no se logra apreciar, sin embargo el sistema se mantiene bajo control. Hay que notar que respecto a la magnitud de la referencia, la

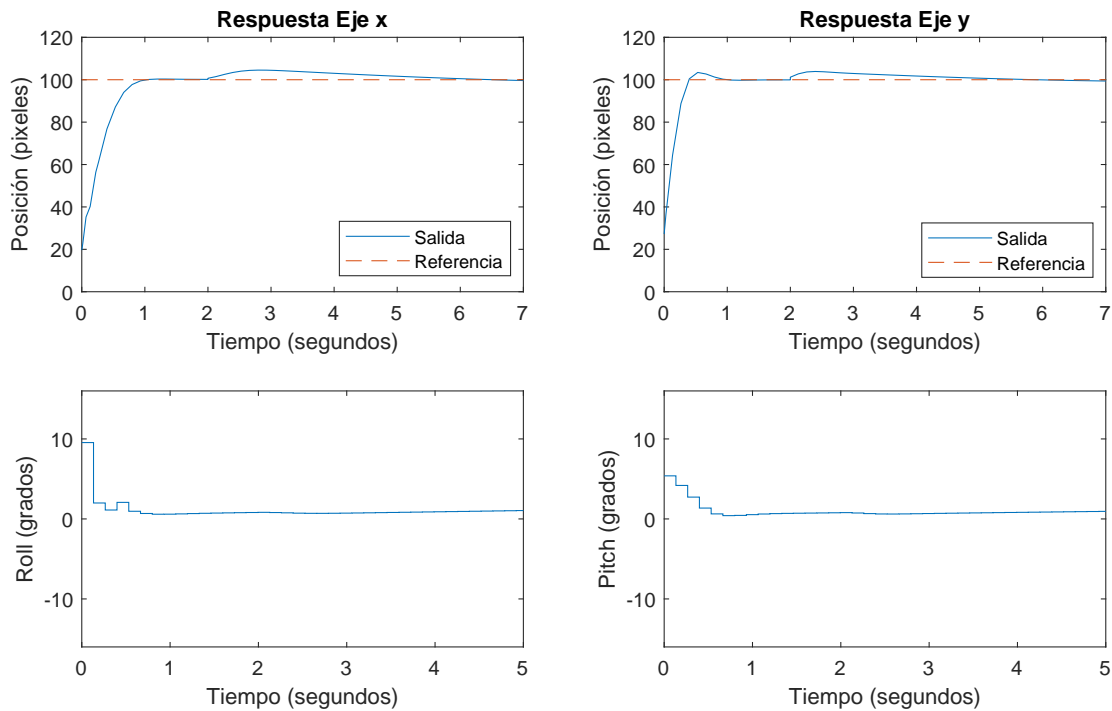


Figura 29. Simulación de perturbaciones.

perturbación es minúscula y ocasiona un cambio significativo en el sistema. Por lo tanto, se puede concluir que a pesar que el controlador reacciona ante las perturbaciones, físicamente estas tendrían que ser muy pequeñas puesto que la plataforma es limitada y ocasionarían la caída de la pelota.

3.2. Implementación

3.2.1. Detección de la pelota

Como se mencionó en el capítulo anterior, para la detección de la pelota se usa transformación a HSV y detección de píxeles mediante umbrales. Se dispone de tres pelotas de tenis de mesa de colores distintos: verde, amarillo y azul, además de dos pelotas metálicas. Los valores umbrales de los rangos de detección en el espacio del color HSV que funcionan con todos los tipos de pelotas anteriores son los que se muestran en la tabla 7.

Tabla 7*Parámetros para rango de detección de píxeles HSV*

Parámetro	Límite inferior	Límite superior
Hue	0	179
Saturation	0	177
Value	234	255

Es importante aclarar que estos valores funcionan exclusivamente para las condiciones de iluminación del lugar donde está ubicado el sistema a la fecha de realizado este trabajo, es decir, los laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE.

3.2.2. Pruebas de funcionamiento

El controlador implementado en el sistema real, es un DMC, inicialmente programado con los parámetros de los mejores resultados de la simulación, pero posteriormente sintonizado y puesto a prueba con los parámetros que se detallan en la tabla 8. En el mismo microcontrolador se agrega la restricción en la salida de la acción de control.

Tabla 8*Parámetros de los controladores DMC implementados*

Parámetro	Eje x	Eje y
λ	13	25
p	7	7
m	5	5

Se realizaron doce ensayos, colocando sin ningún impulso la bola en una esquina de la plataforma y dejando actuar al controlador. De todos ellos, un ensayo puede considerarse fallido, ya que tardó demasiado en lograr alcanzar el setpoint. En ningún caso la pelota cayó de la plataforma. La tabla 9 muestra los parámetros de desempeño de las once pruebas restantes.

La figura 30 muestra el ensayo que más se asemeja a los resultados esperados descritos en la tabla 9, correspondiendo al primer ensayo mostrado en la misma. La figura 31 muestra el ensayo

Tabla 9
Pruebas funcionales del controlador práctico

Ensayo	Eje x				Eje y			
	M_p [%]	e_{ss} [%]	t_r [s]	t_s [s]	M_p [%]	e_{ss} [%]	t_r [s]	t_s [s]
1	39.48	2.52	1.33	4.92	12.34	2.08	1.46	4.12
2	31.82	4.00	1.60	6.52	5.45	6.38	1.73	5.45
3	51.89	12.03	0.931	4.655	0.00	3.59	3.33	4.26
4	0.79	5.44	1.60	6.25	9.65	10.68	1.60	6.12
5	6.53	13.10	2.39	9.18	10.42	9.75	2.79	9.98
6	0.90	8.23	1.86	2.53	1.95	7.98	2.13	2.79
7	12.04	6.087	1.60	4.389	40.39	14.35	1.33	4.38
8	35.09	7.29	1.33	5.32	30.45	10.83	1.06	4.52
9	15.00	21.39	1.995	4.92	0.00	18.92	2.39	3.86
10	8.66	12.15	1.60	5.72	13.11	12.66	1.46	5.45
11	7.38	20.64	2.128	4.66	34.05	0.85	1.064	5.187
\bar{x}	19.05	10.26	1.67	5.37	14.35	8.92	1.85	5.10
σ	17.45	6.32	0.41	1.65	14.19	5.46	0.73	1.86

con el mejor desempeño en cuanto a tiempo de establecimiento y sobreimpulso, y la figura 32 muestra el rendimiento con mayor error en estado estacionario.

Se realizó una prueba de reacción ante perturbaciones del controlador. El procedimiento fue esperar a que llegara al estado estacionario y empujar la pelota hacia los extremos. El resultado de esto se muestra en las figuras 33 y 34.

3.2.3. Análisis de resultados

El controlador, en los casos más favorables, cumple a cabalidad su función en tiempos menores a 6 segundos alcanzando errores promedio de 10.26 %. No obstante, hay mucha varianza en dos parámetros de desempeño: el sobre impulso y el error. Esto puede ser provocado debido a que la predicción falla a causa del desgaste de la plataforma, en la cual se presentan rayaduras, que forman canales por donde la pelota, al ser muy liviana, tiende a circular ocasionando comportamientos no muy susceptibles de predicción y que deberían ser tratados como perturbaciones grandes y con

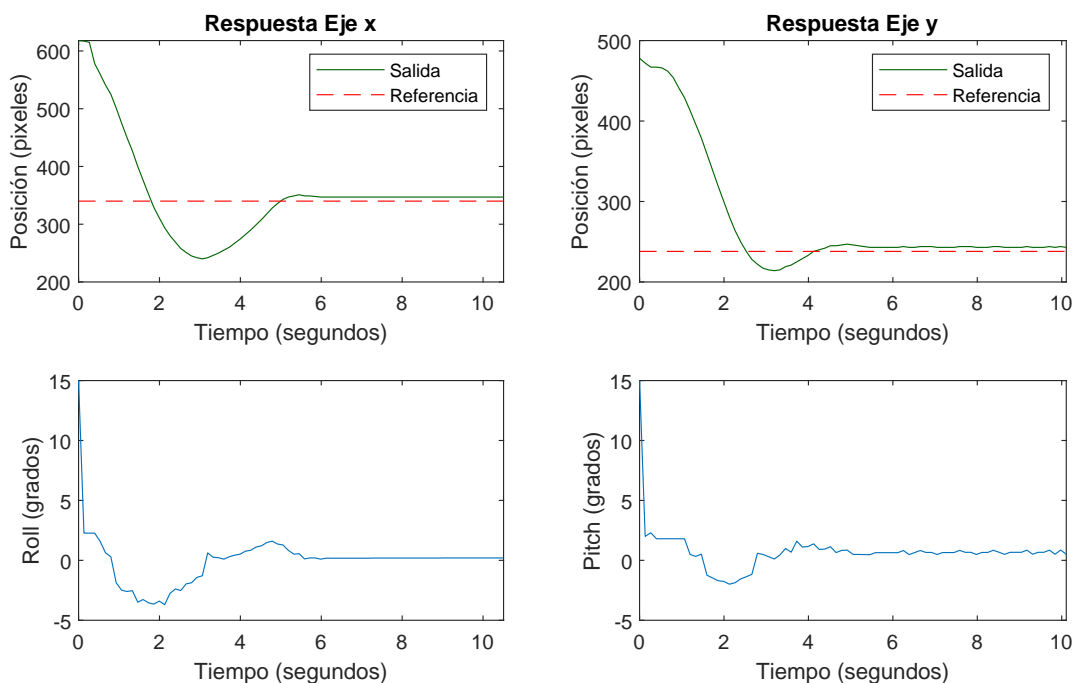


Figura 30. Prueba de funcionamiento con desempeño esperado.

modelos más específicos. A pesar de esto, el sistema controla, y aunque cuando ocurre esto tarda más tiempo, sigue acercándose al setpoint deseado.

En el escenario en el que, alcanzado el estado estacionario, se procede a empujar la pelota, se nota claramente la brevedad con la que el controlador actúa e intenta volver a posicionar la pelota en el punto deseado. Ante este tipo de perturbación, se espera que el sistema se demore alrededor de 6 segundos en volver al estado estable, no obstante, si la perturbación es grande y además la pelota se encuentra con los desgastes de la plataforma, puede tardarse hasta casi 10 segundos como se puede ver en las figuras 33 y 34, concretamente en la primera perturbación. Durante la segunda perturbación, al ser menor, se demoró aproximadamente un 40 % menos.

En ensayos anteriores, concretamente durante la sintonización, se cambiaron los horizontes de control y predicción. Los resultados ante estos cambios fueron los siguientes:

- Un horizonte de predicción menor que 5, hizo que el sistema tarde mucho más en alcanzar el

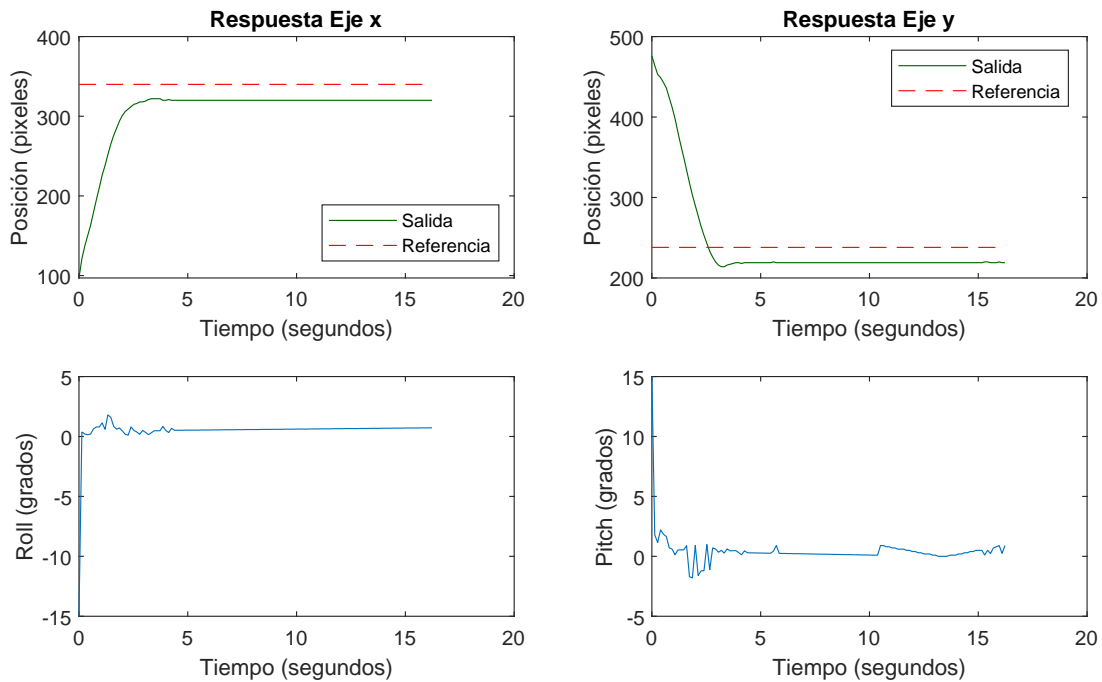


Figura 31. Prueba de funcionamiento con mejor tiempo de establecimiento.

setpoint, y dada la sensibilidad de los actuadores, en muchas ocasiones no lo logra, pues los giros que realiza no son suficientemente significativos para romper la inercia de la pelota.

- Un horizonte de control que se acerque a menos de 3 muestras hacia el horizonte de predicción, hace que el sistema se descontrole, debido a que la acción de control es mayor y el sistema no logra manejar cambios tan agresivos, haciendo que la pelota oscile hasta caer.
- Si se usan valores menores que 2 y 4 para m o p respectivamente, se pierde completamente el control sobre el sistema.

El factor que resulta ser más crítico particularmente para este sistema es λ , el cual básicamente determina la intensidad de la señal de control, y por lo tanto la agresividad del mismo. Disminuyendo la penalización, el sistema tiende a desestabilizarse, principalmente debido al tiempo que le toma a la plataforma alcanzar inclinaciones muy pronunciadas; por tanto, aquí se justifica la restricción de los rangos de giro.

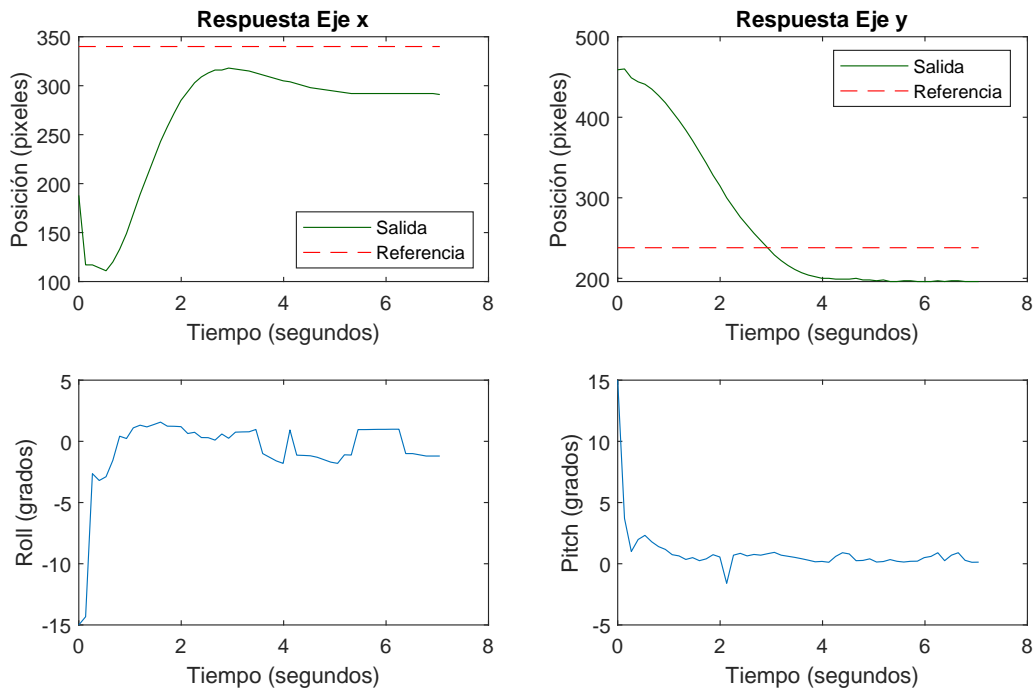


Figura 32. Prueba de funcionamiento con mayor error estacionario.

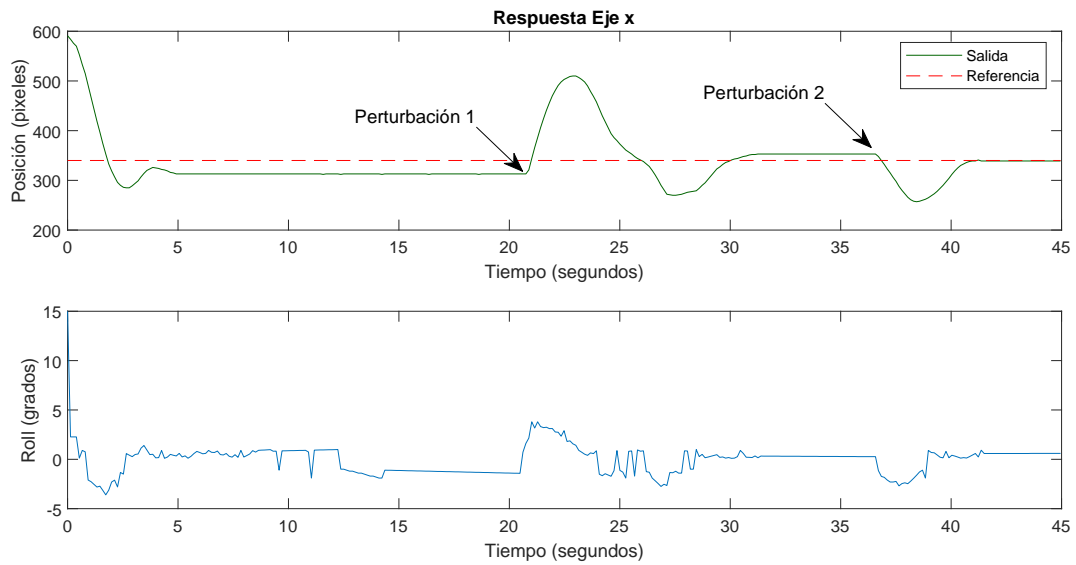


Figura 33. Prueba de funcionamiento con perturbación en eje x.

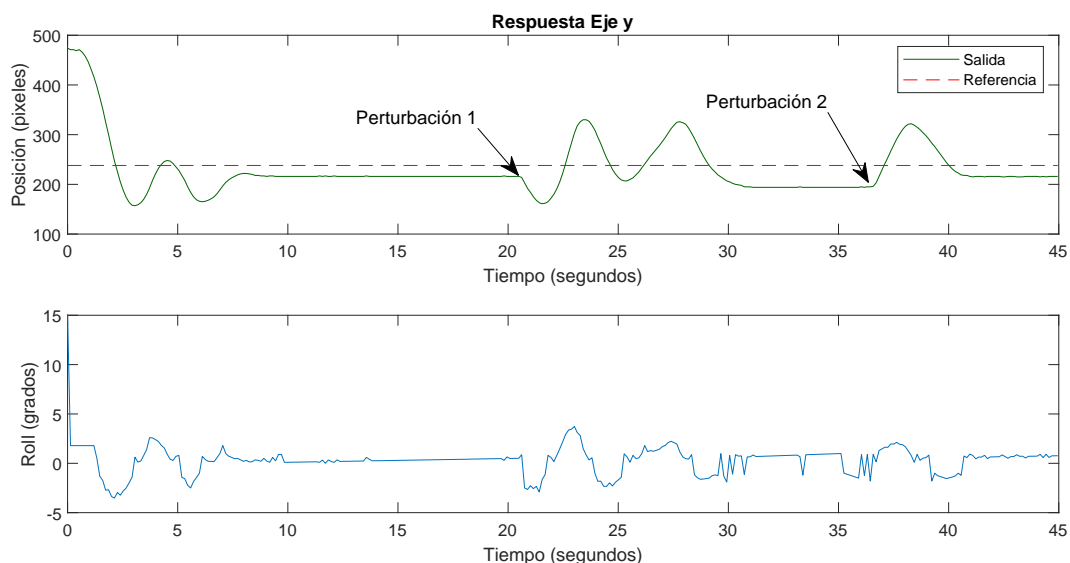


Figura 34. Prueba de funcionamiento con perturbación en eje y.

3.3. Tiempos de ejecución

En esta sección se muestran y analizan los tiempos que tarda cada componente en realizar sus tareas. Los tiempos en la Raspberry Pi 3B fueron tomados con funciones nativas del sistema, mientras que en la STM32F4Discovery fueron tomados midiendo el tiempo en alto de banderas que se activan al iniciar una tarea y desactivan al finalizarla, mediante un osciloscopio. Se recalca que la Raspberry Pi trabaja a las frecuencias base de su modelo, y la STM32F4Discovery fue configurada para trabajar a 168 MHz.

La figura 35 muestra el diagrama de tiempo correspondiente a las dos tareas principales del sistema: adquisición de datos y control. Se verifica que, en efecto, la tarea que se lleva a cabo en más tiempo es precisamente la adquisición y envío de datos (133 ms), lo cual involucra procesamiento de imagen, visión artificial y comunicación serial, y la desempeña la microcomputadora Raspberry Pi 3B. Todas las demás acciones, es decir, recepción de datos seriales, control DMC y cinemática inversa son desempeñadas por la tarjeta microcontrolada STM32F4Discovery, y en todo aquello tarda 70 ms. Se aprecia que el envío y recepción de datos entre los componentes tarda 17.8 ms.

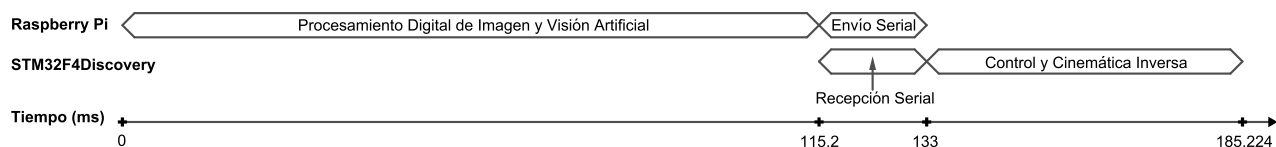


Figura 35. Diagrama de tiempo general de los componentes de adquisición de datos y de control.

La figura 36 muestra el detalle de los tiempos de la STM32F4Discovery. Aquí se puede apreciar que lo que más tiempo consume en realidad es la comunicación serial, puesto que para toma y monitorización de datos, se envían los mismos por un puerto UART para disponibilidad de quien lo requiera, y por otro lado, la comunicación con el driver de los servomotores también utiliza una interfaz serial. De todo el tiempo, solo 1.024 ms están dedicados plenamente a las tareas de control. De esta pequeñísima porción, el control DMC para el eje x tarda $101 \mu s$ y en y tarda $103 \mu s$, y la cinemática inversa tarda $820 \mu s$. Con fines de comparación, un controlador PID digital implementado en el mismo microcontrolador tarda en realizar sus tareas un tiempo de $2 \mu s$.

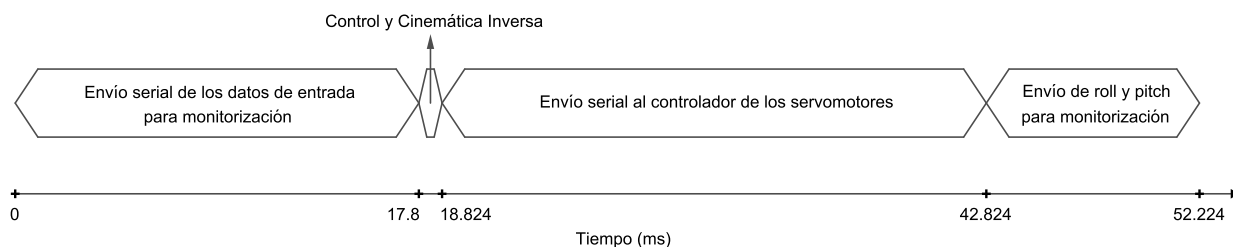


Figura 36. Diagrama de tiempo del detalle de tareas del microcontrolador.

3.4. Conclusiones del Capítulo

En este capítulo se abordó la simulación e implementación de las estrategias de control basadas en matriz dinámica, con resultados satisfactorios, pues, a nivel práctico, se ha logrado poner en marcha el sistema con errores esperados de alrededor de 10% y tiempos de subida y de establecimiento muy cortos, de alrededor de 5 segundos. Se realizaron y documentaron varios ensayos que

otorgaron una perspectiva clara del funcionamiento del controlador, sobre todo, la alta varianza que se obtiene en cuanto a sobre impulso, proporciona una noción de la cantidad de no linealidades, discontinuidades y perturbaciones al que el sistema es susceptible. Adicionalmente, se realizó la medición de los tiempos de ejecución de las tareas del sistema, siendo la microcomputadora en sus tareas de visión artificial la que más tarda, y por tanto definiendo el tiempo de muestreo mínimo del sistema.

CONCLUSIONES y RECOMENDACIONES

Conclusiones

Se ha implementado la estrategia de control predictivo basada en matriz dinámica para una plataforma Stewart de 6 grados de libertad para la aplicación de bola en plataforma, para ello, se ha estudiado la plataforma desde el punto de vista de la robótica, y se han definido las ecuaciones que rigen el comportamiento cinemático de la misma. Por otro lado, se ha analizado la estrategia DMC como tal, y las ventajas que involucra el uso de esta técnica, particularmente, beneficia a este trabajo la robustez que ofrece ante plantas con altos niveles de no linealidad, como es el caso de este robot paralelo.

Se tiene un sistema multivariable, en donde las entradas roll y pitch influyen en salidas que son las posiciones en x e y de la pelota, respectivamente. Dado que los sistemas de posición son ortogonales se pudieron controlar por separado y se pusieron en marcha controladores basados en matriz dinámica, obteniendo un error esperado del 10.25 % para el eje x , 8.92 % para el eje y ; y un tiempo de establecimiento esperado de 5.36 segundos para el eje x y de 5.10 segundos para el eje y . En cuanto al sobreimpulso, los valores tomados durante los ensayos presentan una varianza relativamente alta, y se puede decir que a este parámetro es lo que más afecta la cantidad de perturbaciones y no linealidades presentes en el sistema, pues DMC sigue siendo un controlador basado en un modelo lineal.

Se utilizaron técnicas de procesamiento digital de imágenes, concretamente las transformaciones de espacio de color, y visión artificial para el reconocimiento de la pelota en la plataforma, específicamente mediante detección de contornos. Como resultado del uso de estas técnicas, se pudo reconocer exitosamente pelotas de tenis de mesa de color azul, verde y amarillo, así como esferas metálicas bajo condiciones de iluminación típicas para interiores, como es el caso del laboratorio en donde se encuentra instalado el sistema.

Se identificaron los sistemas en lazo cerrado con el método de mínimos cuadrados no lineal; para ello se dispuso de un controlador PD diseñado para que la pelota no cayera de la plataforma. Como resultado de la identificación se obtuvieron modelos matemáticos que satisfacen en aproximadamente 80 % el comportamiento real del sistema.

Los controladores predictivos que basan su desempeño en una función de costo ofrecen la posibilidad de penalizar ciertas variables que intervienen en el sistema. Particularmente, en el presente trabajo durante la implementación del DMC se penalizó la acción de control en la función del error, lo cual fue muy beneficioso dado que por la naturaleza del sistema, se tienen rangos de trabajo que no se pueden sobrepasar.

Se programó el DMC tanto para el control de la posición de la pelota en el eje x como para el eje y , y el tiempo de ejecución de las tareas que involucra esto están en el orden de los microsegundos, abriendo la posibilidad de utilizar cada vez más controladores predictivos como alternativa a controladores clásicos convencionales. Sin embargo, aunque su tiempo de ejecución no es alto, para este caso particular, puede variar considerablemente si se aumenta el horizonte de control, de predicción, o si el modelo del sistema en lazo abierto tarda mucho en estabilizarse. En cambio estrategias de control clásico, definidas con un conjunto de operaciones, o de ecuaciones de diferencias, para todos los casos se ejecutarán en tiempos muy similares. Particularmente en el caso de este trabajo, el tiempo de ejecución de un PID fue aproximadamente el 2 % del tiempo de un DMC.

Comparando con el trabajo anterior realizado por Vela (2019) en el mismo sistema, los valores esperados en cuanto a error en estado estacionario son similares, oscilando entre 4.83 % a 12.67 %.

Se nota una mejora en el tiempo de establecimiento, pues en promedio con DMC se demora alrededor de 1.5 segundos menos que con la estrategia Fuzzy planteada en dicho trabajo.

Recomendaciones

Se recomienda ante cualquier cambio de posición del sistema, verificar los parámetros de las ventanas del espacio de color HSV para asegurar que la pelota siga siendo detectada. No se recomienda utilizar el sistema en exteriores, sobre todo ante incidencia directa de luz solar puesto que ante la superficie de acrílico causa mucho reflejo y fallas en la detección del objeto en la plataforma.

Se recomienda utilizar el sistema en ambientes con temperaturas frescas, o con buena ventilación, puesto que la microcomputadora Raspberry Pi solo logra mantener el tiempo de muestreo cuando no ocurre estrangulamiento térmico. Esto quiere decir que si este componente sobrepasa una temperatura, automáticamente disminuye la frecuencia de su procesador.

Para trabajos futuros, si se desea mejorar el tiempo de muestreo, se recomienda empezar por recurrir a métodos de captura de imagen que involucren procesamiento concurrente, o en su defecto el uso de varios hilos de procesamiento. Además se recomienda probar otros medios de comunicación entre componentes que no involucren tanto tiempo como la comunicación serial.

Se recomienda, de igual manera para trabajos futuros, implementar un control cinemático para la plataforma que ayude a la cinemática inversa a alcanzar con mayor precisión las posiciones y orientaciones deseadas.

Referencias

- Acedo, J. (2003). *Control Avanzado de Procesos (Teoría y práctica)*. Ediciones Díaz de Santos S. A.
- Aguado Behar, A. & Martínez Iranzo, M. A. (2003). *Identificación y control adaptativo*. Prentice Hall.
- Angeles, J. (2003). *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms* (2da ed.). Springer.
- Arm Ltd. (2019). Cmsis dsp software library. <http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>.
- Arteaga, F. & Contreras, J. (2003). Identificación de sistemas y control de matriz dinámica para la optimización de una planta de endulzamiento de gas. *Revista Ingeniería UC*, 10(2), 79–86.
- Awtar, S. & Craig, K. (2002). Mechatronic design of a ball-on-plate balancing system. *Mechatronics*, 12(2), 217–228.
- Ayala-Taco, J. P., Bernal-Oñate, C. P., Guerra-Jiménez, S., & Fernández-Correa, A. (2014). Identificación de sistemas basado en re-muestreo de señales periódicas implementado en fpga. *Energética*, 43(2-3), 23–29.
- Barrientos, A., Peñín, L. F., Balanger, C., & Aracil, R. (1997). *Fundamentos de Robótica*. McGraw-Hill.

- Basim, M. A., Sharma, N., Vignesh, A., Dinesh, P. B., & Ajith, I. (2018). Design of modified stewart platform for solar tracing applications. *International Journal of Computer Applications*, 180(38), 33–40.
- Bo, Y., Zhongcai, P., & Zhiyong, T. (2011). Fuzzy pid control of stewart platform. *School of Automation Science and Electrical Engineering*, 1, 763–768.
- Boian, R., Bouzit, M., Burdea, G., Lewis, J., & Deutsch, J. (2005). Dual stewart platform mobility simulator. *9th International Conference on Rehabilitation Robotics*, 1(6), 161–169.
- Breuel, T. M. (2008). The ocropus open source ocr system. *Document Recognition and Retrieval XV*, 6815, 120 – 134.
- Brownlee, J. (2019). *Deep Learning for Computer Vision*. Machine Learning Mastery.
- Bueno, C. & Maldonado, A. (2017). Diseño, construcción e implementación de una plataforma tipo stewart para control de una aplicación bola en plataforma a través de visión artificial. Proyecto de grado, Universidad de las Fuerzas Armadas - ESPE.
- Camacho, E. F. & Bordons, C. (2007). *Model Predictive Control* (2da ed.). Springer.
- Campoverde, M. & Guayasamín, R. (2018). Diseño, simulación y comparación de tres controladores predictivos: Control predictivo generalizado, control por matriz dinámica y predictor de smith robusto, aplicados a un reactor de agitación continua y un tanque de mezclado con retardo. Proyecto de grado, Escuela Politécnica Nacional.
- Castaño, S. (2015). Control predictivo basado en modelos dmc. <https://controlautomaticoeducacion.com/control-predictivo/dmc/>.
- Coifman, B., Beymer, D., McLauchlan, P., & Malik, J. (1998). A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 6(4), 271–288.

- Cutler, C. (2003). Control global. <https://www.controlglobal.com/articles/2003/159/?stage=Live>.
- De Luca, A. (2016). Industrial robotics. http://www.diag.uniroma1.it/~deluca/rob1_en/01_IndustrialRobots.pdf.
- De Otaola, P. (2015). Construcción y diseño del sistema de control de posición de una esfera sobre una plataforma móvil. Proyecto de grado.
- Eisele, R. (2019). Inverse kinematics of a stewart platform. <https://www.xarg.org/paper/inverse-kinematics-of-a-stewart-platform/>.
- Forsell, U. (1999). *Closed-loop Identification. Methods, Theory, and Applications*. Linus & Linnea AB.
- Forsyth, D. A. & Ponce, J. (2003). *Computer Vision: A Modern Approach*. Pearson.
- Gadelmawla, E. S. (2011). Computer vision algorithms for measurement and inspection of spur gears. *Measurement*, 44(9), 1669–1678.
- Gawkowski, P., Lawrynczuk, M., Marusak, P., & Sosnowski, J. (2008). Software implementation of explicit dmc algorithm with improved dependability. *Novel Algorithms and Techniques in Telecommunications, Automation & Industrial Electronics*, 1, 214–219.
- Glisson, T. H. (2011). *Introduction to Circuit Analysis and Design*. Springer.
- Gonzalez, R. C. & Woods, R. E. (2008). *Digital Image Processing* (3ra ed.). Prentice Hall.
- Greenspan, H., van Ginneken, B., & Summers, R. M. (2016). Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5), 1153 – 1159.
- Ham, C. & Taufiq, M. M. (2015). Development of a ball and plate system. *122nd ASEE Annual Conference & Exposition*, 122.

- He, J., Gu, H., & Wang, Z. (2013). Solving the forward kinematics problem of six-dof stewart platform using multi-task gaussian process. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227, 161–169.
- Ibrahim, D. (2019). *Arm-Based Microcontroller Projects Using Mbed*. Newnes.
- International Organization for Standardization (2012). Iso 8373:2012 robots and robotic devices - vocabulary. <https://www.iso.org/standard/55890.html>.
- Jain, A. K., Feng, J., & Nandakumar, K. (2010). Fingerprint matching. *Computer*, 43(2), 36–44.
- Jähne, B. & Haußecker, H. (2000). *Computer Vision and Applications*. Academic Press.
- Kapur, S. (2017). *Computer Vision with Python 3*. Packt Publishing Ltd.
- Kassem, A., Haddad, H., & Albitar, C. (2015). Comparison between different methods of control of ball and plate system with 6dof stewart platform. *International Federation of Automatic Control - Papers Online*, 48(11), 47–52.
- Koch, P. M., Kesteven, M., Nishioka, H., Jiang, H., Lin, K.-Y., Umetsu, K., Huang, Y.-D., Raffin, P., Chen, K.-J., Ibañez-Romano, F., & et al. (2009). Dual stewart platform mobility simulator. *The Astrophysical Journal*, 694(2), 1670–1684.
- Krishna, R. (2017). *Computer Vision: Foundations and Applications*. Stanford University.
- Kroymann, T. & Dee, R. (2013). Full motion dynamics 6dof motion simulator. <http://fullmotiondynamics.com/about-the-project/>.
- Landau, I. D. & Zito, G. (2006). *Digital Control Systems: Design, Identification and Implementation*. Springer.
- Lara-Molina, F. A. & Dumur, D. (2010). Architecture of predictive control for a stewart platform manipulator. *World Congress on Intelligent Control and Automation*, 8.

- Lopez-Guede, J. M., Fernandez-Gauna, B., Graña, M., & Oterino, F. (2013). On the influence of the prediction horizon in dynamic matrix control. *International Journal of Control Science and Engineering*, 3(1), 22–30.
- Lundström, P., Lee, J., Morari, M., & Skogestad, S. (1995). Limitations of dynamic matrix control. *Computers & Chemical Engineering*, 19(4), 409–421.
- Mandloi, R. & Shah, P. (2015). Methods for closed loop system identification in industry. *Journal of Chemical and Pharmaceutical Research*, 7(1), 892–896.
- Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3), 90–126.
- Moriones, E. (2016). Diseño, implementación y control de sistema de balance. *Revista SENNOVA*, 2, 135–149.
- Noerpamoengkas, A. (2009). Model based predictive control - dynamic matrix control. <https://noerpamoengkas.files.wordpress.com/2009/01/dynamic-matrix-controller.pdf>.
- Official Organ of the Royal Aero Club (1962). Aviation history.
- Ollero Baturone, A. (2001). *Robótica: Manipuladores y robots móviles*. Marcombo.
- O’Neal, K. (2016). Color models and color applications. <http://slideplayer.com/slide/9806828/>.
- Oravec, M. & Jadlovska, A. (2017). Intelligent positioning plate predictive control and concept of diagnosis system design. *Journal of Manufacturing and Industrial Engineering (MIE)*, 15(1-2), 1–9.
- Pini, A. (2018). Prbs y generación de ruido blanco. <https://www.digikey.com/es/articles/techzone/2018/mar/use-readily-available-components-generate-binary-sequences-white-noise#targetText=En%20las%20comunicaciones%2C%20una%20secuencia,igual%20que%20otras%20se%3%B1ales%20m%3BA%20tiples>.

- Poulsen, N. K. (2019). Stochastic adaptive control: Closed loop identification. <http://www2.imm.dtu.dk/courses/02421/clid.pdf>.
- Ramdani, A. & Grouni, S. (2017). Dynamic matrix control and generalized predictive control, comparison study with imc-pid. *International Journal of Hydrogen Energy*, 30, 1–10.
- Rodríguez, A. (2018). Diseño de un controlador predictivo por matriz dinámica (dmc) de temperatura para una central termosolar de generación eléctrica. Proyecto de grado, Universidad de las Fuerzas Armadas - ESPE.
- Ruchika, N. R. (2013). Model predictive control: History and development. *International Journal of Engineering Trends and Technology*, 4(6), 2600–2602.
- Ruiz, F. (2013). Identificación de sistemas, aspectos básicos de identificación. https://javeriana.edu.co/ruizf/sysid_practica.pdf.
- Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Doyle, F. J. (2011). *Process Dynamics and Control* (3ra ed.). Wiley.
- Shah, M. (1997). *Fundamentals of Computer Vision*. University of Central Florida.
- Smith, A. R. (1978). Color gamut transform pairs. *ACM SIGGRAPH Computer Graphics*, 12(3), 12–19.
- Solem, J. E. (2012). *Programming Computer Vision with Python*. Creative Commons.
- Spacek, L., Vojtesek, J., Gazdos, F., & Kadavy, T. (2018). Ball and plate model for robotic system. *Proceedings European Conference on Modelling and Simulation*, 32.
- Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 100(15), 371–386.

STMicroelectronics (2016). Stm32f4discovery. <https://www.st.com/en/evaluation-tools/stm32f4discovery.html>.

Åström, K. J. & Hägglund, T. (2009). *Control PID Avanzado*. Pearson Education S. A.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

The OpenCV Team (2019). About opencv. <https://opencv.org/about/>.

Vela, S. (2019). Desarrollo de estrategias de control en base a lógica difusa y redes neuronales en un robot paralelo tipo stewart para aplicación de bola en plataforma a través de visión artificial. Proyecto de grado, Universidad de las Fuerzas Armadas - ESPE.

Vernon, D. (1991). *Machine Vision: Automated Visual Inspection and Robot Vision*. Prentice Hall.

Williams, B. (2007). *An Introduction to Robotics*. Ohio University.

Xiao, J. & Buttazzo, G. (2016). Adaptive embedded control for a ball and plate system. *ADAPTIVE 2016: The Eighth International Conference on Adaptive, Self-Adaptive Systems & Applications*, 8, 40–45.