



Diseño y construcción de una plataforma omnidireccional para evaluar controladores modernos en el seguimiento de trayectoria utilizando visión artificial.

Rodríguez Hidalgo, Jhon Carlos

Departamento de Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniero Mecatrónico

Ing. Mendoza Chipantasi, Darío José

17 de marzo del 2020



**DEPARTAMENTO DE ENERGÍA Y MECÁNICA
CARRERA DE MECATRÓNICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“DISEÑO Y CONSTRUCCIÓN DE UNA PLATAFORMA OMNIDIRECCIONAL PARA EVALUAR CONTROLADORES MODERNOS EN EL SEGUIMIENTO DE TRAYECTORIA UTILIZANDO VISIÓN ARTIFICIAL”** fue realizado por el señor **Rodríguez Hidalgo, Jhon Carlos** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, enero 2021



Firmado electrónicamente por:
**DARIO JOSE
MENDOZA
CHIPANTASI**

.....
Ing. Mendoza Chipantasi, Darío José

C.C:0603110834



Document Information

Analyzed document	TESIS FINAL RODRIGUEZ JHON.docx.pdf (D98793818)
Submitted	3/18/2021 5:43:00 PM
Submitted by	
Submitter email	jcrodrguezh@espe.edu.ec
Similarity	5%
Analysis address	djmendoza.espe@analysis.orkund.com



Firmado electrónicamente por:
DARIO JOSE
MENDOZA
CHIPANTASI



Firmado electrónicamente por:
DARIO JOSE
MENDOZA
CHIPANTASI

.....

Ing. Mendoza Chipantasi, Darío José

C.C:0603110834



DEPARTAMENTO DE ENERGÍA Y MECÁNICA
CARRERA DE MECATRÓNICA

RESPONSABILIDAD DE AUTORÍA

Yo, **Rodríguez Hidalgo, Jhon Carlos**, con cédula de ciudadanía N° **0503993024**, declaro que el contenido, ideas y criterios del trabajo de titulación: **DISEÑO Y CONSTRUCCIÓN DE UNA PLATAFORMA OMNIDIRECCIONAL PARA EVALUAR CONTROLADORES MODERNOS EN EL SEGUIMIENTO DE TRAYECTORIA UTILIZANDO VISIÓN ARTIFICIAL**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, enero 2021

.....
Rodríguez Hidalgo, Jhon Carlos

C.C.: 0503993024



**DEPARTAMENTO DE ENERGÍA Y MECÁNICA
CARRERA DE MECATRÓNICA**

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Rodríguez Hidalgo, Jhon Carlos**, con cédula de ciudadanía N° **0503993024**, **autorizo** a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **DISEÑO Y CONSTRUCCIÓN DE UNA PLATAFORMA OMNIDIRECCIONAL PARA EVALUAR CONTROLADORES MODERNOS EN EL SEGUIMIENTO DE TRAYECTORIA UTILIZANDO VISIÓN ARTIFICIAL** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, enero 2021

.....
Rodríguez Hidalgo, Jhon Carlos

C.C.: 0503993024

DEDICATORIA

Dedico este proyecto de titulación en primer lugar a Dios por darme la fortaleza, aprendizaje, valentía y coraje para enfrentar cada obstáculo que se me presentó durante el desarrollo del presente proyecto de titulación y en toda mi formación académica.

A mis padres quienes me dieron vida, educación, apoyo y consejos. A mis compañeros de estudio, a mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

A mis hermanos Jeremi, Heydi y Pamela por su cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias. A toda mi familia porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

A Marco por brindarme todos sus conocimientos, por guiarme siempre por el camino del bien y sobre todo por ser un gran ejemplo digno de admiración e inspiración como persona y como profesional para mí.

A Angelica por el apoyo incondicional que me ha brindado a lo largo de estos años universitarios, acompañándome siempre en cada paso y por estar siempre en las buenas y malas experiencias que me han enseñado a crecer como persona, como hijo, como amigo y como futuro profesional.

Es para mí de gran satisfacción poder dedicarles a ustedes el presente trabajo de tesis, que con mucho esfuerzo, dedicación y paciencia lo he conseguido.

AGRADECIMIENTO

Quiero agradecer a todas las personas que confiaron desde un principio en mi capacidad, conocimiento y fortaleza para poder culminar exitosamente el presente trabajo de titulación.

Agradezco a Marco por ser el eje fundamental durante toda mi formación académica brindándome siempre su apoyo, sus consejos y sus sabios conocimientos que con mucha paciencia siempre me brindó.

De igual manera mis agradecimientos a la Universidad de las Fuerzas Armadas ESPE por abrirme las puertas y permitir mi formación académica. De igual manera a todos los docentes que conforman la carrera de Ingeniería Mecatrónica, quienes con la enseñanza de sus valiosos conocimientos hicieron posible el desarrollo de este trabajo, gracias a cada uno de ustedes por su paciencia y dedicación.

Finalmente quiero expresar mi más grande y sincero agradecimiento al Ing. Darío Mendoza, que representó el pilar fundamental y principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo

Tabla de contenido

Certificación	2
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
agradecimiento	7
Índice de contenido	8
Índice de tablas	14
Índice de figuras	15
Resumen	19
abstract	20
Introducción	21
Antecedentes	21
Planteamiento del Problema	23
Descripción Resumida del Proyecto	24
<i>Estudio del Arte</i>	25
<i>Modelación Diseño y Análisis Estructural</i>	26
<i>Modelación y Simulación</i>	26
<i>Algoritmo de Visión Artificial</i>	27
<i>Algoritmo de Control</i>	27
<i>Entorno Servo Visual</i>	27
<i>Validación</i>	28
Justificación e Importancia	28
Objetivos.....	29
<i>Objetivo General</i>	29

Objetivos Específicos	29
Metas	30
Hipótesis	31
Variables De Investigación	31
Variables Independientes	31
Variables Dependientes	31
Marco teórico	32
Robots Móviles	32
Robots Omnidireccionales	33
Tipos de Ruedas Omnidireccionales	34
Rueda Universal	34
Rueda Mecanum	35
Estructuras de Robots Omnidireccionales	37
Omnidireccional de 3 Ruedas	37
Omnidireccional de 4 Ruedas	38
Omnidireccional de 4 Ruedas Mecanum	39
Locomoción Omnidireccional con Ruedas Mecanum	40
Cinemática del Robot Omnidireccional Mecanum	42
Aspectos Cinemáticos	42
Modelo Cinemático Robot Omnidireccional	43
Matriz de Transformación de Velocidades	45
Control de un Robot Omnidireccional	46
Control de Posición:	47
Seguimiento de Camino:	47
Control de Trayectoria:	48

Controladores Clásicos	48
<i>Control De Dos Posiciones ON/OFF.....</i>	<i>48</i>
<i>Control Proporcional.....</i>	<i>49</i>
<i>Control Proporcional Integral.....</i>	<i>50</i>
<i>Control Proporcional Derivativo</i>	<i>50</i>
<i>Control Proporcional - Integral – Derivativo.....</i>	<i>51</i>
Controladores Modernos.....	52
<i>Control por Lógica Difusa:</i>	<i>52</i>
<i>Control por Redes Neuronales:</i>	<i>52</i>
<i>Control Adaptativo:</i>	<i>53</i>
<i>Control Robusto:</i>	<i>53</i>
<i>Control No Lineal:.....</i>	<i>53</i>
Visión Artificial.....	53
Conceptos Generales.....	53
Modelo de Pinhole	53
Parámetros Intrínsecos de una Cámara	54
Parámetros Extrínsecos de una Cámara	55
Matriz de la Cámara	55
OpenCV	56
Calibración	56
ArUco	58
Marcadores	58
Proceso de Detección	59
Transformaciones 3D	61
Traslación	62

<i>Rotación</i>	63
<i>Ángulos de Euler</i>	64
Diseño y construcción de la plataforma omnidireccional	67
Parámetros de Diseño	67
Cargas de Diseño	68
Análisis Estático	69
Diseño Estructural	71
<i>Carga Máxima Soportada</i>	76
<i>Flexión Máxima</i>	77
<i>Factor de Seguridad</i>	77
<i>Análisis Estructural</i>	78
Teoría de Von Mises	78
Desplazamientos Resultantes	79
Factor de Seguridad	80
Selección de Componentes	81
<i>Ruedas Mecanum</i>	81
Motor	85
Torque	85
Potencia Requerida	86
<i>Driver Motor DC</i>	89
<i>Tarjeta de Desarrollo</i>	90
<i>Módulo de Comunicación</i>	92
<i>Fuente de Alimentación</i>	93
Construcción de la Plataforma Omnidireccional	96
Diseño Electrónico de la Plataforma Omnidireccional	97

<i>Sistema de Alimentación</i>	97
<i>Distribución de pines</i>	98
<i>Tarjeta de Control</i>	99
Modelo Cinemático con Punto de Operación Desplazado	101
Entorno servo visual	105
Adquisición de Imágenes	105
Calibración de la Cámara.....	108
Detección del Marcador ArUco	111
Estimación de Pose del Marcador Aruco	114
Estimación de Pose de la Plataforma Omnidireccional.....	118
<i>Traslación</i>	119
<i>Rotación</i>	121
<i>Ángulos de Euler</i>	123
Control de la Plataforma Omnidireccional	123
<i>Controlador Cinemático No Lineal</i>	124
<i>Controlador Cinemático No Lineal Discreto de Alto Nivel</i>	129
Pruebas y resultados	133
Estimación de la POSE	133
Interfaz Gráfica.....	135
Evaluación de Algoritmos de Control Moderno.....	136
<i>Trayectoria 1</i>	136
<i>Trayectoria 2</i>	140
<i>Trayectoria 3</i>	143
<i>Trayectoria 4</i>	146
<i>Análisis de Robustez</i>	149

<i>Punto de Operación Desplazado</i>	152
<i>Implementación del Controlador Clásico</i>	154
Validación de la Hipótesis	159
Conclusiones y recomendaciones	165
Conclusiones	165
Recomendaciones	167
Bibliografía	169

Índice de tablas

Tabla 1 <i>Comparación entre las Configuraciones de Robots Omnidireccionales</i>	40
Tabla 2 <i>Parámetros de Diseño</i>	68
Tabla 3 <i>Propiedades Mecánicas MDF</i>	73
Tabla 4 <i>Tensiones de Von Mises</i>	79
Tabla 5 <i>Especificaciones Técnicas de la Rueda Mecanum</i>	83
Tabla 6 <i>Especificaciones Técnicas Motor CQRobot 70:1 37D 70L</i>	87
Tabla 7 <i>Especificaciones Técnicas Cytron MDD10A</i>	90
Tabla 8 <i>Características Principales de la Tarjeta de Desarrollo STM32 Nucleo-F446RE</i>	91
Tabla 9 <i>XBEE Pro S2C</i>	93
Tabla 10 <i>Especificaciones Técnicas de la Batería Li-Po.</i>	94
Tabla 11 <i>Especificaciones Cámara Posterior Huawei P30 Lite</i>	107
Tabla 12 <i>Resultados de la Validación de la POSE</i>	134
Tabla 13 <i>Constantes PID</i>	155
Tabla 14 <i>Pruebas de Funcionamiento Trayectoria 1</i>	160
Tabla 15 <i>Pruebas de Funcionamiento Trayectoria 2</i>	160
Tabla 16 <i>Pruebas de Funcionamiento Trayectoria 3</i>	161
Tabla 17 <i>Pruebas de Funcionamiento Trayectoria 4</i>	161
Tabla 18 <i>Datos para Validación de Hipótesis</i>	162

Índice de figuras

Figura 1 <i>Diagrama del Sistema Propuesto</i>	25
Figura 2 <i>Robot Omnidireccional Mecanum</i>	34
Figura 3 <i>Componentes de una Rueda Universal</i>	35
Figura 4 <i>Rueda Omnidireccional Mecanum</i>	36
Figura 5 <i>Robot Omnidireccional de 3 Ruedas</i>	37
Figura 6 <i>Robot Omnidireccional de 4 Ruedas</i>	38
Figura 7 <i>Robot Omnidireccional de 4 Ruedas Mecanum</i>	39
Figura 8 <i>Principales Movimientos de la Plataforma Omnidireccional</i>	41
Figura 9 <i>Representación Esquemática del Robot Omnidireccional</i>	43
Figura 10 <i>Velocidades Lineales del Robot en el Plano</i>	44
Figura 11 <i>Configuración de la Plataforma Omnidireccional</i>	46
Figura 12 <i>Diagrama de Control ON/OFF</i>	49
Figura 13 <i>Diagrama del Control Proporcional</i>	49
Figura 14 <i>Diagrama del Control Proporcional Integral</i>	50
Figura 15 <i>Diagrama del Control Proporcional Derivativo</i>	51
Figura 16 <i>Diagrama del Control PID</i>	51
Figura 17 <i>Modelo de cámara pinhole</i>	54
Figura 18 <i>Resultados del Proceso de Calibración de la Cámara</i>	57
Figura 19 <i>Marcador Estándar ArUco</i>	58
Figura 20 <i>Detección de un Marcador ArUco</i>	60
Figura 21 <i>Sistemas de Coordenadas Marcador - Cámara</i>	61
Figura 22 <i>Transformación de Traslación</i>	62
Figura 23 <i>Transformación de Rotación</i>	63
Figura 24 <i>Ángulos de Euler</i>	64

Figura 25	<i>Diseño CAD Robot Omnidireccional</i>	67
Figura 26	<i>Distribución de Cargas en la Plataforma Omnidireccional</i>	69
Figura 27	<i>Diagrama de Cuerpo Libre</i>	70
Figura 28	<i>Dimensiones de la Base Superior</i>	71
Figura 29	<i>Cargas Distribuidas en la Base</i>	72
Figura 30	<i>Cargas Puntuales en la Base</i>	72
Figura 31	<i>Diagrama de Fuerza Cortante y Momento Flexionante</i>	75
Figura 32	<i>Desplazamientos Resultantes</i>	80
Figura 33	<i>Factor de Seguridad</i>	80
Figura 34	<i>Error de Distribución de Cargas</i>	82
Figura 35	<i>Rueda Omnidireccional Mecanum 4 in</i>	83
Figura 36	<i>Fuerzas Presentes en la Rueda</i>	85
Figura 37	<i>Motor CQRobot 70:1 37D 70L</i>	87
Figura 38	<i>Cytron MDD10A</i>	89
Figura 39	<i>Tarjeta de Desarrollo STM32 Nucleo-F446RE</i>	91
Figura 40	<i>XBEE Pro S2C</i>	92
Figura 41	<i>Zeee Batería Li-Po 4S</i>	95
Figura 42	<i>Ensamble de la Estructura</i>	96
Figura 43	<i>Ensamble de Motores</i>	96
Figura 44	<i>Ensamble de las Ruedas</i>	97
Figura 45	<i>Diagrama de Alimentación</i>	97
Figura 46	<i>Distribución de Pines STM32 NUCLEO-F446RE</i>	98
Figura 47	<i>Diagrama Esquemático del Circuito Electrónico</i>	99
Figura 48	<i>Circuito PCB de la Tarjeta de Control</i>	100
Figura 49	<i>Visualización 3D del PCB</i>	100

Figura 50	Representación Esquemática de la Plataforma Omnidireccional	101
Figura 51	<i>Representación Cinemática con Punto de Operación Desplazado</i>	103
Figura 52	<i>Etapas del Sistema de Visión Artificial</i>	105
Figura 53	<i>Cámaras Huawei P30 Lite</i>	106
Figura 54	<i>Tablero de Ajedrez</i>	108
Figura 55	<i>Imágenes para Calibración de la Cámara</i>	109
Figura 56	<i>Imagen Distorsionada vs Imagen Corregida</i>	110
Figura 57	<i>Marcador ArUco ID 3</i>	112
Figura 58	<i>Detección del Marcador ArUco</i>	113
Figura 59	<i>Sistema de Coordenadas de la Marcador - Cámara</i>	114
Figura 60	<i>Estimación de la POSE de un Marcador ArUco</i>	117
Figura 61	<i>Sistema Servo Visual Propuesto</i>	119
Figura 62	<i>Estimación de Pose de la Plataforma Omnidireccional</i>	121
Figura 63	<i>Diagrama de Bloques del Sistema Servo Visual</i>	124
Figura 64	<i>Diagrama de Bloques del Sistema de Control No Lineal</i>	125
Figura 65	<i>Diagrama de Bloques del Sistema de Control No Lineal Discreto</i>	130
Figura 66	<i>Validación de Datos de POSE Estimados</i>	134
Figura 67	<i>Interfaz Gráfica</i>	135
Figura 68	<i>Trayectoria 1 Elipse</i>	137
Figura 69	<i>Resultados del Seguimiento de una Elipse</i>	137
Figura 70	<i>Resultados Combinados del Seguimiento de la Elipse</i>	138
Figura 71	<i>Señales de Control de Control</i>	138
Figura 72	<i>Errores de Control</i>	139
Figura 73	<i>Trayectoria 2 Infinito</i>	140
Figura 74	<i>Resultados del Seguimiento de la Trayectoria 2</i>	141

Figura 75 <i>Resultados Combinados del Seguimiento de la Trayectoria 2</i>	141
Figura 76 <i>Señales de Control de Control</i>	142
Figura 77 <i>Errores de Control</i>	142
Figura 78 <i>Trayectoria 3 Espiral</i>	143
Figura 79 <i>Resultados del Seguimiento de la Trayectoria 3</i>	144
Figura 80 <i>Resultados Combinados del Seguimiento de la Trayectoria 3</i>	144
Figura 81 <i>Señales de Control de Control</i>	145
Figura 82 <i>Errores de Control</i>	146
Figura 83 <i>Trayectoria 4 - Mariposa</i>	147
Figura 84 <i>Resultados del Seguimiento de la Trayectoria 4</i>	147
Figura 85 <i>Resultados Combinados del Seguimiento de la Trayectoria 4</i>	148
Figura 86 <i>Señales de Control de Control</i>	148
Figura 87 <i>Errores de Control</i>	149
Figura 88 <i>Respuesta Ante Perturbaciones</i>	150
Figura 89 <i>Trayectorias Recorridas</i>	150
Figura 90 <i>Señales de Control</i>	151
Figura 91 <i>Errores de Control</i>	152
Figura 92 <i>Seguimiento con Punto de Operación Desplazado</i>	152
Figura 93 <i>Resultados Obtenidos</i>	153
Figura 94 <i>Señales de Control</i>	153
Figura 95 <i>Errores de Control</i>	154
Figura 96 <i>Control PID de Trayectoria</i>	156
Figura 97 <i>Resultados Obtenidos PID - Discreto</i>	157
Figura 98 <i>Señales de Control PID</i>	157
Figura 99 <i>Errores PID -Discreto</i>	158

RESUMEN

En el presente trabajo de titulación se realiza el diseño y la construcción de una plataforma omnidireccional con el objetivo de evaluar la eficiencia y robustez de los algoritmos de control modernos en el seguimiento de trayectorias a través de la implementación de un sistema servovisual que permite la estimación de la POSE por medio de marcadores binarios ArUco disponibles en la librería de visión artificial OpenCV. Por medio de la investigación preliminar se optó por la configuración Mecanum debido a sus características destacadas frente a otras configuraciones. Los resultados obtenidos fueron contrastados con un algoritmo de control clásico en donde se evalúan las gráficas resultantes de las señales de control y de los errores de control, permitiendo así determinar si el sistema es estable, inestable o asintóticamente estable. El algoritmo de visión artificial fue implementado y compilado en el IDE Qtcreator, el cual hizo posible el desarrollo de una interfaz gráfica que permite al usuario mover la plataforma omnidireccional de forma manual, modificar los parámetros del sistema como son: el desfase del punto de operación tanto de posición como de orientación, seleccionar y configurar la trayectoria deseada, definir el tiempo de implementación del seguimiento de trayectoria, escoger el controlador que se desea implementar, modificar la matriz de ganancia y de peso de los controladores modernos, ubicar el punto de origen del sistema de referencia, visualizar la posición y orientación actual del robot y por último ejecutar el control de trayectoria. Por último, se realiza la validación de la hipótesis por medio la prueba del chi cuadrado el cual permitió afirmar que el sistema servovisual propuesto si permite la evaluación de eficiencia y robustez en el seguimiento de trayectorias.

Palabras Clave:

- **PLATAFORMA OMNIDIRECCIONAL**
- **SISTEMA SERVO VISUAL**
- **CONTROL MODERNO MIMO NO LINEAL**
- **MARCADORES ARUCO**

ABSTRACT

In the present degree work, the design and construction of an omnidirectional platform is carried out with the aim of evaluating the efficiency and robustness of modern control algorithms in the tracking of trajectories through the implementation of a servo-visual system that allows estimation from POSE by means of ArUco binary markers available in the OpenCV artificial vision library. Through preliminary research, the Mecanum configuration was chosen due to its outstanding characteristics compared to other configurations. The results obtained were contrasted with a classical control algorithm where the graphs resulting from the control signals and the control errors are evaluated, thus allowing to determine if the system is stable, unstable or asymptotically stable. The artificial vision algorithm was implemented and compiled in the Qtcreator IDE, which made possible the development of a graphical interface that allows the user to move the omnidirectional platform manually, modify the system parameters such as: the operation point offset both position and orientation, select and configure the desired trajectory, define the implementation time of the trajectory tracking, choose the controller to be implemented, modify the gain and weight matrix of modern controllers, locate the point of origin of the reference system, display the current position and orientation of the robot and finally execute the path control. Finally, the validation of the hypothesis is carried out by means of the chi square test, which includes affirming that the proposed servo-visual system does allow the evaluation of efficiency and robustness in the tracking of trajectories.

Keywords:

- **OMNIDIRECTIONAL PLATFORM**
- **SERVO VISUAL SYSTEM**
- **MODERN NON-LINEAR MIMO CONTROL**
- **ARUCO MARKERS**

Capítulo I

1. Introducción

1.1. Antecedentes

El uso de robots móviles de manera cotidiana ha ido evolucionando progresivamente en los últimos años. En la actualidad se han implementado distintos tipos de robots móviles que a su vez han contribuido al desarrollo de nuevas tecnologías, así como también, la construcción de los mismos ha permitido tener una mejor interacción entre el campo virtual o simulado, y el ambiente real en donde se puede tener una mejor perspectiva del funcionamiento y operación de la robótica móvil. (Salas, 2015)

Una plataforma móvil omnidireccional se caracteriza principalmente por su alta maniobrabilidad en el plano, lo cual significa que posee la libertad de moverse a cualquier punto referencial sin la necesidad de reorientarse. Dicho desplazamiento característico se logra a través de un movimiento sincronizado de las ruedas que comandan el robot móvil, gracias a la electrónica encargada del correcto manejo de señales enviadas a los actuadores. Este tipo específico de robot móvil ha venido siendo objeto de estudio en un rango amplio de aplicaciones debido a su versatilidad en el desarrollo de actividades tanto en el ámbito industrial como en el entorno de investigación. (Peña-Cabrera et al., 2016)

Los sistemas de visión artificial han tenido un amplio crecimiento en las últimas décadas dando paso al desarrollo de un sin número de aplicaciones, entre ellas está incluida la automatización en la navegación móvil eliminando esa dependencia de la geometría del espacio para obtener resultados aceptables. El control servo visual aplicado en robots móviles, hace referencia a la adquisición de datos por medio del

análisis de imágenes utilizando técnicas de visión por computador, con el objetivo de extraer un estimado de la POSE del robot en el plano real para así, mantener un control óptimo en su movimiento. (Arévalo-Vázquez et al., 2015)

Una de las aplicaciones más importantes en la actualidad es la planificación de trayectorias del robot móvil, donde se pretende automatizar la navegación del robot a través de la implementación de distintos lazos de control, donde con mucha frecuencia se lo realiza a través de odometría debido a que se basa únicamente en el estudio de sensores de cuadratura digitales permitiendo así obtener la posición y orientación respecto al plano a un coste computacional bajo. (Cortés et al., 2015). Sin embargo:

La desventaja principal de este método es que si por algún motivo una o más ruedas pierden contacto con el suelo el cálculo odométrico pierde precisión y exactitud lo que se traduce en un error acumulativo en la posición y orientación del robot (Hernández-Paredes et al., 2016, p. 1).

Adicionalmente, el sistema que representa la plataforma omnidireccional es conocido por su característica dinámica no lineal, lo cual dificulta el control del mismo a través de la implementación de los controladores clásicos fundamentados en controladores PID:

Todas las estrategias mencionadas se basan principalmente en la aplicación de controladores PID a lazos simples, linealizando previamente el sistema entorno a un punto de trabajo, lo que supone una aproximación lineal; ya que el modelo de las calderas es no lineal. Estos métodos vienen siendo el estándar de aplicación en la industria de los últimos años. Sin embargo, en muchos casos resultan ser ineficientes provocando grandes consumos de combustible y contaminación (Benites, 2015, p. 5).

1.2. Planteamiento del Problema

En la actualidad la robótica móvil está incluida en la línea de investigación como es: Automática y Robótica, la misma que se ha convertido en una importante fuente de investigación debido a las extensas aplicaciones que tiene en la industria, para lo cual, surge la necesidad de implementar algoritmos de control modernos que permitan evaluar el desempeño del robot móvil en la tarea que fue asignado, debido a que los algoritmos de control clásicos resultan ser poco confiables al intentar controlar este particular sistema no lineal debido a la dinámica del mismo.

La mayoría de estudios que se puede encontrar en el área de la Robótica Móvil están basados en la configuración del tipo diferencial; Sin embargo, la baja maniobrabilidad en el campo representa la principal desventaja de este tipo de configuración, dado que el mismo requiere de una reorientación respecto a su eje central para realizar un cambio de dirección o sentido al momento de querer desplazarse de un punto a otro, limitando considerablemente la cantidad de aplicaciones y tareas que se podrían desarrollar con este tipo específico de configuración tanto en el área industrial como en el de investigación.

Uno de los problemas que frecuentemente se encuentran en la implementación de robots omnidireccionales es el error de estimación de la POSE del robot, debido a que de manera usual se recurre al estudio de su odometría, misma que se basa en el análisis de la información extraída de los encoders acoplados a las llantas; Sin embargo, la fiabilidad de estos datos está internamente ligado al plano en el que circula el robot móvil, puesto que cualquier perturbación o anomalía en el plano puede causar la pérdida de pulsos en el proceso de lectura del encoder. Aun cuando se decida trabajar sobre una superficie netamente plana, el problema seguirá proviniendo de la

lectura de los sensores de cuadratura, puesto que estos no son capaces de detectar en su totalidad el desplazamiento que vienen realizando las ruedas omnidireccionales.

Actualmente no se han desarrollado trabajos que hagan énfasis en el estudio de la planificación de trayectorias de los robots móviles, campo que permitiría automatizar la navegación eficiente del robot a través de la implementación de distintos lazos de control.

Actualmente se encuentra en pleno desarrollo la tecnología basada en visión artificial; No obstante, son pocos los estudios que se han realizado fusionando esta tecnología con la robótica móvil, es decir, implementando técnicas de visión por computador para estimar la POSE y la orientación de un robot móvil con el objetivo de extraer información más precisa y confiable del campo visual.

Tomando de los trabajos relacionados, se demuestra que funcionan únicamente bajo un ambiente estructurado donde la cámara se encuentra en una posición fija necesaria para la correcta parametrización de los píxeles del área de imagen, y al trabajar sobre formato RGB, este es susceptible a los cambios de las condiciones de luminosidad, por lo cual necesitan configurar correctamente los valores de transparencia para cada identificador de color y trabajar bajo condiciones de luminosidad fijas.

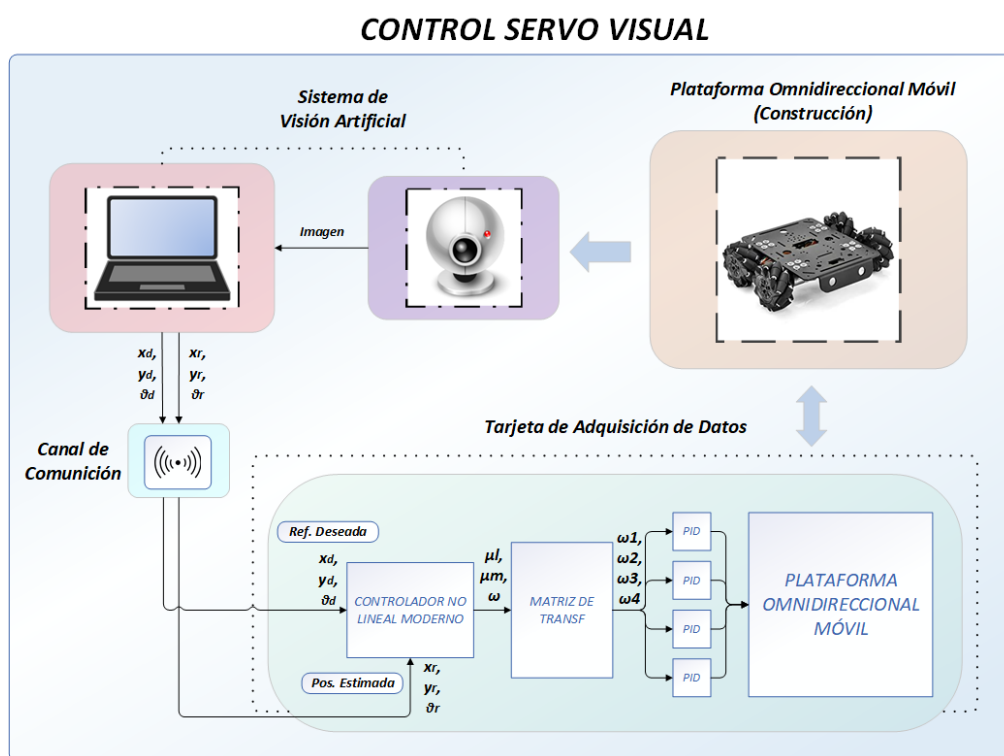
1.3. Descripción Resumida del Proyecto

El presente proyecto tiene como objetivo el diseño y construcción de un sistema servo visual, el cual pretende realizar el control de una plataforma omnidireccional móvil a través de la evaluación de algoritmos de control modernos que permitan disminuir el error de convergencia del robot con respecto a una trayectoria predefinida, por medio de la estimación de la posición de la plataforma haciendo uso de algoritmos de visión artificial con el objetivo de eliminar la dependencia de la geometría del plano en el que

circula el robot móvil. Dicho sistema se caracteriza principalmente por ser un sistema no lineal en la cual evaluará distintos controladores modernos para comparar su desempeño y eficacia en la resolución del problema propuesto. La propuesta del trabajo de titulación se representa en la Figura 1.

Figura 1

Diagrama del Sistema Propuesto



Nota. El gráfico representa un esquema resumido del sistema servovisual propuesto.

Detallando el proyecto de investigación, se lo puede dividir en las siguientes etapas:

1.3.1. Estudio del Arte

Se realizará una investigación bibliográfica el cual permitirá realizar un análisis preliminar donde se tomará en consideración los distintos tipos y configuraciones de

robots omnidireccionales, con el objetivo de tener la información suficiente como para dar paso a una selección del tipo de configuración que permitirá resolver el problema del seguimiento de trayectoria de manera óptima.

1.3.2. Modelación Diseño y Análisis Estructural

En esta etapa se considerará los resultados obtenidos a través de la investigación preliminar y la selección previa del tipo de configuración de la plataforma omnidireccional a implementar, donde se realizará el modelamiento, análisis estructural y construcción de la plataforma omnidireccional, haciendo uso de las prestaciones de un software CAD/CAE que permita tener una mejor proyección visual de la plataforma omnidireccional y a su vez, permita realizar un análisis estructural para evitar fallos debido a fractura, fatiga y sobredimensionamiento de la plataforma.

Posterior al análisis mencionado se procederá a la selección de componentes tanto electrónicos como mecánicos en función de la relación costo-beneficio, finalmente se procederá a la construcción de la plataforma omnidireccional.

1.3.3. Modelación y Simulación

Dicho proyecto parte del análisis de la plataforma omnidireccional móvil haciendo énfasis en los distintos tipos de configuraciones, con el objetivo de hacer una selección minuciosa de la variante que permita desarrollar las aplicaciones propuestas con mayor versatilidad, comodidad y robustez. Por lo que como resultado de la investigación previa se tendrá la información suficiente como para obtener el modelo matemático que describe el comportamiento del sistema, el mismo que será objeto de estudio a través de una simulación en un software que permita analizar y evaluar la respuesta del sistema.

1.3.4. Algoritmo de Visión Artificial

En esta etapa se desarrollará el algoritmo de visión artificial que permitirá estimación de la POSE en el plano real haciendo uso de técnicas de visión por computador en conjunto con filtros complementarios para eliminar el ruido en el proceso de estimación, con el fin de obtener datos más precisos y reales con el propósito de evitar posibles errores en la fase de control de la plataforma omnidireccional, tomando en cuenta que no será necesario un entorno controlado, pero sí deberá cumplir con condiciones mínimas aceptables de trabajo. Para ello, se hará uso de un software que contribuya al desarrollo, compilación y validación del algoritmo de visión artificial.

1.3.5. Algoritmo de Control

La Figura 1 indica el sistema propuesto del cual se puede destacar la implementación de un controlador MIMO avanzado, el cual deberá tomar en consideración como entrada el punto y orientación deseado en el instante actual y, a su vez, la posición y orientación estimada de la plataforma omnidireccional obtenida a través del algoritmo de visión artificial. Dicho controlador evaluará los parámetros mencionados y como resultado a su salida se podrá encontrar las velocidades lineales y angulares que permitirá ejecutar de forma autónoma la traslación del sistema conforme con la referencia predefinida.

1.3.6. Entorno Servo Visual

El diagrama del sistema propuesto es representado en la Figura 1, donde da inicio con la captación de video a través de una cámara para posteriormente ser analizado por el algoritmo de visión artificial, con el propósito de estimar la POSE de la plataforma omnidireccional, la misma que será enviado conjuntamente con las referencias deseadas a la entrada del controlador MIMO por medio de un canal de comunicación inalámbrico. El controlador se encargará de realizar el análisis de sus

entradas y proporcionar las velocidades lineales y angulares, las mismas que serán descompuestas a través de una matriz de transformación en las velocidades angulares de cada motor, donde se implementará un algoritmo de control PID para cada uno de los motores con el propósito de que el mismo alcance la referencia y elimine las posibles perturbaciones debido a la geometría del plano, logrando así, que el robot móvil cumpla con las tareas de locomoción y seguimiento de trayectoria deseado.

1.3.7. Validación

Se evaluarán los resultados obtenidos a partir de los errores de control en el seguimiento de trayectoria de la plataforma omnidireccional móvil bajo la implementación de controladores modernos. Estos resultados serán contrarrestados con los obtenidos en el mismo sistema mencionado, pero con la implementación de un controlador clásico como es el PID.

1.4. Justificación e Importancia

Los sistemas didácticos de marcas comerciales utilizados en el área de la Automática y la Robótica como plantas de pruebas son muy costosos y en general de prestaciones limitadas en cuanto a la interacción entre el investigador y la planta como tal, puesto que no permiten un acceso total a su código de programación impidiendo al desarrollador tener el control de la planta en su totalidad.

Por tal motivo el presente proyecto de titulación plantea el diseño y construcción de una plataforma móvil omnidireccional "Open Source" susceptible a modificaciones, mejoras y expansiones. El cual permitirá evaluar la eficiencia de controladores modernos en referencia a los controladores clásicos, a través del análisis de los errores de control producidos en el seguimiento de trayectoria, haciendo énfasis en el estudio de su posición y orientación durante todo el recorrido.

La plataforma omnidireccional móvil intentará resolver el problema del seguimiento de trayectoria a través de la implementación de algoritmos de control MIMO avanzado, los mismos que contarán como principal fuente de retroalimentación el sistema de visión artificial, el cual permitirá evaluar la eficacia de un sistema servo visual por medio del estudio de la estabilidad en la estimación de la POSE para robots móviles.

La importancia del presente proyecto yace en la contribución al desarrollo de tecnologías basadas en sistemas servo visuales enfocados al área de la robótica móvil, puesto que en la actualidad son pocos los estudios realizados que implementen las técnicas de estimación de POSE a través de algoritmos de visión artificial con el objetivo realizar tareas de locomoción de manera óptima, eficiente y que a su vez ayude a solventar la demanda de soluciones más precisas, eficaces y de sencillo manejo de la industria en la actualidad.

1.5. Objetivos

1.5.1. Objetivo General

- Diseñar y construir una plataforma omnidireccional para evaluar controladores modernos en el seguimiento de trayectoria utilizando visión artificial

1.5.2. Objetivos Específicos

- Investigar acerca de las plataformas omnidireccionales móviles: tipos, variantes, componentes electrónicos y mecánicos; a través de fuentes bibliográficas para realizar un análisis y selección preliminar del tipo de configuración a implementar.
- Realizar el diseño mecánico y electrónico de la plataforma omnidireccional usando herramientas de diseño electrónico y CAD/CAE.

- Construir la plataforma omnidireccional móvil mediante el análisis y el estudio previo realizado.
- Obtener el modelo matemático que describa el comportamiento de la plataforma omnidireccional móvil para evaluar la respuesta del sistema.
- Implementar el entorno servo visual a través del desarrollo de un algoritmo de visión artificial que permita estimar la POSE de la plataforma omnidireccional y a su vez evaluar la estabilidad de los datos obtenidos.
- Analizar la eficiencia y robustez de 2 algoritmos de control MIMO avanzados en el seguimiento de trayectoria de la plataforma omnidireccional móvil con la finalidad de evaluar el comportamiento de los errores de control.
- Contrarrestar los resultados obtenidos a través algoritmos de control moderno con un algoritmo de control PID con el propósito de verificar la eficiencia de los algoritmos de control moderno frente a los algoritmos de control clásicos.

1.6. Metas

- Investigación y estudio preliminar de las plataformas omnidireccionales para realizar una selección del tipo de configuración que será objeto de estudio en el primer mes.
- Realización del diseño mecánico y electrónico de la plataforma omnidireccional en el segundo mes de trabajo.
- Construcción de la plataforma omnidireccional móvil en las 2 primeras semanas del tercer mes de trabajo.
- Obtención del modelo matemático del sistema en las 2 últimas semanas del tercer mes de trabajo.
- Implementación del entorno servo visual en el cuarto mes de trabajo.

- Evaluación de la eficiencia y robustez de 2 algoritmos de control MIMO avanzados en el seguimiento de trayectoria de la plataforma omnidireccional móvil en el quinto mes.
- Validación de resultados a través de la comparativa entre los controladores modernos y un algoritmo de control PID en el sexto mes de trabajo.

1.7. Hipótesis

¿El diseño y construcción de una plataforma omnidireccional permitirá evaluar la eficiencia y robustez de los controladores modernos en el seguimiento de trayectoria a través de la estimación de POSE por medio de visión artificial?

1.8. Variables De Investigación

1.8.1. Variables Independientes

Controlador moderno aplicado a la plataforma omnidireccional

1.8.2. Variables Dependientes

Eficiencia y robustez en el seguimiento de la trayectoria del sistema.

Capítulo II

2. Marco teórico

El presente capítulo está enfocado a exponer de manera resumida el estado del arte referente a las principales variables de investigación del proyecto, las mismas que tratan acerca de; robótica móvil, plataforma omnidireccional, controladores modernos y visión artificial. Dichos componentes que forman parte del estado del arte permitirán una mayor comprensión acerca de los componentes básicos de un sistema de control servovisual y su influencia en la resolución del problema de seguimiento de trayectorias.

2.1. Robots Móviles

Los robots móviles se definen como: “un sistema electromecánico capaz de desplazarse de manera autónoma sin estar sujeto físicamente a un solo punto. Posee sensores que permiten monitorear a cada momento su posición relativa a su punto de origen y a su punto de destino” (Barrientos Sotelo et al., 2007, p. 14).

Gracias a la automatización de procesos se ha podido desarrollar robots con capacidades sensoriales y de locomoción que permiten el desarrollo de tareas sin necesidad de la intervención humana, es decir, dichos frutos de la robótica poseen la aptitud para cumplir una o varias tareas con absoluta autonomía.

Los robots móviles disponen de un sistema de locomoción que puede estar constituido por ruedas, patas u orugas, dichos mecanismos son utilizados para lograr un desplazamiento de un punto a otro en diferentes entornos, condiciones y superficies de trabajo, permitiendo al robot móvil alcanzar el objetivo deseado, como por ejemplo, el transporte autónomo de mercadería en las diferentes cadenas de producción en el ambiente industrial, la exploración en ambientes hostiles o de difícil acceso y la navegación autónoma en áreas distantes (Peña-Cabrera et al., 2016).

Actualmente existe un mayor desarrollo científico referente al sistema de locomoción por ruedas, debido a las ventajas que ofrece respecto a las patas y orugas. Los Robots Móviles con Ruedas (RMR), son más eficientes al desplazarse por superficies lisas, debido a que no consumen tanta energía y a su vez no producen un desgaste en la superficie donde se mueven. Además, los RMR requieren de un menor número de partes y un diseño mecánico relativamente más sencillo, lo cual facilita su diseño, construcción e implementación.

2.2. Robots Omnidireccionales

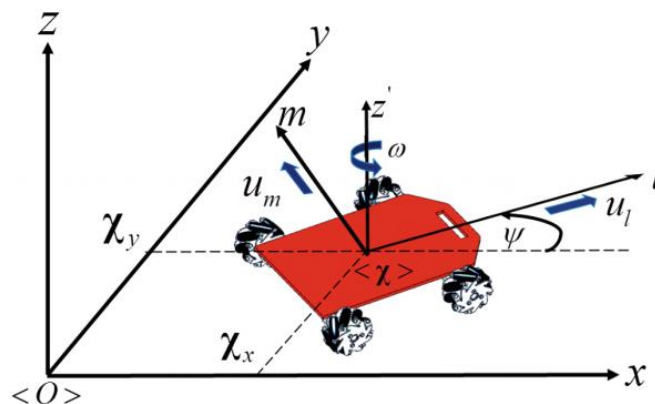
Un robot omnidireccional es aquel que tiene la capacidad de desplazarse de un punto a otro con total libertad sin tener la necesidad de hacer una reorientación previa al movimiento. Se define también como robot holonómico y se caracteriza por tener 3 grados de libertad para movilizarse en el plano como se muestra en la

Figura 2. Existen varios tipos de configuraciones de robots omnidireccionales, sin embargo, la configuración más simple requiere de por lo menos 3 ruedas especiales para lograr este característico movimiento, que a su vez debe acompañarse de control cinemático un poco complejo debido a los posibles deslizamientos resultantes (Álvarez, 2019).

Los robots omnidireccionales poseen una gran ventaja al tener más grados de libertad respecto a lo habitual, puesto que la eficiencia de un robot móvil al realizar una tarea depende en su mayor parte de la movilidad que presente en el ámbito aplicativo en donde el entorno de trabajo puede venir acompañado de ciertos desafíos como, por ejemplo; evitar obstáculos o pasar por un camino estrecho en el que se requiera gran maniobrabilidad.

Figura 2

Robot Omnidireccional Mecanum



Nota. La figura es una representación esquemática donde se puede apreciar sus 3 grados de libertad. Tomado de (Andaluz et al., 2017)

2.3. Tipos de Ruedas Omnidireccionales

Este tipo de ruedas se las puede encontrar más presentes en el área de la robótica, son ruedas especiales con mecanismos ingeniosos que permiten ejercer velocidades en varias direcciones permitiendo al robot realizar movimientos traslacionales, rotacionales o ambos a la vez (A. Ortega, 2019).

Existen varios tipos de ruedas omnidireccionales que por consiguiente permiten tener más de una configuración omnidireccional con características diferentes, pero que cumplen con la misma función. A continuación, se presenta un breve resumen de cada una de ellas:

2.3.1. Rueda Universal

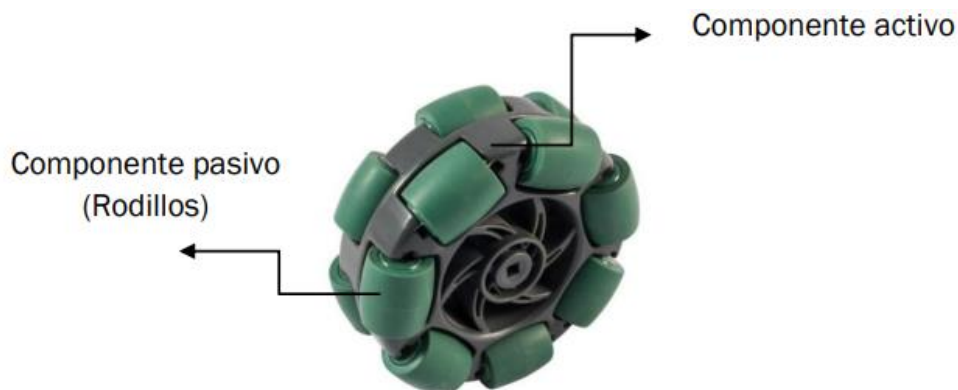
Este tipo especial de rueda omnidireccional provee una combinación entre un movimiento restringido y no restringido durante el giro. Está constituido por rodillos que se ensamblan al eje de rotación en dirección perpendicular al mismo, de este modo la

rueda no solo puede rotar en sentido perpendicular a su eje sino también sentido paralelo (J. F. Ortega & Yapó, 2017).

Se caracterizan por estar provistos por una combinación entre un componente activo y un componente pasivo que conjuntamente permiten aumentar los grados de libertad de la rueda y por ende los del robot omnidireccional. La rueda universal consigue trayectorias rotacionales, laterales y frontales sobre la una misma superficie plana, y aunque en el mercado se pueden encontrar una gran variedad de configuraciones y diseños de este tipo de rueda omnidireccional, todas se basan en el mismo principio de inclusión de rodillos en el contorno de la rueda (Reboto & Rodríguez, 2016).

Figura 3

Componentes de una Rueda Universal



Nota. La figura representa las partes que constituyen una rueda universal omnidireccional. Tomado de (Reboto & Rodríguez, 2016).

2.3.2. Rueda Mecanum

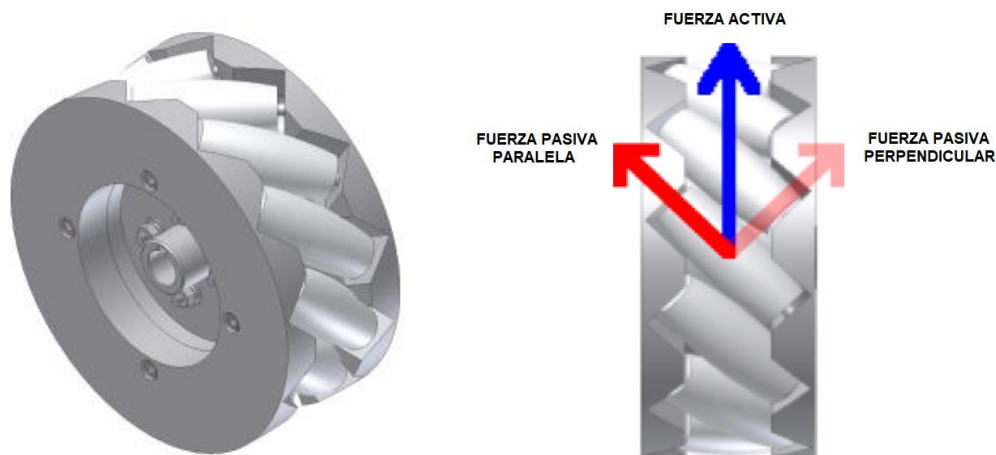
La rueda Mecanum es un tipo de rueda omnidireccional muy especial similar a la rueda mostrada en la Figura 3 con la única diferencia de que en este diseño en

particular los rodillos están ubicados a cierto ángulo respecto al eje del motor, habitualmente dicho ángulo es $\pm 45^\circ$ y los rodillos al igual que en la configuración universal se ubican por todo el contorno de la rueda (Reboto & Rodríguez, 2016).

El principio de funcionamiento de la rueda omnidireccional Mecanum se basa en la fuerza que ejercen los rodillos pasivos con desfase angular, esta componente de fuerza tiene la misma desviación angular que los rodillos ubicados en la rueda, es decir, la fuerza ejercida fluye en dirección paralela a los rodillos, puesto que la fuerza en dirección perpendicular solo sirve para hacer girar los rodillos pasivos. Sin embargo, esta configuración omnidireccional requiere un control de rotación de al menos 4 ruedas para proveer de los 3 grados de libertad al robot omnidireccional (Woods, 2006).

Figura 4

Rueda Omnidireccional Mecanum



Nota. La figura representa la rueda omnidireccional Mecanum y las componentes de fuerza que aparecen al ejercer rotación en la rueda. Adaptado de (Woods, 2006).

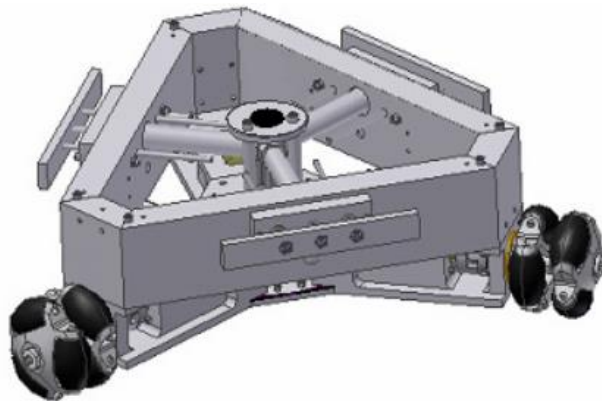
2.4. Estructuras de Robots Omnidireccionales

Una de las difíciles decisiones que se debe tomar es la cantidad de ruedas que se va a implementar en el robot móvil, es decir, que tipo de estructura omnidireccional se va a construir. Las alternativas que comúnmente se puede encontrar son las configuraciones de 3 o 4 ruedas, sin embargo, cada configuración tiene sus ventajas y desventajas. Los robots móviles provistos de 3 ruedas son de mecánica y control relativamente simple pero tienen tracción y estabilidad limitada, por otro lado los robots móviles de 4 ruedas se caracterizan por ser de mecánica y control complejo pero ofrecen mayor tracción y estabilidad (Lalama, 2014).

2.4.1. Omnidireccional de 3 Ruedas

Figura 5

Robot Omnidireccional de 3 Ruedas



Nota. La figura simboliza la configuración triangular de un robot omnidireccional de 3 ruedas universales dobles. Tomado de (Tejera, 2009).

Esta estructura omnidireccional dispone de 3 ruedas universales tal y como se muestra en la Figura 5, en su gran mayoría adoptan una forma triangular por lo que ahí nace su principal desventaja puesto que al tener apenas 3 ruedas no provee un sistema

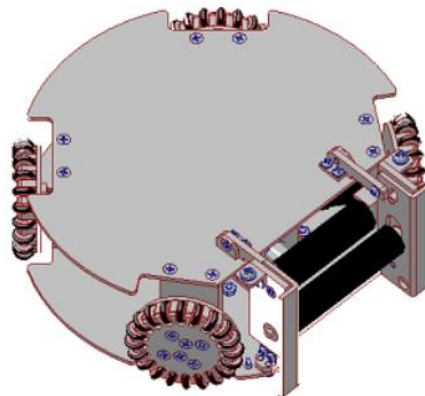
estable es decir, no garantizan un movimiento en línea recta, además proporciona una baja tracción al moverse en el plano, sin embargo, es una buena opción para motivos académicos puesto su diseño mecánico y el control son simples y fáciles de implementar (Pérez, 2017).

2.4.2. Omnidireccional de 4 Ruedas

Esta configuración tiene mucha similitud con la configuración de 3 ruedas, puesto que prácticamente utiliza las mismas ruedas omnidireccionales universales tal y como se indica en la Figura 6. Este tipo de configuración es más estable que el mencionado anteriormente, debido a que posee la ventaja de redundancia, es decir, existen varias combinaciones de velocidades posibles para realizar cualquier movimiento en la superficie plana, añadiendo que su estructura permite la detección de deslizamiento de forma que permita dar solución a dicho problema. Sin embargo, la desventaja que trae esta configuración yace en la falta de tracción al moverse en el plano (Pérez, 2017).

Figura 6

Robot Omnidireccional de 4 Ruedas



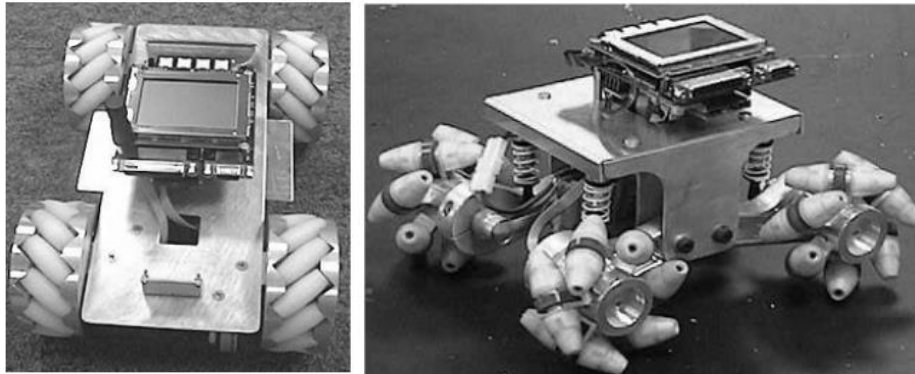
Nota. El presente gráfico representa la configuración omnidireccional con 4 ruedas universales. Tomado de (Tejera, 2009).

2.4.3. Omnidireccional de 4 Ruedas Mecanum

EL robot omnidireccional de 4 ruedas Mecanum es un tipo muy especial de robot omnidireccional, su estructura más básica requiere de 2 ruedas izquierdas y 2 ruedas derechas como se indica en la Figura 7. Puede lograr cualquier tipo de desplazamiento a través de la implementación de un control de velocidad individual en cada uno de los motores. Ofrece una mayor estabilidad con respecto a las configuraciones revisadas anteriormente, sin embargo, sufre una pérdida de velocidad con respecto a la velocidad máxima alcanzable con ruedas convencionales y dispuestas de la misma forma. La capacidad de carga está limitada por la máxima soportada por las ruedas Mecanum al igual que en las ruedas universales, ventajosamente la fricción es relativamente baja ayudando un mejor desplazamiento en el plano (A. Ortega, 2019).

Figura 7

Robot Omnidireccional de 4 Ruedas Mecanum



Nota. La figura representa la estructura de 2 robots omnidireccionales con ruedas Mecanum de mecanismos similares. Tomado de (Tejera, 2009).

Tabla 1

Comparación entre las Configuraciones de Robots Omnidireccionales

	Control	Diseño	Estabilidad	Tracción
Omnidireccional de Tres Ruedas	Simple	Simple	Baja	Baja
Omnidireccional de Cuatro Ruedas	Complejo	Simple	Media	Baja
Omnidireccional de Cuatro Ruedas Mecanum	Complejo	Complejo	Alta	Media

Nota. A pesar de las ventajas de movimiento que ofrecen cada una de las configuraciones se observan ciertas desventajas en cada una de ellas, lo cual limita el desarrollo de aplicaciones. Sin embargo, se observa que la configuración Mecanum resulta ser la más completa.

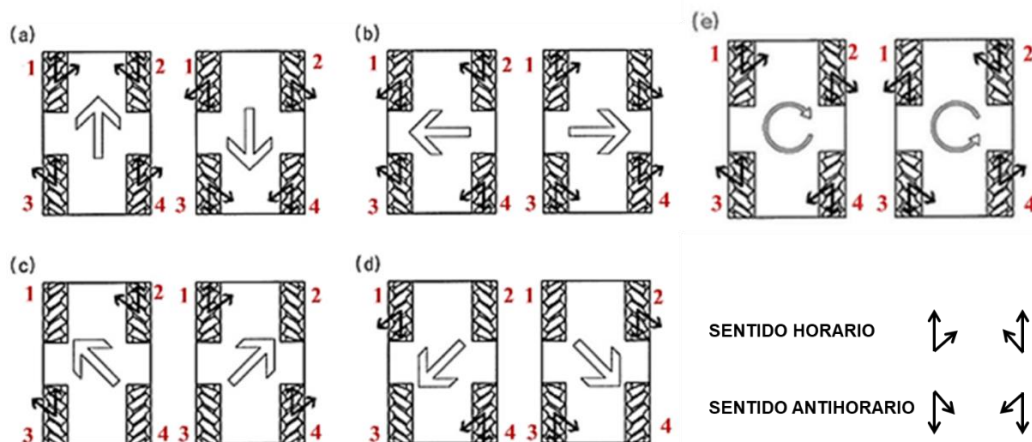
2.5. Locomoción Omnidireccional con Ruedas Mecanum

La plataforma omnidireccional móvil puede realizar 81 movimientos diferentes a través de la combinación de 4 ruedas Mecanum. Sin embargo, algunas de estas combinaciones no pueden ser tomadas en cuenta para el estudio del movimiento del robot omnidireccional debido a que son muy impredecibles a causa de varios factores tales como: desalineamiento de las ruedas, deslizamiento causado por irregularidades en los rodamientos, proporción desigual de par en los motores, etc. Tras la anulación de dichas consideraciones, se puede obtener 51 movimientos funcionales con el robot omnidireccional de ruedas Mecanum (Marosan et al., 2020).

Entre los 51 movimientos nombrados anteriormente la plataforma omnidireccional se caracteriza por tener los siguientes movimientos principales:

Figura 8

Principales Movimientos de la Plataforma Omnidireccional



Nota. El gráfico representa los principales modos de locomoción de una plataforma omnidireccional. Tomado de (Marosan et al., 2020).

- a) Desplazamiento Frontal: Se logra al movilizar las 4 ruedas a la misma velocidad angular y en sentido horario, por otra parte, para dar marcha atrás solamente se requiere cambiar el sentido de giro de las 4 ruedas, es decir en sentido antihorario.
- b) Desplazamiento Lateral: Para un desplazamiento a la izquierda se requiere que las ruedas 1 y 4 giren en sentido antihorario y las ruedas 2 y 3 en sentido horario, de igual manera para un desplazamiento a la derecha, pero en sentido contrario, es decir las ruedas 1 y 4 en sentido horario y las ruedas 2 y 3 en sentido antihorario.
- c) Desplazamiento Diagonal Frontal: Para desplazarse hacia adelante en dirección diagonal a la izquierda se requiere que las ruedas 2 y 3 giren en sentido horario

y las ruedas 1 y 4 no estén accionadas, en cambio para el desplazamiento diagonal a la derecha las rueda 2 y 3 no deben estar accionadas y las ruedas 1 y 4 deben girar en sentido horario.

- d) Desplazamiento Diagonal Anterior: Para desplazarse hacia atrás en dirección diagonal a la izquierda se requiere que las ruedas 1 y 4 giren en sentido antihorario y las ruedas 2 y 3 no estén accionadas, en cambio para el desplazamiento diagonal a la derecha las rueda 1 y 4 no deben estar accionadas y las ruedas 2 y 3 deben girar en sentido antihorario.
- e) Rotación: La rotación antihoraria de toda la plataforma omnidireccional en el mismo eje se obtiene haciendo girar en sentido horario las ruedas 1 y 3 y en sentido antihorario 2 y 4, del mismo modo para el giro antihorario se accionan las ruedas 1 y 3 en sentido antihorario y en sentido horario las ruedas 2 y 4 (Marosan et al., 2020)

2.6. Cinemática del Robot Omnidireccional Mecanum

2.6.1. Aspectos Cinemáticos

La cinemática se encarga de estudiar el movimiento de un robot móvil en función de la geometría de su estructura, excluyendo del análisis a las fuerzas que provocan dicho desplazamiento como la masa y el peso de los componentes del robot móvil.

Dando paso al análisis y simulación del comportamiento cinemático del robot, diseño de controladores o simplemente para determinar en cualquier instante de tiempo la posición, velocidad y aceleración del robot móvil. Sin embargo, para la obtención del modelo cinemático se debe tomar en cuenta las siguientes consideraciones:

- a) La superficie por la que se mueve el robot móvil es plana.
- b) Se considera que todos los elementos del robot móvil son rígidos, es decir que no existen elementos flexibles incluyendo las ruedas.

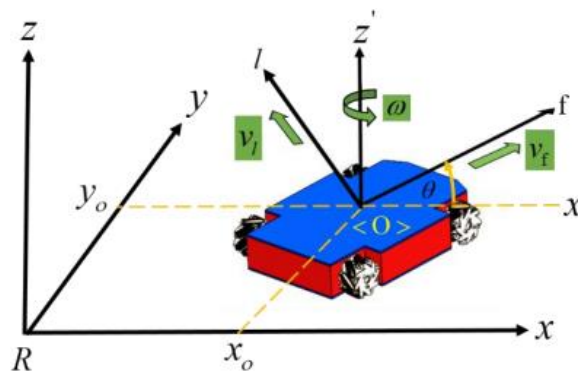
- c) Las ruedas tienen al menos un eje de direccionamiento y en caso de carecer de uno se considera que tiene dirección perpendicular a la superficie por la que se desplaza.
- d) Se desprecia cualquier tipo de fricción ejercida por el suelo o en los componentes mecánicos del robot móvil (García Silla et al., 2016).

2.6.2. Modelo Cinemático Robot Omnidireccional

Para la obtención del modelo cinemático del robot omnidireccional de ruedas Mecanum se considera a toda la plataforma como un cuerpo rígido que se desplaza con respecto a un centro de coordenadas x, y, z , donde el eje z es perpendicular al plano de desplazamiento tal y como se puede observar en la Figura 9.

Figura 9

Representación Esquemática del Robot Omnidireccional



Nota. La figura muestra una representación de las componentes cinemáticas que se toma en cuenta para el análisis del modelo matemático. Tomado de (Álvarez, 2019).

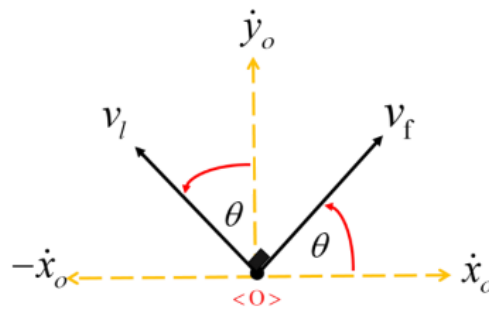
En la Figura 9 se puede observar la ubicación de la plataforma omnidireccional representada mediante las coordenadas x_o, y_o, z_o , que se mueve respecto a un centro

de coordenadas impulsada por 2 velocidades lineales v_l , v_f y una velocidad angular ω , generando así el desplazamiento sobre el plano de análisis xy donde:

- ❖ v_f velocidad lineal frontal expresado en $[m/s]$
- ❖ v_l velocidad lateral expresado en $[m/s]$
- ❖ ω velocidad angular respecto al eje z' expresado en $[rad/s]$.

Figura 10

Velocidades Lineales del Robot en el Plano



Nota. La figura muestra una representación gráfica de la velocidad lateral y frontal en el punto de operación $\langle O \rangle$ desde una vista superior del robot omnidireccional. Adaptado de (Álvarez, 2019).

A través de la Figura 10 se puede identificar claramente la descomposición de las velocidades lineales en el punto de operación $\langle O \rangle$ que en este caso es el centro del robot omnidireccional, de este modo se puede determinar fácilmente las velocidades \dot{x}_o, \dot{y}_o de la plataforma en el eje x, y respectivamente (Álvarez, 2019).

Dicho esto, las velocidades \dot{x}_o, \dot{y}_o están dadas por:

$$\dot{x}_o = v_f \cos \theta - v_l \sin \theta \quad (1)$$

$$\dot{y}_o = v_f \sin \theta + v_l \cos \theta \quad (2)$$

Del mismo modo la velocidad angular del robot se define como:

$$\omega = \dot{\theta} \quad (3)$$

El modelo cinemático del robot omnidireccional con referencia a su centro quedaría definido a partir de las ecuaciones (1), (2) y (3) así:

$$\dot{x}_o = v_f \cos \theta - v_l \sin \theta$$

$$\dot{y}_o = v_f \sin \theta + v_l \cos \theta \quad (4)$$

$$\dot{\theta} = \omega$$

El modelo cinemático representado en la ecuación (4) se puede representar de forma matricial así:

$$\begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_f \\ v_l \\ \omega \end{bmatrix} \quad (5)$$

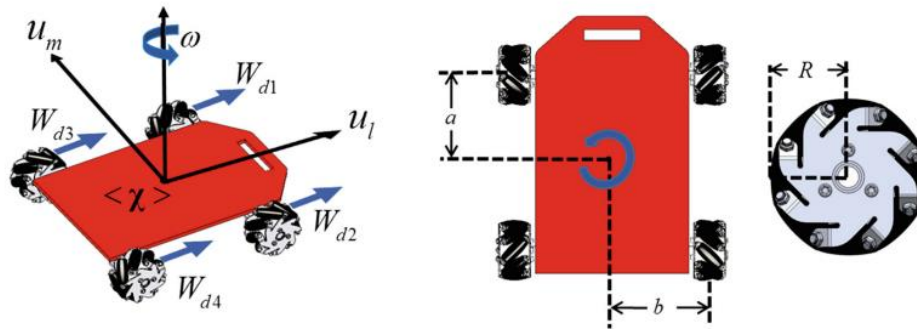
2.7. Matriz de Transformación de Velocidades

La ecuación (5) representa la relación directa entre las velocidades de la plataforma omnidireccional y las velocidades con respecto a cada eje del sistema de referencia. Sin embargo, es indispensable realizar una conversión de velocidades para determinar la velocidad angular de cada una de las ruedas. Esta es una transformación que se construye en función de la configuración de la plataforma omnidireccional, como se muestra en la Figura 11. Dicha transformación se representa como (Andaluz et al., 2017):

$$\begin{bmatrix} W_{d1} \\ W_{d2} \\ W_{d3} \\ W_{d4} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} v_f \\ v_l \\ \omega \end{bmatrix} \quad (6)$$

Figura 11

Configuración de la Plataforma Omnidireccional



Nota. Tomado de (Andaluz et al., 2017)

2.8. Control de un Robot Omnidireccional

Actualmente existen varias maneras de controlar el desplazamiento y la orientación de un robot móvil, partiendo desde la forma más clásica que consiste en un control manual, donde un operador es el encargado de guiar al robot por la trayectoria deseada a través de un mando, una aplicación remota o una interfaz. Sin embargo, lo ideal es que el robot móvil realice la tarea de manera autónoma, optimizando recursos y en el menor tiempo posible, es ahí en donde nace la idea de un control autónomo para este tipo de sistemas.

El control automático de un robot omnidireccional consiste en implementar un algoritmo de control basado en el modelo no lineal del sistema, permitiendo obtener como salida del controlador las velocidades que se deberán enviar al robot para lograr llegar a la referencia deseada, ya sea un punto o una trayectoria parametrizada en el

tiempo. Existen varias maneras de implementar el control de manera autónoma como es el caso de los algoritmos de control clásico por ejemplo el PID y de igual manera, algoritmos de control moderno como, por ejemplo: lógica difusa, control óptimo, control no lineal etc.

Para la implementación de un control automático se requiere analizar la cinemática inversa del sistema omnidireccional con el objetivo de definir las referencias que el robot omnidireccional deberá seguir para el cumplimiento de su tarea. Por lo que es necesario e indispensable de un controlador interno que permita seguir dicha referencia de velocidad, permitiendo disminuir el error de seguimiento producido por la dinámica del robot o a su vez por fallas en la mecánica del mismo (Álvarez, 2019).

La autonomía de un robot omnidireccional depende del sistema de control automático que se implemente, debido a que de esto dependerá el correcto funcionamiento o cumplimiento de la tarea asignada. Existen 3 problemas principales que se puede encontrar al implementar un control autónomo en un robot omnidireccional:

2.8.1. Control de Posición:

Consiste en desplazar al robot omnidireccional desde una posición inicial conocida hasta una posición final deseada. Este tipo de controlador se lo conoce como control por regulación, es decir, tiene una referencia fija.

2.8.2. Seguimiento de Camino:

Este controlador es una combinación del anterior, debido a que se realiza un control de posición para cada uno de los puntos que forman el camino deseado que deberá seguir el robot omnidireccional.

2.8.3. Control de Trayectoria:

El objetivo del control por trayectoria consiste en seguir una curva parametrizada en el tiempo, es decir, la trayectoria de referencia es una función matemática que tiene como variable independiente al tiempo. Este tipo de control se lo conoce como control por seguimiento, debido a que su referencia va evolucionando en cada tiempo de muestreo (Álvarez, 2019).

2.9. Controladores Clásicos

Este tipo de controlador se implementa en la gran mayoría de industrias debido a los buenos resultados que ofrece y a su relativo bajo costo y facilidad de implementación.

Se clasifican según su acción de control (Cevallos, 2016):

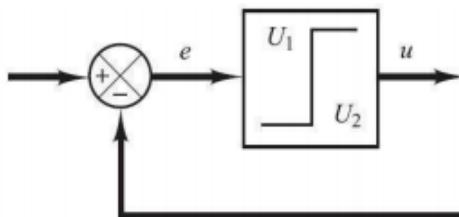
- Control de dos posiciones ON/OFF.
- Control Proporcional.
- Control Proporcional Integral.
- Control Proporcional Derivativo.
- Control Proporcional Integral Derivativo

2.9.1. Control De Dos Posiciones ON/OFF

Se caracteriza por tener únicamente dos estados fijos en la señal de control que proporciona, que por lo general son encendido y apagado o apertura y cierre. El uso de este tipo de controlador se puede apreciar desde el ámbito industrial y doméstico, puesto que es un control sumamente simple, económico y de fácil implementación (Cevallos, 2016).

Figura 12

Diagrama de Control ON/OFF



Nota. La salida $u(t)$ se establece como valor máximo U_1 y valor mínimo U_2 , siendo estos dos valores los únicos que podría tomar dependiendo del valor del error $e(t)$.

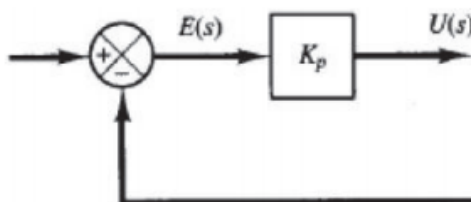
Tomado de (Cevallos, 2016).

2.9.2. Control Proporcional

El control proporcional se lo puede interpretar como un amplificador de ganancia ajustable que consiste en multiplicar el valor del error entre la referencia deseada y el valor real, por un valor de ganancia K_p , permitiendo un aumento de la velocidad de respuesta del sistema para poder alcanzar en menos tiempo el valor de referencia (Álvarez, 2019).

Figura 13

Diagrama del Control Proporcional



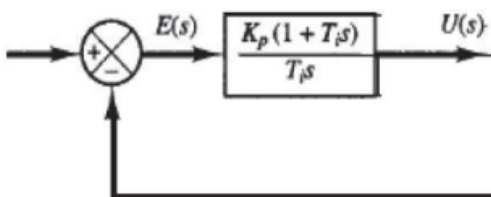
Nota. La figura representa el diagrama del control proporcional donde $E(s)$ representa el error, K_p la ganancia proporcional y $U(s)$ la salida del controlador. Tomado de (Cevallos, 2016).

2.9.3. Control Proporcional Integral

Este tipo de controlador combina una acción proporcional y una acción integral en su señal de control, permitiendo fusionar ambas características en una sola. La acción integral se encarga de eliminar el error en estado estable, que combinado con la velocidad de respuesta del sistema que provee la acción proporcional, se convierte en un controlador bastante eficiente (Cevallos, 2016).

Figura 14

Diagrama del Control Proporcional Integral



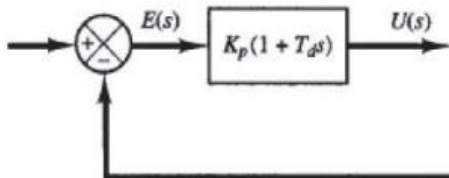
Nota. La figura representa el diagrama del control PI (Proporcional - Integral). Donde T_i es la constante de tiempo integral y K_p la ganancia proporcional, ambas regulables. Tomado de (Cevallos, 2016).

2.9.4. Control Proporcional Derivativo

El controlador proporcional derivativo PD está conformado por una ganancia proporcional y una acción derivativa del error, permitiendo mejorar la estabilidad del sistema, debido a que su acción derivativa representa una ganancia proporcional a la velocidad con la que varía el error del sistema en lazo cerrado (Cevallos, 2016).

Figura 15

Diagrama del Control Proporcional Derivativo



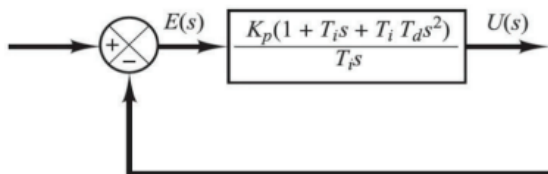
Nota. La figura representa el diagrama del control PD (Proporcional - Derivativo). Donde T_d es la constante de tiempo derivativo y K_p la ganancia proporcional, ambas regulables. Tomado de (Cevallos, 2016).

2.9.5. Control Proporcional - Integral - Derivativo

Este controlador resulta de la combinación de la acción proporcional, integral y derivativa, lo cual lo hace un controlador muy completo, puesto que reúne las características de las tres acciones de control para formar un control bastante eficiente (Cevallos, 2016).

Figura 16

Diagrama del Control PID



Nota. La figura representa el diagrama del control PID (Proporcional - Integral - Derivativo). Donde K_p la ganancia proporcional, T_d es la constante de tiempo derivativo y T_i es la constante de tiempo integral. Tomado de (Cevallos, 2016).

2.10. Controladores Modernos

Los controladores clásicos son una buena opción cuando se tratan de sistemas sencillos y lineales, donde únicamente se evalúa cómo varía una salida con respecto a una entrada. Sin embargo, la realidad es que existen varios procesos que no son tan fáciles de controlar debido a varios factores como, por ejemplo: la dinámica del proceso es no lineal, el proceso está sometido a frecuentes y bruscas perturbaciones o grandes retardos en el sistema. Es por ello que surge la necesidad de implementar un sistema de control moderno que permita realizar un control adecuado en este tipo de sistemas.

A continuación, se presenta una breve descripción de algunos controladores modernos (Cevallos, 2016):

2.10.1. Control por Lógica Difusa:

Es mayormente implementado en el control de sistemas no lineales y variantes en el tiempo, donde es muy difícil extraer el modelo matemático del sistema. Realiza el control del proceso a través de la definición de reglas que permiten al controlador saber cómo reaccionar a diferentes cambios o perturbaciones que puedan surgir en el proceso.

2.10.2. Control por Redes Neuronales:

El comportamiento de este tipo de controlador es análogo al funcionamiento de una neurona del sistema nervioso, requiere de una etapa previa de entrenamiento donde calcular las ganancias o pesos de cada neurona para multiplicarlos por la entrada correspondiente, de este modo produce una salida de control.

2.10.3. Control Adaptativo:

Tiene la capacidad de adaptarse a los cambios producidos en el proceso, permitiendo ajustar automáticamente sus parámetros con el fin de disminuir los errores producidos por dichos cambios.

2.10.4. Control Robusto:

Este tipo de controlador se caracteriza por reaccionar adecuadamente a los errores producidos en el sistema debido a medición o errores en las señales de control.

2.10.5. Control No Lineal:

Se caracteriza por su versatilidad al trabajar con sistemas de dinámica no lineal, sin embargo, requiere del modelo matemático no lineal del sistema para establecer una ley de control que permita estabilizar el sistema en una referencia deseada.

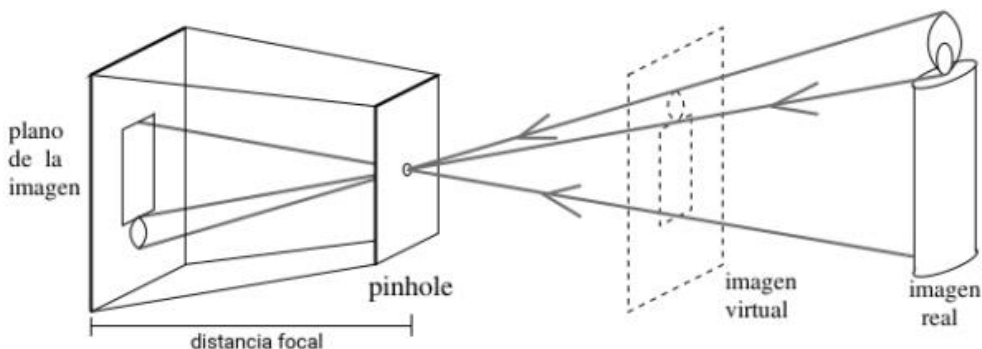
2.11. Visión Artificial

La visión artificial se podría definir como una rama de la inteligencia artificial que consiste en la adquisición, procesamiento y análisis de imágenes obtenidas una de determinada escena del mundo real a través de una cámara.

2.11.1. Conceptos Generales

2.11.1.1. Modelo de Pinhole

Es un modelo simple pero que a su vez permite interpretar la relación matemática a la que está sujeto un punto tridimensional (mundo) y su representación en el plano bidimensional (imagen). El modelo de la cámara de pinhole ideal consiste en “una caja con un pequeño orificio en la parte frontal (el pinhole) a través del cual cruza un único rayo de luz para cada punto existente en la parte trasera interna de la caja (el plano de la imagen)” (Poot Chin, 2019, p. 12).

Figura 17*Modelo de cámara pinhole*

Nota. La figura es una representación esquemática del modelo pinhole ideal. Tomado de (Poot Chin, 2019).

El modelo de la cámara de pinhole ideal es de gran utilidad para comprender el principio básico de cómo una cámara adquiere una imagen, sin embargo, esta es solo una aproximación de primer orden al modelo real de una cámara, debido a que en su representación no se toma en cuenta varios parámetros tales como: el desenfoque, la distorsión, la discretización de píxeles, etc. No obstante, en algunos casos varios de estos parámetros se podrían despreciar por su bajo efecto o simplemente compensarlos matemáticamente (Poot Chin, 2019).

2.11.1.2. Parámetros Intrínsecos de una Cámara

Los parámetros intrínsecos son aquellos valores que determinan el comportamiento de una cámara. Son los encargados de establecer una relación entre el plano de la cámara y el plano de la imagen. Estos valores no son constantes, puesto que varían dependiendo de la cámara que se esté ocupando para la etapa de adquisición. Dichos parámetros conforman una matriz de $\mathbf{K}_{3 \times 3}$ definida como (Tayupanta Zúñiga, 2018):

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Donde los coeficientes f_x y f_y representan las distancias focales de la cámara y c_x , c_y el centro del sensor óptico todo expresado en píxeles.

2.11.1.3. Parámetros Extrínsecos de una Cámara

Los parámetros extrínsecos conforman una matriz de transformación que describe la posición y orientación de la cámara en el mundo tridimensional, dicha matriz de transformación se la puede descomponer en una matriz de rotación y un vector de traslación. Su representación matricial viene dada por (Tayupanta Zúñiga, 2018):

$$[\mathbf{R} | \mathbf{T}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (8)$$

2.11.1.4. Matriz de la Cámara

La representación del modelo de pinhole puede ser interpretado como una transformación lineal a través de la matriz de transformación homogénea $\mathbf{C}_{3 \times 4}$, conocida como matriz de cámara. Para ello se considera un punto en el mundo $P(X, Y, Z)$ y su respectiva proyección en el plano de la imagen denotado como $p(u, v)$, dicha transformación lineal se representa como (Poot Chin, 2019):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (9)$$

La transformación lineal de la ecuación (9) es posible gracias a que la matriz de transformación homogénea $C_{3 \times 4}$, está compuesta por los parámetros intrínsecos y extrínsecos revisados anteriormente, dichos parámetros son valores estimados que se obtienen a través de un proceso de calibración de la cámara. Al expandir la matriz de cámara se obtiene (Poot Chin, 2019):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R} | \mathbf{T}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (11)$$

2.11.2. OpenCV

OpenCV es una librería de visión artificial y aprendizaje automático Open Source que facilita el desarrollo de aplicaciones de visión por computador en tiempo real. Está constituido por más de 2500 algoritmos optimizados que permiten poner en práctica una gran cantidad de algoritmos clásicos y de última generación. Está disponible para Windows, Linux y Android bajo lenguajes de programación como Python, C y C++ (OpenCV, 2020).

2.11.3. Calibración

En la etapa de adquisición de la imagen se puede notar cierta distorsión debido a que el lente de la cámara y el plano de la imagen no son perfectamente paralelos. La calibración de la cámara consiste en un proceso que permite estimar los parámetros intrínsecos y extrínsecos de la cámara con el objetivo de corregir este error.

La calibración geométrica es la herramienta que permite la estimación de los coeficientes de distorsión con los cuales se podrá corregir el error producido al intentar proyectar un punto del mundo real 3D en el plano 2D de la imagen, con el objetivo de facilitar la reconstrucción del entorno lo más cercano posible a la realidad, de este modo se obtienen medidas más precisas y exactas del mundo a partir de imágenes (Jiva, 2019).

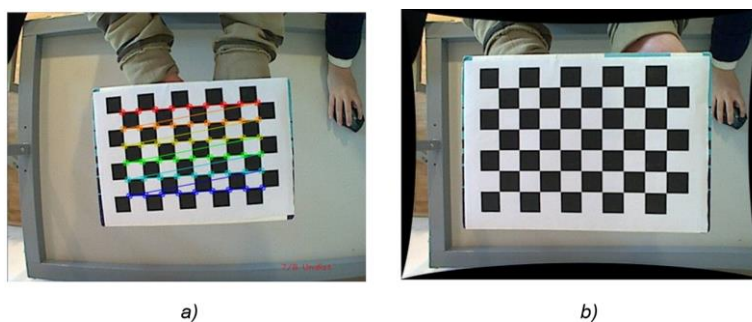
OpenCV incluye funciones que permiten la calibración de la cámara a través de cálculos geométricos básicos que dependerán de la geometría escogida para la etapa de calibración, dichos objetos pueden ser (OpenCV, 2011):

- ❖ Tablero de ajedrez clásico en blanco y negro
- ❖ Patrón de círculo simétrico
- ❖ Patrón de círculo asimétrico

El proceso en general consiste en tomar capturas del objeto seleccionado para calibración y dejar que OpenCV encuentre los respectivos patrones. A continuación, se muestra el resultado después de la calibración de la cámara.

Figura 18

Resultados del Proceso de Calibración de la Cámara



Nota. a) Imagen sin corrección de distorsión b) Imagen con corrección de los errores producidos por distorsión. Adaptado de (OpenCV, 2011).

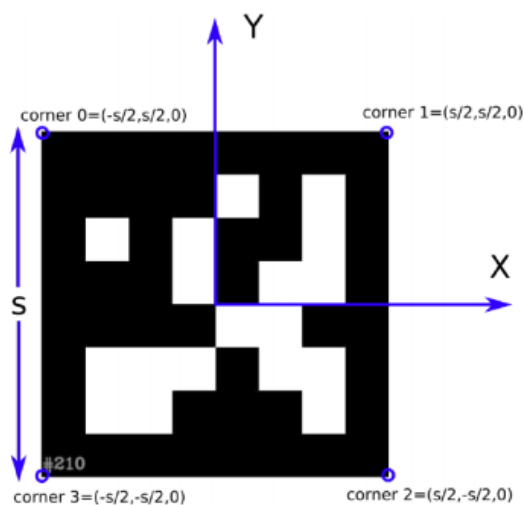
2.11.4. ArUco

ArUco es una librería de código abierto que se basa en la detección de marcadores que conjuntamente con los parámetros obtenidos en el proceso de calibración de la cámara, permiten al usuario determinar la posición y orientación del marcador con respecto al sistema de coordenadas de la cámara o viceversa. Esta librería está escrita en C++ y pertenece a uno de los módulos de contribución externos de OpenCV, así que para trabajar con la librería se requiere de una previa instalación (Jiva, 2019).

2.11.4.1. Marcadores

Figura 19

Marcador Estándar ArUco



Nota. La figura muestra el sistema de coordenadas empleado en el marcador. Tomado de (Jiménez Bravo, 2018)

Los marcadores ArUco son básicamente cuadrados que se conforman por un borde negro que protege el patrón binario único para cada marcador. Dicho patrón binario se construye a partir de un determinado número de bits que dependerá del

diccionario con el que se ha deseado trabajar. Sin embargo, hay que tomar en cuenta que a mayor número de bits se requerirá de una mayor resolución para evitar errores en el proceso de detección de los marcadores ArUco. La Figura 19 muestra un marcador ArUco estándar con sus respectivo sistema de coordenadas (Jiménez Bravo, 2018).

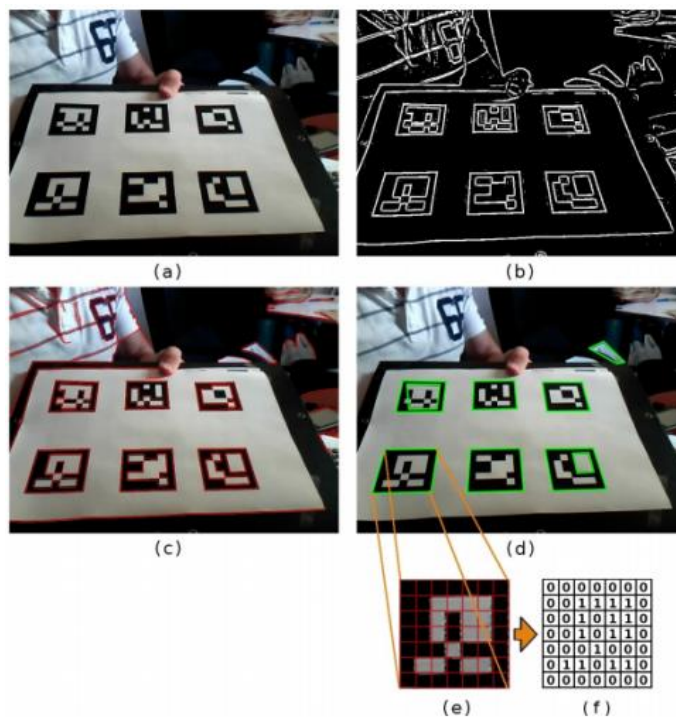
2.11.4.2. Proceso de Detección

El proceso de detección de un marcador ArUco consta de una serie de operaciones que se debe aplicar a la imagen original y un conjunto de pasos que se debe seguir para tener un óptimo resultado. Dichas instrucciones se detallan a continuación (Jiménez Bravo, 2018):

- ✓ Se aplica un umbral adaptativo a la imagen original, se extraen los bordes y se buscan los contornos como se aprecia en la Figura 20 b).
- ✓ A través de un proceso de anulación de los contornos no deseados se eliminan aquellos poco probables que representen un marcador. Para ello se usa una aproximación poligonal para mantener aquellos que tengan 4 esquinas de forma similar a como se muestra en Figura 20 d).
- ✓ Se elimina la distorsión de perspectiva para obtener una vista frontal del marcador como se indica en la Figura 20 e).
- ✓ Se aplica el método Otsu para determinar el umbral óptimo con que se separan los bits blancos y negros.
- ✓ Se analizan los bits para poder determinar si el marcador pertenece al diccionario con el que se está trabajando tal y como se muestra en la Figura 20 f).

Figura 20

Detección de un Marcador ArUco

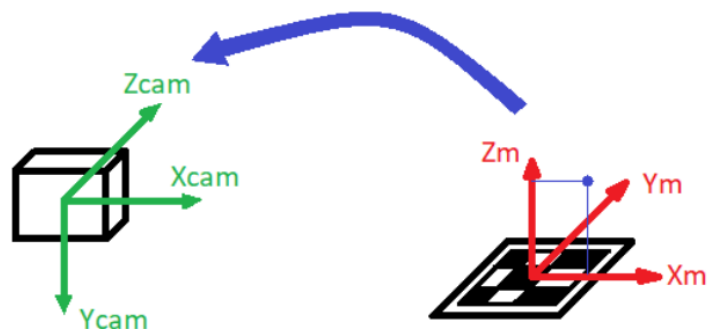


Nota. a) Imagen Original, b) Umbralización, c) Bordes detectados d) Aproximación poligonal, e) Corrección de perspectiva y f) Análisis de bits. Tomado de (Poot Chin, 2019).

Luego del proceso de detección de los marcadores ArUco, se puede extraer los parámetros extrínsecos con respecto a la cámara, los mismos que están conformados por un vector de rotación y un vector de traslación que representan la orientación y la ubicación del sistema de coordenadas del marcador con respecto al sistema de coordenadas de la cámara como se muestra conjuntamente en la Figura 21 (Herguedas Gastón, 2017).

Figura 21

Sistemas de Coordenadas Marcador - Cámara



Nota. La figura representa los sistemas de coordenadas del marcador ArUco con respecto al sistema de coordenadas de la cámara. Adaptado de (Jiménez Bravo, 2018).

Para el cálculo de los parámetros extrínsecos es necesaria una previa estimación de los parámetros intrínsecos de la cámara, es decir obligatoriamente se requiere de una previa calibración de la cámara.

2.12. Transformaciones 3D

Una transformación se define como una operación que permite modificar la posición y orientación de un punto con respecto a un sistema de referencia. Las operaciones más comunes en sistemas tridimensionales son: la traslación, la rotación y el escalado. Dichas operaciones son expresadas a través de una matriz cuadrada de 4x4 que permite la transformación de un punto P como se indica en la ecuación (12):

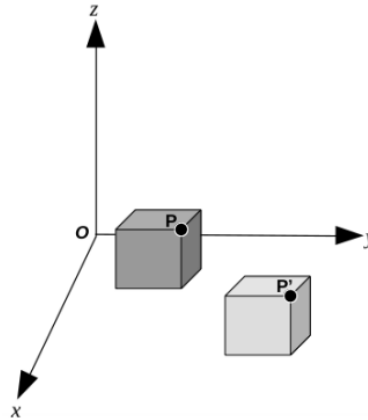
$$\mathbf{P}' = \mathbf{M}\mathbf{P} \quad (12)$$

Donde \mathbf{P}' es el punto transformado, \mathbf{M} es la matriz de transformación y \mathbf{P} es el punto original. A continuación se describen las dos operaciones de transformación más implementadas (Poot Chin, 2019):

2.12.1. Traslación

Figura 22

Transformación de Traslación



Nota. La figura muestra una operación de traslación de un punto P hacia un punto P' . Tomado de (Poot Chin, 2019).

La traslación de coordenadas se define como un desplazamiento de un punto hacia una nueva posición sin modificar la orientación del mismo, tal y como se aprecia en la Figura 22. Esto se obtiene incrementando cada una de las componentes del punto \mathbf{P} una \mathbf{t} como se muestra a continuación:

$$\mathbf{P}' = \mathbf{TP} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (13)$$

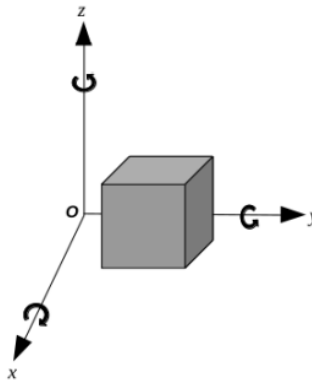
Donde \mathbf{P}' es el punto que ha sido desplazado una distancia \mathbf{t} del punto \mathbf{P} . La traslación también puede ser expresada en forma matricial a través de su representación en coordenadas homogéneas, lo cual permite expresarla de la siguiente forma:

$$\mathbf{P}' = \mathbf{TP} = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (14)$$

2.12.2. Rotación

Figura 23

Transformación de Rotación



Nota. La figura representa la operación de rotación alrededor de los ejes coordenados.

Tomado de (Poot Chin, 2019).

Se define como una transformación que permite la rotación de un punto alrededor de sus 3 ejes del plano cartesiano. Se produce con respecto a un punto en el espacio y no en referencia al origen, como sucede en el caso de la operación de traslación. Se lo puede representar como una expresión matricial $\mathbf{R}_{3 \times 3}$ cuyos componentes permiten la rotación del punto \mathbf{P} :

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (15)$$

La operación de rotación, al igual de la de traslación, se lo puede expresar mediante coordenadas homogéneas como se describe a continuación:

$$\mathbf{P}' = \mathbf{R}\mathbf{P} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (16)$$

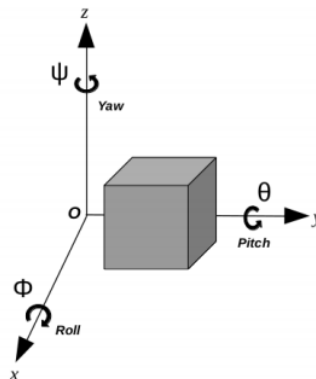
Donde \mathbf{P} es el punto que se va a rotar, \mathbf{R} es la matriz de rotación y \mathbf{P}' es el punto que resulta de la rotación.

2.12.3. Ángulos de Euler

La rotación de un cuerpo en el espacio también puede representarse usando los llamados Ángulos de Euler: $roll(\phi)$, $pitch(\theta)$ y $yaw(\psi)$. Estos ángulos hacen referencia a la rotación con respecto a cada uno de los ejes x , y y z respectivamente. Una mejor apreciación de la correspondencia de los ángulos de Euler y cada uno de los ejes se presenta en la Figura 24 (Jia, 2020).

Figura 24

Ángulos de Euler



Nota. La figura muestra la relación entre los ángulos de Euler y cada uno de los ejes del sistema coordinado. Tomado de (Poot Chin, 2019).

Los ángulos de Euler permiten expresar el giro con respecto a cada uno de los ejes como una transformación que se puede expresar de forma matricial como se muestra a continuación:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

A partir del producto de estas expresiones matriciales se puede definir una matriz de rotación generalizada, cuya expresión matricial se representa de la siguiente manera (Poot Chin, 2019):

$$\mathbf{R}_{x,y,z} = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) & 0 \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) & 0 \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Donde $s(\cdot)$ representa la función seno y $c(\cdot)$ a la función coseno. La matriz

$\mathbf{R}_{x,y,z}$ también puede ser representada de la siguiente forma:

$$\mathbf{R}_{x,y,z} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

De la cual se pueden deducir las siguientes ecuaciones para determinar los correspondientes ángulos de Euler:

$$\phi_{roll} = \arctan\left(\frac{r_{32}}{r_{33}}\right) \quad (22)$$

$$\theta_{pitch} = \arctan\left(\frac{-r_{31}}{\sqrt{(r_{32})^2 + (r_{33})^2}}\right) \quad (23)$$

$$\psi_{yaw} = \arctan\left(\frac{r_{21}}{r_{11}}\right) \quad (24)$$

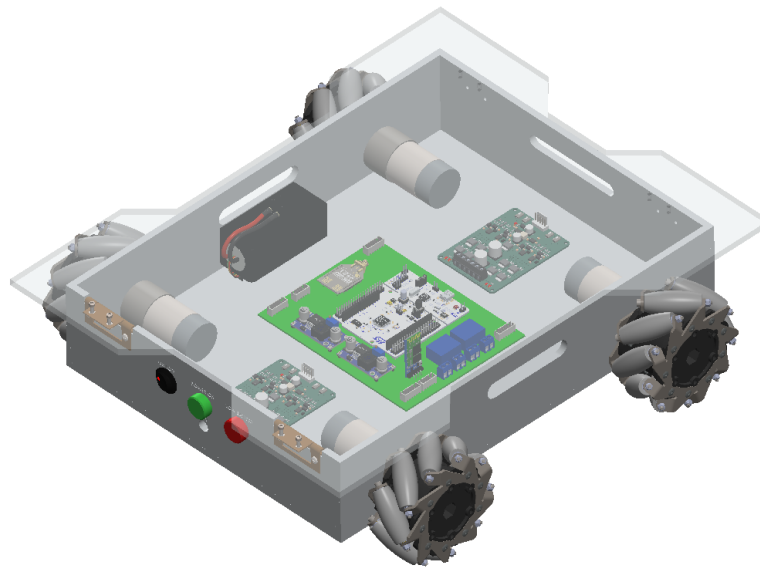
Capítulo III

3. Diseño y construcción de la plataforma omnidireccional

En este capítulo se muestra el diseño y construcción de la plataforma omnidireccional, para lo cual se tomó en consideración los múltiples estudios y diseños existentes acerca de robots omnidireccionales con ruedas Mecanum, con el fin de establecer la forma y el tamaño adecuado de la plataforma omnidireccional, la misma que fue diseñada previamente en el software CAD/CAE Solidworks. A continuación, se observa el diseño final de la estructura del robot omnidireccional.

Figura 25

Diseño CAD Robot Omnidireccional



3.1. Parámetros de Diseño

Para dar inicio con el diseño de la plataforma omnidireccional es necesario definir los parámetros de diseño que permitirán dimensionar y seleccionar los componentes adecuados para la construcción de la plataforma omnidireccional. Los

parámetros establecidos se muestran en la Tabla 2, donde m_p representa el valor máximo de la masa de la plataforma, m_c es el valor de carga máxima que debe transportar la plataforma, f es el factor de carga que se establece a criterio del diseñador y V_{req} es la velocidad requerida para resolver el problema del seguimiento de trayectoria.

Tabla 2

Parámetros de Diseño

Parámetros	Representación	Valor
Masa de la plataforma	m_p	10 kg
Carga máxima	m_c	5 kg
Velocidad lineal	\vec{V}_{req}	0.7 m / s
Factor de Carga	f	10%

La velocidad lineal de 0.7 m / s fue escogida en función de los requerimientos del sistema servo visual, es decir, que no sea un sistema demasiado rápido con el fin de facilitar la detección de la plataforma omnidireccional con la cámara; y evitar colisiones y daños mecánicos producidos por desplazamientos realizados a altas velocidades.

3.2. Cargas de Diseño

La carga total de la plataforma omnidireccional W_T , se define como la sumatoria de todas las cargas establecidas en los parámetros de diseño.

$$W_T = W_p + W_c \quad (25)$$

$$W_T = m_p \cdot g + m_c \cdot g$$

$$W_T = 10\text{kg} \cdot 9.81\text{m/s}^2 + 5\text{kg} \cdot 9.81\text{m/s}^2 = 147.15\text{N}$$

El factor de carga del 10% representa todas las cargas que no han sido consideradas en los parámetros de diseño, con el fin de asegurar un correcto funcionamiento de la plataforma omnidireccional, por lo que la carga de diseño W_D está dada por:

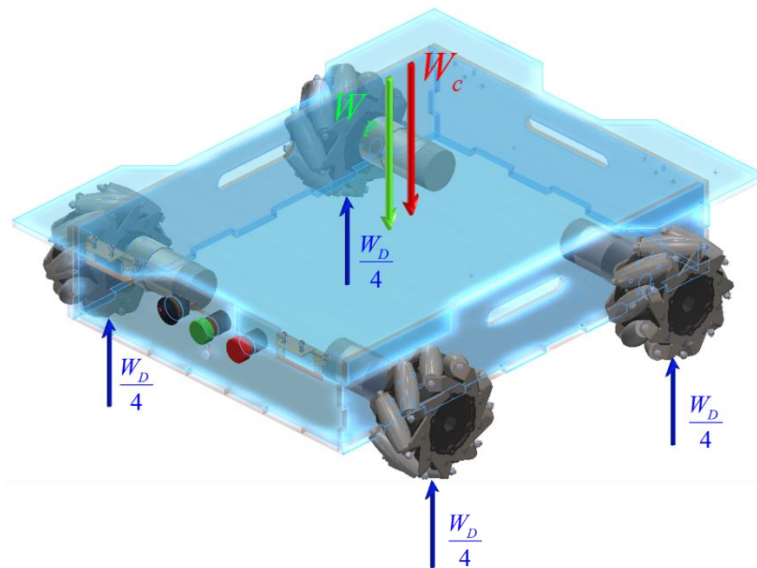
$$W_D = W_T + W_c \cdot f_{\text{carga}} \quad (26)$$

$$W_D = W_T + m_c \cdot g \cdot f_{\text{carga}}$$

$$W_D = 147.15\text{N} + 5\text{kg} \cdot 9.81\text{m/s}^2 \cdot 0.1 = 152.05\text{N} \approx 152\text{N}$$

Figura 26

Distribución de Cargas en la Plataforma Omnidireccional



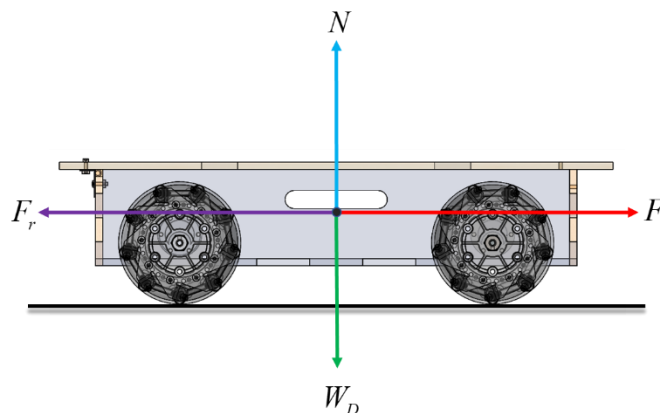
3.3. Análisis Estático

En esta sección se realiza un análisis de las fuerzas que intervienen en la plataforma omnidireccional a través del diagrama de cuerpo libre mostrado en la Figura

27. Donde W_D representa la carga de diseño, N la fuerza normal y F_r la fuerza de rozamiento.

Figura 27

Diagrama de Cuerpo Libre



$$\sum F_y = 0$$

$$\sum F_x = 0$$

$$N - W_D = 0$$

$$F - F_r = 0$$

$$N = W_D$$

$$F = F_r$$

$$N = 152 N$$

La fuerza de rozamiento se calcula a partir del coeficiente de rozamiento entre los rodillos de la rueda omnidireccional y la superficie de contacto.

μ_l : Coeficiente de fricción lateral = 0.6 (Tabla 5).

μ_f : Coeficiente de fricción delantero/inverso = 0.7 (Tabla 5).

$$F_{rl} = \mu_l \cdot N \tag{27}$$

$$F_{rl} = 0.6 \cdot 152 N = 91.2 N$$

$$F_{rl} = \mu_f \cdot N \quad (28)$$

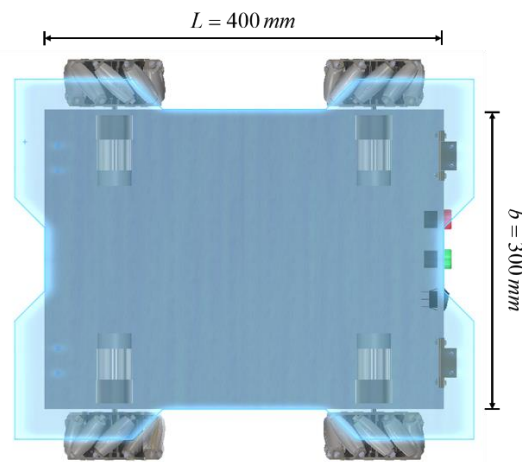
$$F_{rf} = 0.7 * 152 N = 106.4 N$$

3.4. Diseño Estructural

El diseño de la estructura de la plataforma omnidireccional se realiza haciendo énfasis en la base que soportará la carga diseño W_D definida en la ecuación (26), dicha base tiene las dimensiones mostradas a continuación:

Figura 28

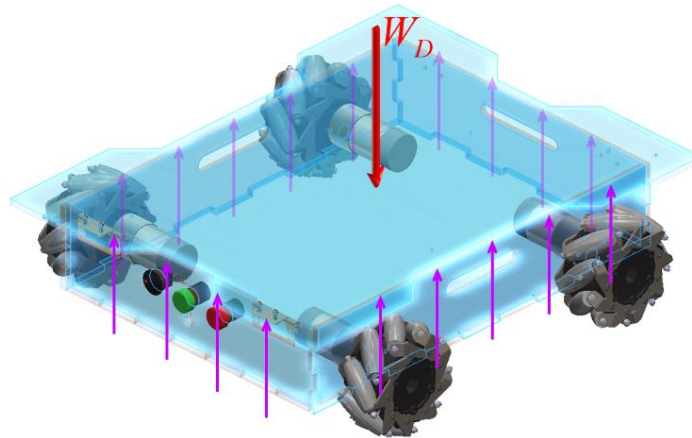
Dimensiones de la Base Superior



La carga aplicada en la base superior de la plataforma omnidireccional, se la puede interpretar como 4 cargas distribuidas en cada uno de los lados de la base, como se muestra en la Figura 29:

Figura 29

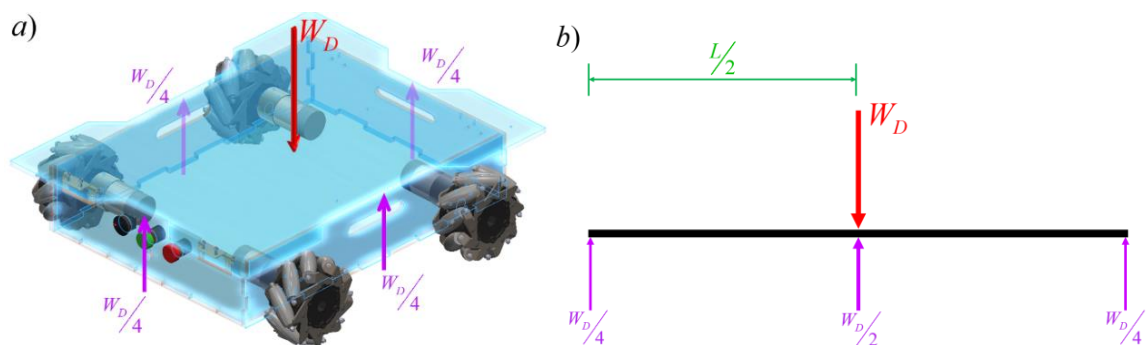
Cargas Distribuidas en la Base



Del mismo modo, cada carga distribuida se puede representar como una carga puntual aplicada en el centro de cada uno de los lados que componen la base, como se muestra en la Figura 30 a), con el fin de trazar el diagrama de fuerzas mostrado en Figura 30 b), desde la vista lateral a la base:

Figura 30

Cargas Puntuales en la Base



Luego de analizar varias alternativas de materiales para el diseño de la estructura de la plataforma omnidireccional se escogió un tablero de Fibra de Densidad Media (MDF), debido a que es un material resistente y con densidad mayor comparado

con otros materiales elaborados a base de madera. Además, es un material liviano, lo cual lo convierte en el candidato perfecto para la aplicación en desarrollo, puesto que, se busca que el peso de la estructura sea lo más liviano posible, con el objetivo de no consumir recursos de los motores en transportar la estructura, sino más bien enfocar dichos recursos en el transporte de carga. Sus propiedades mecánicas se muestran en la Tabla 3.

Tabla 3

Propiedades Mecánicas MDF

Propiedad	Valor
Módulo Elástico	4.0 GPa
Fuerza Compresiva	10 MPa
Radio de Poisson	0.25
Resistencia a la Fluencia	18 MPa
Densidad	0.75 g / cm ³

Nota. Tomado de (MakeltFrom, 2020).

Una vez definido el material con el cual se va a diseñar la estructura se procede a definir el esfuerzo de diseño σ_d como se muestra a continuación:

$$\sigma_d = \frac{S_y}{N} \quad (29)$$

Donde σ_d es el esfuerzo de diseño, S_y es la resistencia a la fluencia y N es el factor de seguridad que para este diseño tiene un valor de 2.

Dado que el material de la estructura fue escogido, el objetivo del diseño es determinar las dimensiones de la sección transversal, es decir, que se requiere determinar el espesor de la base superior mostrada en la Figura 28. Por lo tanto, se conoce que la sección transversal tiene forma rectangular, por lo que el momento de inercia de dicha sección transversal se define como:

$$I_x = \frac{bh^3}{12} \quad (30)$$

Donde I_x es el momento de inercia de la sección transversal con respecto al eje x , b es el ancho y h es el espesor de la base.

Luego, a través de la fórmula de flexión (Mott et al., 2009, p. 358), es posible relacionar el esfuerzo de diseño y las dimensiones de la sección transversal, así:

$$\sigma_{\text{máx}} = \frac{Mc}{I} \quad (31)$$

Donde $\sigma_{\text{máx}}$ es el esfuerzo máximo debido a flexión, M es el momento flexionante, c es la distancia desde el eje centroidal hasta la fibra más externa de la base superior de la estructura, es decir, la mitad del espesor de la base y, por último, I es el momento de inercia de la seccional transversal. Reemplazando las ecuaciones (29) y (30) en la ecuación (31) y reduciendo términos se tiene lo siguiente:

$$\sigma_{\text{máx}} = \frac{Mc}{\frac{bh^3}{12}} \quad (32)$$

$$\frac{S_y}{N} = \frac{M \frac{h}{2}}{\frac{bh^3}{12}} \quad (33)$$

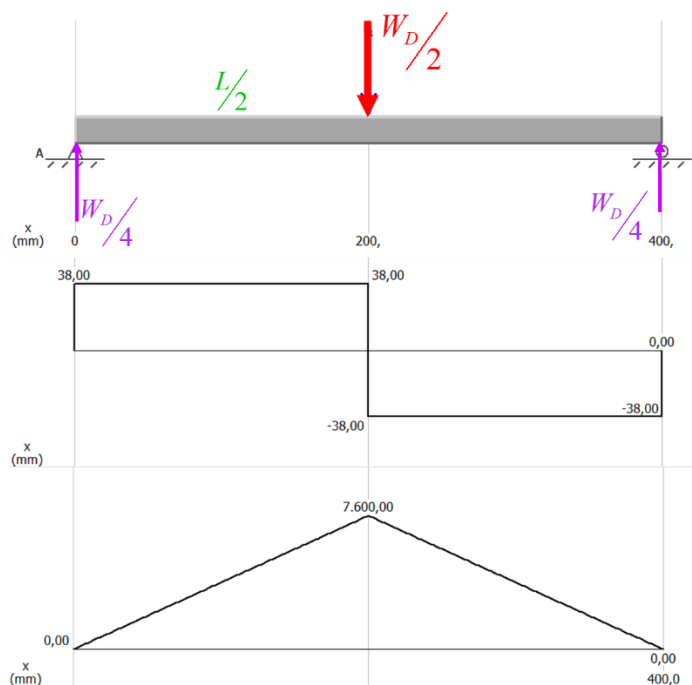
$$\frac{S_y}{N} = \frac{6M}{bh^2} \quad (34)$$

$$h = \sqrt{\frac{6MN}{bS_y}} \quad (35)$$

Para determinar el valor del momento flexionante se requiere trazar el diagrama de fuerza cortante y el diagrama del momento flexionante como se muestra en la Figura 31:

Figura 31

Diagrama de Fuerza Cortante y Momento Flexionante



De la Figura 31 se puede determinar que el momento flexionante producido por una fuerza cortante ubicada en el centro para este caso se define cómo:

$$M = \frac{W_D}{4} \cdot \frac{L}{2} = \frac{W_D L}{8} \quad (36)$$

Finalmente, sustituyendo la ecuación (36) en la ecuación (35) y reemplazando valores se obtiene el valor del espesor de la base como se muestra a continuación:

$$h = \sqrt{\frac{6MN}{bS_y}} \quad (37)$$

$$h = \sqrt{\frac{6 \frac{W_D L}{8} N}{bS_y}} \quad (38)$$

$$h = \sqrt{\frac{3W_D LN}{4bS_y}} \quad (39)$$

$$h = \sqrt{\frac{3 \cdot 152 N \cdot 400 mm \cdot 2}{4 \cdot 300 mm \cdot 18 N/mm^2}} = 4.11 mm$$

En función del valor de espesor determinado, se selecciona el espesor con el que se va a construir la estructura de la plataforma omnidireccional por lo cual se escoge $6 mm$ debido a que es el siguiente valor comercial disponible en el mercado.

3.4.1. Carga Máxima Soportada

Una vez seleccionado el valor del espesor con el que se va a trabajar, se recalcula el valor de la carga que podría soportar la estructura de la plataforma omnidireccional, por lo que, despejando W_D de la ecuación (39) se tiene lo siguiente:

$$h = \sqrt{\frac{3W_D LN}{4bS_y}} \quad (40)$$

$$W_D = \frac{4bh^2S_y}{3LN} \quad (41)$$

$$W_D = \frac{4 \cdot 300 mm \cdot 6^2 mm^2 \cdot 18 N/mm^2}{3 \cdot 400 mm \cdot 2} = 324 N \approx 33 Kg$$

3.4.2. Flexión Máxima

El valor de la flexión máxima que se produce en la base superior cuando se aplica una carga en el centro como se muestra en la Figura 31, se la puede determinar a través de la fórmula de flexión (Mott et al., 2009, p. 734), que se muestra a continuación:

$$y_{\text{máx}} = -\frac{PL^3}{48EI} \quad (42)$$

Donde $y_{\text{máx}}$ es la flexión máxima producida en una viga, P es la fuerza cortante, L es la longitud de la viga, E es el módulo de elasticidad del material y por último I es el momento de inercia de la sección transversal. Sustituyendo la ecuación (30) en la ecuación (42) y reemplazando valores se tiene que la flexión máxima producida en la base superior de la plataforma es:

$$y_{\text{máx}} = -\frac{PL^3}{48E \frac{bh^3}{12}} \quad (43)$$

$$y_{\text{máx}} = -\frac{PL^3}{4Ebh^3}$$

$$y_{\text{máx}} = -\frac{\frac{152}{2} N \cdot 400^3 \text{ mm}^3}{4 \cdot 4000 \frac{N}{\text{mm}^2} \cdot 300 \text{ mm} \cdot 6^3 \text{ mm}^3} = 1.69 \text{ mm}$$

3.4.3. Factor de Seguridad

El factor de seguridad se puede calcular despejando de la ecuación (41) como se muestra a continuación:

$$W_D = \frac{4b h^2 S_y}{3LN} \quad (44)$$

$$N = \frac{4b h^2 S_y}{3LW_D} \quad (45)$$

$$N = \frac{4 \cdot 300mm \cdot 6^2 mm^2 \cdot 18 \frac{N}{mm^2}}{3 \cdot 400mm \cdot 152N} = 4.26$$

El valor de $N = 4,26$ del factor de seguridad determinado permite concluir que el diseño de la estructura con un tablero MDF de $6mm$ es seguro para la carga mencionada anteriormente.

3.4.4. Análisis Estructural

En esta sección se muestra el análisis estático de la estructura de la plataforma omnidireccional. Para ellos se consideran 3 aspectos principales: la teoría de Von Mises, los desplazamientos resultantes y por último el factor de seguridad, los mismos que se describen a continuación.

Dichos análisis se realizan en el software SolidWorks, el cual permite elaborar el mallado de la estructura, aplicar las cargas, definir las sujeciones y por último extraer los resultados y las gráficas correspondientes.

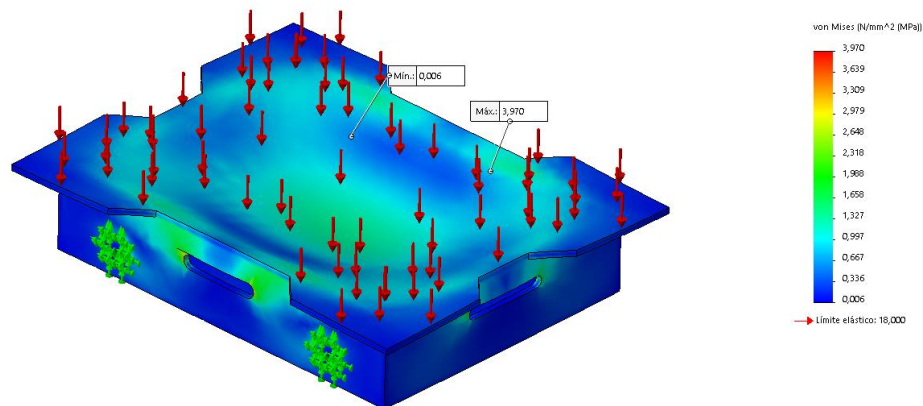
3.4.4.1. Teoría de Von Mises

La teoría de Von Mises o más conocida como teoría de la energía de distorsión máxima, es ampliamente utilizada para evaluar las fallas producidas por una fuerza cortante. Este fallo se produce cuando la energía de distorsión elástica es mayor que el límite de tensión máximo.

La tensión máxima se encuentra en la parte superior de la estructura con un valor de $3,97MPa$, el cual no supera al límite elástico del material de $18MPa$ que se muestra en la Tabla 3 por lo que se puede concluir que el diseño es seguro.

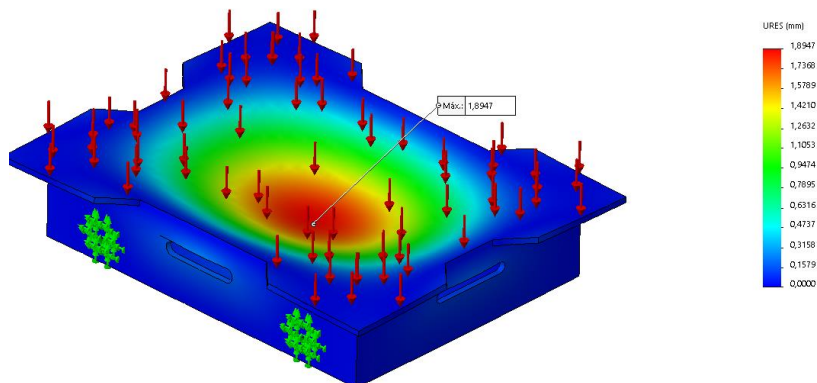
Tabla 4

Tensiones de Von Mises

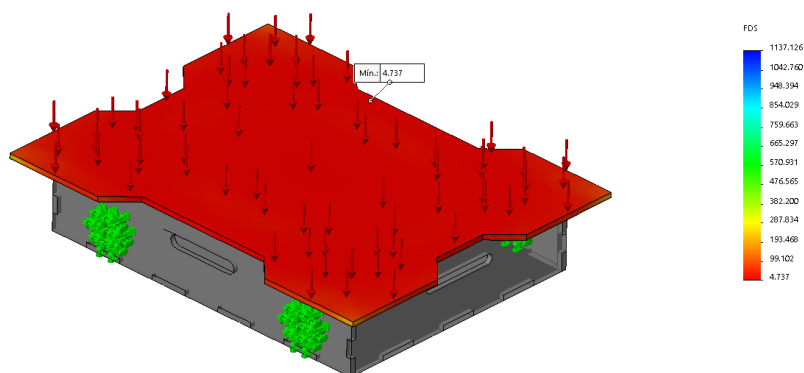


3.4.4.2. Desplazamientos Resultantes

Los desplazamientos resultantes (URES), muestran el valor de deformación en cada uno de los componentes de la estructura producidos por la carga aplicada. El máximo desplazamiento se encuentra en la parte superior de la estructura, como se muestra en la Figura 32, con un valor de $1,8947mm$, dicho valor máximo de deformación es sumamente pequeño tomando en cuenta que es ahí en donde se aplica la carga.

Figura 32*Desplazamientos Resultantes***3.4.4.3. Factor de Seguridad**

El factor de seguridad es el parámetro que define si el diseño es seguro o no, para ello se analiza el valor mínimo de factor de seguridad. En este caso los resultados arrojan un valor mínimo de factor de seguridad igual a 4,77 ,como se muestra en la Figura 33, lo cual nos permite asegurar la validez de la estructura.

Figura 33*Factor de Seguridad*

3.5. Selección de Componentes

3.5.1. Ruedas Mecanum

Para dar inicio con el dimensionamiento de las ruedas omnidireccionales es necesario definir su diámetro acorde a las necesidades y requerimientos del proyecto de investigación. El diámetro seleccionado para la rueda es de $4\text{ in} \approx 0.1016\text{ m}$, debido a que es un tamaño adecuado para la aplicación de seguimiento de trayectorias, el cual permitirá a la plataforma omnidireccional superar pequeñas imperfecciones que se podrían presentar en el plano por el que se moviliza. Además, el diámetro seleccionado representa un tamaño compacto, es decir, no representará mayor carga para la plataforma omnidireccional.

La selección del modelo de la rueda omnidireccional se lo realiza en función de la carga máxima que debe soportar cada rueda en la aplicación destinada. Tomando en cuenta la carga de diseño analizada en las Cargas de Diseño se tiene que:

$$W_D = 152.05\text{ N} \approx 152\text{ N}$$

$$P_{total} = \frac{W_D}{g} \quad (46)$$

$$P_{total} = \frac{152\text{ N}}{9.81\text{ m/s}^2} = 15.49\text{ kg} \approx 15.5\text{ kg}$$

$$P_{rueda} = \frac{P_{total}}{4} \quad (47)$$

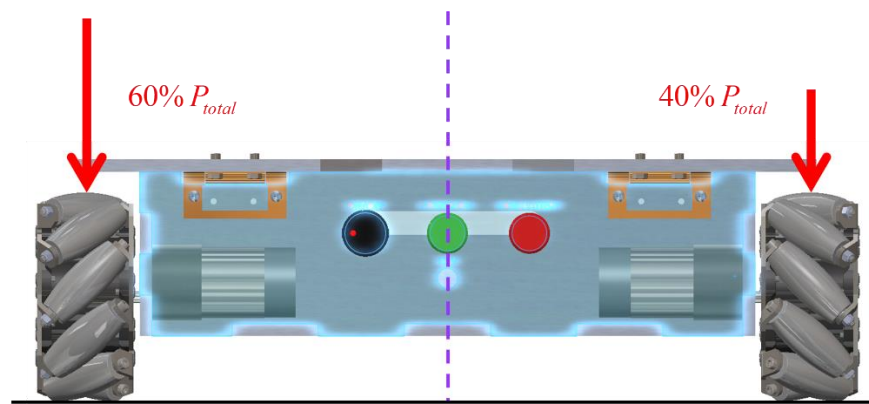
$$P_{rueda} = \frac{15.5\text{ kg}}{4} = 3.87\text{ kg} \approx 4\text{ kg}$$

El valor de P_{total} es el valor de la carga máxima que debe soportar todo el conjunto de ruedas omnidireccionales y de igual manera P_{rueda} es el valor máximo por

cada rueda. Sin embargo, dicho valor máximo de carga es considerado siempre y cuando, las cargas y componentes del robot se ubiquen de manera distribuida sobre la plataforma omnidireccional, para que de este modo la carga equivalente de todo el conjunto se ubique en el centro de gravedad. Por tal motivo se considera un margen de error de distribución de cargas del 10% , como se muestra en la Figura 34, donde se evalúa el caso en el que el 10% de la carga del lado derecho de la plataforma se ubica en el lado izquierdo.

Figura 34

Error de Distribución de Cargas



Por lo tanto, en función del margen de error de distribución de cargas, se puede determinar la carga situada en el lado izquierdo de la plataforma omnidireccional representada como P_{error} , así:

$$P_{error} = P_{total} \cdot 60\% \quad (48)$$

$$P_{error} = 15.5 \text{ kg} \cdot 0.6 = 9.3 \text{ kg}$$

Luego, el peso que soporta cada rueda tomando en cuenta el error de distribución de cargas es:

$$P_{rueda} = \frac{P_{error}}{2} \quad (49)$$

$$P_{rueda} = \frac{9.3 \text{ kg}}{2} = 4.65 \text{ kg}$$

De modo que, el peso que debe soportar solo el conjunto de ruedas Mecanum es:

$$P_{conjunto} = P_{rueda} \cdot 4 \quad (50)$$

$$P_{conjunto} = 4.65 \text{ kg} \cdot 4 = 18.6 \text{ kg} \approx 19 \text{ kg}$$

Basado en el análisis previo las ruedas Mecanum seleccionadas para la plataforma omnidireccional son las 4 IN HD 500 HEX MECANUM WHEEL del fabricante AndyMark. Las especificaciones técnicas de la rueda Mecanum se muestran a continuación:

Figura 35

Rueda Omnidireccional Mecanum 4 in



Nota. Tomado de (AndyMark, 2018).

Tabla 5

Especificaciones Técnicas de la Rueda Mecanum

Característica	Valor
Diámetro	4 in
Capacidad de carga	200 lb / 90.9 kg por rueda
Peso	0.9 lb / 0.4 kg por rueda
Coefficiente delantero/inverso de fricción	0.7
Coefficiente lateral de fricción	0.6
Número de rodillos	9
Material interior del rodillo	nylon (blanco)
Material exterior del rodillo	TPU
Material Corporal Exterior	0.12 in. acero grueso, chapado en zinc
Material Corporal Interior	policarbonato negro moldeado

Nota. Tomado de (AndyMark, 2018).

Luego de la selección de la rueda que se va a implementar, es indispensable determinar el peso máximo que podría cargar todo el conjunto. Esto se realiza haciendo el proceso inverso desarrollado para el cálculo del peso que debe soportar cada rueda en función de los parámetros de diseño. De modo que, partiendo del dato de carga máxima de la rueda especificado en la Tabla 5, y reemplazando en la ecuación (49) se tiene:

$$P_{rueda} = \frac{P_{error}}{2}$$

$$P_{error} = P_{rueda} \cdot 2 \quad (51)$$

$$P_{error} = 90.9 \text{ kg} \cdot 2 = 181.8 \text{ kg}$$

Luego se analiza la ecuación (48), donde se consideraba el error de distribución de carga, por lo que despejando y reemplazando los valores obtenidos se tiene:

$$P_{error} = P_{total} \cdot 60\%$$

$$P_{total} = \frac{P_{error}}{60\%} \quad (52)$$

$$P_{total} = \frac{181.8 \text{ kg}}{60\%} = 303 \text{ kg}$$

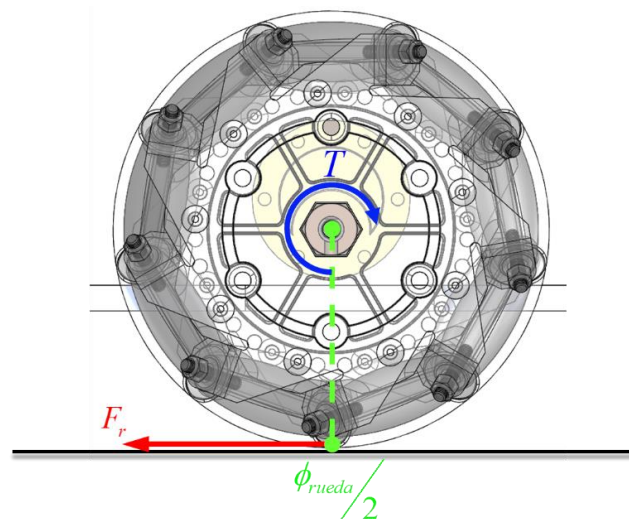
La carga que podría soportar todo el conjunto en base al análisis realizado es de 303 kg.

3.5.2. Motor

3.5.2.1. Torque

Figura 36

Fuerzas Presentes en la Rueda



El torque total requerido para mover la plataforma omnidireccional se define de acuerdo a la ecuación (53):

$$T = F_r \frac{\phi_{rueda}}{2} \quad (53)$$

$$T = 106.4 N \frac{0.1016 m}{2}$$

$$T = 5.4 N \cdot m$$

Por lo tanto, el torque requerido en cada rueda es:

$$T_{rueda} = \frac{T}{4} \quad (54)$$

$$T_{rueda} = \frac{5.4 N \cdot m}{4} = 1.35 N \cdot m \rightarrow 13.77 \text{ kg} \cdot \text{cm}$$

3.5.2.2. Potencia Requerida

$$P_{rueda} = T_{rueda} \cdot \omega_{req} \quad (55)$$

$$\omega_{req} = \frac{2 \cdot \vec{V}_{req}}{\phi_{rueda}}$$

$$\omega_{req} = \frac{2 \cdot 0.7 \text{ m/s}}{0.1016 \text{ m}} = 13.78 \text{ rad/s} \rightarrow 131.59 \text{ RPM}$$

Reemplazando los valores obtenidos en la ecuación (55) se tiene:

$$P_{rueda} = 1.35 \text{ Nm} \cdot 13.78 \text{ rad/s}$$

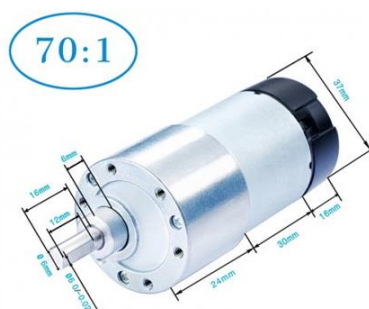
$$P_{rueda} = 19 \text{ W}$$

Como ya se mencionó anteriormente, la plataforma omnidireccional con ruedas Mecanum, requiere de 4 actuadores para realizar el control de su desplazamiento. La selección de estos actuadores se realiza conforme a los datos de velocidad, torque y potencia obtenidos anteriormente. En tal virtud, el motor que cumple con los

requerimientos y consciente los parámetros de diseño es el motor CQRobot 70:1 37D 70L que se muestra en la Figura 37.

Figura 37

Motor CQRobot 70:1 37D 70L



Nota. Tomado de (CQRobot, 2020).

Tabla 6

Especificaciones Técnicas Motor CQRobot 70:1 37D 70L

Característica	Valor
Relación de marchas	70:1
Voltaje	12 V
Velocidad sin carga	157 RPM
Corriente de estancamiento	5.5 A
Par de parada	$28\text{ kg} \cdot \text{cm} \approx 2.75\text{ N} \cdot \text{m}$
Tensión de funcionamiento del codificador	3.3V – 20V
Tipo de codificador	Hall(tipo incremental)
Resolución del codificador	64 CPR
Peso	235 g

Nota. Tomado de (CQRobot, 2020).

Del mismo modo que con la rueda, se realiza el análisis inverso para determinar el valor máximo de carga que podría soportar la plataforma omnidireccional en función de los parámetros especificados en la Tabla 6. Por lo que despejando la ecuación (54) y reemplazando valores se tiene el torque total T es:

$$T_{rueda} = \frac{T}{4}$$

$$T = T_{rueda} \cdot 4 \quad (56)$$

$$T = 1.47 \text{ N} \cdot \text{m} \cdot 4 = 5.88 \text{ N} \cdot \text{m}$$

Despejando la ecuación (53), se puede determinar el valor de la fuerza de fricción F_r , como se muestra a continuación:

$$T = F_r \frac{\phi_{rueda}}{2}$$

$$F_r = T \frac{2}{\phi_{rueda}} \quad (57)$$

$$F_r = 5.88 \text{ N} \cdot \text{m} \frac{2}{0.1016 \text{ m}} = 115.75 \text{ N}$$

Realizando un despeje de la ecuación (28) y reemplazando valores para encontrar el valor de la fuerza normal N , se tiene que:

$$F_{rl} = \mu_f \cdot N$$

$$N = \frac{F_{rl}}{\mu_f} \quad (58)$$

$$N = \frac{115.75 \text{ N}}{0.7} = 165.35 \text{ N}$$

Por último, de acuerdo con el análisis realizado en el eje y del diagrama de cuerpo libre de la Figura 27 se tiene lo siguiente:

$$N = W_D$$

$$W_D = 165.35 \text{ N}$$

Por lo tanto, la carga máxima que podría soportar los motores es:

$$m_D = \frac{165.35 \text{ N}}{9.81 \text{ m/s}^2} = 16.86 \text{ kg}$$

3.5.3. Driver Motor DC

La incorporación de motores DC en la plataforma omnidireccional requiere de un dispositivo que permita controlar la velocidad de los mismos. El driver Cytron MDD10A que se muestra en la Figura 38, es un puente H de doble canal que permite realizar el control bidireccional de hasta 2 motores DC de alta corriente.

Posee componentes de estado sólido que proporcionan un mayor tiempo de respuesta, admite el control de velocidad por señal PWM y no requiere de disipadores de calor. Sus especificaciones técnicas se muestran en la Tabla 7.

Figura 38

Cytron MDD10A



Nota. Tomado de (Cytron, 2017).

Tabla 7*Especificaciones Técnicas Cytron MDD10A*

Característica	Valor
Canal de motor	2
Voltaje de funcionamiento (VDC)	5V – 30V
Corriente máxima (A)	30 A
Corriente continua (A)	10 A
Frecuencia PWM máxima	20 KHz
Niveles lógicos	3.3V – 5V
Protección contra polaridad inversa	<i>no</i>
Frenado regenerativo	<i>si</i>

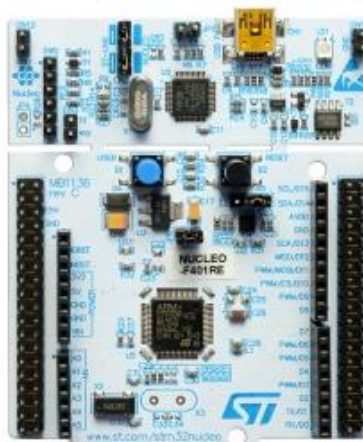
Nota. Tomado de (Cytron, 2017).

3.5.4. Tarjeta de Desarrollo

El control de los elementos que componen la plataforma omnidireccional requiere de la implementación de una tarjeta de control, en este caso se trata de la tarjeta de desarrollo STM32 Nucleo-F446RE que se muestra en la Figura 39, el cual viene incorporado con el MCU STM32F446RE que presenta grandes ventajas frente a otras familias de microcontroladores como AVR o PIC. La Tabla 8 muestra algunas de las características principales de la tarjeta de desarrollo STM32 Nucleo-F446RE

Figura 39

Tarjeta de Desarrollo STM32 Nucleo-F446RE



Nota. Tomado de (STMicroelectronics, 2020).

Tabla 8

Características Principales de la Tarjeta de Desarrollo STM32 Nucleo-F446RE

Característica	Valor
Microcontrolador	<i>STM 32 en paquete LQFP64</i>
Núcleo del CPU	<i>Arm Cortex – M4 core</i>
Frecuencia máxima	<i>180 MHz</i>
Potencia de Consumo	<i>345 mW</i>
ADC	<i>3 convertidores A / D de 12 bits con 24 Canales</i>
E/S Digitales	<i>114 puertos de E / S con Capacidad de Interrupción</i>
Temporizadores	<i>17 Timers 32bits, 2x watchdog, 1x SysTick timer</i>
Interfaces de Comunicación	<i>4 USART, 4 I²C ,4 SPI, 2 SAI, 2 CAN</i>
Depurador/programador	<i>ST – LINK a bordo, USB almacenamiento masivo</i>

Nota. Tomado de (STMicroelectronics, 2020).

En general, la línea familia de microcontroladores STM32 incorpora un núcleo ARM Cortex de alto rendimiento. La tarjeta de desarrollo en cuestión ofrece una alta frecuencia de rendimiento de 180 MHz, memorias incluidas de alta velocidad, acceso a periféricos como ADC, temporizadores e interfaces de comunicación UART, SPI, I2C, SPI y CAN. Todas estas características reunidas hacen que esta familia de microcontroladores sea un candidato idóneo para el desarrollo de aplicaciones tales como: control de motores, inversores, PLC, impresoras, escáner, equipos médicos, etc. (STMicroelectronics, 2020).

3.5.5. Módulo de Comunicación

El sistema de comunicación se realiza a través del protocolo de comunicaciones inalámbricas ZigBee por medio de los módulos XBEE S2C. Estos módulos manejan el protocolo IEEE 802.15.4 permitiendo la transmisión de datos a grandes distancias y a una baja latencia a través de redes punto a punto y multipunto. Además, se puede conectar directamente al PC por medio de un conversor UART a USB para su respectiva configuración o comunicación. Son ideales para el envío y recepción de datos provenientes de un microcontrolador o microprocesador. La Tabla 9 muestra sus principales especificaciones técnicas.

Figura 40

XBEE Pro S2C



Nota. Tomado de (Digi International Inc, 2021).

Tabla 9*XBEE Pro S2C*

Característica	Valor
Voltaje de alimentación	2.7V – 3.6V
Rango en interiores	90m
Rango en exteriores	3200m
Potencia	63mW
Interfaz de datos serie	UART / SPI
Protocolo	IEEE 802.15.4
Dimensiones	2.199 x 3.4 x 0.305 cm

Nota. Tomado de (Digi International Inc, 2021).

3.5.6. Fuente de Alimentación

El funcionamiento autónomo de la plataforma omnidireccional requiere de un sistema de alimentación que provea de energía suficiente a los componentes, tanto de la etapa de control como de la etapa de potencia. Motivo por el cual se vio la necesidad de implementar una batería Li-Po, debido a las buenas características y excelente rendimiento que proporciona.

Para seleccionar la batería Li-Po adecuada, se parte del análisis de la potencia requerida por todo el sistema.

$$P_T = P_{motores} + P_{stm32} + P_{xbee} \quad (59)$$

Donde P_T es la potencia total requerida por el sistema, $P_{motores}$ representa la potencia máxima de consumo de los 4 motores, P_{stm32} es la potencia de consumo del microcontrolador STM32F446RE y P_{xbee} es la potencia de consumo del módulo de

comunicación XBEE S2C. Desarrollando y reemplazando los valores conocidos en la ecuación (59), se tiene lo siguiente:

$$P_T = 4 \cdot V_{motor} \cdot I_{motor} + P_{stm32} + P_{xbee} \quad (60)$$

$$P_T = 4 \cdot 12V \cdot 5.5A + 345mW + 65mW$$

$$P_T = 4 \cdot 12V \cdot 5.5A + 345mW + 65mW$$

$$P_T = 264W + 0.345W + 0.065W$$

$$P_T = 264.41W \approx 265W$$

La batería que se escogió en función de la potencia total requerida por la plataforma omnidireccional ZEEE que se muestra en la Figura 41, sus especificaciones técnicas se detallan en la Tabla 10.

Tabla 10

Especificaciones Técnicas de la Batería Li-Po.

Característica	Valor
Número de Celdas	4S
Voltaje Típico	14.8V
Capacidad Típica	9000mAh \approx 133Wh
Tasa de Descarga Máxima	100C
Peso	735 gr

Nota. Tomado de (Zeee, 2021).

Figura 41

Zeee Batería Li-Po 4S



Nota. Tomado de (Zeee, 2021).

A continuación, se realiza el cálculo de la autonomía que tendrá la plataforma omnidireccional por medio del uso de la presente batería.

$$t_a = \frac{C_{bateria}}{P_T} \quad (61)$$

Donde t_a es el tiempo de autonomía expresado en horas, $C_{bateria}$ es la capacidad típica de la batería especificada en la Tabla 10, y P_T es la potencia total requerida por la plataforma omnidireccional. Reemplazando valores en la ecuación (61) se tiene que el tiempo de autonomía es:

$$t_a = \frac{133Wh}{265W} = 0.502 \text{ horas} \approx 30 \text{ minutos}$$

El tiempo de autonomía determinado es aceptable, tomando en cuenta que el cálculo se realizó con el máximo consumo de los motores, es decir, cuando se aplica la carga máxima. Sin embargo, en este no es el caso debido a que, se está trabajando a menos de la mitad de la capacidad de carga del motor, razón por la cual luego de varias pruebas se determinó que el tiempo de autonomía en la práctica es:

$$t_a = 2 \text{ horas} \approx 120 \text{ minutos}$$

3.6. Construcción de la Plataforma Omnidireccional

El proceso de construcción de la plataforma omnidireccional comienza por el corte láser de las diferentes partes que componen la plataforma omnidireccional en un tablero de densidad media (MDF) de $6mm$ de espesor. Posteriormente se realiza el ensamble de las partes de la estructura como se muestra en Figura 42.

Figura 42

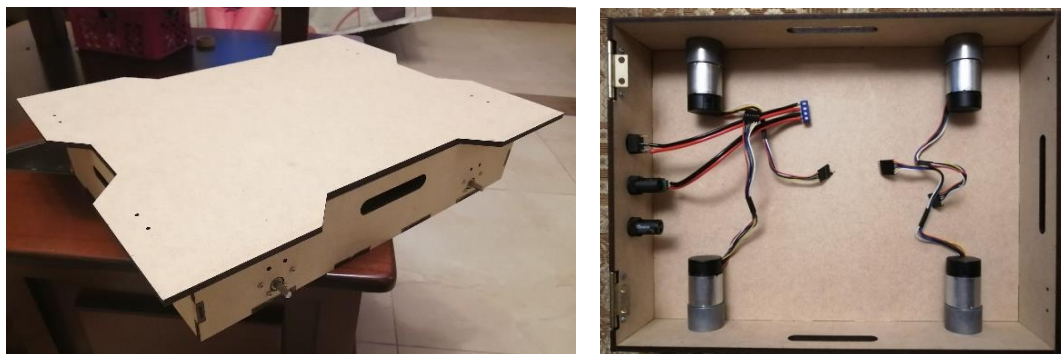
Ensamble de la Estructura



Luego, se realiza el ensamble de los 4 motores a la estructura como se muestra a continuación en la Figura 43:

Figura 43

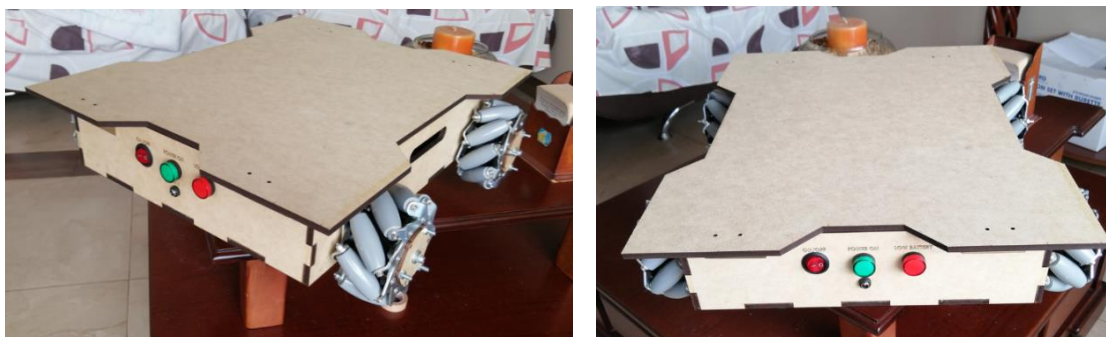
Ensamble de Motores



Finalmente se realiza el ensamble de las 4 ruedas a los respectivos motores respetando la configuración descrita en la Figura 11, como se muestra a continuación:

Figura 44

Ensamble de las Ruedas

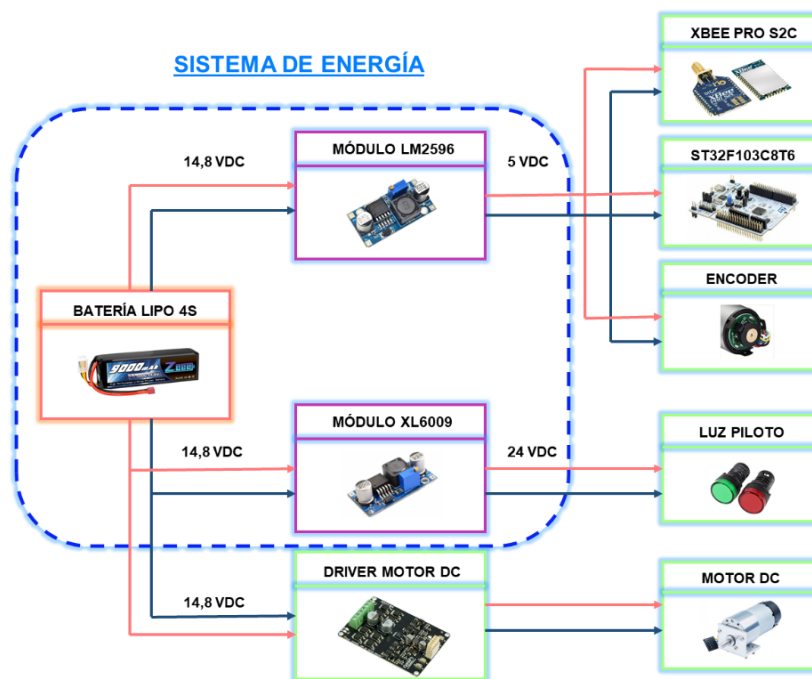


3.7. Diseño Electrónico de la Plataforma Omnidireccional

3.7.1. Sistema de Alimentación

Figura 45

Diagrama de Alimentación

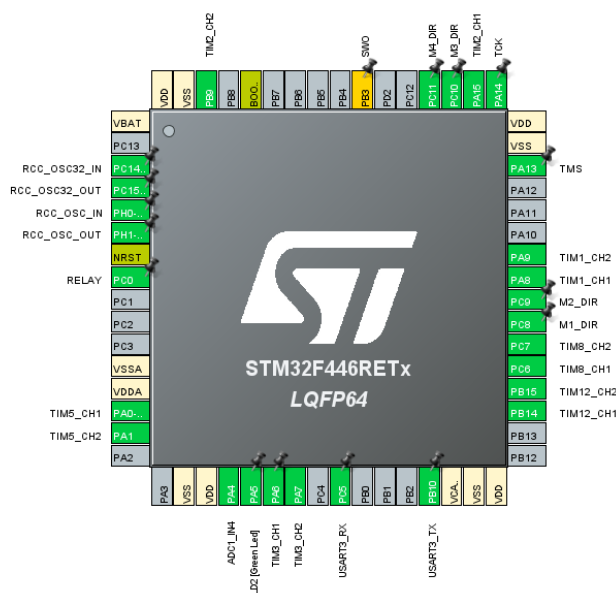


Para dar inicio con el diseño electrónico de la plataforma omnidireccional, se define el sistema de alimentación que proveerá de energía a todo el sistema. Para ello, se optó por implementar la batería seleccionada en la Fuente de Alimentación, y dos módulos electrónicos extra con el fin de entregar los niveles de voltaje adecuados para cada uno de los dispositivos que componen la plataforma omnidireccional. Dichos módulos son el módulo convertidor DC-DC reductor (*LM 2596*) y un módulo convertidor DC-DC elevador (*XL6009*) como se muestra en la Figura 45.

3.7.2. Distribución de pines

Figura 46

Distribución de Pines STM32 NUCLEO-F446RE



Luego de definir el sistema de alimentación se procede a identificar los puertos digitales de entrada y salida de la tarjeta de desarrollo STM32 Nucleo-F446RE, necesarios para realizar el control de la plataforma omnidireccional y a su vez, establecer la comunicación con el PC mediante los módulos Zigbee, es decir, se

necesita 8 entradas digitales para la adquisición de las señales de los 2 canales del encoder de cada uno de los 4 motores, 4 salidas digitales para indicar el sentido de giro al puente H, 4 salidas de PWM para el control de velocidad de los motores, una salida digital para el control de encendido de la luz piloto y por último 1 puerto UART para establecer la comunicación serial con el módulo Zigbee.

La configuración de periféricos, distribución de pines y la programación de la tarjeta de desarrollo se realiza en el software STM32Cube IDE, el cual permite tener una interacción más amigable con el microcontrolador para su configuración como se muestra en la Figura 46.

3.7.3. Tarjeta de Control

El diseño de la tarjeta de control se realizó en el Software Proteus, el cual permitió realizar el diseño del diagrama esquemático del circuito electrónico mostrado en la Figura 47, y su respectivo PCB representado en la Figura 48.

Figura 47

Diagrama Esquemático del Circuito Electrónico

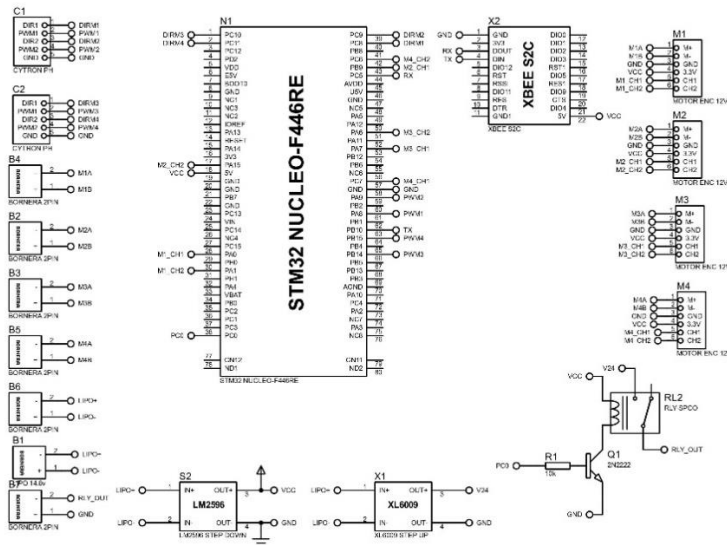
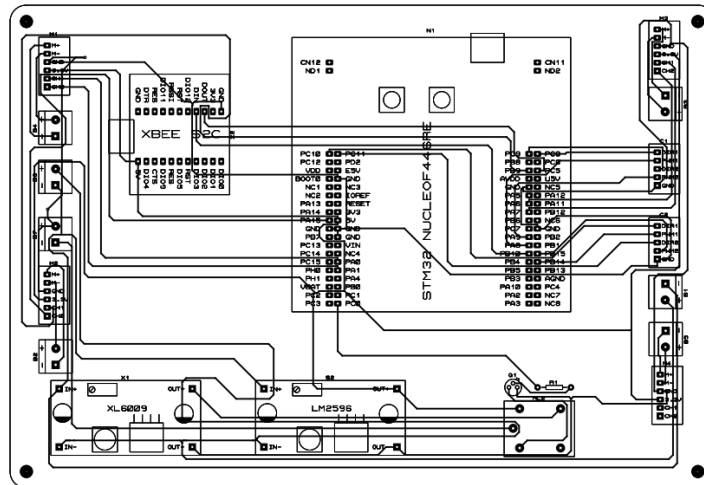


Figura 48

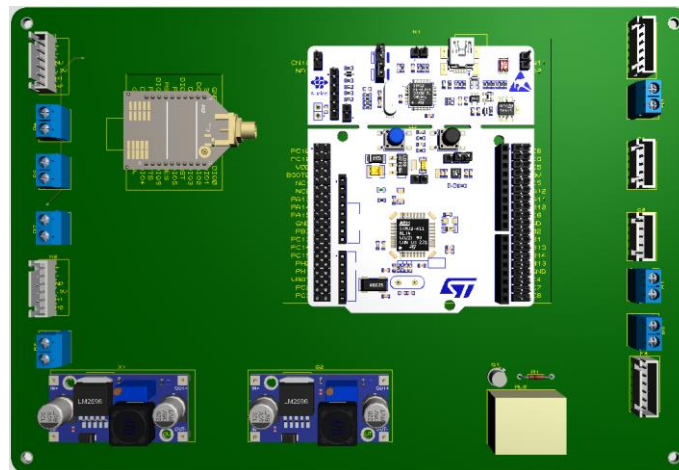
Circuito PCB de la Tarjeta de Control



Además, se realizó la inclusión del diseño CAD de cada dispositivo para tener una apreciación real de la tarjeta de control como se muestra en la Figura 49.

Figura 49

Visualización 3D del PCB



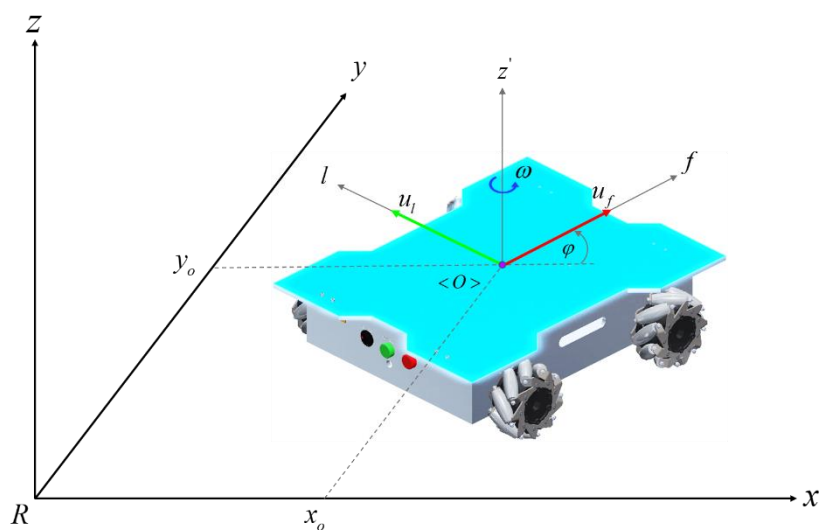
3.8. Modelo Cinemático con Punto de Operación Desplazado

El modelo cinemático de la plataforma omnidireccional descrito en el Modelo Cinemático Robot Omnidireccional, se obtuvo analizando las velocidades que intervienen en el desplazamiento del robot omnidireccional, permitiendo llegar a una expresión matricial que relaciona las velocidades propias del robot con las velocidades lineales de los respectivos ejes x e y . Sin embargo, dicho modelo es válido siempre y cuando el punto de operación con el que se desea trabajar coincida con el marco inercial $\langle O \rangle$ de la plataforma omnidireccional, como se muestra en la Figura 50, lo cual limita a siempre tener que trabajar con un mismo punto de operación.

Para dar solución a esta limitante, se realiza un análisis combinado, en el cual se toma en consideración el caso en el que el punto de operación coincide con $\langle O \rangle$, y a su vez, cuando el punto de operación está desplazado una distancia a , del marco inercial $\langle O \rangle$.

Figura 50

Representación Esquemática de la Plataforma Omnidireccional



Nota. La figura es una representación esquemática de la plataforma omnidireccional cuando se toma como punto de operación el marco inercial $\langle O \rangle$.

Realizando el mismo análisis cinemático descrito en el Modelo Cinemático Robot Omnidireccional para la Figura 50, se llega a las siguientes expresiones:

$$\dot{x}_o = u_f \cos(\varphi) - u_l \sin(\varphi) \quad (62)$$

$$\dot{y}_o = u_f \sin(\varphi) + u_l \cos(\varphi) \quad (63)$$

$$\dot{\varphi} = \omega \quad (64)$$

Donde:

\dot{x}_o, \dot{y}_o : Velocidad lineal en x e y respectivamente, del centro de la plataforma $\langle O \rangle$.

u_l, u_f : Velocidad lateral y frontal respectivamente, del centro de la plataforma $\langle O \rangle$.

$\dot{\varphi}, \omega$: Velocidad angular.

Luego, para obtener el modelo cinemático con el punto de control desplazado, se parte del análisis de la posición de la plataforma omnidireccional en cada uno de los ejes del sistema de coordenadas R , tomando en consideración el desplazamiento a .

Por lo que analizando la

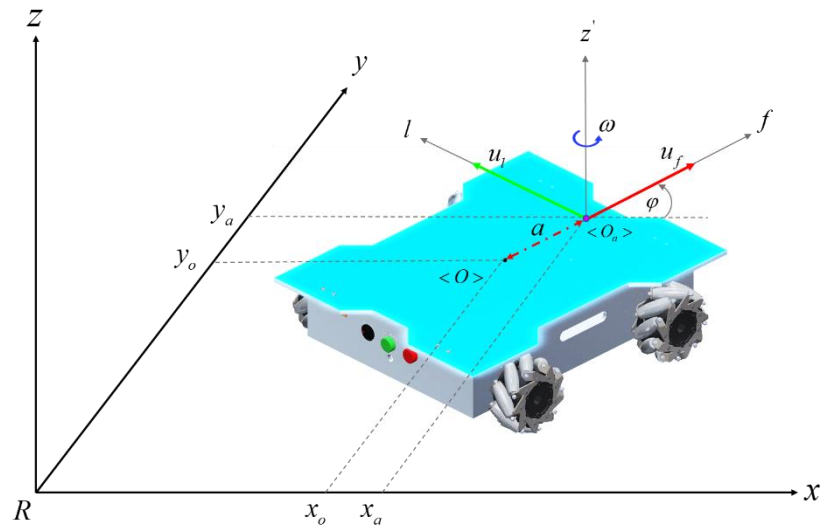
Figura 51 se tiene lo siguiente:

$$x_a = x_o + a \cos(\varphi) \quad (65)$$

$$y_a = y_o + a \sin(\varphi) \quad (66)$$

Figura 51

Representación Cinemática con Punto de Operación Desplazado



Para obtener las velocidades con respecto a este nuevo punto de operación, se aplica la derivada parcial a las ecuaciones (65) y (66) como se muestra a continuación:

$$\dot{x}_a = \dot{x}_o - a \dot{\varphi} \sin(\varphi) \quad (67)$$

$$\dot{y}_a = \dot{y}_o + a \dot{\varphi} \cos(\varphi) \quad (68)$$

Reemplazando las ecuaciones (62) y (63), en las ecuaciones (67) y (68) respectivamente se obtiene:

$$\dot{x}_a = u_f \cos(\varphi) - u_l \sin(\varphi) - a \dot{\varphi} \sin(\varphi) \quad (69)$$

$$\dot{y}_a = u_f \sin(\varphi) + u_l \cos(\varphi) + a \dot{\varphi} \cos(\varphi) \quad (70)$$

Por último, del gráfico representado en la

Figura 51 se sabe que $\dot{\varphi} = \omega$, por lo tanto, el modelo matemático con el punto de operación desplazado queda definido por las siguientes expresiones:

$$\dot{x}_a = u_f \cos(\varphi) - u_l \sin(\varphi) - a \omega \sin(\varphi) \quad (71)$$

$$\dot{y}_a = u_f \sin(\varphi) + u_l \cos(\varphi) + a \omega \cos(\varphi) \quad (72)$$

$$\dot{\varphi} = \omega \quad (73)$$

Las ecuaciones (71),(72) y (73) se las puede expresar de forma matricial de la siguiente manera:

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & -a \sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) & a \cos(\varphi) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_f \\ u_l \\ \omega \end{bmatrix} \quad (74)$$

El modelo cinemático del robot omnidireccional se puede escribir en forma abreviada como se muestra a continuación:

$$\dot{\mathbf{h}}(t) = \mathbf{J}(\varphi)\mathbf{u}(t) \quad (75)$$

Donde $\mathbf{J}(\varphi)$ es la matriz Jacobiana que establece la relación entre las velocidades de la plataforma omnidireccional en el punto de operación $\langle O_a \rangle$, y las velocidades respecto a cada eje del sistema de referencia.

Capítulo IV

4. Entorno servo visual

En el presente capítulo se muestra el proceso referente a la etapa de visión artificial, donde se detallan cada uno de los subprocesos como la adquisición de imágenes, calibración de la cámara, detección de un marcador ArUco y su correspondiente estimación de POSE. Dichas aplicaciones se llevaron a cabo en el entorno de desarrollo integrado (IDE) QtCreator, haciendo uso de la librería para visión artificial OpenCV en el lenguaje de programación C++. Finalmente se definirá los 2 controladores que van a ser objeto de estudio y se analizará la estabilidad de cada uno de ellos.

Figura 52

Etapas del Sistema de Visión Artificial



4.1. Adquisición de Imágenes

La primera etapa del proceso de visión artificial es la Adquisición de la imagen, para ello es necesario definir el dispositivo que permitirá realizar dicha actividad, debido a que en el mercado se encuentra una gran variedad de cámaras, cada una con propiedades específicas y enfocadas a diversas actividades.

En este caso se optó por aprovechar la cámara posterior disponible en un smartphone, específicamente la del Huawei P30 Lite mostrado en la Figura 53, debido a que posee excelentes características (Tabla 11) que van acorde a las necesidades del proyecto de investigación, ya que viene incorporado con un triple sensor que ofrece una excelente resolución en foto y video, facilitando la obtención de información más precisa y exacta de la ubicación de las esquinas del marcador ArUco, lo cual ayudará a disminuir el error en la detección y estimación del mismo.

Figura 53

Cámaras Huawei P30 Lite



Nota. Tomado de (HUAWEI, 2021).

Además, dicha selección se fundamenta en que actualmente la presencia de un smartphone en el diario vivir se ha vuelto sumamente habitual, lo cual facilitará el desarrollo de proyectos futuros relacionados con el tema de investigación propuesto.

Tabla 11*Especificaciones Cámara Posterior Huawei P30 Lite*

Característica	Valor
Cámara de tres lentes	24 MP +8 MP (<i>lente ultra gran angular</i>)+ 2 MP (<i>lente bokeh</i>)
Autofocus	<i>Si</i>
Sensor	<i>CMOS BSI</i>
Video	<i>FHD 1080p 60fps / 30fps</i>
Foto	<i>5632x4224 pixeles</i>

Nota. Tomado de (HUAWEI, 2021).

Sin embargo, para lograr obtener las imágenes captadas por la cámara del smartphone en la PC, se requiere de la implementación de un software. En este caso se ha optado por utilizar el software iVCam, el cual permite lo siguiente:

- ✓ Recepción de video en tiempo real con baja latencia.
- ✓ Configuración de velocidad de fotogramas por segundo.
- ✓ Soporte de tamaños de video 360p, 480p, 540p, 720p, 1080p y 4K.
- ✓ Configuración de calidad de video.
- ✓ Configuración de calidad de audio.
- ✓ Conexión Wi-Fi o USB.
- ✓ Conmutación delantera/trasera.

De este modo el software iVCam permitirá la adquisición de la imagen aprovechando la máxima resolución y velocidad de fps que proporciona la cámara del smartphone (*1080p / 60fps*).

4.2. Calibración de la Cámara

El proceso de calibración de la cámara es indispensable para poder obtener los parámetros intrínsecos de la cámara, necesarios para poder conocer con exactitud la proyección que tiene un punto 3D del mundo en el espacio 2D de la imagen. Como ya se mencionó en la Calibración existen 3 objetos calibradores que son:

- ❖ Tablero de ajedrez clásico en blanco y negro
- ❖ Patrón de círculo simétrico
- ❖ Patrón de círculo asimétrico

En este caso se utilizará el tablero de ajedrez mostrado en la Figura 54, el cual mide *25 cm* de ancho y *17.5 cm* de alto, es decir, cada recuadro tiene *2.5 cm* por lado.

Figura 54

Tablero de Ajedrez



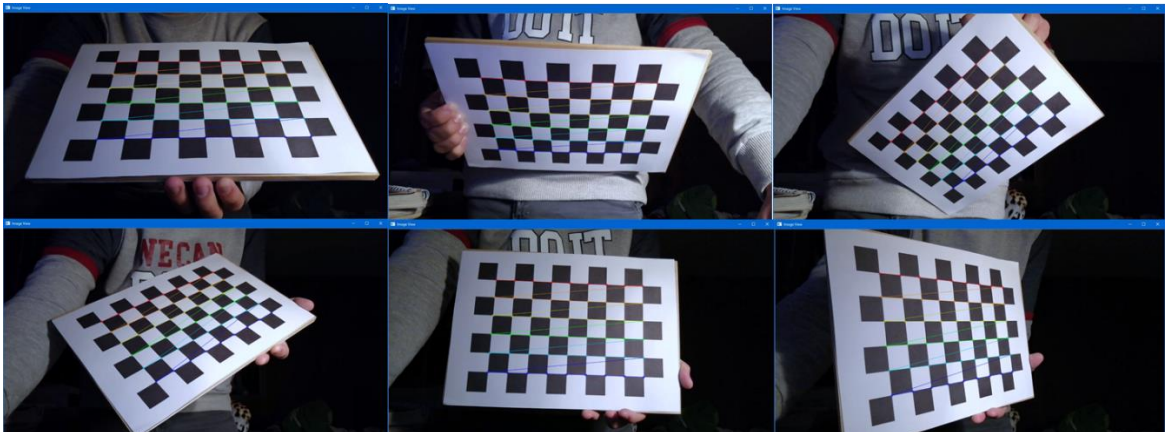
Luego, el proceso de calibración de la cámara se lleva a cabo por medio de la utilización del código fuente disponible en la carpeta de instalación de la librería de OpenCV. El código mencionado se encarga de: determinar la matriz de distorsión, determinar la matriz de la cámara, tomar los archivos de entrada ya sea, video o

imagen, guardar los resultados obtenidos en un archivo XML/YAML y por último calcular el error de re-proyección.

El código de calibración de la cámara requiere que el usuario especifique el número de esquinas internas tanto de ancho como de alto, el tamaño del lado de un recuadro, el número imágenes con las que se va a trabajar y por último el tiempo de espera entre cada captura. De acuerdo con las recomendaciones de OpenCV se debe trabajar con al menos 10 capturas en diferentes posiciones del tablero de ajedrez para obtener una buena aproximación de los parámetros intrínsecos de la cámara.

Figura 55

Imágenes para Calibración de la Cámara



Nota. La figura representa algunas de las imágenes tomadas para el proceso de calibración de la cámara.

Tomando en cuenta que se va a trabajar con el tablero de ajedrez de la Figura 54, los valores con los que se especificó en el código de calibración fueron:

- ❖ Esquinas internas de ancho: 9
- ❖ Esquinas internas de alto: 6
- ❖ Número de capturas de todo el proceso: 12 *frames*

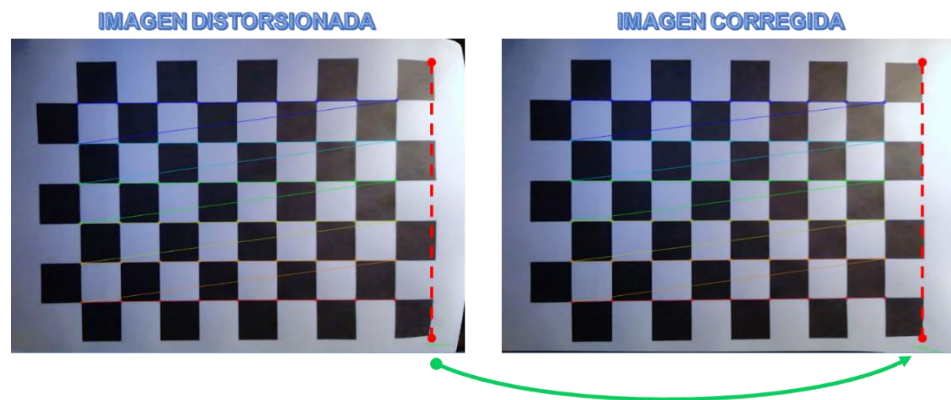
- ❖ Intervalo de tiempo entre cada captura: 5 *seg*

Finalmente se ejecuta el código y se coloca el tablero de ajedrez en diversas posiciones frente a la cámara como se muestra en la Figura 55.

Una vez terminado el proceso de calibración de la cámara, el código permite evidenciar el efecto de la distorsión en la imagen de entrada como se muestra en la Figura 56. El efecto que tiene la distorsión en la imagen varía dependiendo de la cámara, por lo que dicha distorsión podría verse con mayor intensidad en algunos modelos de cámaras más que en otros.

Figura 56

Imagen Distorsionada vs Imagen Corregida



Una vez que se finaliza la aplicación, se genera el archivo YAML, el cual contiene lo siguiente:

- Matriz de la Cámara:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (76)$$

- Matriz de Coeficientes de Distorsión axial y radial:

$$\mathbf{C}_{coef} = [k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3] \quad (77)$$

c. Error de re-proyección

Este último dato que proporciona el archivo YAML, es el encargado de determinar qué tan buena es la aproximación de los parámetros encontrados, mientras más cercano a 0 sea el error de re-proyección, más confiables serán los resultados encontrados. Por lo que el proceso de calibración de la cámara deberá repetirse hasta conseguir una buena aproximación en base a este dato. A continuación, se muestra a manera de ejemplo los resultados obtenidos del proceso de calibración realizado para la cámara mencionada anteriormente.

a. Matriz de la Cámara:

$$\mathbf{K} = \begin{bmatrix} 1.5210181054622121e+03 & 0 & 9.7378643950557341e+02 \\ 0 & 1.5210181054622121e+03 & 5.5368555274148105e+02 \\ 0 & 0 & 1 \end{bmatrix}$$

d. Matriz de Coeficientes de Distorsión axial y radial:

$$\mathbf{d} = \begin{bmatrix} 1.9230200035757158e-01 \\ -1.4198943374882129e+00 \\ -2.8164028016595716e-03 \\ 1.6822106632135511e-03 \\ 2.4581124234371332e+00 \end{bmatrix}$$

e. Error de re-proyección

$$error = 0.4053826900597293$$

4.3. Detección del Marcador ArUco

Para dar inicio con la detección de un marcador ArUco primero hay que definir el diccionario con el cual se va a trabajar. El diccionario es un conjunto de marcadores que

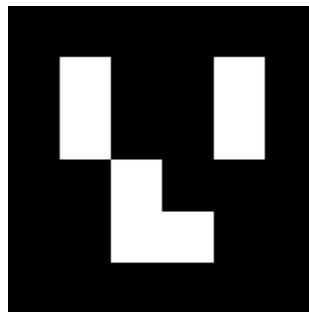
poseen un número determinado de bits en su codificación binaria. El módulo ArUco contiene diccionarios predefinidos de acuerdo a su tamaño y a la dimensión en bits de los marcadores que lo componen.

La selección del diccionario con el cual se va a trabajar es de gran importancia, debido a que el número de bits que se va a detectar se define acorde al diccionario escogido, por lo que hay que tomar en cuenta que a un mayor número de bits que tenga el marcador en su codificación binaria, mayor será la resolución requerida en la etapa de adquisición, para una correcta detección del marcador.

Tomando en consideración lo antes mencionado, el diccionario con el cual se va a trabajar es el *DICT_4X4_50*, el cual contiene 50 marcadores de *4x4bits*. Luego, el marcador con el cual se decida trabajar es a libre elección, por lo que en este caso se hará la detección del marcador 3 (Figura 57) del diccionario mencionado.

Figura 57

Marcador ArUco ID 3



El proceso de detección del marcador se realiza haciendo uso de la función `cv::aruco::detectMarkers()`, incluida en el módulo ArUco. La función mencionada anteriormente recibe los siguientes parámetros:

detectMarkers(image, dictionary, corners, ids, parameters);

1. *image* : Es la imagen de entrada que contiene el marcador.
2. *dictionary* : Es el diccionario al cual pertenece el marcador que se desea detectar.
3. *corners* : Es la estructura de salida que contiene las coordenadas de las esquinas los marcadores detectados
4. *ids* : Es la lista de ids correspondiente de los marcadores detectados.
5. *parameters* : Son los parámetros opcionales de configuración para el proceso de detección de marcadores.

La función descrita facilita el proceso de detección de los marcadores ArUco. Sin embargo, hay que tomar en cuenta que internamente realiza el mismo proceso descrito en el ArUco, por lo que el proceso de detección también se podría hacer manualmente.

Figura 58

Detección del Marcador ArUco



Luego, es posible dibujar los contornos y el id del marcador detectado sobre la imagen a través de la función `cv::aruco::drawDetectedMarkers()`, como se muestra en Figura 58. Esta función recibe la imagen `outputImage`, sobre la cual se va a dibujar y,

por último, los parámetros de salida de la función `cv::aruco::detectMarkers()`, tales como `markerCorners` y `markerIds`. Dicha función está definida así:

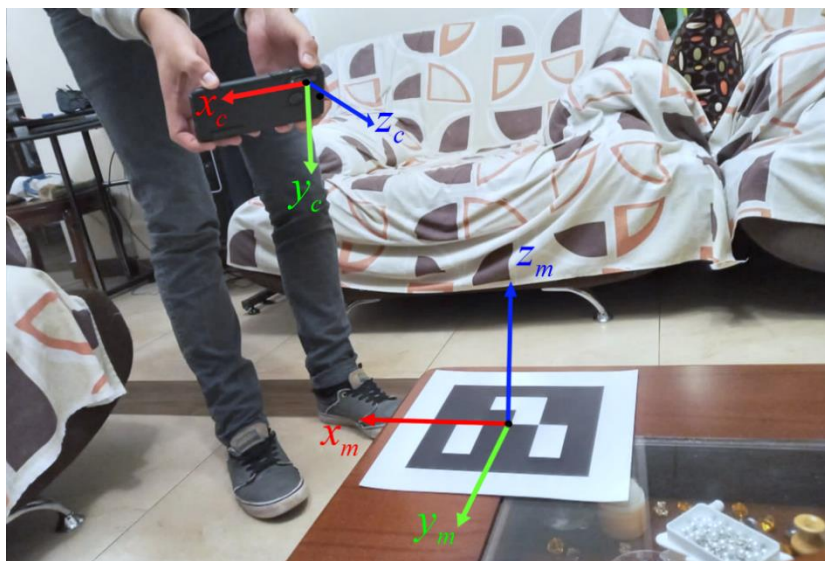
```
drawDetectedMarkers(outputImage,markerCorners,markerIds);
```

4.4. Estimación de Pose del Marcador Aruco

Luego de realizar el proceso de calibración de la cámara y la detección del marcador ArUco, es posible realizar la estimación de la POSE del marcador Aruco. La estimación de la POSE de un marcador ArUco se la puede describir como una transformación 3D del sistema de coordenadas del marcador al sistema de coordenadas de la cámara como se muestra en la Figura 59. Lo cual se puede interpretar como una transformación de rotación y traslación.

Figura 59

Sistema de Coordenadas de la Marcador - Cámara



Nota. La figura muestra el sistema de coordenadas del marcador y el sistema de coordenadas de la cámara ubicados en sus respectivos centros.

El proceso de estimación de la POSE del marcador consiste en determinar los vectores de traslación y rotación que permiten la transformación 3D mencionada anteriormente. Por fortuna, el módulo ArUco cuenta con una función que realiza el proceso de estimación de la POSE de todos los marcadores detectados en la etapa anterior. La función `cv::aruco::estimatePoseSingleMarkers()` encargada de realizar el proceso en cuestión, recibe los siguientes parámetros:

`estimatePoseSingleMarkers(markerCorners, markerLength, cameraMatrix, distCoeffs, rvecs, tvecs);`

- *markerCorners* : Son las coordenadas de las esquinas de los marcadores detectados respectivamente.
- *markerLength* : Es la longitud de uno de los lados del marcador.
- *cameraMatrix, distCoeffs* : Es la matriz y los coeficientes de distorsión de la cámara respectivamente, obtenidos en la etapa de calibración.
- *rvecs, tvecs* : Son los vectores de rotación y traslación estimados, correspondientes a cada uno de los marcadores detectados, estos vectores tienen la siguiente forma:

$$\mathbf{tvec} = [t_x, t_y, t_z] \quad (78)$$

$$\mathbf{rvec} = [r_x, r_y, r_z] \quad (79)$$

Haciendo uso de la función mencionada anteriormente, se realizó el proceso de estimación de la POSE del marcador mostrado en la Figura 58, el cual tiene una longitud de 0.15 m por lado. Obteniendo como resultado el vector de traslación \mathbf{tvec} y el vector de rotación \mathbf{rvec} , respecto al sistema de coordenadas de la cámara. Dichos resultados se muestran a continuación:

$$\mathbf{tvec} = [-0.0517, -0.01673, 0.6125]m$$

$$\mathbf{rvec} = [2.3918, 1.0253, -0.2930]$$

Además, la rotación del marcador con respecto a la cámara, también se puede representar como una matriz de rotación. Dicha matriz se puede determinar a partir un vector de rotación gracias a la fórmula de rotación rodrigues que se muestra a continuación:

$$\mathbf{R} = \cos(\theta)\mathbf{I} + (1 - \cos(\theta))\mathbf{r}\mathbf{r}^T + \sin(\theta) \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \quad (80)$$

Donde:

\mathbf{R} : Es la matriz de rotación.

\mathbf{r} : Es el vector unitario del vector de rotación.

θ : Es la norma del vector de rotación.

Obteniendo como resultado una matriz con la forma:

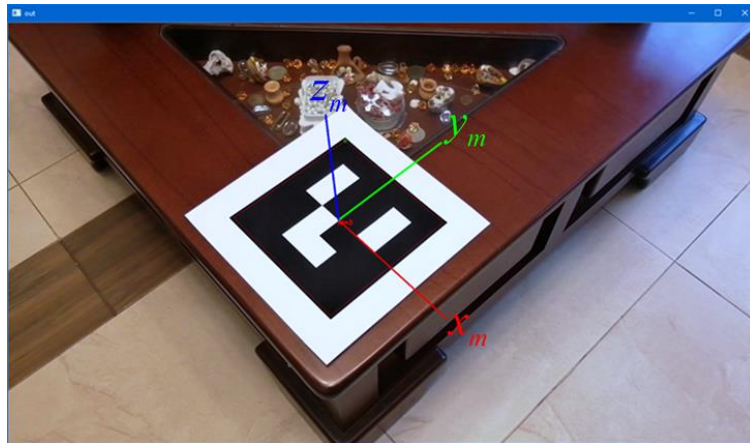
$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (81)$$

OpenCV cuenta con una función `cv::Rodrigues()`, la cual realiza el cálculo de la matriz de rotación a partir de un vector de rotación y viceversa. De este modo, es posible determinar la matriz de rotación \mathbf{R}_{MC} , a partir del vector de rotación \mathbf{rvec} determinado anteriormente, dicha matriz se muestra a continuación:

$$\mathbf{R}_{MC} = \begin{bmatrix} 0.6905 & 0.7233 & 0.0047 \\ 0.6115 & -0.5803 & -0.5377 \\ -0.3862 & 0.3742 & -0.8430 \end{bmatrix}$$

Figura 60

Estimación de la POSE de un Marcador ArUco



Nota. La figura representa el resultado de la estimación de POSE del marcador ArUco, permitiendo apreciar la correcta ubicación de sus ejes coordenados x_m, y_m, z_m .

Una vez obtenido el vector de rotación y el vector de traslación del marcador, es posible graficar los ejes x, y, z , sobre el centro marcador a través de la función `cv::aruco::drawAxis()` como se muestra en la Figura 60. Para ello, es necesario especificar la imagen `outputImage`, en la que se va a graficar los ejes, la matriz de la cámara `cameraMatrix`, los coeficientes de distorsión de la cámara `distCoeffs`, los vectores `rvec` y `tvec`, y por último, la longitud de los ejes `length`. Dicha función se la implementa de la siguiente manera:

```
drawAxis(outputImage, cameraMatrix, distCoeffs, rvec, tvec, length);
```

4.5. Estimación de Pose de la Plataforma Omnidireccional

La estimación de POSE de la plataforma omnidireccional se realiza por medio de la implementación de los marcadores binarios ArUco, debido a lo fácil que resulta estimar los vectores de traslación y rotación de dichos marcadores. Sin embargo, dichos vectores están estimados con respecto al sistema de coordenadas de la cámara, lo cual es poco práctico para la resolución de problema de seguimiento de trayectorias de la plataforma omnidireccional, debido a que la plataforma omnidireccional siempre se estaría desplazando con respecto a un sistema de coordenadas ubicado en el centro de la cámara y no, en el plano por el que se desplaza.

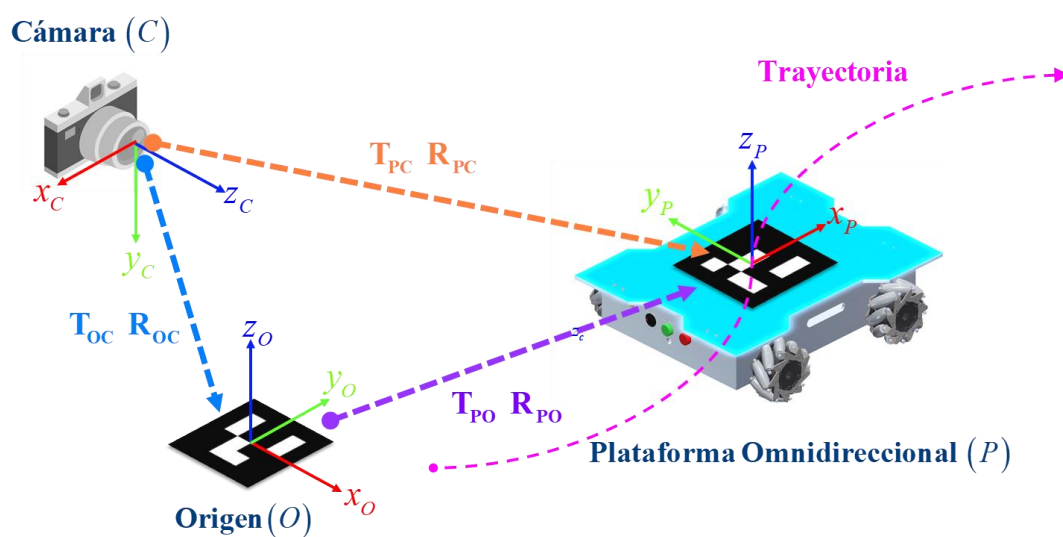
El hecho de que la plataforma omnidireccional se desplace con respecto a un sistema de coordenadas fijado al plano en el que se encuentra, facilita la resolución del seguimiento de trayectorias, debido a que, resulta más fácil identificar manualmente los errores acontecidos en el proceso de estimación de la POSE realizada por cualquier método y, además, es mucho más factible proponer trayectorias parametrizadas en el tiempo con origen en el sistema de coordenadas mencionado.

Por tal motivo, la estimación de la POSE de la plataforma omnidireccional se realiza con respecto a un marcador ubicado en el mismo plano de desplazamiento. Lo cual permite fijar un sistema de coordenadas en el mismo plano de la plataforma omnidireccional y a su vez, estimar la POSE con respecto a este centro de coordenadas. El sistema servo visual propuesto se muestra en la

Figura 61.

Figura 61

Sistema Servo Visual Propuesto



Nota. La figura es una representación del desplazamiento de la plataforma omnidireccional con respecto al sistema de coordenadas (O) , donde T y R son los vectores de traslación y rotación respectivamente de los sistemas de coordenadas mostrados.

4.5.1. *Traslación*

Analizando la

Figura 61, se puede definir la posición de la plataforma omnidireccional con respecto a la cámara de la siguiente manera:

$$\mathbf{T}_{PC} = \mathbf{R}_{OC} \mathbf{T}_{PO} + \mathbf{T}_{OC} \quad (82)$$

Donde:

\mathbf{T}_{PC} : Es la posición de la plataforma omnidireccional P , con respecto a la cámara C .

\mathbf{R}_{OC} : Es la matriz de rotación del marcador de origen O , con respecto a la cámara C .

\mathbf{T}_{PO} : Es el vector de traslación de la plataforma P , con respecto al origen O .

\mathbf{T}_{OC} : Es el vector de traslación del origen O , con respecto a la cámara C .

Sin embargo, \mathbf{T}_{PC} , \mathbf{R}_{OC} y \mathbf{T}_{OC} de la ecuación (82), son datos conocidos que se obtienen al estimar la POSE de cada uno de los marcadores mostrados en la

Figura 61, con respecto a la cámara. De modo que la posición de la plataforma omnidireccional P , con respecto al marcador de origen O , se expresa como:

$$\mathbf{T}_{PC} = \mathbf{R}_{OC} \mathbf{T}_{PO} + \mathbf{T}_{OC}$$

$$\mathbf{T}_{PC} - \mathbf{T}_{OC} = \mathbf{R}_{OC} \mathbf{T}_{PO}$$

$$\mathbf{R}_{OC}^{-1} (\mathbf{T}_{PC} - \mathbf{T}_{OC}) = \mathbf{R}_{OC}^{-1} \mathbf{R}_{OC} \mathbf{T}_{PO}$$

$$\mathbf{T}_{PO} = \mathbf{R}_{OC}^{-1} (\mathbf{T}_{PC} - \mathbf{T}_{OC}) \quad (83)$$

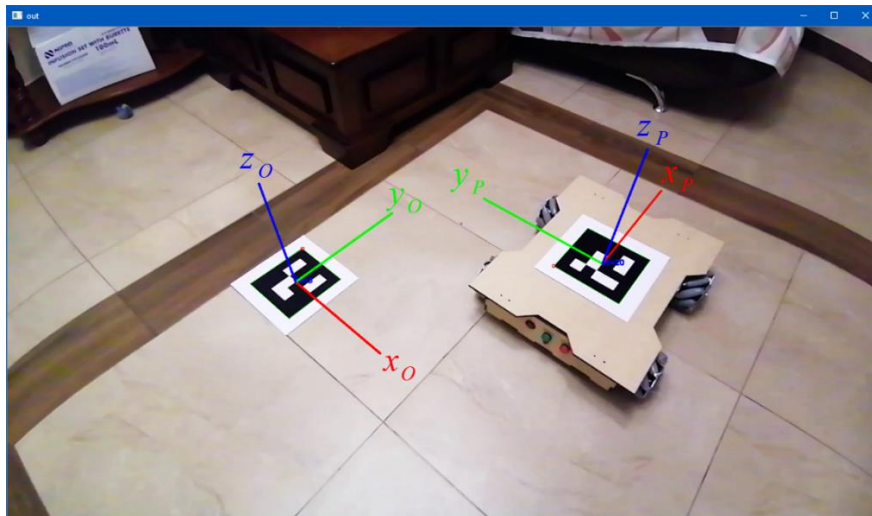
Una de las características principales de una matriz de rotación es que son matrices ortogonales, es decir, la inversa de la matriz de rotación coincide con su transpuesta. Por lo que reescribiendo la ecuación (83), se tiene lo siguiente:

$$\mathbf{T}_{PO} = \mathbf{R}_{OC}^T (\mathbf{T}_{PC} - \mathbf{T}_{OC}) \quad (84)$$

La ecuación (84), permite determinar la posición de de la plataforma omnidireccional con respecto a un segundo marcador, en este caso el segundo marcador está fijado al plano de desplazamiento de la plataforma como se muestra en la Figura 62.

Figura 62

Estimación de Pose de la Plataforma Omnidireccional



4.5.2. Rotación

Del mismo modo, analizando la

Figura 61, se puede definir la orientación de la plataforma omnidireccional con respecto a la cámara de la siguiente manera:

$$\mathbf{R}_{PC} = \mathbf{R}_{OC} \mathbf{R}_{PO} \quad (85)$$

Donde:

\mathbf{R}_{PC} : Es la posición de la matriz de rotación de la plataforma omnidireccional P , con respecto a la cámara C .

\mathbf{R}_{OC} : Es la matriz de rotación del marcador de origen O , con respecto a la cámara C .

\mathbf{R}_{PO} : Es la posición de la matriz de rotación de la plataforma omnidireccional P , con respecto al origen O .

Del mismo modo, las matrices de rotación \mathbf{R}_{OC} y \mathbf{R}_{PC} se obtiene mediante la estimación de los respectivos marcadores. Por lo tanto, a partir de la ecuación (85), se puede definir lo siguiente:

$$\mathbf{R}_{PC} = \mathbf{R}_{OC} \mathbf{R}_{PO}$$

$$\mathbf{R}_{OC}^{-1} \mathbf{R}_{PC} = \mathbf{R}_{OC}^{-1} \mathbf{R}_{OC} \mathbf{R}_{PO}$$

$$\mathbf{R}_{PO} = \mathbf{R}_{OC}^{-1} \mathbf{R}_{PC} \quad (86)$$

Así mismo, por la característica ortogonal de una matriz de rotación, la ecuación (86) se reescribe de la siguiente manera :

$$\mathbf{R}_{PO} = \mathbf{R}_{OC}^{-T} \mathbf{R}_{PC} \quad (87)$$

La ecuación (87), permite determina la rotación de la plataforma omnidireccional respecto a un segundo marcador ArUco, en este caso el segundo marcador está fijado al plano de desplazamiento de la plataforma como se muestra en la Figura 62.

4.5.3. Ángulos de Euler

Una vez que se conoce la matriz de rotación de la plataforma omnidireccional P , con respecto al marcador ArUco ubicado en el sistema de coordenadas de referencia y , considerando que dicha matriz está expresada de la forma:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Es posible determinar los ángulos de Euler $roll(\phi)$, $pitch(\theta)$ y $yaw(\psi)$, a través de la implementación de las ecuaciones (17), (18) y (19) respectivamente. Cabe recalcar que para la resolución del seguimiento de trayectorias sólo se requiere conocer el valor estimado de rotación con respecto al eje z , es decir, el ángulo $yaw(\psi)$.

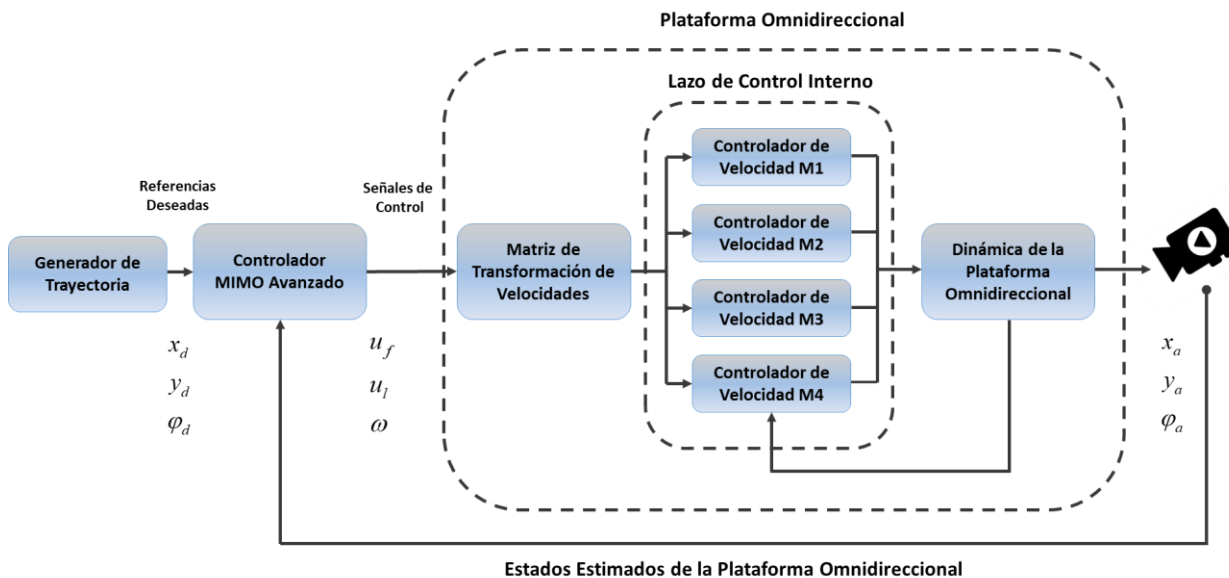
$$\psi_{yaw} = \arctan\left(\frac{r_{21}}{r_{11}}\right)$$

4.6. Control de la Plataforma Omnidireccional

El modelo matemático de la plataforma omnidireccional se caracteriza por ser un particular sistema MIMO con comportamiento no lineal, motivo por el cual se dificulta el análisis de este tipo de sistemas. Actualmente, existen varios algoritmos que permiten realizar el control de estos sistemas MIMO no lineales, una buena alternativa son los controladores cinemáticos, los cuales permiten disminuir la cota de los errores de control de visión artificial.

Figura 63

Diagrama de Bloques del Sistema Servo Visual



En este apartado se hará el estudio de 2 controladores cinemáticos con el objetivo de analizar su comportamiento y su estabilidad al intentar dar solución al problema de seguimiento de trayectorias del sistema servo visual mostrado en la Figura 63. Dicho sistema representa un gran desafío para los algoritmos de control modernos mencionados debido a los posibles errores de medición que podrían surgir en la etapa Controlador Cinemático No Lineal

4.6.1. Controlador Cinemático No Lineal

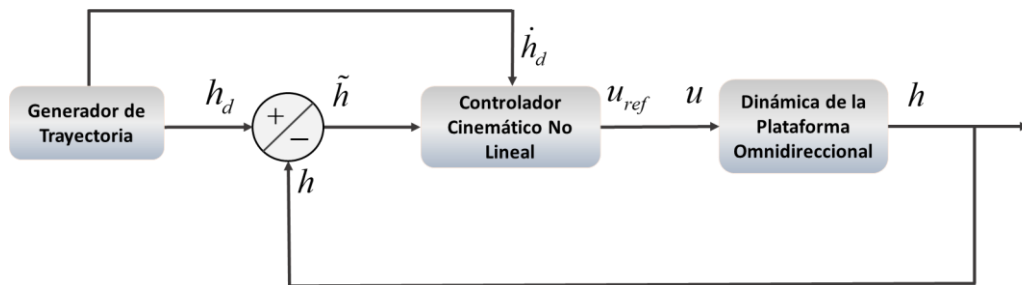
El primer controlador moderno que se va a estudiar es un Controlador Cinemático No Lineal, el cual se basa en el comportamiento cinemático inverso de la plataforma omnidireccional, las velocidades de referencia deseadas y una matriz de peso para tener control sobre la velocidad con la que evoluciona el sistema como se muestra en la Figura 64. Entonces, partiendo del modelo cinemático de la plataforma omnidireccional representado en la ecuación (75), se tiene:

$$\dot{\mathbf{h}} = \mathbf{J}\mathbf{u} \quad (88)$$

Donde $\dot{\mathbf{h}}$ representa el vector de velocidades respecto a cada eje del sistema de referencia, \mathbf{J} es la matriz jacobiana y \mathbf{u} es el vector de velocidades de la plataforma omnidireccional.

Figura 64

Diagrama de Bloques del Sistema de Control No Lineal



En función del modelo cinemático descrito se plantea la siguiente ley de control.

$$\mathbf{u}_{ref} = \mathbf{J}^{-1}(\dot{\mathbf{h}}_d + \mathbf{K}\tilde{\mathbf{h}}) \quad (89)$$

Donde:

\mathbf{u}_{ref} : Vector de velocidades de control.

\mathbf{J}^{-1} : Matriz Jacobiana Inversa.

$\dot{\mathbf{h}}_d$: Vector de velocidades deseadas.

\mathbf{K} : Matriz de ganancia.

$\tilde{\mathbf{h}}$: Vector de errores de control.

Análisis de Estabilidad

El análisis de estabilidad de sistemas no lineales, de manera general, se basa la teoría Lyapunov, la cual consiste en definir una función candidata de Lyapunov que cumpla con las siguientes condiciones y analizar su derivada parcial:

- $V(x)$ Es una función de estados continua y diferenciable.
- $V(x)$ Es una función de estados definida positiva, es decir:

$$\begin{cases} V(x) = 0 & \text{si } x = 0 \\ V(x) > 0 & \forall x \neq 0 \end{cases}$$

Por lo general la función candidata de Lyapunov es función del error cuadrático medio mostrada a continuación:

$$V(\tilde{\mathbf{h}}) = \frac{1}{2} \tilde{\mathbf{h}}^T \tilde{\mathbf{h}} \quad (90)$$

Reemplazando el vector de estados en la ecuación (90) se tiene lo siguiente:

$$V(\tilde{\mathbf{h}}) = \frac{1}{2} \begin{bmatrix} \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 \end{bmatrix} \begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{bmatrix} \quad (91)$$

Si se reemplaza un vector cualquiera y un vector nulo en la ecuación (91), se observa que cumple con las condiciones mencionadas, por lo tanto, la función representada en la ecuación (90) es una función candidata de Lyapunov.

El método sigue con el análisis de la derivada parcial de la función candidata de Lyapunov, por lo tanto, aplicando la derivada parcial a la ecuación (90) se tiene:

$$\dot{V}(\tilde{\mathbf{h}}) = \frac{d}{dt} V(\tilde{\mathbf{h}}) \quad (92)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \frac{1}{2} \left[\dot{\tilde{\mathbf{h}}}^T \tilde{\mathbf{h}} + \tilde{\mathbf{h}}^T \dot{\tilde{\mathbf{h}}} \right] \quad (93)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \frac{1}{2} \left(\begin{array}{c} \left[\begin{array}{ccc} \dot{\tilde{h}}_1 & \dot{\tilde{h}}_2 & \dot{\tilde{h}}_3 \end{array} \right] \left[\begin{array}{c} \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{array} \right] + \left[\begin{array}{ccc} \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 \end{array} \right] \left[\begin{array}{c} \dot{\tilde{h}}_1 \\ \dot{\tilde{h}}_2 \\ \dot{\tilde{h}}_3 \end{array} \right] \end{array} \right) \quad (94)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \frac{1}{2} \left[\dot{\tilde{h}}_1 \tilde{h}_1 + \dot{\tilde{h}}_2 \tilde{h}_2 + \dot{\tilde{h}}_3 \tilde{h}_3 + \tilde{h}_1 \dot{\tilde{h}}_1 + \tilde{h}_2 \dot{\tilde{h}}_2 + \tilde{h}_3 \dot{\tilde{h}}_3 \right] \quad (95)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \frac{1}{2} \left[2\dot{\tilde{h}}_1 \tilde{h}_1 + 2\dot{\tilde{h}}_2 \tilde{h}_2 + 2\dot{\tilde{h}}_3 \tilde{h}_3 \right] \quad (96)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \dot{\tilde{h}}_1 \tilde{h}_1 + \dot{\tilde{h}}_2 \tilde{h}_2 + \dot{\tilde{h}}_3 \tilde{h}_3 \quad (97)$$

$$\dot{V}(\tilde{\mathbf{h}}) = \tilde{\mathbf{h}}^T \dot{\tilde{\mathbf{h}}} \quad (98)$$

La teoría de Lyapunov especifica que un sistema es asintóticamente estable siempre y cuando $\dot{V}(\tilde{\mathbf{h}})$ sea una función definida negativa, es decir, que cumpla con las siguientes condiciones:

$$\begin{cases} \dot{V}(x) = 0 & \text{sii } x = 0 \\ \dot{V}(x) < 0 & \forall x \neq 0 \end{cases}$$

Sin embargo, para definir la estabilidad del sistema, se necesita determinar la ecuación en lazo cerrado del sistema mostrado en la Figura 64. Para ello se debe definir el error de control según el diagrama de bloques así:

$$\tilde{\mathbf{h}} = \mathbf{h}_d - \mathbf{h} \quad (99)$$

Aplicando la derivada parcial a la ecuación (99) y despejando $\dot{\tilde{\mathbf{h}}}$:

$$\dot{\tilde{\mathbf{h}}} = \dot{\mathbf{h}}_d - \dot{\mathbf{h}}$$

$$\dot{\mathbf{h}} = \dot{\mathbf{h}}_d - \dot{\tilde{\mathbf{h}}} \quad (100)$$

Luego, tomando en cuenta que la salida del controlador es igual a la entrada de control de la plataforma omnidireccional, es decir:

$$\mathbf{u}_{ref} = \mathbf{u} \quad (101)$$

Reemplazando la ecuación (101) en la ecuación (88):

$$\dot{\mathbf{h}} = \mathbf{J}\mathbf{u}_{ref} \quad (102)$$

Y reemplazando la ecuación (89) en la ecuación (102) y resolviendo:

$$\begin{aligned} \dot{\mathbf{h}} &= \mathbf{J}\mathbf{J}^{-1}(\dot{\mathbf{h}}_d + \mathbf{K}\tilde{\mathbf{h}}) \\ \dot{\mathbf{h}} &= \dot{\mathbf{h}}_d + \mathbf{K}\tilde{\mathbf{h}} \end{aligned} \quad (103)$$

Reemplazando la ecuación (100) en la ecuación (103) y reduciendo términos semejantes se obtiene la ecuación en lazo cerrado del sistema mostrado en la Figura 64:

$$\begin{aligned} \dot{\mathbf{h}}_d - \dot{\tilde{\mathbf{h}}} &= \dot{\mathbf{h}}_d + \mathbf{K}\tilde{\mathbf{h}} \\ \dot{\tilde{\mathbf{h}}} &= -\mathbf{K}\tilde{\mathbf{h}} \end{aligned} \quad (104)$$

Finalmente, se reemplaza la ecuación en lazo cerrado de la ecuación (104) en la derivada parcial de la función candidata de Lyapunov de la ecuación (98):

$$\dot{V}(\tilde{\mathbf{h}}) = -\tilde{\mathbf{h}}^T \mathbf{K} \tilde{\mathbf{h}} \quad (105)$$

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \quad (106)$$

$$\dot{V}(\tilde{\mathbf{h}}) = -k_1\tilde{h}_1^2 - k_2\tilde{h}_2^2 - k_3\tilde{h}_3^2 \quad (107)$$

Analizando $\dot{V}(\tilde{\mathbf{h}})$ de la ecuación (107) de acuerdo a las condiciones de estabilidad de la teoría de Lyapunov, se concluye que el sistema es Globalmente Uniformemente Asintóticamente Estable siempre y cuando la matriz \mathbf{K} sea una matriz diagonal definida positiva, es decir, que todos sus autovalores sean mayores a cero, o sea, $\lambda_k > 0$. Esto es debido a que solo para dicho caso la función $\dot{V}(\tilde{\mathbf{h}})$ es una función definida negativa, por lo tanto, cumple con las condiciones de estabilidad del método.

Finalmente se añade la función \tanh a la ley de control descrita en la ecuación (89), con el fin de saturar los errores de control y lograr un comportamiento más suave del sistema, por lo que el controlador queda definido de la siguiente manera:

$$\mathbf{u}_{ref} = \mathbf{J}^{-1} \left(\dot{\mathbf{h}}_d + \mathbf{K} \tanh(\tilde{\mathbf{h}}) \right) \quad (108)$$

4.6.2. Controlador Cinemático No Lineal Discreto de Alto Nivel

El segundo controlador que va a ser objeto de estudio es un Controlador Cinemático No Lineal Discreto de Alto Nivel. Este controlador tiene su fundamento teórico a partir de las cadenas de Markov y fue diseñado tomando en consideración la velocidad con la que evoluciona el sistema de un estado a otro. Además, al igual que el controlador no lineal continuo, también considera el comportamiento cinemático inverso, la referencia deseada y el tiempo de muestreo para ofrecer una señal de control adecuada para el seguimiento de trayectoria.

Para poder trabajar con este controlador, es necesario definir el modelo cinemático discreto de la plataforma omnidireccional, entonces, partiendo del modelo cinemático continuo descrito en la ecuación (88), se tiene:

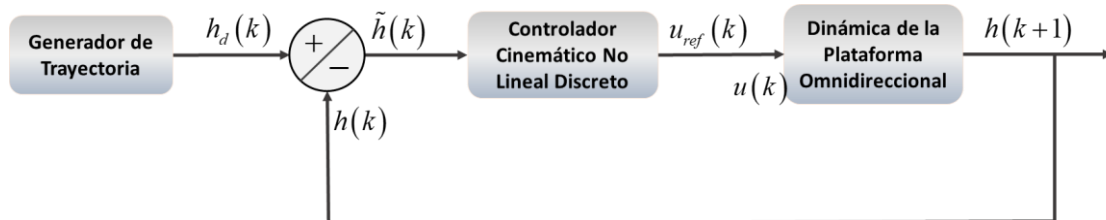
$$\dot{\mathbf{h}} = \mathbf{J}\mathbf{u}$$

$$\frac{\mathbf{h}(k+1) - \mathbf{h}(k)}{T_o} = \mathbf{J}(k)\mathbf{u}(k)$$

$$\mathbf{h}(k+1) = \mathbf{h}(k) + T_o\mathbf{J}(k)\mathbf{u}(k) \quad (109)$$

Figura 65

Diagrama de Bloques del Sistema de Control No Lineal Discreto



La ecuación (109) representa el modelo discretizado de la plataforma omnidireccional, por lo que en función del modelo matemático descrito (Andaluz et al., 2017), propone la siguiente ley de control.

$$\mathbf{u}_{\text{ref}}(k) = \frac{1}{T_o} \mathbf{J}^{-1}(k) \left[\mathbf{h}_d(k+1) - \mathbf{h}(k) - \mathbf{W}(\mathbf{h}_d(k) - \mathbf{h}(k)) \right] \quad (110)$$

Donde:

\mathbf{u}_{ref} : Vector de Velocidades de Control

\mathbf{J}^{-1} : Matriz Jacobiana Inversa

\mathbf{h}_d : Referencia Deseada

\mathbf{h} : Vector de Estados del Sistema

\mathbf{W} : Matriz de Peso

Análisis de Estabilidad

El análisis de estabilidad para este controlador moderno no lineal discreto, consiste en analizar el comportamiento de los errores de control luego de varios instantes de muestreo. Dicho esto, el análisis parte de la siguiente consideración:

$$\mathbf{u}_{ref}(k) = \mathbf{u}(k) \quad (111)$$

Reemplazando la ecuación (111) en el modelo cinemático discreto representado en la ecuación (109) se obtiene:

$$\mathbf{h}(k+1) - \mathbf{h}(k) = T_o \mathbf{J}(k) \mathbf{u}_{ref}(k) \quad (112)$$

Reemplazando la ley de control descrita en la ecuación (110) en la ecuación (112) y reduciendo términos semejantes se tiene lo siguiente:

$$\mathbf{h}(k+1) - \mathbf{h}(k) = T_o \mathbf{J}(k) \frac{1}{T_o} \mathbf{J}^{-1}(k) [\mathbf{h}_d(k+1) - \mathbf{h}(k) - \mathbf{W}(\mathbf{h}_d(k) - \mathbf{h}(k))] \quad (113)$$

$$\mathbf{h}(k+1) - \mathbf{h}(k) = \mathbf{h}_d(k+1) - \mathbf{h}(k) - \mathbf{W}(\mathbf{h}_d(k) - \mathbf{h}(k)) \quad (114)$$

$$\mathbf{h}(k+1) - \mathbf{h}_d(k+1) = -\mathbf{W}(\mathbf{h}_d(k) - \mathbf{h}(k)) \quad (115)$$

$$\tilde{\mathbf{h}}(k+1) = \mathbf{W}(\tilde{\mathbf{h}}(k)) \quad (116)$$

Mediante la expresión obtenida en la ecuación (116), es posible analizar la evolución de los errores de control de un instante anterior a un instante actual, lo cual determinará la estabilidad del sistema. Por lo que dando valores de k a dicha expresión se tiene:

$$\tilde{\mathbf{h}}(2) = \mathbf{W}(\tilde{\mathbf{h}}(1)) \quad (117)$$

$$\tilde{\mathbf{h}}(3) = \mathbf{W}(\tilde{\mathbf{h}}(2)) \quad (118)$$

$$\tilde{\mathbf{h}}(4) = \mathbf{W}(\tilde{\mathbf{h}}(3)) \quad (119)$$

Reemplazando la ecuación (117) en la ecuación (118):

$$\begin{aligned} \tilde{\mathbf{h}}(3) &= \mathbf{W}(\mathbf{W}(\tilde{\mathbf{h}}(1))) \\ \tilde{\mathbf{h}}(3) &= \mathbf{W}^2(\tilde{\mathbf{h}}(1)) \end{aligned} \quad (120)$$

Ahora, reemplazando la ecuación (120) en la ecuación (119) se obtiene la siguiente expresión:

$$\begin{aligned} \tilde{\mathbf{h}}(4) &= \mathbf{W}(\mathbf{W}^2(\tilde{\mathbf{h}}(1))) \\ \tilde{\mathbf{h}}(4) &= \mathbf{W}^3(\tilde{\mathbf{h}}(1)) \end{aligned} \quad (121)$$

De modo que se puede extraer una expresión generalizada de la evolución de los errores de control con respecto al error en el instante $k = 1$ de la siguiente manera:

$$\tilde{\mathbf{h}}(n+1) = \mathbf{W}^n(\tilde{\mathbf{h}}(1)) \quad (122)$$

De la cual se puede concluir que en un instante infinito con una matriz de ganancia diagonal \mathbf{W} , el sistema es Globalmente Uniformemente Asintóticamente Estable siempre y cuando:

$$h(k) \rightarrow 0 \quad \text{sii} \quad 0 < w < 1$$

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \quad (123)$$

Capítulo v

5. Pruebas y resultados

En este capítulo se realizarán las pruebas y se analizarán los resultados obtenidos con el objetivo de evaluar el correcto funcionamiento de la plataforma omnidireccional al evaluar algoritmos de control moderno en el seguimiento de trayectorias. Para ello, se parte del análisis de confiabilidad de los datos de POSE estimados a través del proceso de visión artificial. Luego se realizarán múltiples ensayos que pondrán a prueba el desempeño de los controladores modernos para posteriormente contrarrestar estos resultados con un algoritmo de control clásico y, finalmente se realizará el proceso de validación de la hipótesis.

5.1. Estimación de la POSE

Con anterioridad se realizó la descripción del proceso de estimación de POSE a través de la implementación de los marcadores binarios ArUco, donde se indicó que la fiabilidad de los datos estimados se relacionaba con el error de re-proyección resultante del proceso de calibración de la cámara. Sin embargo, es preciso evaluar la confiabilidad de los datos en la práctica, debido a que la eficiencia y el correcto funcionamiento dependen de dichos valores.

Los resultados obtenidos de las 10 pruebas realizadas colocando los marcadores a diferentes distancias entre sí. Se muestran en la Tabla 12 donde x_e, y_e, φ_e representan los datos estimados, x_r, y_r, φ_r son los datos reales obtenidos de forma manual y por último e_x, e_y y e_φ representan el error correspondiente a cada una de las variables.

Figura 66*Validación de Datos de POSE Estimados***Tabla 12***Resultados de la Validación de la POSE*

$x_r [m]$	$y_r [m]$	$\varphi_r [^\circ]$	$x_e [m]$	$y_e [m]$	$\varphi_e [^\circ]$	$e_x [m]$	$e_y [m]$	$e_\varphi [^\circ]$
0	1.5	0	-0.02	1.52	-0.17	0.02	-0.02	0.17
0.5	0.5	90	0.53	0.48	91.13	-0.03	0.02	-1.13
1	0	0	0.99	-0.14	0.02	0.01	0.14	-0.02
0.75	0.75	45	0.77	0.75	46.12	-0.02	0	-1.12
1.2	0	180	1.19	0.01	179.65	0.01	-0.01	0.35
0	0.5	20	0.03	0.56	21.53	-0.03	-0.06	-1.53
2.5	1.5	0	2.53	1.49	0.11	-0.03	0.01	-0.11
1.1	1	-45	1.08	1.03	-45.5	0.02	-0.03	0.5
1.5	1.5	-90	1.51	1.51	-89.7	-0.01	-0.01	-0.3
1.75	2	60	1.74	2.01	61.5	0.01	-0.01	-1.5

Como se puede observar el error presente en x, y no supera el valor de los 3 cm en todas las pruebas realizadas, por otro lado, el error de estimación para del ángulo yaw es relativamente bajo debido a que no supera el valor de los 2° tanto para ángulos positivos como para ángulos negativos. La obtención de estos resultados es posible gracias al factor de error de re-proyección de 0.4 obtenido en la Calibración de la Cámara, debido a que en la documentación de OpenCV para el módulo ArUco, se menciona que mientras más cercano a cero sea este coeficiente más confiable serán los valores estimados tanto de posición como de orientación.

5.2. Interfaz Gráfica

Figura 67

Interfaz Gráfica



El sistema servovisual es controlado mediante la interfaz gráfica mostrada en la Figura 67, la cual permite al usuario mover la plataforma omnidireccional de forma manual, modificar los parámetros del sistema como son: el desfase del punto de

operación tanto de posición como de orientación, seleccionar y configurar la trayectoria deseada, definir el tiempo de implementación del seguimiento de trayectoria, escoger el controlador que se desea implementar, modificar la matriz de ganancia y de peso de los controladores modernos, ubicar el punto de origen del sistema de referencia, visualizar la posición y orientación actual del robot y por último ejecutar el control de trayectoria.

5.3. Evaluación de Algoritmos de Control Moderno

En este apartado se realizarán las pruebas que ayudarán a demostrar la eficiencia y robustez de los algoritmos de control moderno analizados en el Control de la Plataforma Omnidireccional. Para ello se realizarán las pruebas de seguimiento de trayectoria con 4 curvas parametrizadas en el tiempo, una evaluación de respuesta ante perturbaciones de los controladores modernos, y un último estudio de la eficiencia de los controladores cuando se desfasa el punto de operación del centro inercial de la plataforma omnidireccional. Finalmente se implementará un algoritmo de control clásico con el fin de contrarrestar los resultados de seguimiento obtenidos con control moderno.

5.3.1. Trayectoria 1

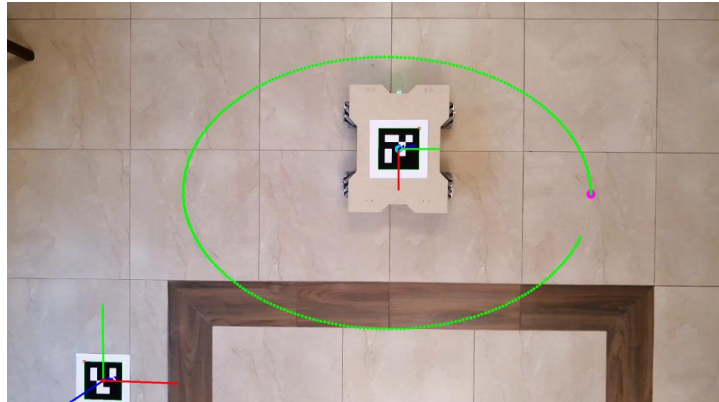
Para la trayectoria 1 se tomó en consideración la ecuación paramétrica de la elipse descrita en la ecuación (124) y representada en la

Figura 68.

$$\begin{aligned}x &= 0.75 \cos(0.2t) + 1.1 \\y &= 0.5 \cos(0.2t) + 0.75\end{aligned}\tag{124}$$

Figura 68

Trayectoria 1 Elipse



Dado que la curva paramétrica es una curva suave, no presentó gran problema la resolución del seguimiento de trayectoria con la implementación de los dos algoritmos de control moderno como se puede observar Figura 69, obteniendo como resultado curvas muy similares tanto del algoritmo de Control No Lineal como del Control No Lineal Discreto. Dichos resultados contrastados se observan en la

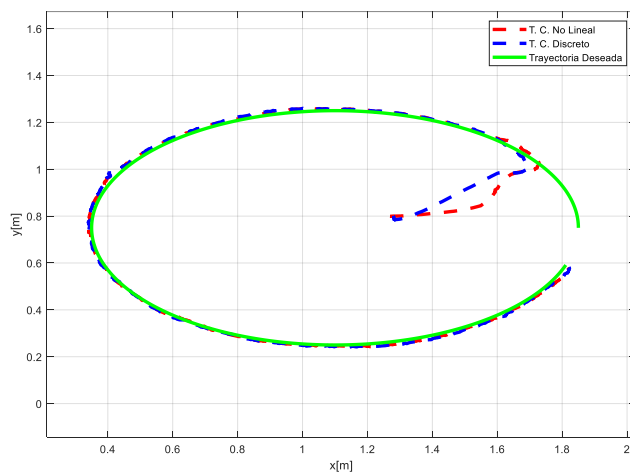
Figura 70.

Figura 69

Resultados del Seguimiento de una Elipse

a) *Controlador No Lineal*b) *Controlador No Lineal Discreto***Figura 70**

Resultados Combinados del Seguimiento de la Elipse

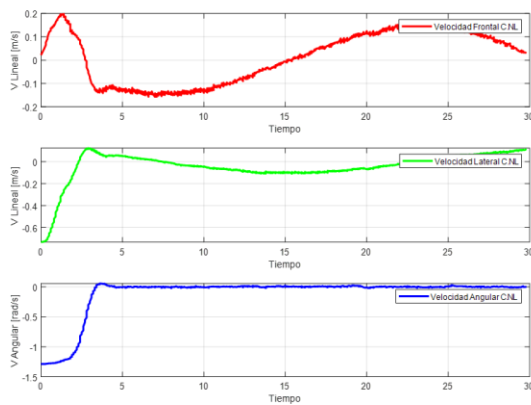


Las señales de control proporcionadas por ambos controladores mostradas en la Figura 71 no reflejan un comportamiento brusco en el sistema, lo cual beneficiará a la plataforma omnidireccional, puesto que no se forzarán los actuadores al seguir dicha referencia

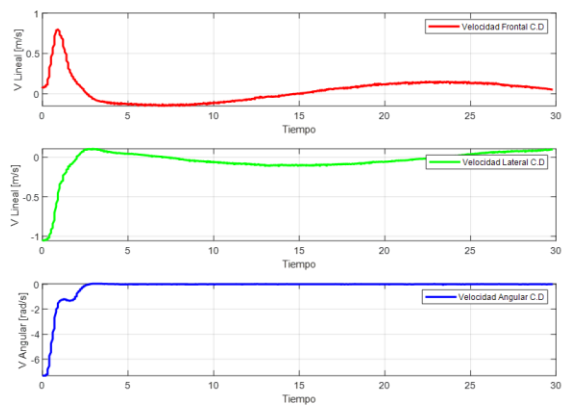
Figura 71

Señales de Control de Control

a) Controlador No Lineal



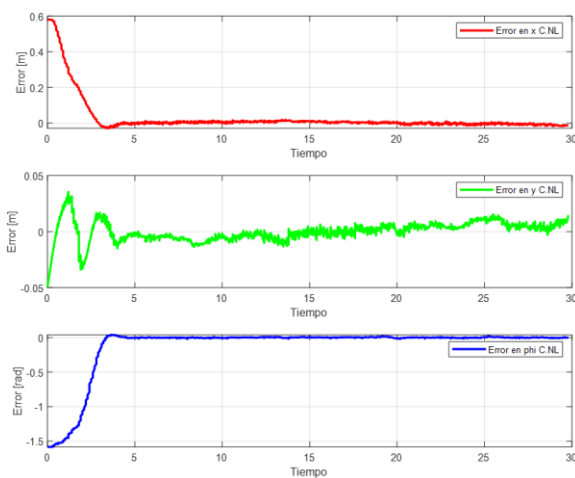
b) Controlador No Lineal Discreto



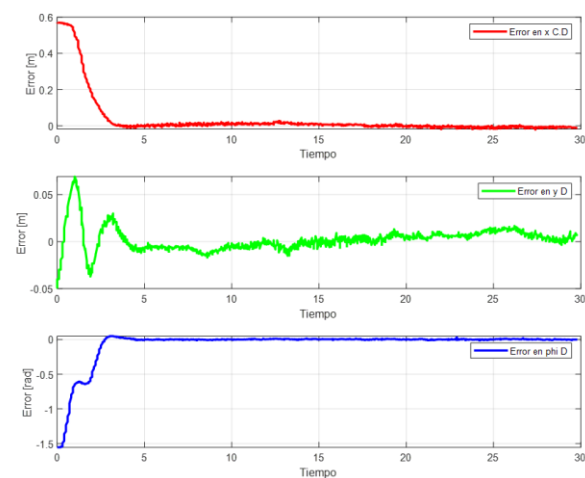
El comportamiento de los errores de control permite determinar que el sistema se vuelve asintóticamente estable, debido a que se puede apreciar en la Figura 72 como los errores de control se enclavan en 0 en un tiempo menor a 5 segundos.

Figura 72*Errores de Control*

a) Controlador No Lineal



b) Controlador No Lineal Discreto



5.3.2. Trayectoria 2

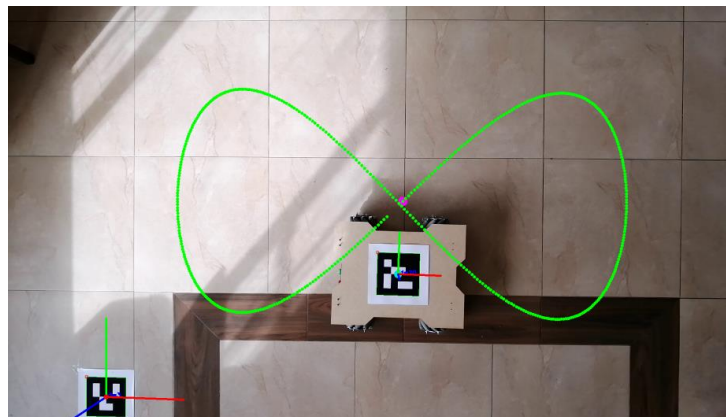
Para el ensayo 2 se tomó en consideración la ecuación paramétrica que forma un infinito, las ecuaciones paramétricas se describen en la ecuación (125) y representan en la

Figura 73.

$$\begin{aligned} x &= 0.8\sin(0.13t) + 1.1 \\ y &= 0.8\sin(0.13t)\cos(0.13t) + 0.75 \end{aligned} \quad (125)$$

Figura 73

Trayectoria 2 Infinito



Al igual que en el caso anterior, se observó un comportamiento muy aceptable en el seguimiento de la trayectoria descrita por parte de ambos controladores modernos como se observa en la Figura 74, debido a que la trayectoria parametrizada no presenta

cambios de pendiente bruscos, lo cual permite un comportamiento suave del sistema (Figura 75).

Figura 74

Resultados del Seguimiento de la Trayectoria 2

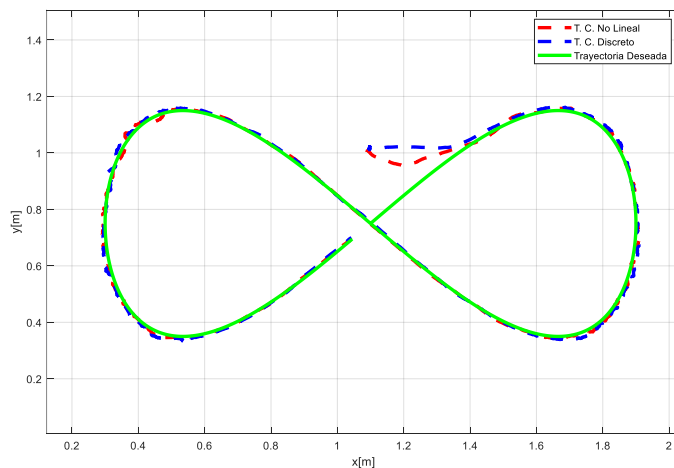
a) Controlador No Lineal

b) Controlador No Lineal Discreto



Figura 75

Resultados Combinados del Seguimiento de la Trayectoria 2

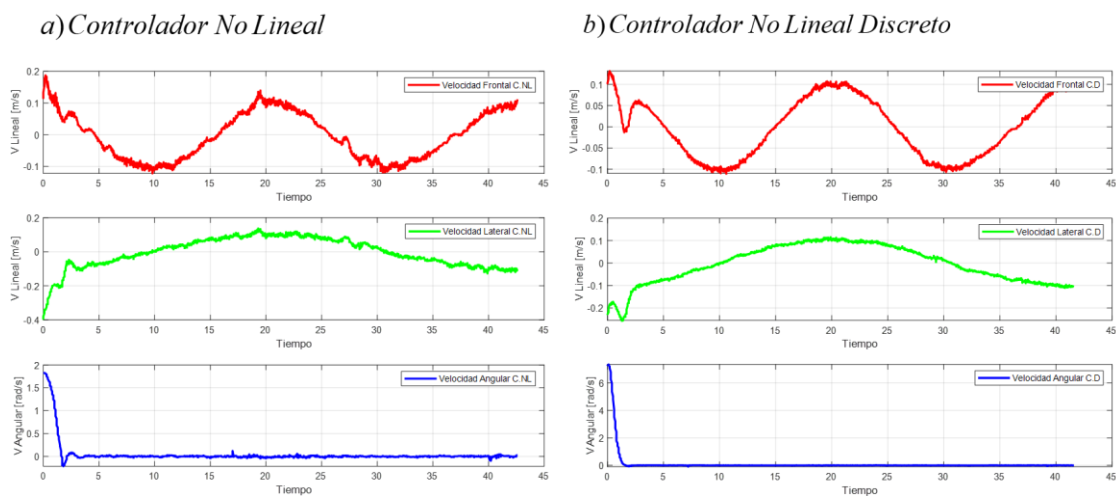


Las señales de control proporcionadas por ambos controladores mostradas en la Figura 76 no reflejan un comportamiento brusco en el sistema. Sin embargo, el controlador No Lineal Discreto refleja un incremento en las señales de control con

respecto al controlador No Lineal en la velocidad angular proporcionada al sistema con el fin de llegar a la referencia del ángulo proporcionado en el menor tiempo posible.

Figura 76

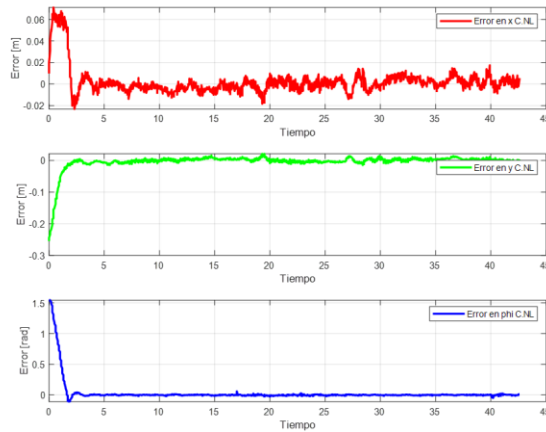
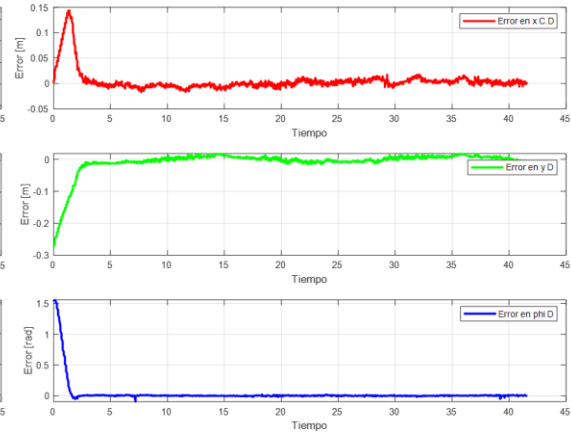
Señales de Control de Control



El comportamiento de los errores de control permite determinar que el sistema se vuelve asintóticamente estable en el caso del controlador Discreto y un comportamiento estable en el controlador No Lineal, debido a que se presentan ligeras oscilaciones del error en el eje x con una magnitud de $\pm 2\text{cm}$ como se muestra en la Figura 77

Figura 77

Errores de Control

a) *Controlador No Lineal*b) *Controlador No Lineal Discreto*

5.3.3. Trayectoria 3

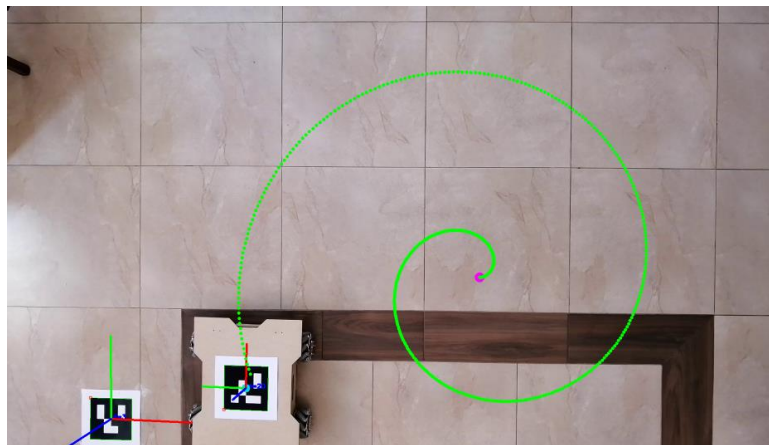
Para el ensayo 3 se implementó la ecuación paramétrica de un espiral, las ecuaciones paramétricas se describen en la ecuación (126) y se representan en la

Figura 78.

$$\begin{aligned} x &= 0.018t \cos(0.21t) + 1.3 \\ y &= 0.018t \sin(0.21t) + 0.75 \end{aligned} \tag{126}$$

Figura 78

Trayectoria 3 Espiral

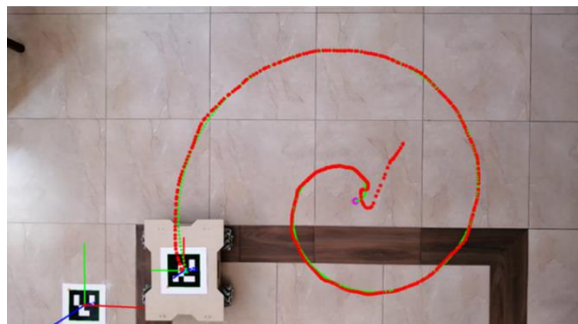


La implementación de la curva paramétrica descrita no presentó mayor complicación en ambos controladores modernos a pesar de que dicha curva se caracteriza por tener un crecimiento progresivo de su amplitud y por ende de la velocidad de la curva (Figura 79). El comportamiento de ambos controladores es muy similar. Sin embargo, se observa una mejor compensación del desfase inicial en el controlador discreto como se puede observar en la Figura 80.

Figura 79

Resultados del Seguimiento de la Trayectoria 3

a) Controlador No Lineal



b) Controlador No Lineal Discreto

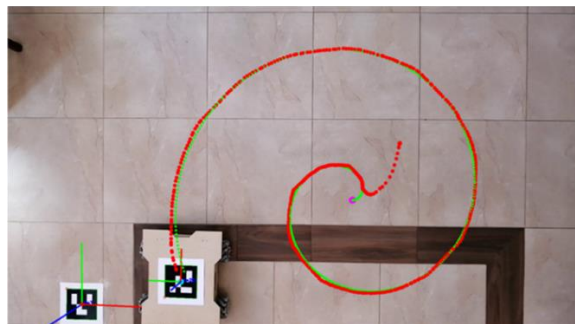
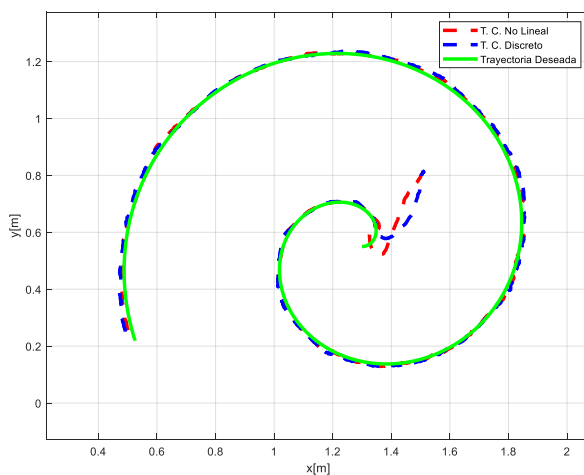


Figura 80

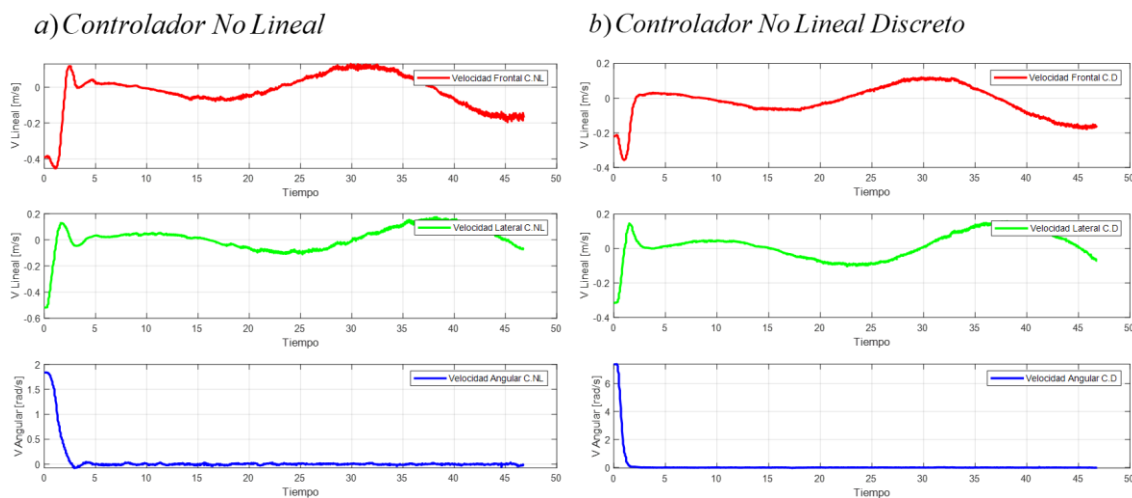
Resultados Combinados del Seguimiento de la Trayectoria 3



Las señales de control proporcionadas por ambos controladores describen un comportamiento similar a excepción de la velocidad angular, debido a que el controlador discreto refleja un comportamiento más brusco al compensar el error en el ángulo (Figura 81)

Figura 81

Señales de Control de Control

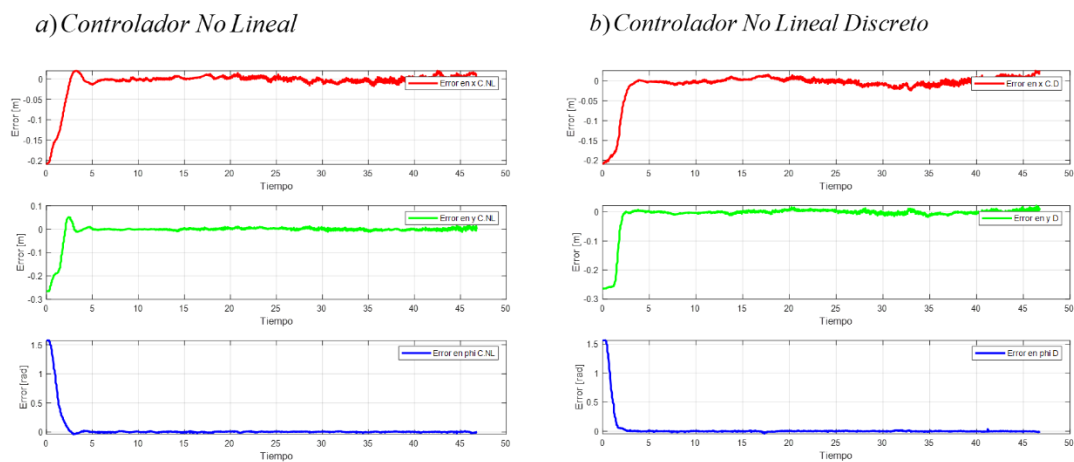


El comportamiento de los errores de control permite determinar que el sistema se vuelve asintóticamente estable a partir de los 5 segundos transcurridos, lo cual

permite demostrar la eficiencia de ambos controladores modernos en el control de una trayectoria con crecimiento progresivo como se evidencia en la Figura 82.

Figura 82

Errores de Control



5.3.4. Trayectoria 4

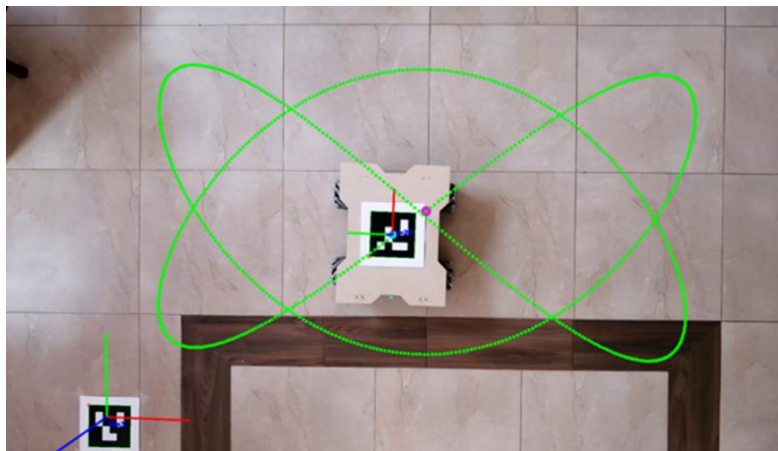
Para el ensayo 4 se implementó la ecuación paramétrica de una curva compuesta por una sinusoidal en cada eje, pero con diferente frecuencia angular, las ecuaciones paramétricas se describen en la ecuación (127) y se representan en la

Figura 83.

$$\begin{aligned}
 x &= 0.88 \sin(0.1t) + 1.1 \\
 y &= 0.47 \sin(1.5t) + 0.75
 \end{aligned}
 \tag{127}$$

Figura 83

Trayectoria 4 - Mariposa

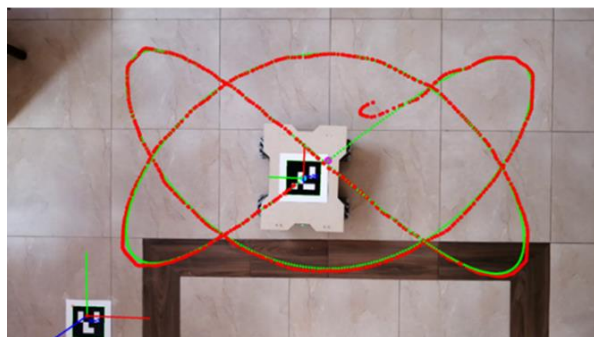


La implementación de la curva paramétrica descrita no presentó mayor complicación en ambos controladores modernos (Figura 85), a pesar de representar una curva paramétrica sumamente compleja con curvas más angostas que en casos anteriores. Sin embargo, el controlador discreto resulta ser un tanto más estable que el controlador No Lineal, debido a la baja presencia de oscilaciones en el recorrido de la trayectoria, como se puede apreciar en la Figura 84.

Figura 84

Resultados del Seguimiento de la Trayectoria 4

a) Controlador No Lineal



b) Controlador No Lineal Discreto

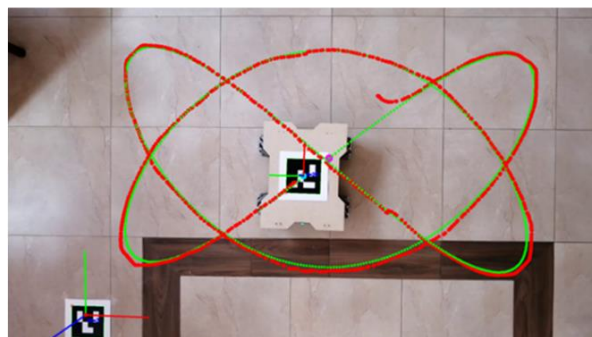
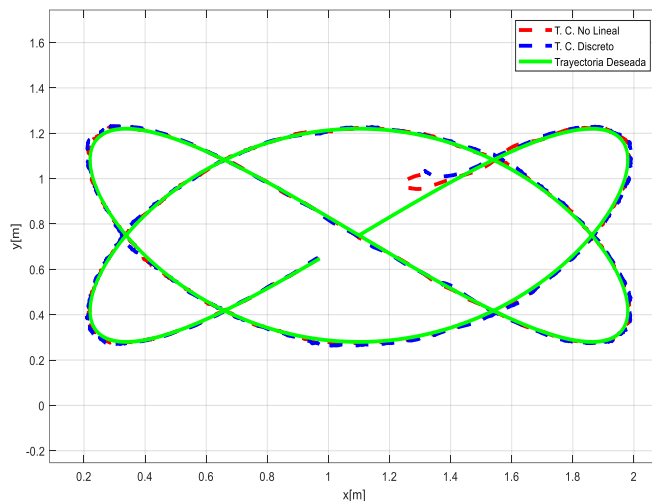


Figura 85

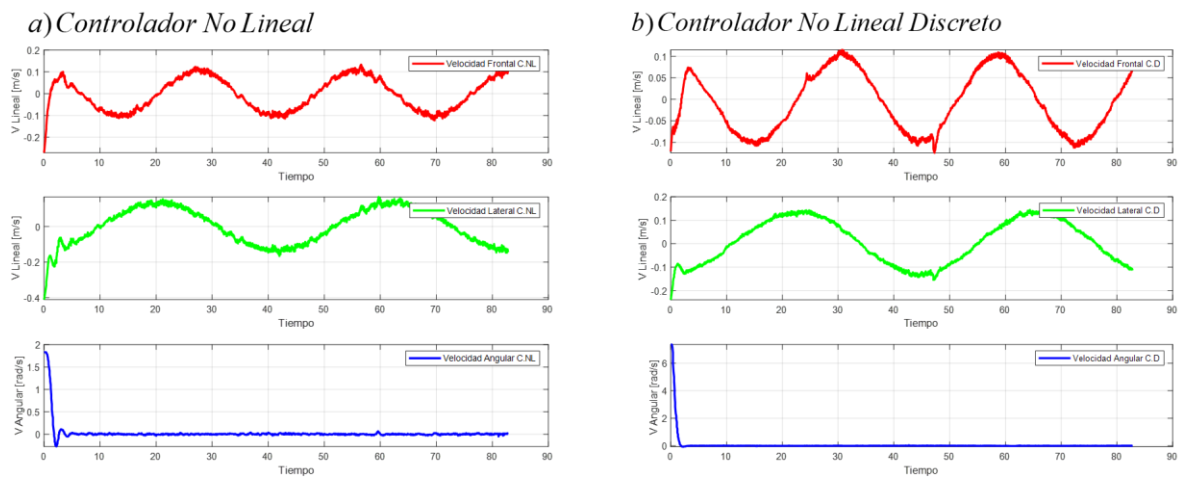
Resultados Combinados del Seguimiento de la Trayectoria 4



Las señales de control proporcionadas por ambos controladores describen un comportamiento similar. Sin embargo, el controlador discreto no presenta oscilaciones en las velocidades lineales, lo cual lo convierte en un controlador más estable que el controlador No Lineal (Figura 86)

Figura 86

Señales de Control de Control

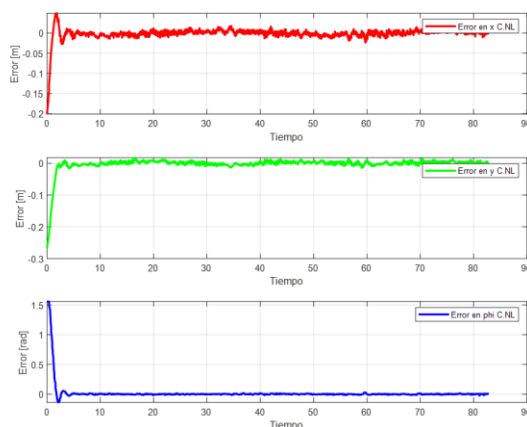


El comportamiento de los errores de control que se muestran en la Figura 87, permite determinar que el sistema se vuelve asintóticamente estable debido al enclavamiento de los errores de control en 0, lo cual permite demostrar la eficiencia de ambos controladores modernos en el seguimiento de la trayectoria descrita.

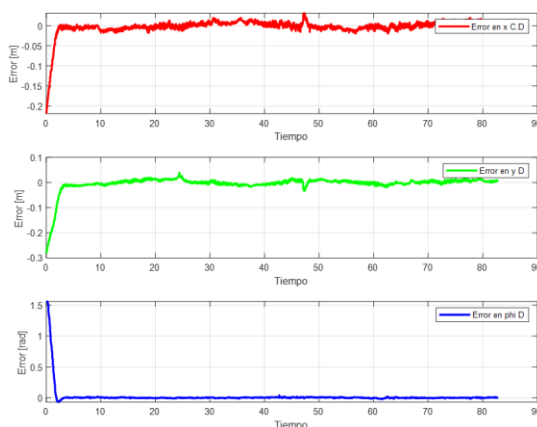
Figura 87

Errores de Control

a) Controlador No Lineal



b) Controlador No Lineal Discreto



5.3.5. Análisis de Robustez

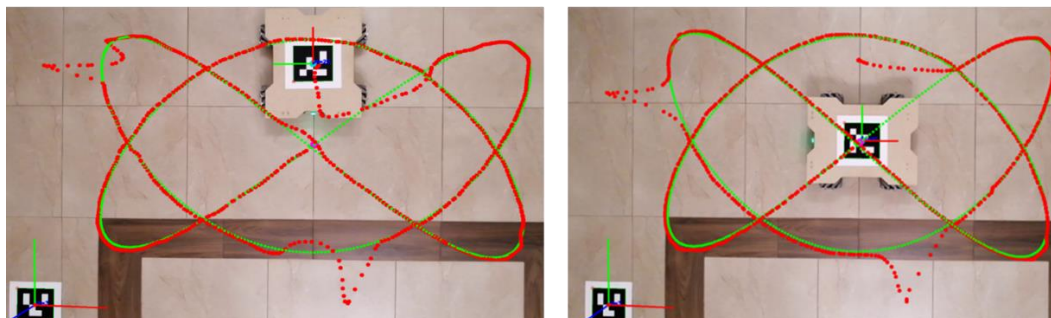
El análisis de robustez de los controladores modernos se realiza evaluando la respuesta del sistema ante la aplicación de una perturbación. En este caso la perturbación aplicada es el desfase de posición de forma manual de la plataforma omnidireccional con respecto a la trayectoria de seguimiento como se muestra en la Figura 88.

Figura 88

Respuesta Ante Perturbaciones

a) Controlador No Lineal

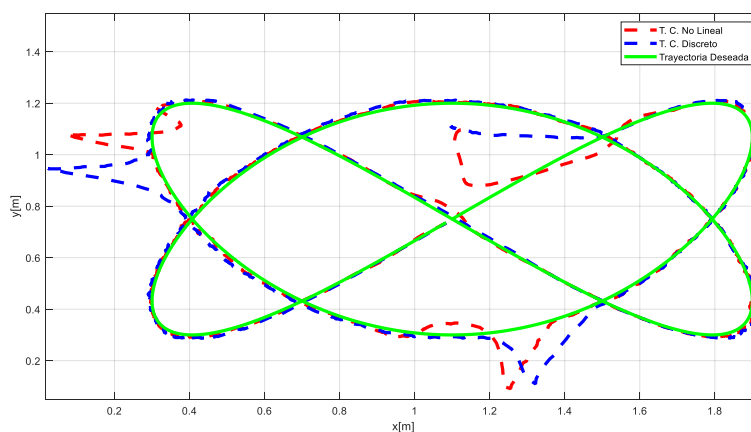
b) Controlador No Lineal Discreto



Para este análisis se optó por aplicar 2 perturbaciones al sistema en diferentes puntos de la trayectoria. Las trayectorias recorridas por la plataforma omnidireccional con el uso del controlador No lineal y el controlador Discreto se muestran en la Figura 89, donde se puede apreciar que el controlador No Lineal presenta un ligero sobre impulso al intentar eliminar la perturbación razón por la cual demora un poco más en estabilizar el sistema a diferencia del controlador Discreto que muestra un comportamiento mucho más robusto.

Figura 89

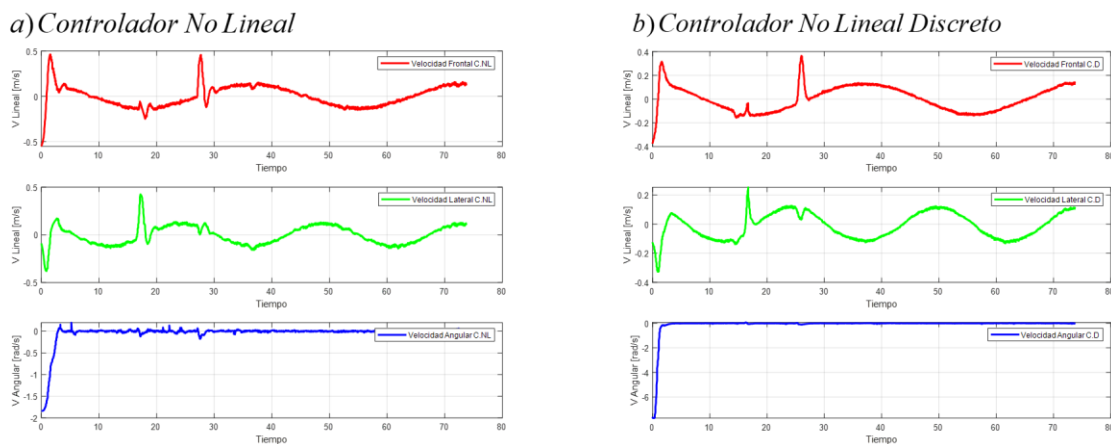
Trayectorias Recorridas



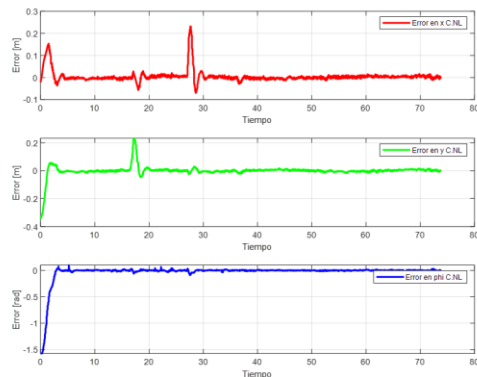
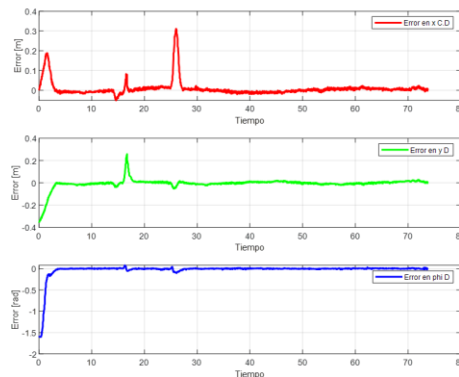
El comportamiento de las señales de control permite confirmar lo antes mencionado, debido a que el controlador discreto no presenta oscilaciones en sus señales de control cuando se le aplica una perturbación a diferencia del controlador No Lineal como se puede apreciar en la Figura 90.

Figura 90

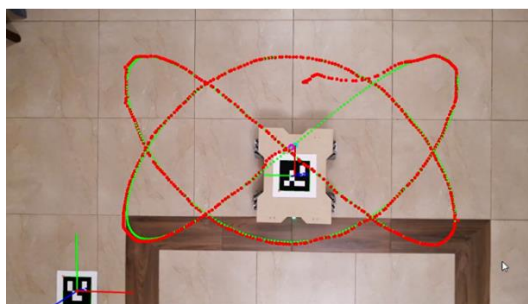
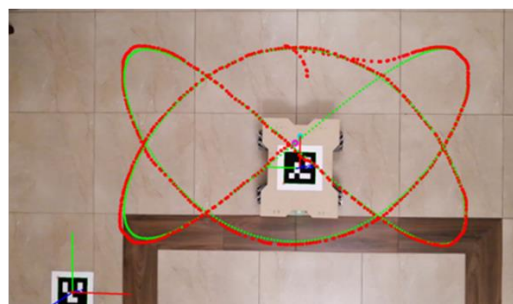
Señales de Control



La gráfica de los errores de control permite evidenciar que ambos controladores logran eliminar la perturbación aplicada. Sin embargo, el controlador discreto demora menos tiempo en llevar a la estabilidad al sistema y no presenta oscilaciones en las señales en el recorrido de la trayectoria, lo cual permite demostrar la superioridad del controlador discreto frente al controlador Lineal como se muestra en la Figura 91.

Figura 91**Errores de Control***a) Controlador No Lineal**b) Controlador No Lineal Discreto***5.3.6. Punto de Operación Desplazado.**

Para evaluar la estabilidad del sistema ante cambios en la cinemática de la plataforma omnidireccional se implementó el control de trayectoria con algoritmos de control moderno, con el punto de operación desplazado $a = 0.15m$. Con el fin de poner a prueba la eficiencia de los controladores en cuestión debido a que el desfase del punto de operación incrementa el nivel de dificultad en el seguimiento de trayectoria. Para ello fue necesaria la implementación del modelo cinemático descrito en la ecuación (74). Obteniendo los siguientes resultados:

Figura 92**Seguimiento con Punto de Operación Desplazado***a) Controlador No Lineal**b) Controlador No Lineal Discreto*

El comportamiento observado es muy similar a los obtenidos anteriormente, con la diferencia que si se produce un pequeño desfase en las partes de la trayectoria donde existe una curva cerrada. Sin embargo, los resultados mostrados en la Figura 93 son muy aceptables, debido a que ambos controladores convergen bastante bien a la curva deseada.

Figura 93

Resultados Obtenidos

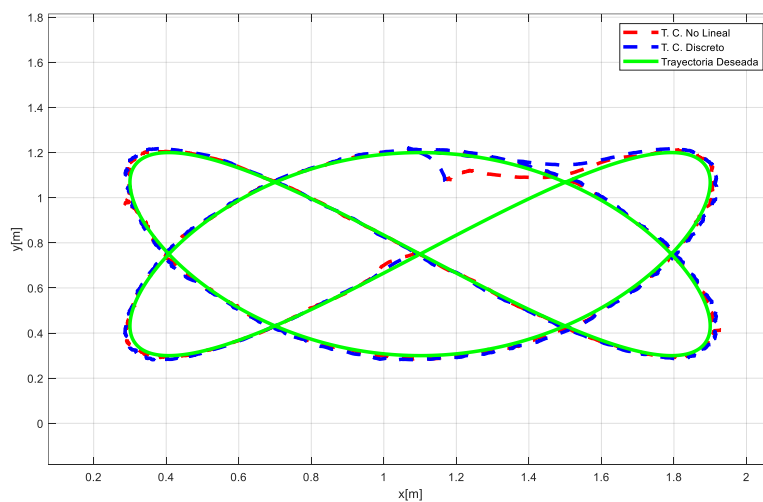
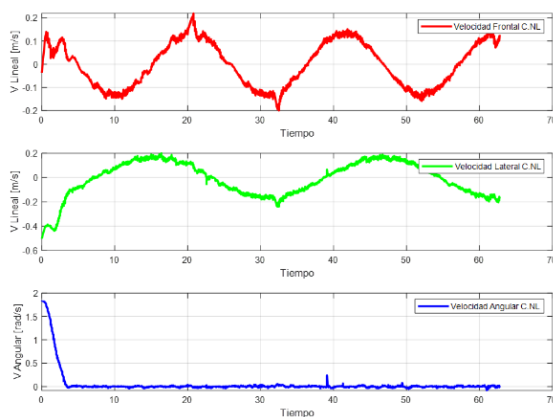


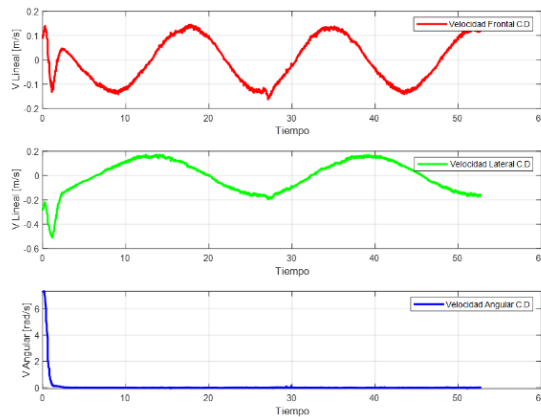
Figura 94

Señales de Control

a) Controlador No Lineal



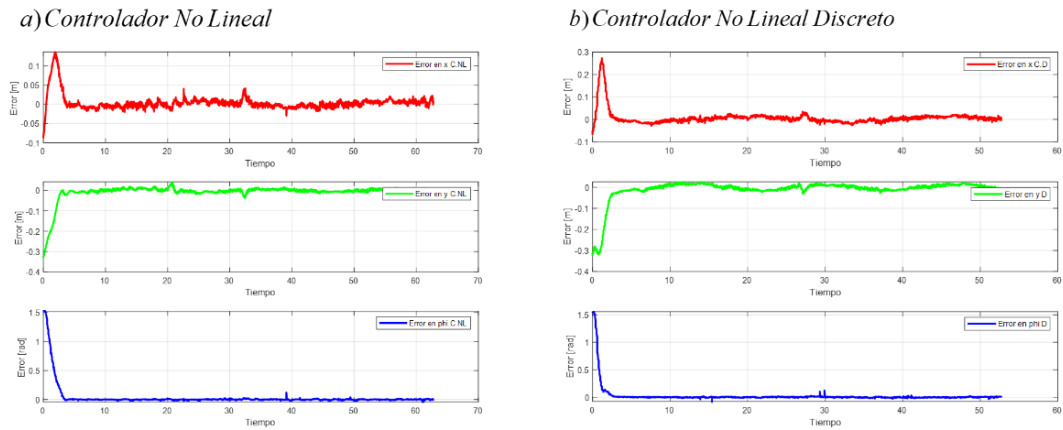
b) Controlador No Lineal Discreto



Las señales de control mostradas en la Figura 94 son bastante similares para ambos controladores. Sin embargo, el controlador Discreto refleja una curva mucho más suave y estable que el controlador no lineal, dicho argumento se puede confirmar en la gráfica de los errores de control (Figura 95), donde se puede observar ligeras oscilaciones en la gráfica proporcionada por el controlador No Lineal.

Figura 95

Errores de Control



5.3.7. Implementación del Controlador Clásico

Para validar el funcionamiento y la superioridad de los controladores modernos MIMO No Lineales en la resolución del seguimiento de trayectoria, se realizó la implementación de un algoritmo de control clásico como es el PID que tiene la forma que se describe a continuación:

$$u(t) = k_p \tilde{h}(t) + k_i \int \tilde{h}(t) + k_d \frac{d}{dt} \tilde{h}(t) \quad (128)$$

Donde:

$u(t)$: Representa la señal de control.

k_p, k_i, k_d Son las constantes proporcional, integral y derivativa respectivamente del controlador PID.

$\tilde{h}(t)$: Es error entre la referencia y el estado actual del sistema.

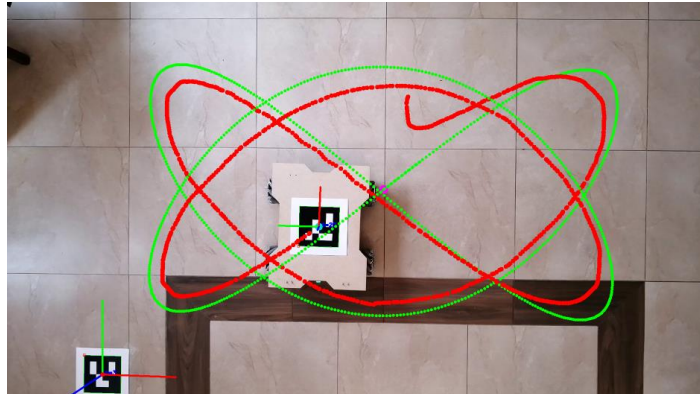
El controlador tradicional se implementó para cada uno de los grados de libertad del sistema propuesto, con el fin de realizar el control de posición tanto en x, y y de la orientación en φ , es decir, para realizar el control de la plataforma omnidireccional se precisa de 3 controladores independientes debido a que el algoritmo de control tradicional está limitado al control de sistemas SISO. El cálculo de las constantes del controlador PID se determinó en la práctica a través del método prueba-error obteniendo los siguientes resultados:

Tabla 13

Constantes PID

	k_p	k_i	k_d
Posición en x	2.52	13.45	0.15
Posición en y	2.52	13.45	0.15
Orientación en φ	1.17	10.23	0

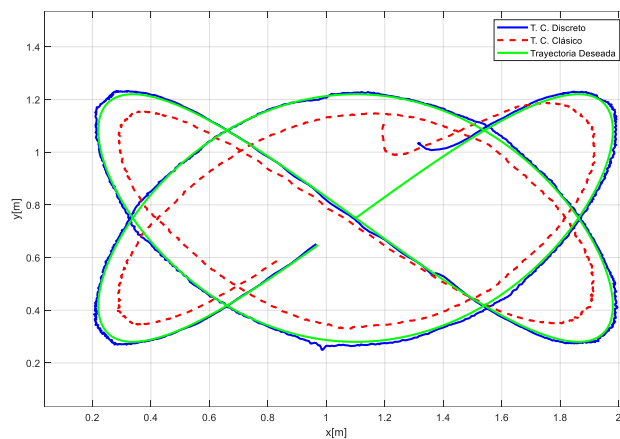
Sin embargo, a pesar de la sintonización previa el control tradicional resultó ser muy poco eficiente en el control del seguimiento de trayectoria de la plataforma omnidireccional, debido a que el controlador PID no toma en consideración la cinemática propia de la plataforma omnidireccional y al aplicar el control con una referencia variable aumenta el grado de dificultad y no se obtienen buenos resultados como se puede apreciar en la Figura 96.

Figura 96*Control PID de Trayectoria*

A pesar de no obtener buenos resultados, se puede evidenciar como el control tradicional intenta seguir la referencia en toda la trayectoria. Para tener una mejor apreciación de la eficiencia de un controlador clásico frente a un controlador moderno se realizó el control de la misma trayectoria tanto para el control PID como para el Controlador No Lineal Discreto, obteniendo los resultados mostrados en la Figura 93, donde se puede confirmar la eficiencia del controlador moderno frente al controlador tradicional en la resolución del problema de seguimiento de trayectoria debido a que el control PID presenta un error en estado estable que no le permite llegar a la referencia de manera adecuada a diferencia del control discreto que si converge a la curva de manera eficiente.

Figura 97

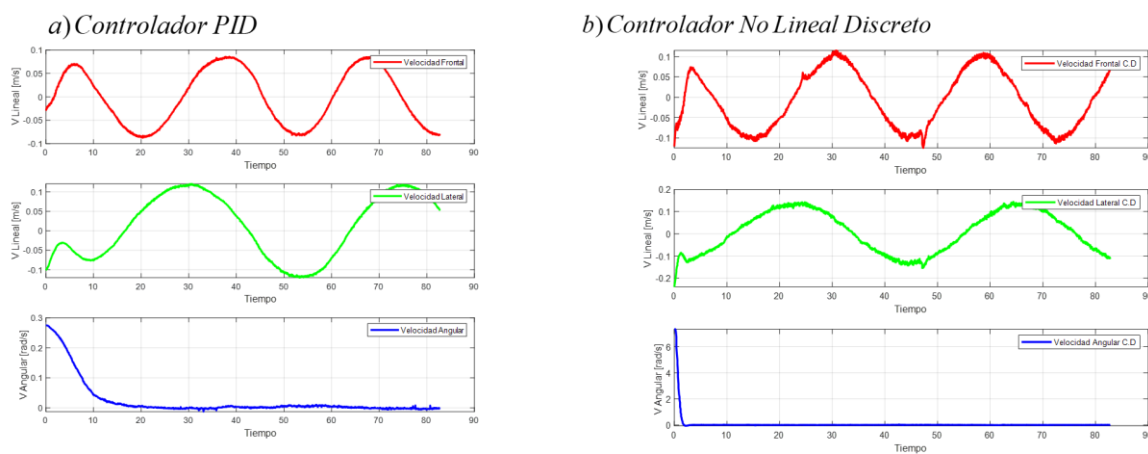
Resultados Obtenidos PID - Discreto



Las señales de control (Figura 98), son un poco similares sin embargo el comportamiento de ambos controladores es diferente como se puede apreciar en la Figura 97.

Figura 98

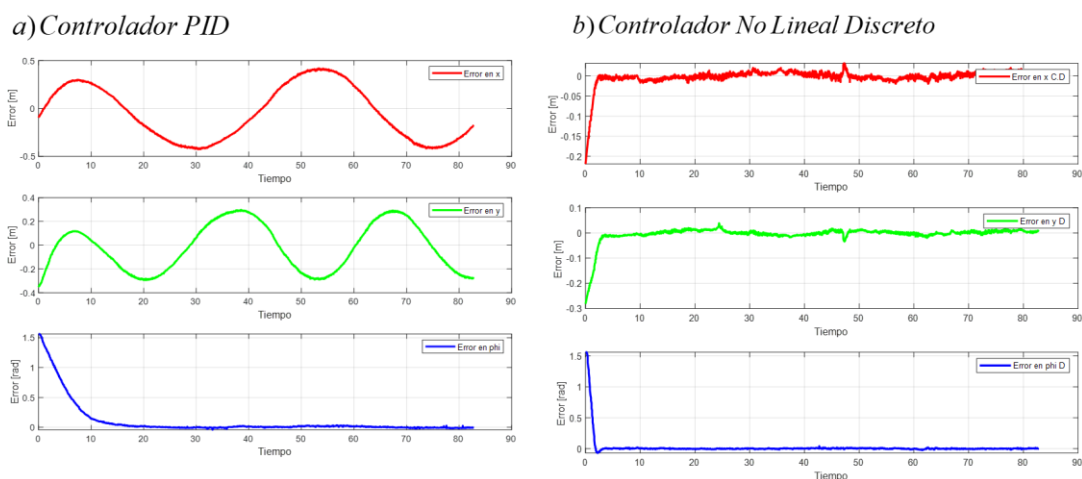
Señales de Control PID



Los errores de control permiten confirmar la eficiencia del controlador moderno frente al tradicional, debido a la magnitud del error presente en el controlador PID durante todo el recorrido de la trayectoria.

Figura 99

Errores PID -Discreto



Conclusión

Luego de realizar múltiples pruebas entre los controles modernos, se puede visualizar la superioridad del controlador discreto frente al controlador No lineal, debido a que el controlador discreto resulta ser más eficiente en la resolución del control de trayectoria, arrojando curvas mucho más suaves y libres de oscilación, además en la práctica se evidenció un comportamiento menos agresivo por parte del controlador discreto, lo cual permite realizar el seguimiento de trayectoria sin la necesidad de forzar al sistema con señales de control agresivas u oscilantes. Estos resultados evidentemente se deben a que el controlador discreto toma en consideración la referencia deseada en un instante de muestreo, lo cual permite al controlador reaccionar de manera eficiente ante perturbaciones y a su vez converger de mejor manera a la curva deseada.

5.4. Validación de la Hipótesis

Para realizar las pruebas de funcionamiento se implementaron 4 curvas parametrizadas en el tiempo, cada una analizada considerando las siguientes variables: velocidad lenta, rápida, con perturbación, sin perturbación. Dichos escenarios se evaluaron bajo la implementación de 3 controladores: 2 modernos mimo no lineales y un controlador clásico. Además, cada ensayo se realizó 3 veces, es decir, que se obtuvieron 144 muestras para realizar la validación de hipótesis.

Para definir calificar cada una de las muestras como una prueba funcional o no, se utilizó el valor del error cuadrático medio como se muestra en la ecuación:

$$e = \frac{1}{n} \sum_{n=0}^{\infty} \mathbf{e}^T \mathbf{e} \quad (129)$$

Mediante el error cuadrático es posible determinar la eficiencia de un controlador en el seguimiento de trayectoria, por lo que, luego de varias pruebas realizadas se optó por escoger un valor umbral del error cuadrático de $e = 0.001$, debido a que, luego de varias pruebas se demostró que dicho valor se refleja cuando el seguimiento de trayectoria arroja una convergencia aceptable respecto a la referencia.

Los resultados obtenidos se muestran en las siguientes tablas, donde se podrá observar el resumen de los datos obtenidos en cada prueba realizada para la validación de hipótesis.

Se detalla el significado de cada variable a utilizar:

$T1 \rightarrow$ Trayectoria 1

$T4 \rightarrow$ Trayectoria 4

$T2 \rightarrow$ Trayectoria 2

$Lento \rightarrow$ velocidad lenta

$T3 \rightarrow$ Trayectoria 3

$Rápida \rightarrow$ velocidad rápida

$SP \rightarrow$ sin perturbación

$N \rightarrow$ control no lineal

$P \rightarrow$ con perturbación

$D \rightarrow$ control no lineal discreto

$C \rightarrow$ control clásico

Tabla 14

Pruebas de Funcionamiento Trayectoria 1

	T1												TOTAL
	LENTO						RÁPIDO						
	SP			P			SP			P			
	C	N	D	C	N	D	C	N	D	C	N	D	
FUNCIONA	2	3	3	1	2	3	2	3	3	1	2	3	28
NO FUNCIONA	1	0	0	2	1	0	1	0	0	2	1	0	8
TOTAL	3	3	3	3	3	3	3	3	3	3	3	3	36

Tabla 15

Pruebas de Funcionamiento Trayectoria 2

	T2												TOTAL
	LENTO						RÁPIDO						
	SP			P			SP			P			
	C	N	D	C	N	D	C	N	D	C	N	D	
FUNCIONA	2	2	3	1	2	2	1	3	2	2	2	2	24
NO FUNCIONA	1	1	0	2	1	1	2	0	1	1	1	1	12
TOTAL	3	3	3	3	3	3	3	3	3	3	3	3	36

Tabla 16*Pruebas de Funcionamiento Trayectoria 3*

		T3												
		LENTO						RÁPIDO						
		SP			P			SP			P			
		C	N	D	C	N	D	C	N	D	C	N	D	TOTAL
FUNCIONA		2	2	3	2	2	3	2	1	3	2	1	2	25
NO FUNCIONA		1	1	0	1	1	0	1	2	0	1	2	1	11
TOTAL		3	3	3	3	3	3	3	3	3	3	3	3	36

Tabla 17*Pruebas de Funcionamiento Trayectoria 4*

		T4												
		LENTO						RÁPIDO						
		SP			P			SP			P			
		C	N	D	C	N	D	C	N	D	C	N	D	TOTAL
FUNCIONA		2	2	3	1	3	3	2	2	3	1	1	3	26
NO FUNCIONA		1	1	0	2	0	0	1	1	0	2	2	0	10
TOTAL		3	3	3	3	3	3	3	3	3	3	3	3	36

Obtenido los datos de las Tabla 14,

Tabla 15, Tabla 16 y, Tabla 17. Se procede a realizar la validación de hipótesis mediante el método de Chi Cuadrado el cual permite apreciar si la Hipótesis alternativa o nula son válidas.

Tabla 18*Datos para Validación de Hipótesis*

	T1	T2	T3	T4	TOTAL
FUNCIONA	28	24	25	26	103
NO FUNCIONA	8	12	11	10	41
TOTAL	36	36	36	36	144

El método del Chi Cuadrado requiere de la definición de las hipótesis alternativa y nula, por lo tanto, las hipótesis establecidas para la validación del presente proyecto son:

Hipótesis Nula (Ho): La plataforma no permite evaluar controladores.

Hipótesis alternativa (Hi): La plataforma permite evaluar controladores.

$$ft(\text{frecuencia teórica}) = \frac{(\text{total de la fila } i) \cdot (\text{total de la columna } j)}{\text{número total de datos}}$$

Se obtiene la frecuencia teórica de cada uno de los componentes de la tabla

$$28 \rightarrow \frac{36 \cdot 103}{144} = 25.75 \quad 24 \rightarrow \frac{36 \cdot 103}{144} = 25.75 \quad 12 \rightarrow \frac{36 \cdot 41}{144} = 10.25$$

$$8 \rightarrow \frac{36 \cdot 41}{144} = 10.25 \quad 11 \rightarrow \frac{36 \cdot 41}{144} = 10.25 \quad 10 \rightarrow \frac{36 \cdot 41}{144} = 10.25$$

$$25 \rightarrow \frac{36 \cdot 103}{144} = 25.75 \quad 26 \rightarrow \frac{36 \cdot 103}{144} = 25.75$$

Fórmula del Chi-Cuadrado Calculado

$$(x_{calc})^2: \sum \frac{(f - ft)^2}{ft}$$

Donde:

$f \rightarrow$ frecuencia observada

$ft \rightarrow$ frecuencia teórica

Se calcula el Chi Cuadrado

$$(x_{calc})^2 = \sum \frac{(f - ft)^2}{ft}$$

$$(x_{calc})^2 = \frac{(28 - 25.75)^2}{25.75} + \frac{(8 - 10.25)^2}{10.25} + \frac{(24 - 25.75)^2}{25.75} + \frac{(12 - 10.25)^2}{10.25} + \frac{(25 - 25.75)^2}{25.75} \\ + \frac{(11 - 10.25)^2}{10.25} + \frac{(26 - 25.75)^2}{25.75} + \frac{(10 - 10.25)^2}{10.25}$$

$$(x_{calc})^2 = 8.416765$$

Se calcula el grado de libertad (v)

$$v = (\text{numero de filas} - 1) \cdot (\text{numero de columnas} - 1)$$

$$v = (2 - 1) \cdot (4 - 1)$$

$$v = 3$$

Se trabajó con un nivel de significancia de 5% equivalente a 0.05

Teniendo 3 grados de libertad y un valor de margen de error de 0.05 se obtiene el valor de Chi Cuadrado en su tabla de distribución de

$$(x_{calc})^2 = 7.8147$$

Se deduce que el Chi Cuadrado calculado es mayor que el Chi Cuadrado obtenido se anula la hipótesis nula, de esta manera se valida la hipótesis alternativa que dice que la plataforma permite evaluar controladores, demostrando así que el trabajo realizado si permite cumplir con el objetivo planteado.

Capítulo VI

6. Conclusiones y recomendaciones

6.1. Conclusiones

- Existen varias configuraciones de robots omnidireccionales que se pueden clasificar según el tipo de rueda que utilizan, según su número de ruedas y según su configuración mecánica. Cada una de estas configuraciones permiten al robot móvil realizar un desplazamiento con 3 grados de libertad sobre el plano por el que circula, Sin embargo cada configuración posee sus ventajas y desventajas que vienen relacionadas con la tracción que ofrecen, su complejidad al realizar control sobre el sistema, la estabilidad que presentan en su desplazamiento y la complejidad del diseño mecánico, muchas características limitan el número de aplicaciones a las que podría estar destinado un robot omnidireccional. Sin embargo, el robot omnidireccional de ruedas Mecanum es la configuración que ofrece una mayor tracción y alta estabilidad en su desplazamiento permitiendo el desarrollo de aplicaciones con mayor facilidad siempre y cuando se tome en cuenta que esta configuración presenta una mayor complejidad en el control y requiere de un diseño mecánico más complejo.
- El diseño y la construcción de la plataforma omnidireccional en un tablero de fibra de densidad media, fue muy acertado para la resolución del seguimiento de trayectorias, debido a que no se requiere de un diseño robusto, sino por el contrario, se requiere de una estructura liviana que facilite el desplazamiento de la plataforma omnidireccional sin obligar a los actuadores a realizar un esfuerzo innecesario en el transporte de su estructura, pero que a su vez permita el alojamiento de todos los componentes en su interior. Por otro lado, no se presentaron inconvenientes en la reproducción del diseño mecánico en el tablero de MDF debido a que se utilizó el corte por láser para la elaboración de las diferentes partes que conforman la estructura.

- La estimación de pose por medio de los marcadores binarios pertenecientes al módulo ArUco disponible en la librería de OpenCV que ofrecen resultados altamente confiables, lo cual permitió la implementación de dicha estimación como medio de realimentación del sistema servo visual propuesto. Sin embargo, la fiabilidad de los datos de POSE está sumamente ligada al proceso de calibración de la cámara, específicamente al factor de re-proyección resultante, debido a que este parámetro es el encargado de evaluar la confiabilidad de los parámetros intrínsecos estimados en este proceso, concluyendo que mientras más cercano a cero sea dicho factor más fiables y precisos serán los resultados estimados de la POSE del marcador binario. Por otro lado, el uso de un marcador binario extra para el proceso de estimación de la POSE de la plataforma omnidireccional, permite fijar el sistema de coordenadas al plano por el cual se desplaza el robot móvil, facilitando el proceso de identificación manual de errores presentes en el proceso de estimación de los vectores de traslación y rotación. Además, resulta mucho más factible proponer trayectorias parametrizadas en el tiempo con origen en el sistema de coordenadas.

- La implementación de una interfaz gráfica para la manipulación, movimiento y configuración de todo el sistema fue muy acertada debido a que, por medio de esta, es posible la modificación de los parámetros de las curvas paramétricas y a su vez, hace posible la visualización de resultados al implementar la superposición de las curvas obtenidas en tiempo real, permitiendo dar un criterio de funcionalidad del seguimiento de trayectoria con respecto a la referencia deseada.

- Luego de realizar varias pruebas en las cuales se puso en prueba la eficiencia y robustez de los algoritmos de control moderno, se pudo evidenciar un mejor rendimiento por parte del controlador discreto, debido a las señales mucho más estables que se obtuvieron, lo cual permitió una mejor convergencia de la curva real con respecto a la

curva deseada. Dichos resultados fueron contrarrestados con un algoritmo de control clásico permitiendo evidenciar la superioridad de los controladores modernos frente al control tradicional ya que la mayoría de sistemas tienen varias variables a controlar, debido a que dicho controlador está orientado al control de sistemas SISO. Sin embargo, se pudo apreciar una convergencia medianamente aceptable con respecto a la curva deseada, sin embargo no es recomendable la implementación de dicho controlador para la resolución de trayectorias en las cuales se requiera alta precisión.

6.2. Recomendaciones

- La implementación de encoders en el proceso medición de la velocidad de cada uno de los motores resulta muy eficiente y confiable siempre y cuando se realice un correcto acondicionamiento de las señales, debido a que por la velocidad característica resultante, la señal rebota y se producen errores en la medición lo cual puede llevar a la inestabilidad a todo el sistema. Sin embargo, dicho problema queda resuelto a través de la implementación de filtro pasa bajo.
- El proceso de calibración de la cámara es indispensable para poder obtener los parámetros intrínsecos de la cámara, necesarios para poder conocer con exactitud la proyección que tiene un punto 3D en el plano de cámara. Por tal motivo es posible realizar el proceso de calibración hasta obtener un error de re-proyección lo más cercano a cero posible, debido a que en dicho caso los resultados obtenidos en el proceso de estimación serán mucho más confiables y precisos.
- Debido a la baja disponibilidad de los componentes en el país se recomienda realizar la adquisición de los componentes con el tiempo suficiente debido a que el proceso podría resultar muy tardío debido a la importación de los componentes. Sin embargo, los algoritmos de control moderno podrían ser implementados en un sistema

con características similares, lo cual permite tener un criterio aproximado del funcionamiento del sistema con la implementación de los algoritmos de control MIMO no lineales en el seguimiento de trayectorias de un robot móvil.

Bibliografía

- Álvarez, S. (2019). *Navegación autónoma de un robot móvil omnidireccional en trayectorias predeterminadas con evitación de obstáculos*.
<http://repositorio.uta.edu.ec/handle/123456789/29173>
- Andaluz, V. H., Carvajal, C. P., Alex, S. G., Zambrano, V. D., & Pérez, J. A. (2017). Navigation and dynamic control of omnidirectional platforms. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10454 LNAI, 661–672.
https://doi.org/10.1007/978-3-319-64107-2_54
- AndyMark. (2018, December 20). 4 in. HD Mecanum Wheels - AndyMark Inc.
[https://www.andymark.com/products/4-in-hd-mecanum-wheel-set-options?Bore=1%2F2 Inch Hex&quantity=1&Mecanum Wheel=Left](https://www.andymark.com/products/4-in-hd-mecanum-wheel-set-options?Bore=1%2F2+Inch+Hex&quantity=1&Mecanum+Wheel=Left)
- Arévalo-Vázquez, E., Zúñiga-López, A., Villegas-Cortez, J., & Avilés-Cruz, C. (2015). Implementación de reconocimiento de objetos por color y forma en un robot móvil. In *21 Research in Computing Science* (Vol. 91).
[https://www.polibits.cidetec.ipn.mx/2015_91/Implementacion de reconocimiento de objetos por color y forma en un robot movil.pdf](https://www.polibits.cidetec.ipn.mx/2015_91/Implementacion+de+reconocimiento+de+objetos+por+color+y+forma+en+un+robot+movil.pdf)
- Barrientos Sotelo, V. R., García Sánchez, J. R., & Silva Ortigoza, R. (2007). Robots Móviles: Evolución y Estado del Arte. In *redalyc.org* (pp. 12–17). Polibits.
<https://www.redalyc.org/pdf/4026/402640448003.pdf>
- Benites, N. R. (2015). *Diseño y simulación de controladores multivariables clásico y avanzado para una caldera industrial*.
<http://repositorio.unac.edu.pe/handle/UNAC/1059>

Cevallos, C. A. (2016). *Diseño e implementación de un sistema de auto-piloto para el control de un hexacóptero* [Escuela Politécnica Nacional].

<http://bibdigital.epn.edu.ec/handle/15000/16468>

Cortés, U., Castañeda, A., Benítez, A., & Díaz, A. (2015). Control de Movimiento de un Robot Móvil Tipo Diferencial Robot úβot-32b. In *amca.mx*.

http://amca.mx/memorias/amca2015/files/0058_JuAT3-01.pdf

CQRobot. (2020, April 20). *70:1 Metal Gearmotor*.

https://www.cqrobot.com/index.php?route=product/product&product_id=1085&search=motor&description=true&page=2

Cytron. (2017, January). *10Amp 5V-30V DC Motor Driver (2 Channels)*.

<https://www.cytron.io/p-10amp-5v-30v-dc-motor-driver-2-channels>

Digi International Inc. (2021, January). *Multipoint Wireless Networking OEM RF Module | Digi International*. <https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-802-15-4#specifications-xbees2c802154>

García Silla, D., Vargas Soto, J. E., Díaz Delgado, G., & González Gutiérrez, A. (2016). *Diseño Mecatrónico de un Robot Móvil*. *Mecatronica.Net*.

<https://www.mecatronica.net/emilio/CapLibros/2016-Cap30-Libro-IMEM.pdf>

Herguedas Gastón, R. (2017). Sistema de navegación autónoma de un dirigible. [Universidad de Zaragoza]. In *zagan.unizar.es*.

<https://zagan.unizar.es/record/63606>

Hernández-Paredes, J., ... L. S.-H.-C., & 2016, undefined. (2016). Control de Movimiento y Seguimiento de Trayectorias de Un Robot Móvil Diferencial a través de la Estimación Odométrica. In *200.79.179.163*. www.iteshu.edu.mx.

- HUAWEI. (2021, January 21). *HUAWEI P30 Lite - HUAWEI Latin*.
<https://consumer.huawei.com/latin/phones/p30-lite/>
- Jia, Y.-B. (2020). *Rotation in the Space*. Com S 477/577 Notes.
<http://web.cs.iastate.edu/~cs577/handouts/rotation.pdf>
- Jiménez Bravo, R. (2018). *Sistema de seguimiento de objetos usando OpenCv, ArUco y Filtro de Kalman extendido* [Escuela Técnica Superior de Ingeniería Universidad de Sevilla]. <https://idus.us.es/handle/11441/85466>
- Jiva, E. I. (2019). *Desarrollo de la teleoperación de robots industriales y colaborativos mediante técnicas avanzadas de visión artificial* [Universitat Politècnica de València]. <https://riunet.upv.es/handle/10251/130010>
- Lalama, L. (2014). *Implementación de un Robot Móvil Terrestre Omnidireccional, Semiautónomo y telecontrolado a través de un teléfono Inteligente Android* [Universidad de Guayaquil]. <http://repositorio.ug.edu.ec/handle/redug/6521>
- MakeltFrom. (2020, May 30). *Medium Density Fiberboard (MDF) :: MakeltFrom.com*.
<https://www.makeitfrom.com/material-properties/Medium-Density-Fiberboard-MDF>
- Marosan, L. A., Constantin, G., Biris, C., & Girjob, C. (2020). A Model Locomotion System for Mobile Platforms Omnidirectional Serving the Industrial Environment. *Content.Sciendo.Com*. <https://doi.org/10.2478/aucts-2020-0004>
- Mott, R., Salas, R., Flores, M., & Martínez, E. (2009). *Resistencia de materiales*.
<https://repository.uaeh.edu.mx/revistas/index.php/sahagun/article/download/2361/2363?inline=1>
- OpenCV. (2011). *OpenCV: Camera calibration With OpenCV*.
https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

- OpenCV. (2020). *About - OpenCV*. <https://opencv.org/about/>
- Ortega, A. (2019). ALGORITMO DE CONTROL DE UN ROBOT OMNIDIRECCIONAL DE TRES RUEDAS [Universidad Politécnica de Madrid]. In *oa.upm.es*.
http://oa.upm.es/58728/1/TFM_ALBERTO_ORTEGA_ANTON.pdf
- Ortega, J. F., & Yapo, L. M. (2017). *Construcción de un Robot Móvil Híbrido Omnidireccional* [Universidad Politécnica Salesiana].
<http://dspace.ups.edu.ec/handle/123456789/14048>
- Peña-Cabrera, M., Andrade, J., & Ortega, J. (2016). Caracterización y control del desplazamiento de una plataforma omnidireccional. *Academia.Edu*, 1–2.
https://www.academia.edu/download/54033763/caracterizacion_de_plataforma_AC_CA_2016_1_sept_F6F3__1.pdf
- Pérez, D. (2017). *Construcción de un robot móvil esférico controlado inalámbricamente para la ejecución de proyectos de investigación formativa y generativa de la facultad de ingeniería en sistemas electrónica e industrial*.
<http://repositorio.uta.edu.ec/handle/123456789/25192>
- Poot Chin, M. E. (2019). *Desarrollo de un sistema de visión computacional para el vuelo autónomo de un Vehículo Aéreo No Tripulado* [UNIVERSIDAD AUTONOMA DE YUCATAN]. <https://clir-lab.org/images/Carlos/thesis.pdf>
- Reboto, E., & Rodríguez, Q. (2016). *Optimización de la gestión de pedidos online en un almacén mediante la introducción de un robot móvil*.
<https://core.ac.uk/download/pdf/211101612.pdf>
- Salas, L. E. (2015). *Diseño de un sistema de navegación autónomo para robots móviles usando fusión de sensores y controladores neuro difusos*. 3–10.

<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/6191>

STMicroelectronics. (2020, August 19). *NUCLEO-F446RE - STM32 Nucleo-64 development board with STM32F446RE MCU, supports Arduino and ST morpho connectivity - STMicroelectronics*. <https://www.st.com/en/evaluation-tools/nucleo-f446re.html#overview>

Tayupanta Zúñiga, W. E. (2018). *NAVEGACIÓN AUTÓNOMA DE MINI-DRONES PARA MULTIROBOTS SLAM 3D EN ENTORNOS ESTÁTICOS CON ALGORITMOS DE INTELIGENCIA DE EJAMBRE*. Universidad de las Fuerzas Armadas ESPE.

Tejera, G. (2009). *Control y Comportamiento de Robots Omnidireccionales Estado del Arte*. <http://www.ng.edu.uy/~pgomni>

Woods, B. (2006). *Omni-Directional Wheelchair Honours Thesis* [The University Of Western Australia]. <http://robotics.ee.uwa.edu.au/theses/2006-Wheelchair-Woods.pdf>

Zeee. (2021, January). *Zeee Power*.

<http://www.zeeepower.com/pro.asp?keyno=1466&bascode=101&name=Zeee6000mAh-30C-4S-14.8V>

ANEXOS