



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO EN CONFIGURACIÓN DIFERENCIAL PARA LOCALIZACIÓN EN INTERIORES USANDO ODOMETRÍA”

Trabajo de titulación

Tipo: Dispositivo Tecnológico

Presentado para optar el grado académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES

AUTORES: HENRY VINICIO PILAMALA SISLEMA

JORGE LUIS HILAÑO CHANATASI

TUTOR: ING. FAUSTO RAMIRO CABRERA AGUAYO

Riobamba – Ecuador

2021

© 2021, Henry Vinicio Pilamala Sislema; & Jorge Luis Hilaño Chanatasi

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Nosotros, Henry Vinicio Pilamala Sislema y Jorge Luis Hilaño Chanatasi, declaramos que el presente trabajo de titulación es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 18 de febrero 2021



Henry Vinicio Pilamala Sislema
160066331-2



Jorge Luis Hilaño Chanatasi
180472262-5

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES**

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo Dispositivo Tecnológico. **“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO EN CONFIGURACIÓN DIFERENCIAL PARA LOCALIZACIÓN EN INTERIORES USANDO ODOMETRÍA”**, realizado por los señores: **HENRY VINICIO PILAMALA SISLEMA** y **JORGE LUIS HILAÑO CHANATASI**, ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Paúl Patricio Romero Riera PRESIDENTE DEL TRIBUNAL	_____	2021-02-18
Ing. Fausto Ramiro Cabrera Aguayo DIRECTOR DE TRABAJO DE TITULACIÓN	_____	2021-02-18
Ing. Jorge Luis Paucar Samaniego MIEMBRO DEL TRIBUNAL	_____	2021-02-18

DEDICATORIA

Quiero dedicar esta tesis a Mariana mi ángel que siempre me cuidó, fue madre para mí y toda mi familia y sé que desde arriba está orgullosa, a mi familia, en especial a mi padre Marco y mi madre Angélica por ayudarme y apoyarme en cada meta, sueño y objetivo planteado, por ser mi fuerza y alentarme a seguir cuando pensaba que no podría más.

Henry

A Dios que me da salud, fuerza y fe para empezar un nuevo día, a mi familia en especial a mis padres Mario y Fanny que han trabajado arduamente para que pueda ser un gran profesional y me han enseñado buenos valores para ser una gran ser humano. A mi esposa Dorys y mi hija Emma que son la inspiración en todo lo que hago.

Jorge

AGRADECIMIENTO

A la Escuela Superior de Chimborazo por abrirme las puertas y haberme dado las destrezas suficientes para llegar a ser un gran profesional. A los profesores que gracias a su sabiduría y conocimiento han logrado impregnar en mí un nuevo horizonte, a mi compañera de vida Viviana por apoyarme en el transcurso de mi etapa universitaria y por último a mis amigos y compañeros por el apoyo incondicional que me han dado.

Henry

Agradezco infinitamente a la Escuela Superior Politécnica de Chimborazo y la Escuela de Ingeniería Electrónica en Control y Redes Industriales por haberme alojado por varios años y haber sido un lugar que me abra los ojos hacia un nuevo mundo de investigación y tecnología, también agradezco a mis profesores por todas sus enseñanzas y experiencias, finalmente a todos mis amigos por sus consejos y compañía brindada en varios momentos de la vida estudiantil.

Jorge

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ANEXOS.....	xv
ÍNDICE DE ABREVIATURAS.....	xvi
RESUMEN.....	xvii
SUMMARY.....	xviii
INTRODUCCIÓN.....	1

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL.....	3
1.1 Robots móviles.....	3
<i>1.1.1 Definición.....</i>	<i>3</i>
<i>1.1.2 Vehículos con ruedas.....</i>	<i>4</i>
<i>1.1.2.1 Ackerman.....</i>	<i>4</i>
<i>1.1.2.2 Configuración diferencial.....</i>	<i>4</i>
<i>1.1.2.3 Triciclo clásico.....</i>	<i>5</i>
<i>1.1.2.4 Configuración Omnidireccional.....</i>	<i>5</i>
<i>1.1.2.5 Análisis comparativo de vehículos con ruedas.....</i>	<i>6</i>
<i>1.1.3 Locomoción por patas.....</i>	<i>7</i>
1.2 Localización espacial.....	7
<i>1.2.1 Posición.....</i>	<i>7</i>
<i>1.2.2 Orientación.....</i>	<i>8</i>
1.3 Cinemática en robótica.....	9
1.4 Odometría.....	10
<i>1.4.1 Definición.....</i>	<i>10</i>
<i>1.4.2 Estimación de la posición y orientación.....</i>	<i>11</i>
<i>1.4.3 Inconvenientes de la odometría.....</i>	<i>13</i>
<i>1.4.3.1 Errores sistemáticos.....</i>	<i>13</i>
<i>1.4.3.2 Errores no Sistemáticos.....</i>	<i>13</i>
1.5 Sistema de control.....	13
<i>1.5.1 Control PID.....</i>	<i>14</i>
<i>1.5.1.1 Acción proporcional.....</i>	<i>15</i>
<i>1.5.1.2 Acción Integral.....</i>	<i>15</i>

1.5.1.3	<i>Acción Derivativa</i>	15
1.5.2	<i>Estabilidad de Lyapunov</i>	15
1.5.2.1	<i>Sistema estable</i>	16
1.5.2.2	<i>Sistema asintóticamente estable</i>	17
1.6	Algoritmos de planificación de movimiento	17
1.6.1	<i>Método probabilístico PRM</i>	18
1.6.2	<i>Algoritmo A*</i>	18
1.6.3	<i>Algoritmo A* híbrido</i>	19
1.7	Dispositivos hardware	20
1.7.1	<i>Motor DC</i>	20
1.7.2	<i>Driver de motor</i>	21
1.7.3	<i>Tarjeta de desarrollo</i>	21
1.7.4	<i>Sensores</i>	22
1.7.4.1	<i>Encoders</i>	22
1.7.4.2	<i>Sensor medición inercial</i>	25
1.8	Software	25
1.8.1	<i>SolidWorks</i>	25
1.8.2	<i>Arduino Ide</i>	26
1.8.3	<i>Matlab</i>	26
1.9	Comunicación	27
1.9.1	<i>Comunicación Serial</i>	27
1.9.2	<i>Comunicación Inalámbrica</i>	27
1.9.3	<i>Módulos radio frecuencia</i>	27

CAPÍTULO II

2.	PROPUESTA Y DISEÑO DEL PROTOTIPO	28
2.1	Metodología científica	29
2.2	Investigación	29
2.2.1	<i>Alcance de la investigación</i>	30
2.2.2	<i>Limitaciones</i>	30
2.3	Requerimientos del prototipo	30
2.4	Concepción del sistema	31
2.5	Modelo Cinemático de un Robot móvil en configuración diferencial	31
2.5.1	<i>Cinemática Directa</i>	32
2.5.2	<i>Cinemática Inversa</i>	33
2.6	Control de seguimiento de trayectoria	34
2.7	Planificadores de hoja de ruta para mapas conocidos	36
2.7.1	<i>Generación de mapas</i>	36

2.7.2	<i>Uso del planificador probabilístico PRM.....</i>	38
2.7.3	<i>Uso del planificador A* híbrido.....</i>	42
2.8	Simulación del sistema usando Matlab	43
2.8.1	<i>Importación del diseño 3D de robot móvil a Matlab</i>	43
2.8.2	<i>Desarrollo de funciones para simular movimiento del diseño 3D.....</i>	44
2.8.3	<i>Simulación de seguimiento de trayectoria</i>	45
2.9	Diseño del prototipo	48
2.9.1	<i>Estructura mecánica.....</i>	48
2.10	Sistema electrónico.....	51
2.10.1	<i>Motores y encoders</i>	51
2.10.2	<i>Driver para motor</i>	51
2.10.3	<i>Arduino Nano</i>	52
2.10.4	<i>Módulo de radio frecuencia</i>	53
2.10.5	<i>Diseño de placa principal electrónica</i>	54
2.10.6	<i>Diseño PCB.....</i>	55
2.10.7	<i>Cálculo de batería.....</i>	55
2.11	Construcción del prototipo de robot móvil.....	57
2.11.1	<i>Impresión 3D del prototipo de robot móvil.</i>	57
2.11.2	<i>Desarrollo de placa electrónica.....</i>	57
2.11.3	<i>Acoplamiento de los elementos constitutivos del prototipo</i>	58
2.12	Lectura Encoders de cuadratura.....	59
2.13	Control de velocidad en motores	62
2.14	Odometría en Arduino.....	66
2.15	Comunicación inalámbrica R.F. entre Matlab y Arduino.....	67

CAPÍTULO III

3.	VALIDACIÓN DEL PROTOTIPO	68
3.1	Ambiente de pruebas	68
3.2	Evaluación de la funcionalidad prototipo	69
3.2.1	<i>Pruebas de control de velocidad.....</i>	69
3.2.2	<i>Pruebas de posición con odometría.....</i>	72
3.2.3	<i>Prueba de trayectoria en mapa con planificador PRM.....</i>	78
3.2.4	<i>Prueba de trayectoria en mapa con planificador A* Híbrido</i>	84
3.2.5	<i>Análisis de errores de posicionamiento.....</i>	90
3.3	Pruebas de rendimiento del prototipo.....	91
3.3.1	<i>Pruebas de duración de batería.....</i>	91
3.3.2	<i>Pruebas de tiempo de ejecución de trayectorias</i>	91

CAPÍTULO IV

4.	EVALUACIÓN ECONÓMICA.....	93
4.1	Análisis de costos.....	93
	CONCLUSIONES.....	94
	RECOMENDACIONES.....	95
	GLOSARIO	
	BIBLIOGRAFÍA	

ÍNDICE DE TABLAS

Tabla 1-1:	Escala de valoración de dificultad	6
Tabla 2-1:	Evaluación de complejidad de sistemas de locomoción por ruedas	6
Tabla 1-2:	Datos técnicos del motor	51
Tabla 2-2:	Datos técnicos driver L298N	52
Tabla 3-2:	Características de tarjetas de desarrollo.....	53
Tabla 4-2:	Características de tarjetas de desarrollo.....	54
Tabla 5-2:	Elementos de consumo de corriente del robot.....	56
Tabla 6-2:	Estados anterior vs actual.	60
Tabla 1-3:	Puntos de prueba en el mapa.	68
Tabla 2-3:	Mediciones de posicionamiento.	72
Tabla 3-3:	Prueba t-student X odometría vs X patrón.....	74
Tabla 4-3:	Prueba t-student Y odometría vs Y patrón	75
Tabla 5-3:	Prueba t-student X patrón vs X deseado.	76
Tabla 6-3:	Prueba t-student Y patrón vs Y deseado	77
Tabla 7-3:	Posiciones de la trayectoria 1 con algoritmo PRM.....	78
Tabla 8-3:	Posiciones de la trayectoria 2 con algoritmo PRM.....	80
Tabla 9-3:	Posiciones de la trayectoria 3 con algoritmo PRM.....	82
Tabla 10-3:	Posición y orientación de la trayectoria 1 con Algoritmo A* Híbrido	84
Tabla 11-3:	Posición y orientación de la trayectoria 2 con Algoritmo A*híbrido	86
Tabla 12-3:	Posición y orientación de la trayectoria 3 con algoritmo A* Híbrido.	88
Tabla 13-3:	Promedio de errores absolutos de posicionamiento.....	90
Tabla 14-3:	Tiempo de descarga de batería.	91
Tabla 15-3:	Tiempo de ejecución trayectorias	92
Tabla 1-4:	Costos del prototipo.....	93

ÍNDICE DE FIGURAS

Figura 1-1:	Configuración Ackerman.....	4
Figura 2-1:	Configuración Diferencial	5
Figura 3-1:	Configuración Triciclo.....	5
Figura 4-1:	Configuración Omnidireccional.....	6
Figura 5-1:	Locomoción por Patas.....	7
Figura 6-1:	Localización en el plano	8
Figura 7-1:	Orientación en el plano	8
Figura 8-1:	Relación de cinemática de un brazo robótica.....	9
Figura 9-1:	Relación de cinemática de un robot móvil.....	9
Figura 10-1:	Desplazamiento circular de un robot móvil	10
Figura 11-1:	Desplazamiento inclinado de un robot móvil	11
Figura 12-1:	Lazo cerrado de control	14
Figura 13-1:	Estabilidad en el sentido de Lyapunov	16
Figura 14-1:	Mapa de ocupación con muestras probabilísticas	18
Figura 15-1:	Funcionamiento algoritmo A*	19
Figura 16-1:	Funcionamiento algoritmo A* híbrido.....	20
Figura 17-1:	Diagrama simplificado de una máquina de DC.	21
Figura 18-1:	Circuito electrónico de un controlador de motor de dc.....	21
Figura 19-1:	Encoder Magnético	23
Figura 20-1:	Encoder incremental	24
Figura 21-1:	Encoder absoluto.....	24
Figura 22-1:	Esquema del sistema de medición inercial	25
Figura 1-2:	Fases de desarrollo del prototipo.	29
Figura 2-2:	Concepción del sistema.....	31
Figura 3-2:	Geometría del robot móvil en configuración diferencial	32
Figura 4-2:	Trayectoria del robot móvil.....	34
Figura 5-2:	Localización del robot en una trayectoria	36
Figura 6-2:	Mapa con obstáculos.....	37
Figura 7-2:	Mapa de ocupación binaria en metros.....	37
Figura 8-2:	Rutas posibles en puntos al azar	38
Figura 9-2:	Trayectoria de finida por el algoritmo	39
Figura 10-2:	Vector con las coordenadas de la ruta.....	39
Figura 11-2:	Mapa Inflado.....	40
Figura 12-2:	Ruta nueva en el mapa inflado.	40

Figura 13-2:	Ruta nueva con distancia de conexión modificada.....	41
Figura 14-2:	Ruta con planificador A* híbrido.....	43
Figura 15-2:	Archivos STL.....	43
Figura 16-2:	Función de parámetros de las partes. stl.	44
Figura 17-2:	Función de graficadora de las partes .stl.	44
Figura 18-2:	Trayectoria generada mediante el Algoritmo PRM	47
Figura 19-2:	Trayectoria generada mediante el Algoritmo A* híbrido	47
Figura 20-2:	Dimensiones de la base del robot móvil	49
Figura 21-2:	Modelo de la base del robot móvil.....	49
Figura 22-2:	Disposición de motores en la base del robot móvil.....	50
Figura 23-2:	Ruedas locas para soporte.....	50
Figura 24-2:	Diseño general del prototipo	50
Figura 25-2:	Motor dc con encoder integrado	51
Figura 26-2:	Driver L298N.....	52
Figura 27-2:	Holybro Telemetry Radio	53
Figura 28-2:	Esquema general de conexiones	54
Figura 29-2:	Diseño de PCB y disposición de elementos.....	55
Figura 30-2:	Vista3D del diseño en PCB.....	55
Figura 31-2:	Batería Lipo 4 celdas	56
Figura 32-2:	a) Impresión de la base, b) Impresión de la estructura externa.....	57
Figura 33-2:	Placa electrónica del sistema.....	58
Figura 34-2:	Serigrafía de Placa electrónica del sistema.	58
Figura 35-2:	Elementos de placa electrónica del sistema.	58
Figura 36-2:	Estructura externa del robot.	59
Figura 37-2:	a) disposición interna de los elementos, b) Vista general del robot.....	59
Figura 38-2:	Señal digital del Encoder en sentido horario.....	60
Figura 39-2:	Señal digital del Encoder en sentido anti-horario	60
Figura 40-2:	Diagrama de bloques para un control PID de velocidad de un motor	62
Figura 41-2:	Estimación del modelo de proceso del motor	63
Figura 42-2:	Parámetros del Controlador PID	64
Figura 1-3:	Ambiente de pruebas.....	68
Figura 2-3:	Trayectoria generada al punto 1 con planificador PRM	79
Figura 3-3:	Trayectoria generada al punto 2 con planificador PRM	81
Figura 4-3:	Trayectoria generada al punto 3 con planificador PRM	83
Figura 5-3:	Trayectoria generada al punto 1 con planificador A* Híbrido	85
Figura 6-3:	Trayectoria generada al punto 2 con planificador A* Híbrido	87
Figura 7-3:	Trayectoria generada al punto 3 con planificador A* Híbrido	89

ÍNDICE DE GRÁFICOS

Gráfico 1-2:	Algoritmo de mapa de ocupación.....	38
Gráfico 2-2:	Algoritmo de uso del planificador de rutas PRM.....	41
Gráfico 3-2:	Algoritmo del planificador de rutas A* híbrido.	42
Gráfico 4-2:	Modelo simulado en Matlab.	45
Gráfico 5-2:	Diagrama de flujo de simulación.....	46
Gráfico 6-2:	Desplazamiento del robot al usar el Algoritmo PRM.....	48
Gráfico 7-2:	Desplazamiento del robot al usar el Algoritmo A* híbrido.....	48
Gráfico 8-2:	Proceso de lectura del encoder en arduino	61
Gráfico 9-2:	Evolución de la velocidad angular total del robot en el tiempo.....	63
Gráfico 10-2:	Estimación de la respuesta en lazo abierto del motor.....	64
Gráfico 11-2:	Respuesta en Lazo cerrado del motor.....	64
Gráfico 12-2:	Diagrama de flujo de Control PID de motores	65
Gráfico 13-2:	Lectura de posición y orientación en Arduino	66
Gráfico 14-2:	Diagrama de flujo de comunicación del sistema.	67
Gráfico 1-3:	Respuesta de velocidad $w_r = w_l$	69
Gráfico 2-3:	Trayectoria del robot móvil $w_r = w_l$	70
Gráfico 3-3:	Respuesta de velocidad $w_r > w_l$	70
Gráfico 4-3:	Trayectoria del robot móvil cuando $w_r > w_l$	71
Gráfico 5-3:	Respuesta de velocidad $w_r < w_l$	71
Gráfico 6-3:	Trayectoria del robot móvil $w_r < w_l$	72
Gráfico 7-3:	Medidas de posición en el eje X.....	77
Gráfico 8-3:	Medidas de posición en el eje Y.	78
Gráfico 9-3:	Trayectoria recorrida al punto 1 con planificador PRM.....	80
Gráfico 10-3:	Trayectoria recorrida al punto 2 con planificador PRM.....	82
Gráfico 11-3:	Trayectoria recorrida al punto 3 con planificador PRM.....	84
Gráfico 12-3:	Trayectoria recorrida al punto 1 con planificador A* Híbrido.....	86
Gráfico 13-3:	Trayectoria recorrida al punto 2 con planificador A* Híbrido.....	88
Gráfico 14-3:	Trayectoria recorrida al punto 3 con planificador A* Híbrido.....	90

ÍNDICE DE ANEXOS

- ANEXO A:** ESQUEMA DE CONEXIONES DEL CIRCUITO DEL ROBOT MÓVIL EN CONFIGURACIÓN DIFERENCIAL
- ANEXO B:** VISTAS DE LA ESTRUCTURA DEL ROBOT
- ANEXO C:** INTERFAZ GRÁFICA
- ANEXO D:** CONSTRUCCIÓN DEL AMBIENTE DE PRUEBAS
- ANEXO E:** DESPLAZAMIENTO DEL ROBOT MÓVIL EN EL AMBIENTE DE PRUEBAS
- ANEXO F:** HOJA TÉCNICA DE DATOS DEL ARDUINO NANO
- ANEXO G:** HOJA TÉCNICA DE DATOS DEL DRIVER L298N
- ANEXO H:** HOJA TÉCNICA DE DATOS DEL MOTOR DC TSINY CON ENCODERS
- ANEXO I:** HOJA TÉCNICA DE DATOS DEL INTEGRADO L7805
- ANEXO J:** HOJA TÉCNICA DE DATOS DEL INTEGRADO L7812
- ANEXO K:** HOJA TÉCNICA DE DATOS DEL MÓDULO DE COMUNICACIÓN “HOLYBRO TELEMETRY RADIO”

ÍNDICE DE ABREVIATURAS

DC:	Corriente Directa
GSM:	Sistema Global para las comunicaciones móviles
UMTS:	Sistema Universal de las telecomunicaciones móviles
WiFi:	Wireless Fidelity/Fidelidad Inalámbrica
RF:	Radiofrecuencia
PPR:	Pulsos por revolcion
PRM:	Probabilistic RoadMap/ Hoja de ruta Probabilista
PID:	Proporcional Integral Derivativo
CAD:	Computer Aided Design/ Diseño asistido por computadora
2D:	Dos dimensiones
3D:	Tres dimensiones
PWM:	Pulse Width Modulation/ Modulación por ancho de pulso
MDF:	Medium density fibreboard / Tablero de fibra de densidad media

RESUMEN

El objetivo de la presente investigación fue el diseño y construcción de un robot móvil autónomo en configuración diferencial para la localización en interiores al usar odometría. El desarrollo del sistema robótico comenzó con la modelación matemática del robot móvil en configuración diferencial y un control de seguimiento de trayectoria basado en la estabilidad de Lyapunov para obtener las velocidades que debe tener el robot para seguir una trayectoria deseada, toda la base matemática descrita se ha programado en Matlab con el fin de realizar la simulación del desplazamiento del robot al aplicar dos algoritmos de planificación de rutas, PRM y A* Híbrido los cuales generaran la trayectoria óptima para llegar a un punto deseado al evitar obstáculos dentro de un mapa conocido. La etapa de diseño contempló el estudio de los componentes mecánicos y electrónicos que conforman el prototipo. En Arduino se realizó un control PID para que los motores lleguen a las velocidades calculadas con el control de seguimiento de trayectoria, además se obtienen la posición y orientación mediante la odometría las cuales se envían a Matlab para que se pueda ver su desplazamiento real en una simulación 3D. La comunicación entre Matlab y Arduino se hizo mediante dos módulos de radiofrecuencia. Las pruebas se han realizado en un tablero plano de madera MDF para evitar fricción de las llantas del robot con la superficie. Las pruebas realizadas en tres diferentes trayectorias han dado como resultado un promedio de error de desplazamiento de ± 1.1 cm en el eje X, y ± 1.6 cm en el eje Y con respecto a las posiciones deseadas, se concluye que el mejor planificador es el algoritmo A*Híbrido al ofrecer mayor suavidad en las curvas de sus trayectorias y un menor tiempo de ejecución para llegar a una posición deseada.

Palabras clave: <ROBOT MÓVIL>, <ODOMETRÍA>, <CINEMÁTICA>, <ALGORITMO HOJA DE RUTA PROBABILÍSTICA (PRM)>, <ALGORITMO A* HÍBRIDO>, <ENCODERS>, <SEGUIMIENTO DE TRAYECTORIA>, <ESTABILIDAD DE LYAPUNOV>



1/02/2021

0595-DBRAI-UPT-2021

SUMMARY

The aim of this research was to design and construct an autonomous mobile robot in a differential configuration for indoor localization using odometry. The robotic system development began with the mathematical modeling of the mobile robot in differential configuration and a trajectory tracking control based on Lyapunov stability to obtain the speeds that the robot must have to follow the desired trajectory. All the mathematical bases described have been programmed in Matlab to perform the robot displacement simulation by applying two-path planning algorithms, PRM and Hybrid A*, to generate the optimal trajectory to reach the desired point by avoiding obstacles within a known map. The design stage included the study of the mechanical and electronic components that make up the prototype. Arduino made a PID control for the motors to reach the speeds calculated with the trajectory tracking power. Also, they obtain the position and orientation through the odometry sent to Matlab to show its real displacement in a 3D simulation. The communication between Matlab and Arduino was done using two radio frequency modules. The tests have been performed on a flat MDF wooden board to avoid friction of the robot tires with the surface. The tests performed on three different trajectories have resulted in an average displacement error of ± 1.1 cm on the X-axis and ± 1.6 cm on the Y-axis concerning the desired positions. We conclude that the best planner is the A*Hybrid algorithm by offering more excellent smoothness in the curves of its trajectories and a shorter execution time to reach the desired position.

KEYWORDS: <MOBILE ROBOT>, <ODOMETRY>, <KINEMATICS>, <PROBABILISTIC PATH SHEET (PRM) ALGORITHM>, <ALGORITHM A* HYBRID>, <ENCODERS>, <TRACE FOLLOWING>, <LYAPUNOV STABILITY>.

INTRODUCCIÓN

La robótica móvil ha conseguido grandes avances en la industria, a nivel doméstico y en otras áreas de trabajo. Estos robots móviles trabajan bajo principios odométricos contiguo con otros métodos y técnicas para estimar su localización dentro de su espacio de trabajo. Existen empresas internacionales dedicadas al progreso de estos robots para su manejo en diversas áreas como la compañía MIR con los prototipos MIR200 al MIR1000 los cuales son óptimos para las instrucciones internas de transporte y logística, las ultimas han sido desarrolladas para transporte de pallets y cargas pesadas. Sus prototipos incluyen odometría en base a encoders y odometría visual para estimar su localización (Mobile Industrial Robots, 2020).

El robot móvil en configuración diferencial es un tipo de robot muy fácil de usar gracias a su fácil control ya que el robot se mueve de acuerdo a la diferencia de velocidades en sus llantas por lo que puede moverse en líneas rectas o curvas (Arellano, 2015, p. 17). El punto en que se enfoca la presente investigación es abordar el cómo localizar a un prototipo de robot diferencial en interiores de un mapa conocido mediante odometría y que este pueda realizar diferentes trayectorias de forma autónoma.

En investigaciones previas (Vaca, 2018, p.5) y (Herrera y Topa, 2018, p.4) se ha usado visión artificial con un costo elevado para su implementación, en cambio un robot móvil también puede tener un alto nivel de autonomía con el uso de planificadores de rutas para encontrar la trayectoria más corta en un mapa con obstáculos, además se usa un control de seguimiento de trayectoria basado en la estabilidad de Lyapunov para disminuir el error que se da entre la trayectoria deseada calculada por los planificadores y la trayectoria real recorrida por el robot, el control de seguimiento de trayectoria obtiene velocidades que debe tener cada motor para realizar la trayectoria deseada por lo que también se implementa un control PID de velocidad para que cada motor llegue a su consigna de velocidad, todos estos controles mejoran el cálculo de la localización mediante la odometría y tiene un costo menor que implementar un robot móvil con visión artificial.

Existen varios tipos de robots y cada uno se puede desempeñar en diferentes ambientes para la presente investigación abordamos un ambiente en interiores el cual se caracteriza por ser ambientes cerrados como casas, edificios, fabricas, etc (Zurita et al., 2019, p.1081). El robot debe localizarse y desplazarse autónomamente hacia un punto deseado dentro de un mapa conocido, estas funciones son primordiales para diferentes actividades como las que realizan los robots de limpieza, transporte, logística y rescate.

OBJETIVOS

OBJETIVO GENERAL

Diseñar y construir un robot móvil autónomo en configuración diferencial para la localización en interiores usando odometría.

OBJETIVOS ESPECIFICOS

- Investigar la operatividad y métodos de funcionamiento de los robots móviles autónomos en configuración diferencial para localización en interiores usando odometría.
- Definir los requerimientos que debe cumplir el robot móvil autónomo en configuración diferencial para la localización en interiores usando odometría.
- Realizar el diseño y simulación del robot móvil en configuración diferencial que cumple con los requerimientos planteados.
- Construir el prototipo de robot móvil en configuración diferencial para la localización en interiores seleccionando el software y hardware que permita cumplir con los requerimientos planteados.
- Evaluar el funcionamiento del prototipo de robot móvil en configuración diferencial para diferentes trayectorias en interiores.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 Robots móviles

Los robots como maquinas aparecen en los años 50 como solución a las actividades y trabajos que resultaban peligrosos y cansados en la industria, sobre todo para las actividades repetitivas, la robótica empieza con el desarrollo de brazos articulados con el fin de imitar las extremidades humanas para el trabajo arduo siendo estos al principio controlados por un operado, posteriormente teleoperados y actualmente hay brazos robóticos industriales que pueden ser preprogramados para actividades (Barrientos et al., 2007, p.7).

Los robots móviles nacen de la necesidad de extender el campo que abarca la robótica a fin de no establecerse en el desarrollo de manipuladores articulados posados sobre una superficie rígida. Con esta nueva línea de investigación se buscó el incremento la autonomía de estos sistemas que ofrecen varias prestaciones para su uso en exploración y en la industria. Este nuevo enfoque de desarrollo de la robótica ha incorporado grandes avances en temas de autonomía, planificación y trayectorias (Baturone, 2005, p.8).

1.1.1 Definición

Los robots móviles pueden ser definidos como sistemas electromecánicos capaces de trasladarse de forma autónoma a un punto fijado en un entorno de trabajo (Ortigoza et al., 2007, p.3), la locomoción de los mismo puede ser de algunas formas como la representada en los vehículos con ruedas, robots articulados, con patas, acuáticos y aéreos.

Otra definición válida para robot móviles deriva de la teoría clásica de control (Aguilera et al., 2007, p.1): “Un robot móvil es un vehículo de propulsión autónoma y movimiento (re) programado por medio de control automático para realizar una tarea específica.”.

1.1.2 Vehículos con ruedas

Entre los robots móviles el desplazamiento sobre superficies sólidas y libre de obstáculos hace que la opción óptima a la movilización sea el uso o la implementación de vehículos con ruedas, al aprovechar estas características del terreno para la obtención de velocidades relativamente altas, pero a la vez existe la probabilidad de presencia de eventos no deseados como vibraciones o deslizamientos. Este tipo de sistemas móviles a la vez presentan dificultades sobre los terrenos blandos (Baturone, 2005, p.8).

1.1.2.1 Ackerman

Este tipo de configuración resulta de los vehículos convencionales, siendo así la utilización de cuatro ruedas convencionales. La movilidad de esta configuración consiste en modificar la orientación de las ruedas delanteras, al cumplir esta acción y a velocidades bajas se obtiene un rodamiento puro y se evita deslizamientos esto debido a que las ruedas delanteras giran con radio diferentes, esto logra una trayectoria curva con centro instantáneo de rotación P_1 común para ambas ruedas. En la figura 1 se puede observar que ($\theta_1 > \theta_0$), esta condición permite evitar los deslizamientos al ser ligeramente mayor el ángulo interno θ_1 (Marín et al., 2013, p.2).

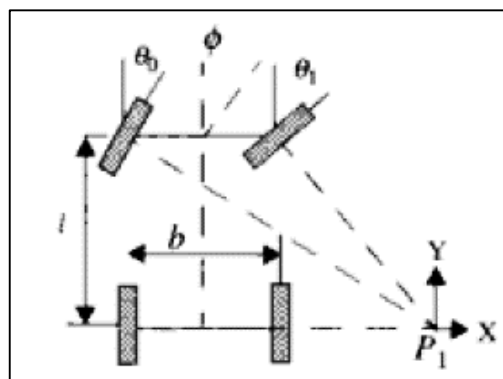


Figura 1-1: Configuración Ackerman

Fuente: Baturone,2005

1.1.2.2 Configuración diferencial

Esta configuración consiste en un cuerpo rígido caracterizado por el uso de dos ruedas laterales no deformables y no orientables en la cual su movilidad consiste en la diferencia de velocidades de estas además generalmente se hace uso de otras ruedas únicamente para soporte de su estructura. Las ruedas laterales deben estar separadas una distancia definida, las ruedas deberán ser del mismo radio y tener una forma que evite los deslizamiento y las vibraciones (Silva et al., 2010, p.3).

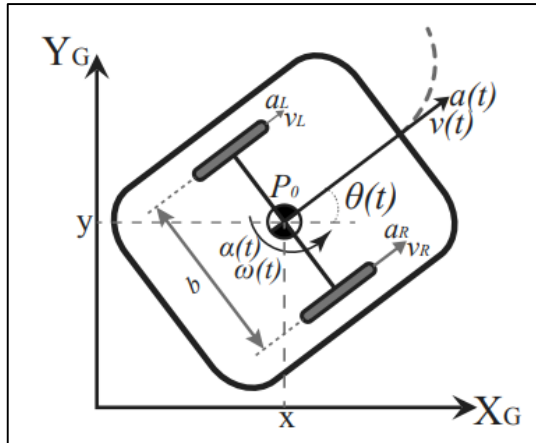


Figura 2-1: Configuración Diferencial

Fuente: Ortigoza,2007

1.1.2.3 Triciclo clásico

Esta configuración está basada en una morfología de tres ruedas una delantera y dos traseras laterales pasivas. La movilidad de esta se basa en la acción de la rueda delantera al controlar su tracción y por ende su dirección. En superficies inclinadas su centro de gravedad tiende a desplazarse (Reyes y Reyes, 2018, p.1).

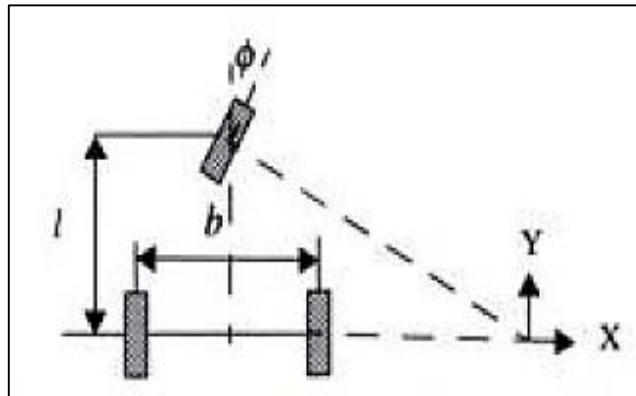


Figura 3-1: Configuración Triciclo

Fuente: Baturone, 2005

1.1.2.4 Configuración Omnidireccional

Esta configuración un movimiento fácil, consta tres ruedas omnidireccionales con rodillos dispuestos en 90 grados en relación a su rueda principal, permitiendo movilidad lateral. Las tres forman un triángulo equilátero como base del móvil (Bohórquez y Machuca, 2018,p.24).

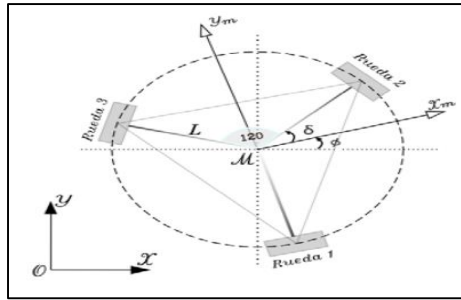


Figura 4-1: Configuración Omnidireccional

Fuente: Bendezú, 2018

1.1.2.5 Análisis comparativo de vehículos con ruedas

Las configuraciones mostradas anteriormente tienen sus ventajas y desventajas dependiendo el uso a cuál van destinadas, estas características pueden influir mucho al momento de elegir una para determinada tarea y saber escoger la correcta con bases técnicas es recomendable y ayuda a evitar problemas futuros. A continuación, se muestra una tabla en la escala de Likert de acuerdo a la complejidad de su configuración y costo. La configuración diferencial presenta la ponderación más baja ubicándose como una configuración de dificultad media, tanto en control como implementación, además de ser de bajo costo (Vásconez, 2018, p.11).

Tabla 1-1: Escala de valoración de dificultad

Dificultad baja	Dificultad media	Dificultad alta
1	2	3
0-33%	34-66%	67-100%

Fuente: Vásconez, 2010

Realizado por: Pilamala Henry & Hilaño Jorge, 2020

Tabla 2-2: Evaluación de complejidad de sistemas de locomoción por ruedas

Características	Ackerman	Triciclo	Diferencial	Omnidireccional
Control de posición	2	1	1	3
Implementación	2	2	1	3
Maniobrabilidad	3	2	2	1
Estabilidad	1	2	1	1
Costo	2	2	1	3
Total de incidencias	10	9	6	11
Ponderación total	66.67%	60%	40%	73.33%

Fuente: Vásconez, 2010

Realizado por: Pilamala Henry & Hilaño Jorge, 2020

1.1.3 Locomoción por patas

Este tipo de movilidad aísla el cuerpo robótico del suelo mediante puntos de soporte. Con las modificaciones que este puede tener presenta factibilidad de evadir obstáculos en caminos complicados siendo esta su ventaja ante los vehículos móviles que pierden su beneficios en terrenos difíciles (Baturone, 2005, p.34).

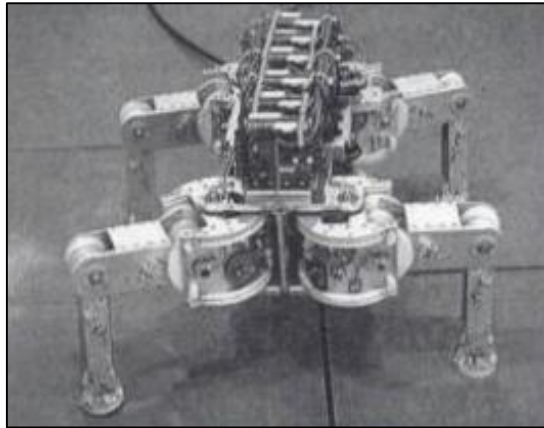


Figura 5-1: Locomoción por Patas

Fuente: Baturone, 2005

1.2 Localización espacial

Los robots móviles se desplazan en el plano por lo que desprecia la posibilidad de un espacio tridimensional salvo que en algún caso sea necesario su consideración. Se considera un plano para el cual es necesario dos datos posición y orientación, para la posición son datos de coordenadas (x,y) y para la orientación un ángulo θ .

1.2.1 Posición

Se considera al plano de trabajo como un sistema $\{A\}$, la posición del sistema será representada en un vector ${}^A P = \begin{bmatrix} P_x \\ P_y \end{bmatrix}$ siendo los vectores unitarios \hat{x}_A, \hat{y}_A . Otra forma de representación es en coordenadas polares con nombre r del modulo de distancia a un punto en el sistema y con θ grados de \hat{x}_A hacia r (Baturone, 2005, p.40).

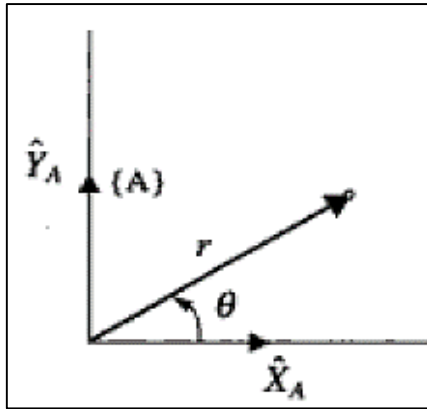


Figura 6-1: Localización en el plano

Fuente: Baturone, 2005

1.2.2 Orientación

La orientación en el plano del sistema $\{A\}$ se la hace a partir de un ángulo θ desde el eje \hat{x}_A al punto en cuestión, ahora al considerar un nuevo movimiento a partir sistema anterior se traza un nuevo sistema $\{B\}$ de igual manera con vectores unitarios \hat{x}_B, \hat{y}_B . Al expresar los vectores unitarios del sistema B en el sistema A aparecen un término importante la matriz de rotación (Baturone, 2005, p.41).

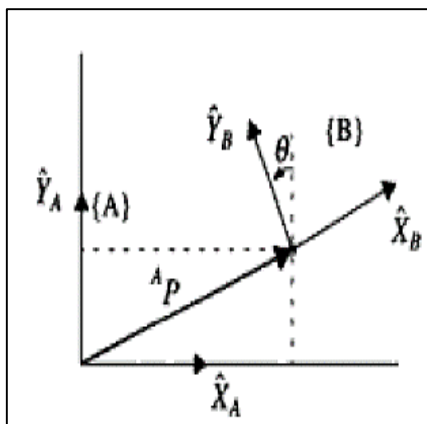


Figura 7-1: Orientación en el plano

Fuente: Baturone, 2005

La matriz de rotación se define como ${}^A_B R = [A_{\hat{x}_B}, A_{\hat{y}_B}]$, al considerar el ángulo θ la matriz 1 resulta:

$${}^A_B R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (1)$$

1.3 Cinemática en robótica.

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia (Barrientos et al., 2007, p.93). La cinemática de robots se ocupa de la configuración de los robots en su espacio de trabajo, las relaciones entre sus parámetros geométricos y las restricciones impuestas en sus trayectorias. Las ecuaciones cinemáticas dependen de la estructura geométrica del robot, por ejemplo, un robot fijo puede tener una estructura cartesiana, cilíndrica, esférica o articulada, y un robot móvil puede tener una, dos, tres o más ruedas con o sin restricciones en su movimiento (Tzafestas, 2013, p.31).

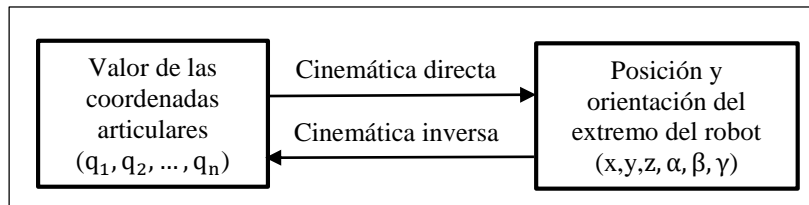


Figura 8-1: Relación de cinemática de un brazo robótica

Fuente: Barrientos,2007

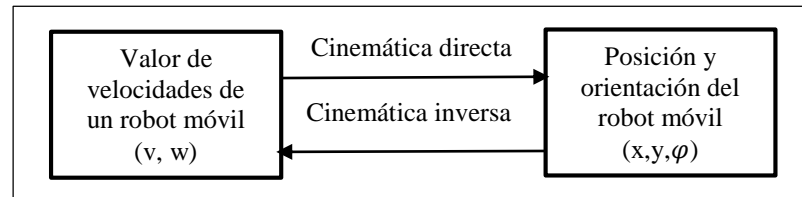


Figura 9-1: Relación de cinemática de un robot móvil

Fuente: Barrientos,2007

A la cinemática de un robot se la puede analizar de dos formas, la primera es mediante la cinemática directa y consiste en determinar cuál es la posición y orientación del extremo final de un brazo robótico conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot, para el caso de un robot móvil consiste en determinar la posición que se ha desplazado el robot conociendo los valores de velocidad en las llantas ; la segunda forma de análisis es mediante la cinemática inverso para resolver la configuración que debe adoptar un brazo robótico para una posición y orientación del extremo conocidas, en un robot móvil la cinemática inversa determina las velocidades que debe tener el robot para poder realizar un desplazamiento deseado (Barrientos et al., 2007, pp. 93-94).

1.4 Odometría

1.4.1 Definición

Es la técnica que estudia la estimación de la localización de un robot móvil mediante el uso de sensores en sus ruedas como los encoders que permiten saber el desplazamiento lineal a partir de datos discretos de las revoluciones. Con el uso de ecuaciones geométricas que caracterizan a la configuración del robot móvil se puede estimar la posición relativa en el plano respecto a una posición inicial conocida (Vaca y Rosero, 2018 ,p.1).

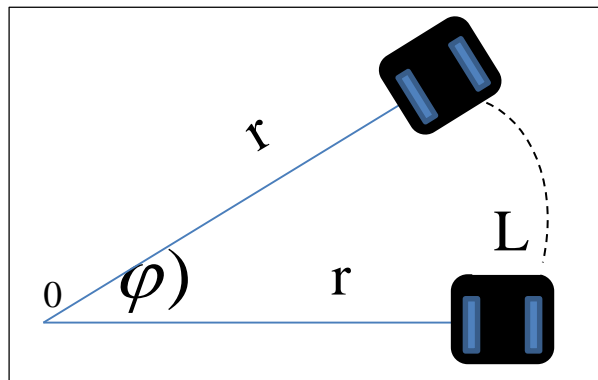


Figura 10-1: Desplazamiento circular de un robot móvil

Realizado por: Pilamala, H; Hilaño, J, 2020

Los desplazamientos de cada rueda del robot móvil son los siguientes.

$$\Delta s_{L,i} = \frac{\pi D}{nC_e} N_{L,i} \quad (2)$$

$$\Delta s_{R,i} = \frac{\pi D}{nC_e} N_{R,i} \quad (3)$$

D : Diámetro de la rueda

n : Reducción eje rueda

C_e : Resolución del encoder

$N_{L,i}$: Cantidad de pulsos leídos por el encoder izquierdo para tiempo de muestreo i

$N_{R,i}$: Cantidad de pulsos leídos por el encoder izquierdo para tiempo de muestreo i

$\Delta s_{L,i}$: Desplazamiento de la llanta izquierda

$\Delta s_{R,i}$: Desplazamiento de la llanta derecha

Mediante el análisis geométrico de la figura 10-1 se obtiene el desplazamiento angular y el desplazamiento lineal en el punto central del robot en función de los desplazamientos lineales de cada llanta.

$$\Delta\varphi_i = \frac{\Delta S_{L,i}}{r} \quad (4)$$

$$\Delta\varphi_i = \frac{\Delta S_{R,i}}{r + L} \quad (5)$$

$$\Delta\varphi_i = \frac{\Delta S_{c,i}}{r + \frac{L}{2}} \quad (6)$$

$$\Delta\varphi_i = \frac{\Delta S_{R,i} - \Delta S_{L,i}}{L} \quad (7)$$

$$\Delta S_{c,i} = \frac{\Delta S_{R,i} + \Delta S_{L,i}}{2} \quad (8)$$

L : Distancia de separación entre las llantas

$\Delta\varphi_i$: Desplazamiento angular

$\Delta S_{c,i}$ Desplazamiento del robot en torno a su centro

1.4.2 Estimación de la posición y orientación

Una vez obtenido los desplazamientos anteriores se procede a calcular la posición x, y y la orientación φ en el instante $i - 1$ (Tzafestas, 2013, pp. 490-491). Las ecuaciones surgen del análisis trigonométrico de la figura 11-1.

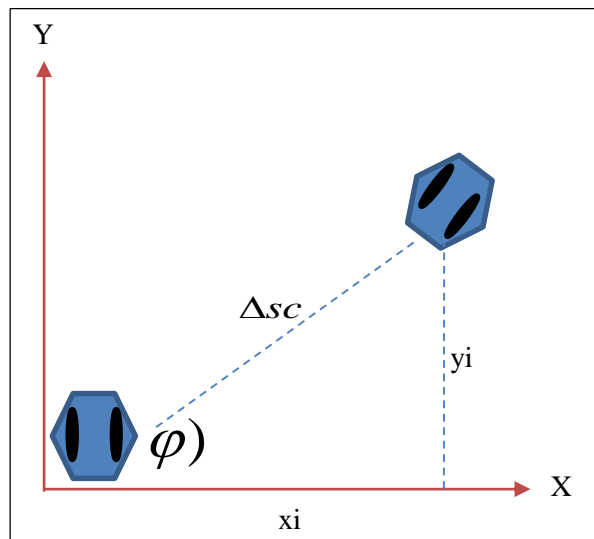


Figura 11-1: Desplazamiento inclinado de un robot móvil

Realizado por: Pilamala, H; Hilaño, J, 2020

$$x_i = x_{i-1} + \Delta S_{c,i} \cos\varphi_i \quad (9)$$

$$y_i = y_{i-1} + \Delta S_{c,i} \sen\varphi_i \quad (10)$$

$$\varphi_i = \varphi_{i-1} + \Delta\varphi_i \quad (11)$$

De acuerdo con Watanabe y Yuta (1990, p.2011) $v(t)$ es la velocidad promedio o lineal y $\omega(t)$ es la velocidad angular total que se genera con el movimiento las dos ruedas del robot móvil diferencial, las ecuaciones 12, 13, 14 determinan la posición y orientación del robot en un instante de tiempo.

$$x(t + \tau) = x(t) + \int_t^{t+\tau} v(t) \cos\theta(t) dt \quad (12)$$

$$y(t + \tau) = y(t) + \int_t^{t+\tau} v(t) \sen\theta(t) dt \quad (13)$$

$$\theta(t + \tau) = \theta(t) + \int_t^{t+\tau} \omega(t) dt \quad (14)$$

La relación integral puede ser aproximada como una ecuación diferencial con un intervalo de muestreo τ (Watanabe y Yuta, 1990, p.2012).

$$x(t + \tau) = x(t) + \tau v(t) \cos\theta(t) \quad (15)$$

$$y(t + \tau) = y(t) + \tau v(t) \sen\theta(t) \quad (16)$$

$$\theta(t + \tau) = \theta(t) + \tau v(t) w(t) \quad (17)$$

En las ecuaciones 15, 16, 17 se toma en cuenta las velocidades tanto lineal como angular a diferencia de las ecuaciones 9, 10, 11 que están en función del desplazamiento lineal y angular, sin embargo, la odometría también permite calcular estas velocidades mediante las siguientes ecuaciones.

$$\omega_R = \frac{2\pi}{tnC_e} N_R \quad (18)$$

$$\omega_L = \frac{2\pi}{tnC_e} N_L \quad (19)$$

$$v_R = \frac{D\pi}{tnC_e} N_R \quad (20)$$

$$v_L = \frac{D\pi}{tnC_e} N_L \quad (21)$$

$$v = \frac{v_r + v_l}{2} = \frac{(\omega_r + \omega_l)R}{2} \quad (22)$$

$$\omega = \frac{v_r - v_l}{L} = \frac{(\omega_r - \omega_l)R}{L} \quad (23)$$

D : Diámetro de la rueda de la rueda

R : Radio de la rueda

L : Distancia de separación entre las llantas

n : reducción eje rueda
 C_e : Resolución del encoder
 N_L : Cantidad de pulsos leídos por el encoder izquierdo
 N_R : Cantidad de pulsos leídos por el encoder izquierdo
 t : Tiempo de muestreo
 v_r : Velocidad lineal llanta derecha.
 v_l : Velocidad lineal llanta izquierda.
 ω_r : Velocidad angular llanta derecha
 ω_l : Velocidad angular llanta izquierda
 v : Velocidad lineal total
 ω : Velocidad angular total

1.4.3 Inconvenientes de la odometría

1.4.3.1 Errores sistemáticos

Los errores sistemáticos son los que aparecen a partir del propio sistema como un mal dimensionamiento de la estructura del sistema, diferencia entre los radios de las llantas, las llantas no son uniformes y producen vibración. En fin, tanto el software como hardware pueden presentar problemas o dificultades que pueden afectar al correcto funcionamiento del sistema al generar errores que con el tiempo son acumulativos (Arellano, 2015, p. 10).

1.4.3.2 Errores no Sistemáticos

Los errores no sistemáticos son parte del entorno del sistema, y por lo tanto no se pueden controlar o son muy difíciles de gestionar como por ejemplo que una rueda patine, que una superficie se a resbaladiza y genere un desplazamiento involuntario afectando la estimación de la posición del robot móvil (Arellano, 2015, p.11).

1.5 Sistema de control

Los sistemas de control están presente en todos los procesos repetitivos que requieren un resultado deseado o prefijado, en la industria estos sistemas están formados por un conjunto de dispositivos y elementos encargados de controlar procesos para evitar las posibilidades de fallo (Kuo, 1996, p.2).

1.5.1 Control PID

El controlador Proporcional-Integral-Derivativo es un tipo de control robusto que ofrece solución y fácil manejo a una alta gama de procesos en la industria. El PID es un controlador que incorpora tres términos o partes, el mismo se ha convertido casi en un estándar en el área de control automático desde inicios de siglo pasado. Este controlador no ha desaparecido con los años, al contrario, ha evolucionado con el desarrollo de la tecnología, ha pasado de ser analógico a digital y en la actualidad se puede encontrar su implementación en todo tipo de equipo de control (Visioli, 2006, p.1).

Este controlador se lo usa en sistemas de control de lazo cerrado en procesos donde mediante un sensor la salida, $H(s)$, del proceso puede ser medida y comparada con un punto de referencia $R(s)$, set point o valor deseado del proceso $G(s)$ el cual se resta de la salida medida por el sensor y se obtiene una señal de error $E(s)$ la cual va ser la entrada o señal de actuación al controlador $C(s)$, con el fin de reducir el error, todo esto se facilita al trabajar en términos de frecuencia (Ogata, 2003, p.7).

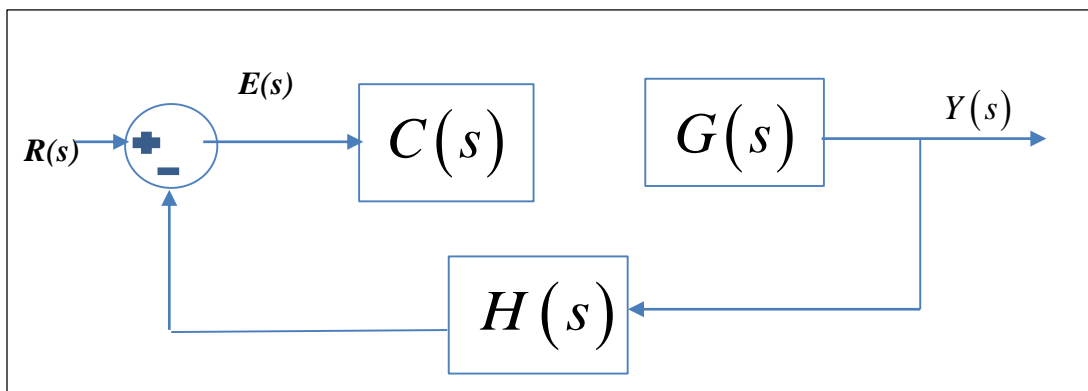


Figura 12-1: Lazo cerrado de control

Fuente: Propia

La estructura clásica de un controlador PID consta de tres acciones de control que aporta un efecto en la variable del proceso para corregirla u actuar sobre ella, disminuyendo el error con respecto a la consigna. Su forma clásica en función de transferencia se representa en la ecuación 24.

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (24)$$

Donde K_p es conocido como ganancia proporcional, T_i el tiempo integral y T_d el tiempo derivativo, todas estas son constantes (Visioli 2006, p.7).

1.5.1.1 Acción proporcional

La acción de control en sencilla, si la variable de error es grande la acción de control también (Visioli, 2006, p.5) y trabaja de la siguiente manera como lo indica la ecuación 25.

$$u(t) = K_p \cdot e(t) \quad (25)$$

Y su función de transferencia lo indica la ecuación 26.

$$C(s) = K_p \quad (26)$$

1.5.1.2 Acción Integral

La acción integral se basa en la integral de la señal del error de control, con el propósito de eliminar el error en estado estacionario, a la señal de error se aplica una integral y se multiplica por una constante K_i (Visioli, 2006, p.5), esto de acuerdo con la ecuación 27.

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad (27)$$

Y su función de transferencia está dado por la ecuación 28:

$$C(s) = \frac{K_i}{s} \quad (28)$$

1.5.1.3 Acción Derivativa

La acción de control derivativa ejerce un efecto predictivo de corrección sobre el control de error, manteniendo el error al mínimo y corrigiéndolo al mismo tiempo que se produce (Visioli, 2006, p.6), según la ecuación 29.

$$u(t) = K_d \frac{de(t)}{dt} \quad (29)$$

Y su función de transferencia se denota según la ecuación (30).

$$C(s) = K_d s \quad (30)$$

1.5.2 Estabilidad de Lyapunov

Un sistema estable es un sistema dinámico con una respuesta acotada para una entrada acotada. La característica de que un sistema realimentado en lazo cerrado sea estable o inestable trata sobre la estabilidad absoluta, dado el sistema en lazo cerrado estable se puede constatar el grado de estabilidad, a este grado de estabilidad trata sobre la estabilidad relativa (Dorf y Bishop, 2005, p.312).

La estabilidad de un sistema depende del error entre el sistema de referencia basado en el modelo y el sistema real. Si se aplica el controlador diseñado basado en un modelo al sistema real, éste puede no ser estable, por lo que se debe diseñar el sistema de control definiendo primeramente el rango de posibles errores y después diseñar el controlador, de esta forma si el error del sistema está en dicho rango, el sistema de control diseñado permanezca estable (Ogata 2003, p.2).

La teoría de estabilidad de Lyapunov es una de las herramientas más importantes en el análisis de sistemas no lineales, estos sistemas pueden presentar oscilaciones de amplitud y periodo fijos sin excitación externa. Además, los cambios cualitativos de los parámetros de un sistema no lineal generan cambios cualitativos de las propiedades del sistema. En los sistemas lineales, pequeñas diferencias en las condiciones iniciales pueden causar sólo pequeñas diferencias en la salida (Anzures et al., 2008, p.6). La estabilidad de Lyapunov puede determinar a un sistema de las siguientes formas:

1.5.2.1 Sistema estable

Se dice que el equilibrio $x = 0$ del sistema $\dot{x} = f(x)$ es estable en el sentido de Liapunov, o simplemente estable (Gordillo 2009, pg 6), en caso de que para cada $\varepsilon > 0$ existe un valor de $\delta > 0$ tal como la ecuación 31.

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon \forall t \geq 0 \quad (31)$$

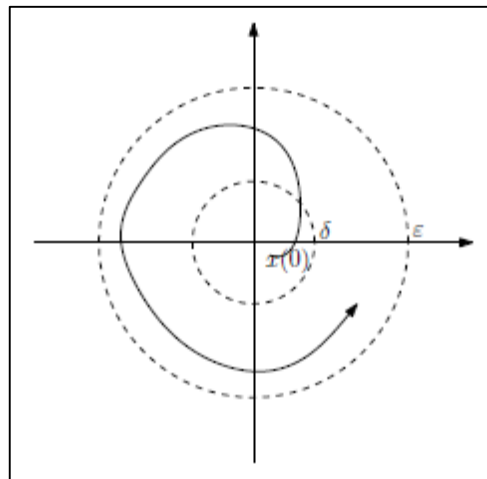


Figura 13-1: Estabilidad en el sentido de Lyapunov

Fuente: Gordillo, 2009

Esta definición presenta que, si la condición inicial está suficientemente cerca del equilibrio, la trayectoria no se sobrepase bastante del mismo. En contexto, la definición va más allá de esta definición ya que establece que si se especifica cuanto, ε , se puede desviar la trayectoria del equilibrio, existe una distancia máxima, δ , de la condición inicial al equilibrio que garantiza el desempeño de esa especificación. En la figura 1 se observa en

el plano una trayectoria que parte adentro del círculo de radio δ y no sale del círculo de radio ε , si el sistema es estable esta trayectoria no debe pasar de este círculo aunque pase el tiempo ni ninguna otra trayectoria que parta del círculo de radio δ (Gordillo 2009, pp. 6-7).

1.5.2.2 Sistema asintóticamente estable

Se dice que el equilibrio $x = 0$ del sistema $\dot{x} = f(x)$ es asintóticamente estable si es localmente estable en el sentido de Lyapunov y además existe un valor de $\delta > 0$ tal que la ecuación (32)

$$\|x(0)\| < \delta \implies \lim_{t \rightarrow \infty} x(t) = 0 \quad (32)$$

El concepto de estabilidad asintótica reduce las ambigüedades de estabilidad en el sentido de Lyapunov imponiendo que las trayectorias finalmente converjan hacia el equilibrio (Gordillo 2009, p.7).

El resultado fundamental de Lyapunov es llegar a un teorema que define el denominado método directo de Lyapunov (Gordillo 2009, pg 7), Este método no requiere el conocimiento del tiempo de respuesta del sistema (Tzafestas 2013, p.144).

El teorema trata lo siguiente: Considérese el sistema $\dot{x} = f(x)$ y supóngase que existe una función continuamente diferenciable $V: D \rightarrow R$ tal que

$$V(0) = 0 \quad (33)$$

$$V(x) > 0, x \in D, x \neq 0, \quad (34)$$

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \leq 0, x \in D \quad (35)$$

Entonces, el equilibrio $x = 0$ es estable en el sentido de Lyapunov. Si además se cumple en la ecuación (14).

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0, x \in D, x \neq 0 \quad (36)$$

el equilibrio $x = 0$ es asintóticamente estable.

1.6 Algoritmos de planificación de movimiento

Los problemas de movimiento del robot ante un entorno que presenta obstáculos deben ser resueltos mediante una previa planificación de posible rutas o espacios libres de colisión, a diferencia de un brazo robótico articulado en el que se supone un espacio libre para su movimiento (Siciliano et al. 2010, p.523).

1.6.1 Método probabilístico PRM

Los planificadores probabilísticos son de los métodos más eficaces en la generación de hojas de ruta en mapas de ocupación, poseen una alta eficiencia en mapas de configuración de dimensiones altas. Se basan en técnicas de muestreo que busca posibles configuraciones de redes conectadas en los espacios libres de obstáculos dentro de mapas, obteniendo así posibles hojas de rutas que solucionen el problema de planificación de movimiento del robot diferencial (Siciliano et al. 2010, p.541).

El algoritmo comienza generando puntos aleatorios en el mapa de ocupación utilizado una distribución de probabilidad uniforme, posterior se verifica que las muestras no generen colisión en el mapa. En caso de que una de las muestras se presente en un espacio de colisión se descarta, si una de las muestras no presenta colisión se agrega a la hoja de ruta y se conecta a través de rutas locales (Siciliano et al. 2010, p.542).

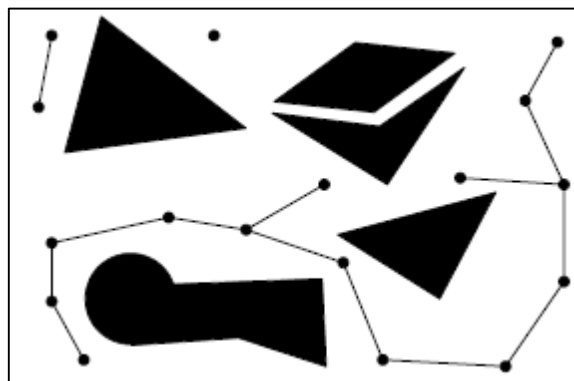


Figura 14-1: Mapa de ocupación con muestras probabilísticas

Fuente: Siciliano, 2010

1.6.2 Algoritmo A*

Este algoritmo diseñado por Peter E. Hart utiliza la división en rejillas de mapas de ocupación, generando así un espacio de trabajo que resulta en celdas usadas para trazar trayectorias libres de obstáculos. La elección de la mejor ruta se basa en funciones de costo entre los puntos inicio y final (Hart et al., 1968, p.100).

Es un algoritmo de búsqueda de rutas iterativo el cual mediante celdas libres encontradas genera caminos o rutas desde un punto Niñicial hasta un punto Nobjetivo. De forma iterativa explora celdas o nodos vecinos hasta llegar al punto deseado, lo hace de forma de árbol de expansión con origen en Niñicial, el conjunto de nodos expandidos en árbol forman las diferentes rutas y la condición para la selección de la mejor ruta posible dentro del mapa se basa en funciones de

costo asociadas a las mismas, cada nodo explorado se le añade su función de costo y se lo registra es decir el nodo que represente menor costo en el listado es la mejor ruta (Hart et al,1968, p.101).

La búsqueda en las casillas vecinas es gracias al operador de expansión al comprobar que cada celda esté libre de obstáculos o colisión, en el caso de que se encuentre libre de colisión se le asigna un coste asociado estimado mediante una función de coste.

Esta estimación es de forma heurística con respecto al coste asociado. La ecuación (37) de función de costo en la cual se basa costa de dos términos el primero $g(N)$ representa el costo de camino o nodo desde Niñicial hasta un punto N y el segundo término es $h(N)$ que representa una aproximación heurística el coste de la ruta de menor coste entre N y Nobjetivo (Hart et al.,1968, p.102).

$$f(N) = g(N) + h(N) \quad (37)$$

En la figura se puede observar el funcionamiento del algoritmo, el cuadrado azul es Niñicial y el amaillo es Nobjetivo, los cuadros azules son la búsqueda por la función de expansión, la roja representa la ruta a N y la gris es la ruta selescionada por el menor coste que tiene (Preciado, 2018, p.43).

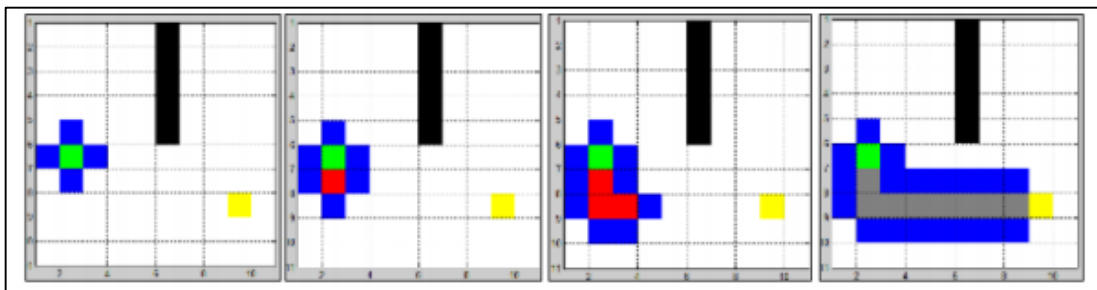


Figura 15-1: Funcionamiento algoritmo A*

Fuente: Preciado, 2018

1.6.3 Algoritmo A* híbrido

El algoritmo modificado de A*, conocido como A* híbrido tiene el mismo funcionamiento que su predecesor, pero con nuevas consideraciones que mejora su función y calidad de ruta para vehículos no holonómicos. Esta versión del algoritmo toma en consideración la cinemática del vehículo para seleccionar los nodos potenciales vecinos como siguientes en la ruta (Sedighi et al, 2019, p.2).

El algoritmo trabaja en mapas de ocupación binaria que les permite trabajar de forma fácil con funciones de costo para cada cuadrícula, pero el algoritmo aún necesita una mejor función de costo heurístico que le permita tener una mejor comprensión del mapa global y su ruta de rastreo correcta (Sedighi aet al., 2019, p.2).

El algoritmo usa un mapa de cuadrícula de ocupación en que el entorno se ilustra de manera discreta para planificar el movimiento del vehículo en un dominio continuo a diferencia de su antecesor que para pasar al siguiente nodo sucesor elige una maniobra lineal lo que facilita el movimiento a vehículos holonómicos.

La necesidad de amortiguar las rutas trazadas y eliminar esquinas rectas de 90° hace que este algoritmo sea eficiente para vehículos diferenciales o de configuración Ackerman al suavizar sus giros y maniobras a realizar.

Este método mejorado usa la cinemática del vehículo para predecir el movimiento del vehículo al considerar parámetros del vehículo como la velocidad, marcha, ángulo de dirección y otros parámetros. Este método ayuda al planificador a seleccionar el nodo sucesor correcto que un robot no holonómico puede seguir, usa la misma función de costo que el algoritmo normal A-star para encontrar la ruta para llegar al nodo objetivo, además, agregará algunos costos más a eso dependiendo del movimiento del vehículo y sus condiciones de movimiento (Sedighi et al, 2019, p.2).

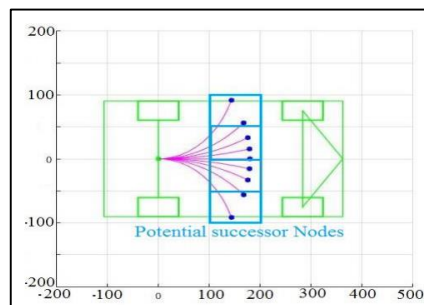


Figura 16-1: Funcionamiento algoritmo A* híbrido

Fuente: Sedighi, S.,2019

1.7 Dispositivos hardware

1.7.1 Motor DC

Los motores de cd son máquinas eléctricas que convierte la energía eléctrica de DC en energía mecánica (Chapman, 2005, p. 345). El motor DC puede contener una caja reductora que puede aumentar o disminuir la velocidad de rotación del rotor, también permite entregar un buen torque en un pequeño tamaño y bajo voltaje (Frías, 2018, p.21) .

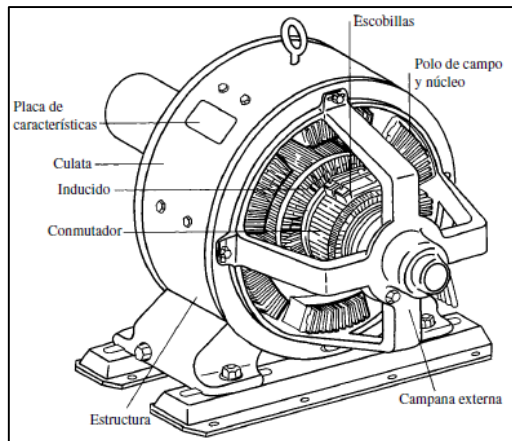


Figura 17-1: Diagrama simplificado de una máquina de DC.
Fuente: Chapman, 2005

1.7.2 Driver de motor

Es un circuito electrónico encargado del control del motor de una manera sencilla y eficaz permitiendo controlar el sentido de giro y la velocidad por medio de salidas digitales y PWM enviadas desde un microcontrolador o sistema embebido. Existen gran diversidad de driver que permiten maniobrar motores DC, servomotores, motores paso a paso, entre otros. Para realizar el dimensionamiento del driver se debe tomar en cuenta el consumo de voltaje y corriente de los motores (Vásconez, 2018, p.21).

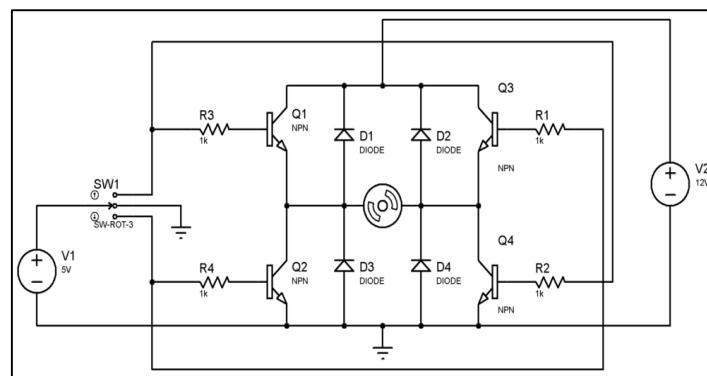


Figura 18-1: Circuito electrónico de un controlador de motor de dc.
Fuente: <https://n9.cl/jczln>

1.7.3 Tarjeta de desarrollo

Una tarjeta de desarrollo en el ámbito de la ingeniería es una herramienta para el diseño y construcción de sistemas digitales o analógicos, siendo dispositivo muy valioso para optimizar los procesos de diseño al disminuir del tiempo de comprobación (González y Silva, 2013, p.2).

Estas tarjetas están conformadas por placas electrónicas en conjunto con un lenguaje de programación lo cual facilita el desarrollo varias aplicaciones, el avance de la tecnología ha hecho posible crear diversas tarjetas de desarrollo y con características específicas como entradas y salidas digitales como analógicas, puerto USB, memoria para el almacenamiento de datos, entre otras características que varían según el tipo de placa, entre las más utilizadas están Raspberry Pi, Intel Galileo y Arduino (González y Carrillo, 2019, p.14).

1.7.4 Sensores

Un sensor es un dispositivo de entrada que provee una salida manipulable de la variable física medida. Los sensores entregan señales eléctricas a la salida ya sean análogas o digitales, siendo estas las más utilizadas en los sistemas de medición actuales (Corona et al., 2014, p.2).

Los sensores diseñados para robots se parecen a los sensores del sistema sensorial humano los cuales envían señales de entrada al cerebro para su procesamiento, utilización y operación. El manejo de sensores en robótica es de vital importancia para cerrar los circuitos de control de retroalimentación que garantiza la ejecución eficaz y autónoma de robots en aplicaciones de la vida real. Mediante los sensores se pueden crear varios métodos de detección del ambiente en que se mueve el robot al generar capacidades de mayor nivel e inteligencia que superan las tareas de ejecución repetitivas preprogramadas (Tzafestas, 2013, p.101).

En la odometría el objetivo es saber la posición y obtener una lectura correcta del desplazamiento de un robot móvil en base a un sistema de referencia adaptado al entorno en que se mueve el robot, para saber esta posición se deben utilizar sensores de posicionamiento que se muestran a continuación.

1.7.4.1 Encoders

Los encoders son sensores que generan señales digitales en respuesta al movimiento, son usados en conjunto con dispositivos mecánicos tales como engranajes, ruedas de medición o flechas de motores, estos pueden ser utilizados para medir movimientos lineales, velocidad y posición (Romero, 2013, p.1). Los encoders están contruidos principalmente en base a dos sensores:

- Sensores magnéticos o de efecto Hall:

El encoder basado en sensores magnéticos determina, sin contacto, la velocidad de rotación y las posiciones a partir de la medición del campo magnético generado por un imán multipolar y un sensor magnético de efecto Hall (Ojeda y Zúñiga, 2018, p.790). El efecto Hall determina que, si una corriente eléctrica fluye a través de un conductor en presencia de un campo magnético, se

producirá una fuerza transversal que busca equilibrar el efecto de dicho campo al generar un voltaje Hall medible en los extremos del conductor (Corona et al., 2014, p.172).

Este encoder tiene dos sensores de efecto Hall que generan dos señales de pulsos digitales desfasada en 90° eléctricos o en cuadratura gracias a un disco magnético giratorio, montado en el eje trasero del motor, a estas señales de salida, se les llama comúnmente A y B. Estos dos sensores se activan y desactivan en una secuencia que permite saber la dirección y el número de desplazamientos que han ocurrido en el encoder (Reyes, 2011, p.83).

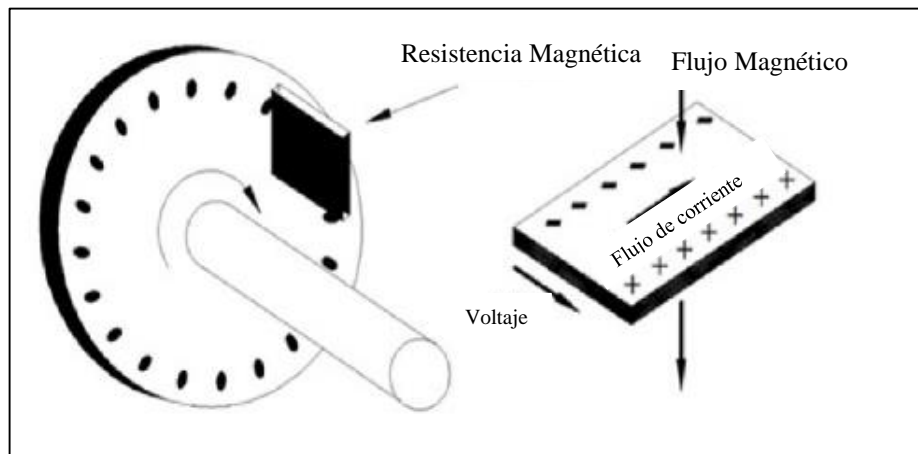


Figura 19-1: Encoder Magnético

Fuente: <https://n9.cl/k9ni7>

- Sensor óptico:

El encoder consta de un disco de sector con segmentos en blanco y negro junto con un diodo emisor de luz y un fotodiodo. El fotodiodo detecta la luz reflejada durante un segmento blanco, pero no durante un segmento negro (Tzafestas, 2013, p.104). Los encoders basados en sensores ópticos se dividen en dos tipos:

- Encoder incremental o de cuadratura:

Es el dispositivo de retroalimentación más común de los robots. Se monta un codificador en cada eje de unión del motor. Un encoder diferencial tiene una segunda pista agregada para generar un pulso que ocurre una vez por revolución (señal de índice) que se utiliza para indicar una posición absoluta. Para obtener información sobre la dirección de rotación, las líneas en el disco se leen por dos elementos de fotodiodo diferentes que miran el patrón del disco con un desplazamiento mecánico de un cuarto del tono del par de líneas entre ellos. A medida que el disco gira, los dos fotodiodos generan señales que tienen una diferencia de 90 fases. Estas dos señales generalmente se denominan señales de cuadratura A y B. La dirección en el sentido de las agujas del reloj

generalmente se define como positiva cuando el canal A se mueve antes que el canal B (Tzafestas, 2013, p.105).

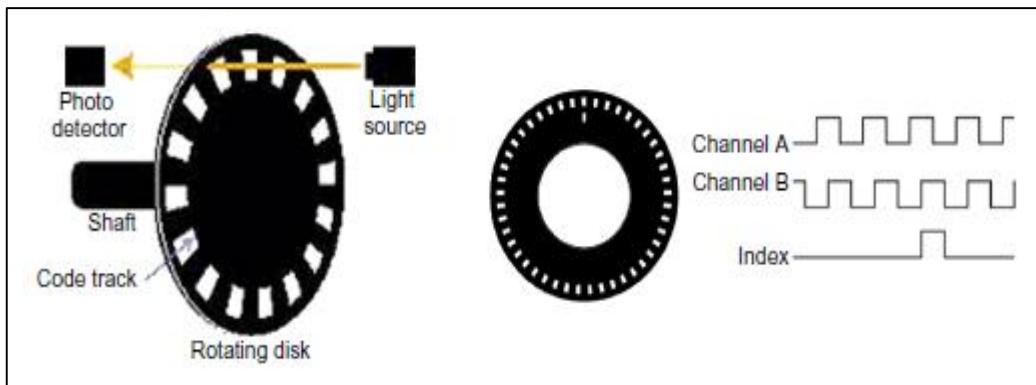


Figura 20-1: Encoder incremental
Fuente: Tzafestas S. 2014

- Encoder absoluto:

Consiste en un disco de vidrio óptico en el que están dispuestos círculos concéntricos (pistas), cada pista tiene una secuencia alterna de sectores transparentes y sectores mate obtenidos por depósito de una película metálica. Se emite un haz de luz en correspondencia con cada pista que es interceptada por un fotodiodo o un fototransistor ubicado en el lado opuesto del disco. Mediante una disposición adecuada de los sectores transparentes y mate, es posible convertir un número finito de posiciones angulares en datos digitales correspondientes. El número de pistas determina la longitud de la palabra y, por lo tanto, la resolución del codificador (Siciliano et al. 2010, pp.210-211).

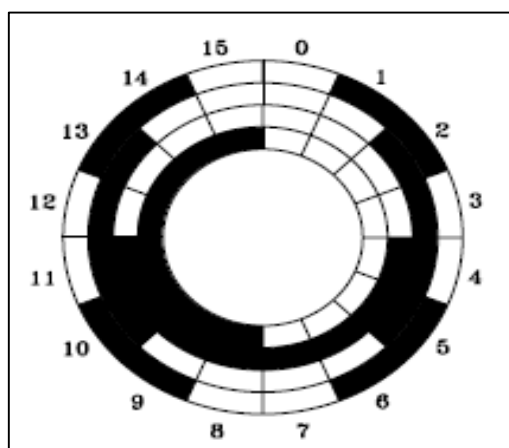


Figura 21-1: Encoder absoluto
Fuente: Siciliano B. 2010

1.7.4.2 Sensor medición inercial

Una unidad de medición inercial (IMU) es un dispositivo electrónico que mide la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giroscopos. El término IMU es usado ampliamente para referirse a una caja que contiene tres acelerómetros y tres giroscopos. Los acelerómetros están situados de tal forma que sus ejes de medición son ortogonales entre sí, estos miden la aceleración inercial, también conocida como fuerzas G. Los tres giroscopos están posicionados en un patrón ortogonal similar, midiendo la posición rotacional en referencia a un sistema de coordenadas seleccionada en forma arbitraria (Luengas et al.,2017, p.95).

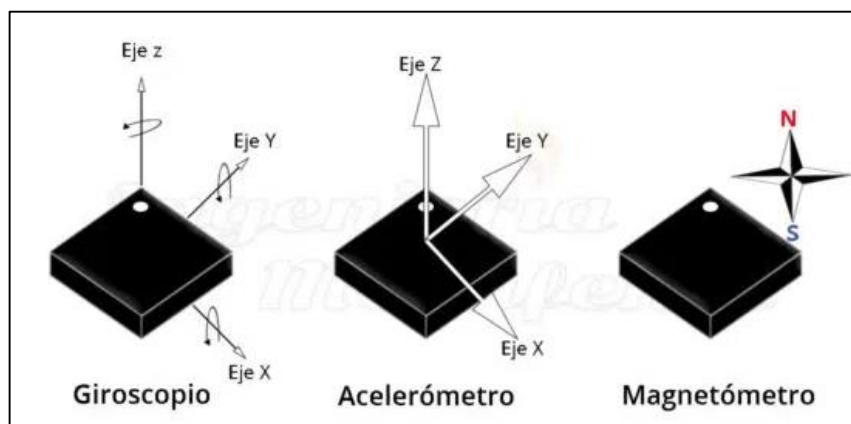


Figura 22-1: Esquema del sistema de medición inercial
Fuente: (Mecafenix, 2018)

1.8 Software

1.8.1 SolidWorks

El software CAD SolidWorks es una aplicación para realizar diseño mecánico de forma automatizada que facilita a los diseñadores croquizar ideas con rapidez, experimentar con operaciones y cotas, producir modelos y dibujos detallados (Frías, 2018, p.70).

Solid Works ocupa un método de diseño en 3D que implica diseñar una pieza desde el croquis inicial hasta el resultado final, a partir de este modelo se puede crear dibujos en 2D o componentes de relaciones de posición que tengan piezas o sub ensamblajes para crear ensamblajes en 3D, también se puede diseñar dibujos en 2D a partir de los ensambles en 3D (SolidWorks, 2015).

1.8.2 Arduino Ide

Es un software de desarrollo libre escrito en Java, es de código abierto estando disponible para ser instalado en cualquier sistema operativo, este software se usa en todas las placas de Arduino, contiene un editor de texto, un área de mensajes, una consola de texto, barra de herramientas y una serie de menús. El lenguaje de programación utilizado por Arduino es C y C++, preservando su sintaxis de declaración de variables, operadores, punteros y estructuras (Paredes, 2017, p.52).

Este software facilita la programación de los diversos sensores y dispositivos, también permite seleccionar el tipo de tarjeta Arduino, seleccionar librerías y ejemplos, compilación, facilitando el manejo de las características de cada dispositivo para Arduino, cuenta una comunicación serial USB 3.0 facilitando la comunicación entre hardware y software (González y Carrillo, 2019, p.33).

El entorno de programación de Arduino consta de un sketch está formado por dos funciones esenciales: `setup()` y `loop()`; la primera inicializa los requerimientos del sistema y solo es ejecutada una vez, la segunda es una ejecución infinita es decir un bucle donde contiene todas las funciones y procesos del sistema (Paredes, 2017, pp.52-53).

1.8.3 Matlab

Matlab (una abreviatura de MATrix LABoratory) es un sistema basado en el cálculo matricial para desarrollar aplicaciones matemáticas y de ingeniería. Matlab es una clase de lenguaje diseñado únicamente para realizar manipulaciones matriciales. Todas las variables que se manejen en Matlab son una matriz o un array rectangular de números, Matlab tiene un extenso conjunto de rutinas para obtener salidas gráficas (Ogata, 1998, p.1).

Específicamente en el área de ingeniería Matlab permite realizar simulaciones de sistemas mecatrónicas y robots manipuladores, el proceso de simulación resulta importante cuando no se dispone de una adecuada infraestructura experimental. Sin embargo, la simulación depende de un buen modelo matemático que refleje todos los fenómenos físicos del sistema real, la simulación es flexible ya que permite detectar posibles deficiencias en el diseño del modelado. Una vez encontrado el modelo dinámico la simulación facilita el proceso para analizar, estudiar y comprender el comportamiento de la dinámica del sistema siendo una etapa fundamental para el diseño de algoritmos de control. Cuando el modelo matemático del sistema mecatrónico o robot manipulador es lo suficientemente completo entonces la simulación proporciona un medio virtual del sistema real (Cortés, 2012, p.6).

1.9 Comunicación

1.9.1 Comunicación Serial

La comunicación serial es un protocolo estándar que envía y recibe bytes de información a un bit a la vez, uno detrás de otro en serie. Transmite datos en formato ASCII. En comparación a una comunicación en paralelo, que envía un byte completo es más lenta, pero más sencilla y puede alcanzar distancias mayores (Vásconez, 2018, p.25).

1.9.2 Comunicación Inalámbrica

Este tipo de comunicación no utiliza cables de interconexión sus equipos, en la actualidad se está viviendo lo que puede significar una revolución en el uso de las tecnologías de la información, las Redes Inalámbricas o Wireless Networks, se están introduciendo en el mercado de consumo gracias a unos precios populares y a un conjunto de entusiastas, mayoritariamente particulares, que han visto las enormes posibilidades de esta tecnología. La revolución de los ordenadores personales y el espectacular desarrollo de Internet están haciendo que la informática sea tan común en la vida diaria como lo es el teléfono. En este entorno, se puede nombrar algunas tecnologías de comunicación inalámbrica como GSM, UMTS, WiFi, Bluetooth, etc (Frías, 2018, pp.27-28).

1.9.3 Módulos radio frecuencia

En la industria actual, muchos de los sistemas de telemetría emplean RF ya que presenta ciertas características muy convenientes en ciertas aplicaciones, son dispositivos de precio accesible y se destacan por su facilidad para usarlos, sobre todo destacan en una de las características que otros medios de comunicación inalámbrica como bluetooth y wifi tienen limitaciones, el alcance de transmisión y recepción (Chimarro y Villagómez, 2014, p. 15).

CAPÍTULO II

2. PROPUESTA Y DISEÑO DEL PROTOTIPO

En el presente capítulo se describe las fases y el desarrollo metodológico de la propuesta de diseño y construcción de un robot móvil autónomo en configuración diferencial para la localización en interiores usando odometría. En la propuesta se define el diseño de un robot móvil autónomo, se utiliza la odometría como método de estimación de la posición del robot y finalmente se destaca que la propuesta se utiliza en la localización de interiores para diferentes trayectorias. En el presente trabajo de investigación se detalla la metodología científica como tecnológica que permiten cumplir con el ciclo de vida del proyecto de ingeniería y sus objetivos.

La primera fase contiene los requerimientos necesarios para el correcto desempeño del robot móvil, en la segunda fase se realiza el modelamiento matemático y control de estabilidad del robot móvil en configuración diferencial para que siga una trayectoria deseada, en la tercera fase se utiliza algoritmos de planificación de rutas en un mapa conocido de interiores localizando al robot en puntos deseados. Al cumplir las tres primeras fases se tienen los elementos necesarios para proceder a la cuarta fase en que se realiza la simulación del desplazamiento del robot móvil hacia un punto deseado dentro de un mapa conocido, todo este proceso de simulación se lo realiza utilizando el software Matlab. La quinta fase trata sobre el diseño y construcción tanto del sistema mecánica como electrónico del robot. En la sexta fase se desarrollará programas en Arduino Ide tanto para aplicar la odometría como para realiza un control PID de velocidad para los motores. Finalmente, en la séptima fase se hará la comunicación inalámbrica y bidireccional entre Matlab y Arduino con el fin de realizar el procesamiento del modelo matemático y control de estabilidad en Matlab mientras que la odometría y el control de velocidad de los motores serán ejecutados en Arduino.

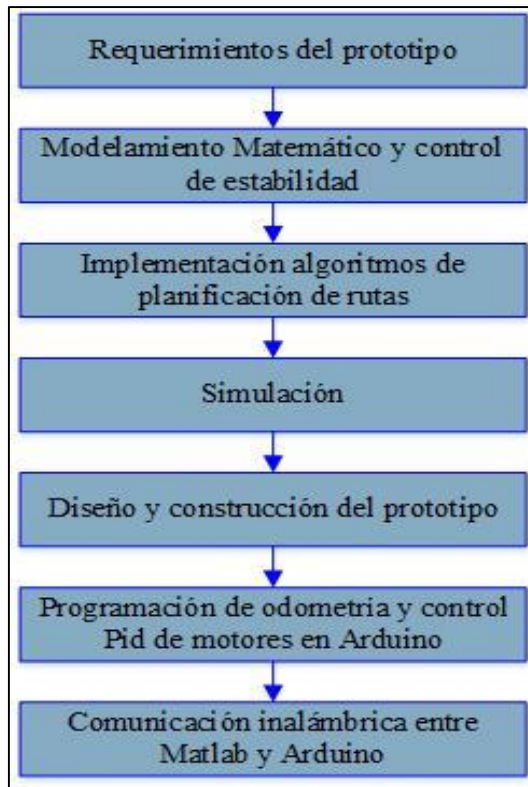


Figura 1-2: Fases de desarrollo del prototipo.

Realizado por: Pilamala, H; Hilaño, J, 2020

2.1 Metodología científica

La presente investigación tiene una metodología científica de carácter experimental y aplicada puesto que es realizada en un laboratorio, tomarse las muestras de velocidades y posición del robot, siendo estas comparadas con simulaciones por computadora del prototipo. Además, se manipula mediante computador los datos de entrada como lo son punto de inicio, punto final, velocidad del robot, trayectoria o caminos a seguir en interiores.

2.2 Investigación

Al ser investigación de tipo aplicada, la recopilación de la información acerca robots móviles y sus diferentes configuraciones conllevó a la determinación de una preferencia por la configuración diferencial en robots móviles tanto por su facilidad en el análisis matemático como en su implementación y por ende su funcionalidad y operatividad para el caso de estudio del presente proyecto.

De los resultados de la revisión bibliográfica de obtuvieron algunos métodos que permiten lograr el objetivo del proyecto, pero se escogió el uso de algoritmos planificadores de hojas de ruta en base a mapas de ocupación binaria para la localización en el plano.

2.2.1 Alcance de la investigación

El alcance de este trabajo de investigación se formula a nivel de robótica y control de sistemas que busca obtener que el robot se localice usando odometría, control PID, estabilidad de Lyapunov dentro un espacio específico de un mapa conocido, a través de diferentes rutas o trayectorias obtenidos por planificadores.

2.2.2 Limitaciones

Debido a que este proyecto de investigación es realizado bajo requerimientos que plantean un escenario conveniente para el desarrollo del mismo, el prototipo no consta con más sensores que puedan percibir su entorno para su estimación de posición, de la misma manera se procura que el escenario limite o reduzca el efecto de los errores sistemáticos que afecten a la precisión de la estimación de la posición en el plano del robot en interiores.

2.3 Requerimientos del prototipo

En el presente trabajo de investigación se han determinado algunos parámetros que se debe implementar para el desarrollo correcto del prototipo y el entorno en el que este trabaja, por motivo de simular un ambiente idóneo en el cual el robot pueda desempeñarse de forma correcta, disminuyendo así el efecto que producen errores sistemáticos y no sistemáticos que se presente experimento de localización en el plano dentro de un mapa conocido.

- El prototipo debe ser capaz de llegar a un punto deseado en el plano referencial al cual este sea colocado
- La superficie del entorno donde el robot se mueva debe ser lo más lisa posible para evitar perturbaciones del sistema y errores en la estimación de localización.
- Las llantas del robot deben ser adherentes en lo posible al suelo, evitando que estas patinen o resbalen ocasionando así que los errores en la estimación se incrementen.
- El plano en el que el robot se va a considerar obstáculos ya conocidos por el robot en su planificador de ruta hacia un punto deseado.
- Las velocidades a las que el robot se debe mover deben ser bajas debido a que no se considera un sistema de frenado para el robot.

2.4 Concepción del sistema

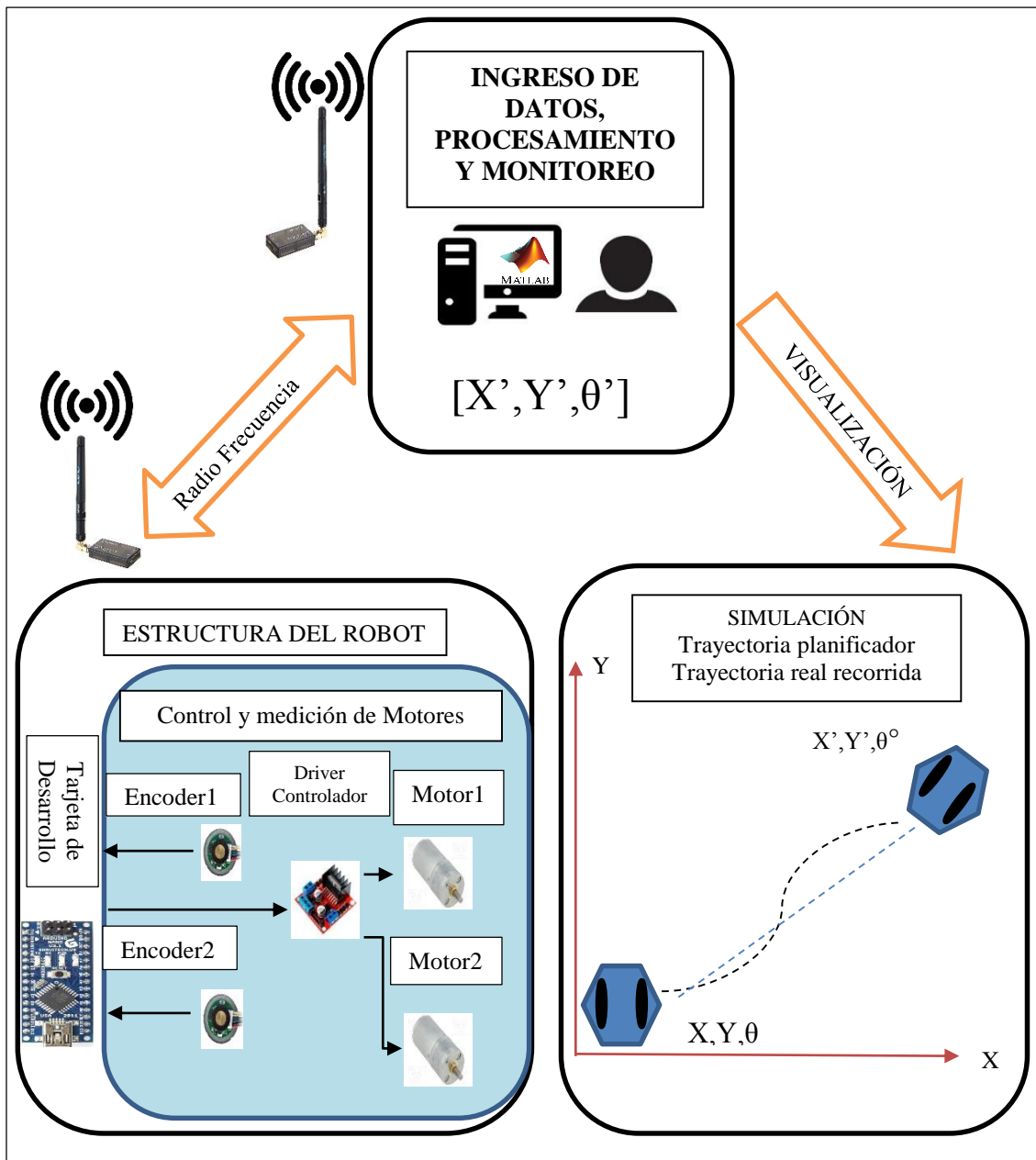


Figura 2-2: Concepción del sistema

Realizado por: Pilamala, H; Hilaño, J, 2020

2.5 Modelo Cinemático de un Robot móvil en configuración diferencial

El inicio del análisis de un sistema de control es su representación por un modelo matemático, este modelo se representa como un operador entre entradas y salidas del sistema, o como un conjunto de ecuaciones diferenciales (Varela, 2016, p.15).

2.5.1 Cinemática Directa

La figura 3-2 muestra la posición del robot móvil respecto a los ejes xy y el ángulo de rotación φ en un instante dado. La cinemática directa de un robot móvil toma como entrada las velocidad lineal y angular total para obtener como salida las derivadas de la posición y orientación tal como se describe en la ecuación 38.

J es la matriz Jacobiana la cual representa en un robot móvil la relación de las velocidades con el desplazamiento de la posición del robot en el espacio de tareas (Tzafestas, 2013,p. 33).

$$\dot{h} = J\dot{q} \quad (38)$$

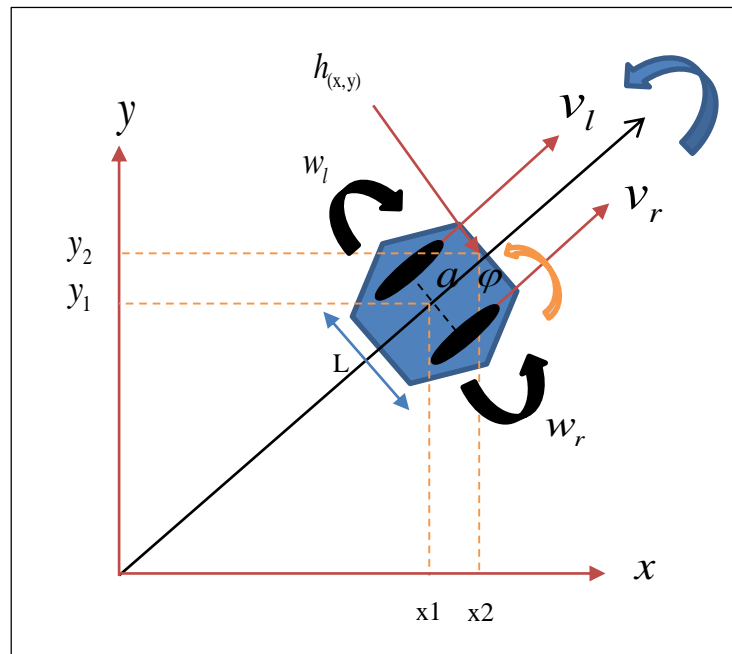


Figura 3-2: Geometría del robot móvil en configuración diferencial

Realizado por: Pilamala, H; Hilaño, J, 2020

El modelo cinemático se lo hará respecto al punto de control desplazado una distancia a del centro, este modelo es el más comúnmente aplicado porque siempre se sitúa alguna herramienta o efector final en la estructura del robot móvil a una distancia del centro.

La posición se obtiene analizando la figura 3-2 mediante razones trigonométricas simples, una vez obtenida la posición se aplica la derivada generando velocidades en los ejes xy y una velocidad angular respecto a la variación del ángulo de rotación como se muestra en las siguientes ecuaciones.

$$hx = x_1 + x_2 \quad (39)$$

$$hy = y_1 + y_2 \quad (40)$$

$$hx = x_1 + a \cos\varphi \quad (41)$$

$$hy = y_1 + a \sin\varphi \quad (42)$$

$$\dot{hx} = v \cos\varphi - a\omega \sin\varphi \quad (43)$$

$$\dot{hy} = v \sin\varphi + a\omega \cos\varphi \quad (44)$$

$$\dot{\varphi} = \omega \quad (45)$$

Estas ecuaciones son representadas en la siguiente expresión matricial siendo este el modelo cinemático directo.

$$\begin{bmatrix} \dot{hx} \\ \dot{hy} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -a \sin\varphi \\ \sin\varphi & a \cos\varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (46)$$

2.5.2 Cinemática Inversa

La cinemática inversa de un robot móvil toma como entradas las derivadas de posición y orientación para obtener como salida las velocidades lineal y angular que debe tener el robot para realizar una trayectoria deseada.

Para obtener la cinemática inversa del robot móvil se aplica la inversa de la matriz Jacobiana de la siguiente forma.

$$\dot{q} = J^{-1}\dot{h} \quad (47)$$

$$J = \begin{bmatrix} \cos\varphi & -a \sin\varphi \\ \sin\varphi & a \cos\varphi \\ 0 & 1 \end{bmatrix} \quad (48)$$

Al ser la matriz Jacobiana no cuadrada la solución es encontrar la matriz pseudoinversa, esta matriz se define como:

$$J^{-1} = (J^T * J)^{-1} * J^T \quad (49)$$

$$J^{-1} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -a \sin\varphi & a \cos\varphi & 1 \\ \frac{1}{a^2 + 1} & \frac{1}{a^2 + 1} & \frac{1}{a^2 + 1} \end{bmatrix} \quad (50)$$

Una vez definida la matriz Jacobiana inversa, el modelo cinemático inverso se presenta de la siguiente forma.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\varphi & \operatorname{sen}\varphi & 0 \\ -a \operatorname{sen}\varphi & a \cos\varphi & 1 \\ \frac{1}{a^2 + 1} & \frac{1}{a^2 + 1} & \frac{1}{a^2 + 1} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \quad (51)$$

Para obtener la velocidad angular que debe tener cada motor en función las velocidades totales calculadas en el modelo cinemático inverso se debe analizar las ecuaciones 22 y 23 con el fin de obtener las siguientes ecuaciones.

$$\omega_r = \frac{1}{R} \left(v + \frac{\omega L}{2} \right) \quad (52)$$

$$\omega_l = \frac{1}{R} \left(v - \frac{\omega L}{2} \right) \quad (53)$$

R : Radio de la rueda

L : Distancia entre las dos ruedas

v : Velocidad lineal total

ω : Velocidad angular total

v_r : Velocidad lineal llanta derecha.

v_l : Velocidad lineal llanta izquierda.

ω_r : Velocidad angular llanta derecha

ω_l : Velocidad angular llanta izquierda

2.6 Control de seguimiento de trayectoria

El control de seguimiento de trayectoria en robots consiste en conseguir que el robot siga como referencia una curva parametrizada en el tiempo. El objetivo de este tipo control es conseguir que el robot alcance y siga con un error pequeño los estados deseados variantes con el tiempo.

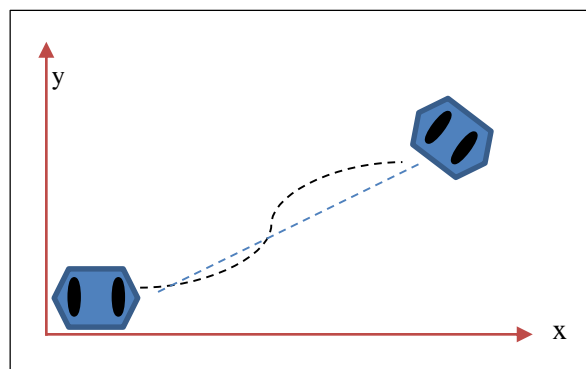


Figura 4-2: Trayectoria del robot móvil

Realizado por: Pilamala, H; Hilaño, J, 2020

El robot móvil deberá seguir una trayectoria h al punto de interés en términos de \dot{h} y sus condiciones iniciales de posición y orientación, mediante la cinemática inversa se determina el conjunto de velocidades \dot{q} que reproduzca dicha trayectoria. Estas velocidades serán ejecutadas en un sistema robótico real por lo que siempre se debe tener en cuenta el error de la posición entre la trayectoria deseada y la real (Siciliano et al., 2010, p.132).

$$h_e = h_d - h \quad (54)$$

Derivando el error tenemos

$$\dot{h}_e = \dot{h}_d - \dot{h} \quad (55)$$

$$\dot{h} = \dot{h}_d - \dot{h}_e \quad (56)$$

Reemplazando la ecuación (56) en la ecuación (47) obtenemos la cinemática inversa tomando en cuenta el error de posición y orientación.

$$\dot{q} = J^{-1}(\dot{h}_d - \dot{h}_e) \quad (57)$$

El análisis de estabilidad del robot móvil se lo hará mediante la estabilidad de Lyapunov, se comienza asumiendo el seguimiento perfecto de velocidad.

$$\dot{q}_{ref} = \dot{q} \quad (58)$$

Se considera la Función candidata de Lyapunov mediante el modelo de los errores cuadráticos descrita en la ecuación 59 (Varela, 2016, p.68). A partir de esta función se aplica las condiciones para la estabilidad asintótica de Lyapunov descrita en las ecuaciones 33, 34, 35, 36.

$$V(h_e) = \frac{h_e^T h_e}{2} \quad (59)$$

$$\dot{V} = h_e^T \dot{h}_e \quad (60)$$

$$\dot{h}_e = -K h_e \quad (61)$$

$$\dot{V} = -h_e^T K h_e < 0 \quad (62)$$

K es una matriz definida positiva generalmente diagonal.

Reemplazando la ecuación (61) en la ecuación (57) se obtiene el modelo cinemático inverso basado en Lyapunov para que el robot siga una trayectoria deseada.

$$\dot{q}_{ref} = J^{-1}(\dot{h}_d + K h_e) \quad (63)$$

Para que el robot se mueva a una velocidad constante v_{max} se analiza figura 5-2 cuando el robot se aproxima a un punto p_d de una trayectoria, las ecuaciones que rigen el movimiento del robot en este punto son las siguientes.

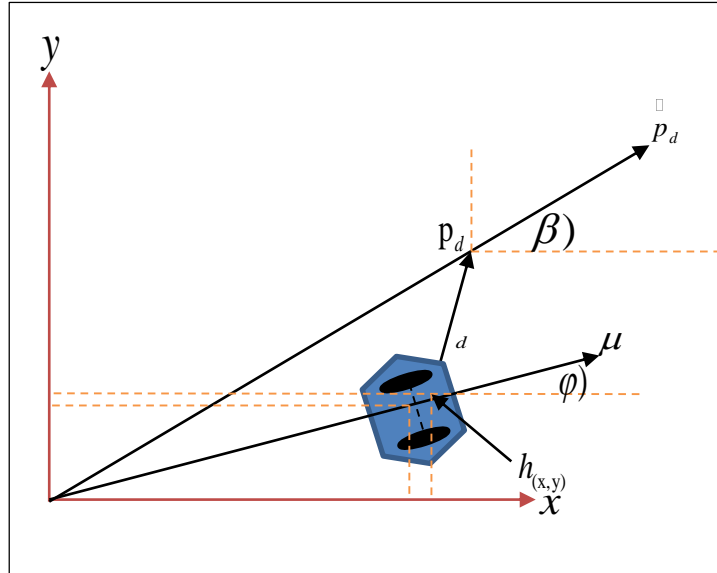


Figura 5-2: Localización del robot en una trayectoria

Realizado por: Pilamala, H; Hilaño, J, 2020

$$\dot{q} = J^{-1}(\dot{p}_d + Kh_e) \quad (64)$$

$$\dot{p}_x = |\dot{p}_d| \cos \beta \quad (65)$$

$$\dot{p}_y = |\dot{p}_d| \sen \beta \quad (66)$$

$$\beta = \tan^{-1} \frac{\Delta p_{yd}}{\Delta p_{xd}} \quad (67)$$

$$|\dot{p}_d| = v_{max} \quad (68)$$

Para obtener el modelo cinemático inverso del robot basado en Lyapunov para seguimiento de trayectoria a una velocidad constante se considera la ecuación 63 con un cambio de velocidad \dot{h}_d por \dot{p}_d puesto que el punto de interés p_d es el que se encuentra en la trayectoria.

$$\dot{q} = J^{-1}(\dot{p}_d + Kh_e) \quad (69)$$

2.7 Planificadores de hoja de ruta para mapas conocidos

2.7.1 Generación de mapas

El mapa en el cual el prototipo se moviliza presenta ciertos obstáculos los cual se deberá evitar para llegar al punto de consigna definido por el usuario. El mapa se lo puede realizar de acuerdo

con los requerimientos del entorno en el cual el robot debe desplazarse. Se debe tomar en cuenta que la escala de la imagen del mapa que se quiere realizar sea la correcta con respecto al entorno original, para esto se puede utilizar herramientas CAD o generar una imagen con obstáculos al azar para realizar las pruebas respectivas.

El mapa 2D necesita previamente ser binarizado para establecer en una matriz las zonas ocupadas del mapa y designarla con valores de “1” en caso de que la zona este ocupada y valores de “0” en caso de que este vacío. El mapa generado puede ser un archivo binario, lógico o .mat para generar un mapa de ocupación.

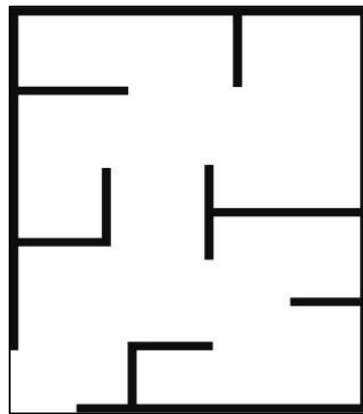


Figura 6-2: Mapa con obstáculos

Realizado por: Pilamala, H; Hilaño, J, 2020

Para la definición del mapa de ocupación binaria se hace uso del toolbox de Matlab “Robotics System Toolbox™”. Con los comandos de mapa de ocupación del toolbox se puede generar de acuerdo al tamaño real del entorno una resolución que otorgue los valores en metro reales (Mathworks, 2015).

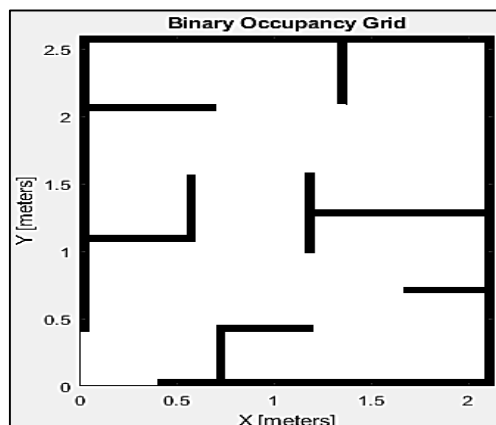


Figura 7-2: Mapa de ocupación binaria en metros

Realizado por: Pilamala, H; Hilaño, J, 2020

El método de realizar la asignación de un mapa de ocupación dentro del proceso del programa se lo puede ver en el siguiente diagrama de flujo del gráfico 1-2.

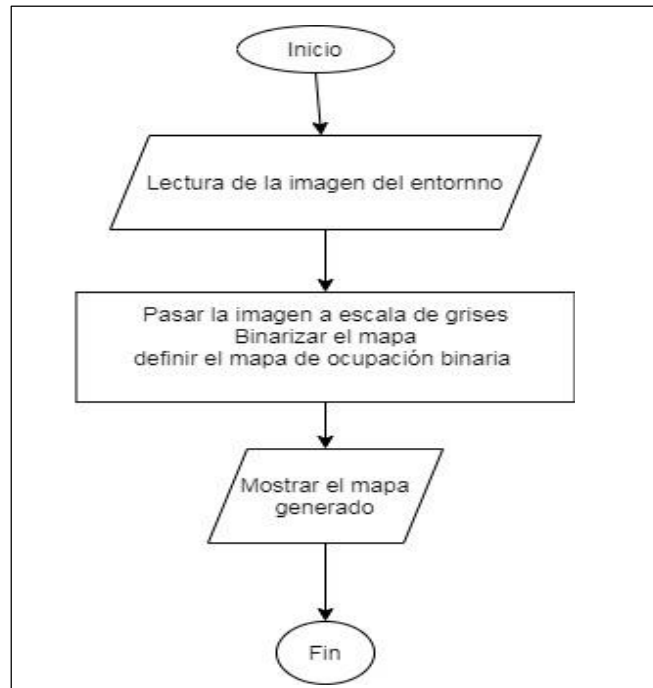


Gráfico 1-2. Algoritmo de mapa de ocupación

Realizado por: Pilamala, H; Hilaño, J, 2020

2.7.2 *Uso del planificador probabilístico PRM*

Obtenido el mapa se usa un algoritmo probabilístico de planificación de rutas basado en mapas de ocupación (MathWorks,2020). El algoritmo genera un número especificado de puntos al azar en los espacios libres del mapa y los conecta generando varias rutas que unen a todos los puntos. Un número mayor de puntos permite obtener una ruta más eficiente que un número menor de puntos que generará una ruta más simple y poco convencional como se muestra en la figura 8-2.

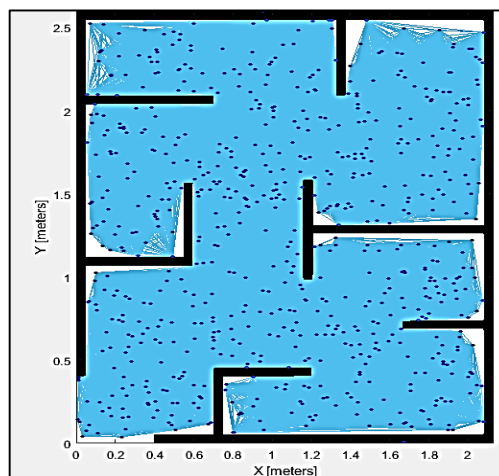


Figura 8-2: Rutas posibles en puntos al azar

Realizado por: Pilamala, H; Hilaño, J, 2020

Se indica un punto de inicio y un punto final de llegada, indiferentemente de que si uno de estos se encuentre en un punto generado al azar. El algoritmo busca unir ambas coordenadas dadas con los puntos más cercanos y a la vez hace una búsqueda exhaustiva de la mejor ruta para unir los puntos de partida y llegada como se muestra en la figura 9-2.

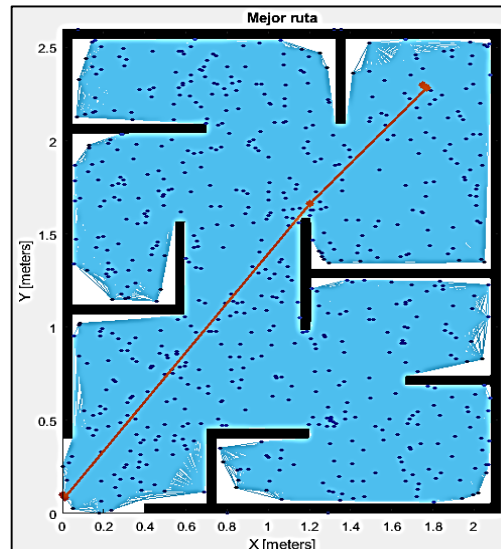


Figura 9-2: Trayectoria de finida por el algoritmo

Realizado por: Pilamala, H; Hilaño, J, 2020

Obtenida la trayectoria, esta se guarda en un vector de puntos de la red probabilística que fueron unidos por el algoritmo. Los puntos que participaron en la trayectoria son coordenadas en el plano (x,y) que serán enviadas al robot para que este se desplace por la misma como se muestra en la figura 10-2.

	1	2	
1	0	0.1000	
2	0.0130	0.0875	
3	1.2024	1.6617	
4	1.7692	2.2827	
5	1.7500	2.3000	
6			

Figura 10-2: Vector con las coordenadas de la ruta

Realizado por: Pilamala, H; Hilaño, J, 2020

Para evitar que el robot colisiones con los obstáculos del mapa se utiliza una técnica fácil que consiste en sobre dimensionar o “inflar” a los obstáculos con un radio especificado en el mapa para considerar el diámetro que tiene el robot y no solo el punto de control al que se requiere localizar en el mapa como se muestra en la figura 11-2.

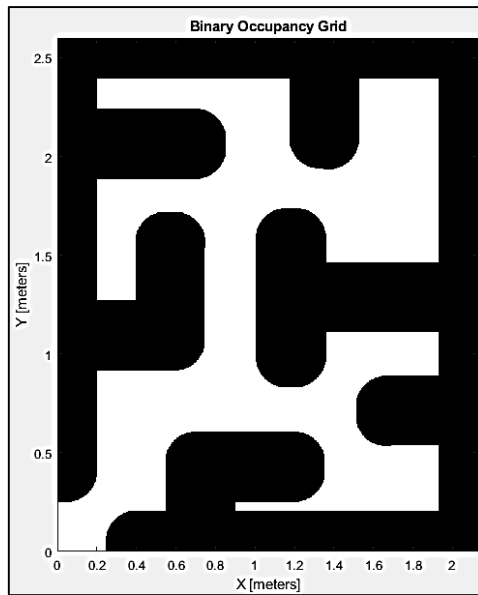


Figura 11-2: Mapa Inflado

Realizado por: Pilamala, H; Hilaño, J, 2020

Entonces se procede a recalcular la ruta con el algoritmo PRM, la cual cambiara de acuerdo a los espacios disponibles del nuevo mapa. En la figura 12-2 se observa el mapa original con respecto a su versión inflada.

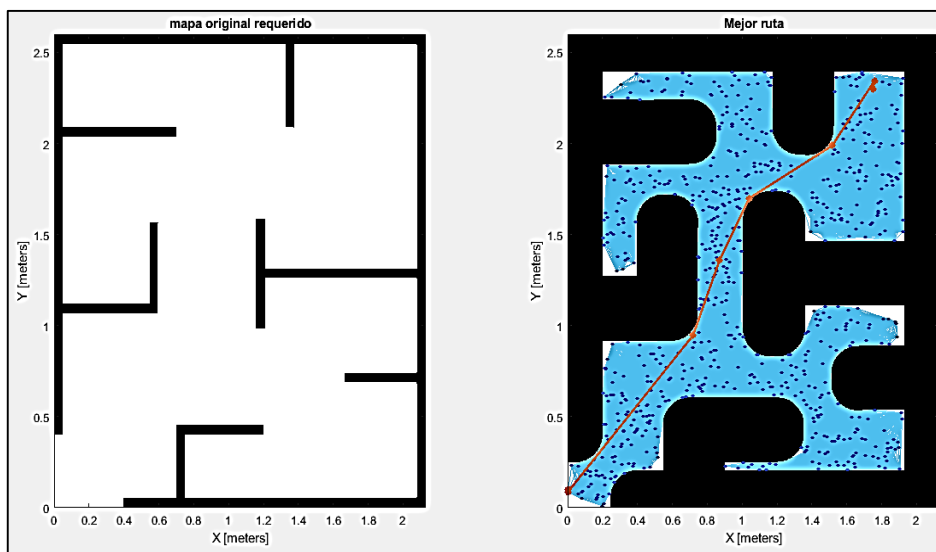


Figura 12-2: Ruta nueva en el mapa inflado.

Realizado por: Pilamala, H; Hilaño, J, 2020

Para mejorar la ruta, o aumentar más puntos dentro de la ruta trazada por el algoritmo planificador se modifica una propiedad de distancia de conexión del planificador que permite ubicar puntos en la ruta a una distancia específica una de la otra, mejorando la eficiencia de la ruta (figura 13-2). El procedimiento del uso de este algoritmo mediante el software Matlab 2020 se lo muestra en el gráfico 2-2.

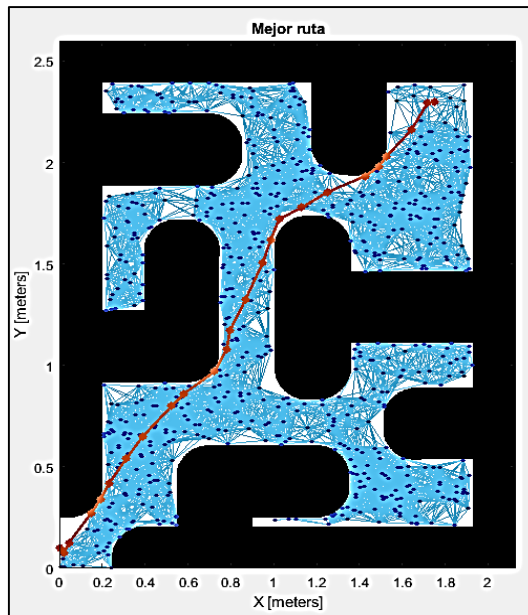


Figura 13-2: Ruta nueva con distancia de conexión modificada.

Realizado por: Pilamala, H; Hilaño, J, 2020

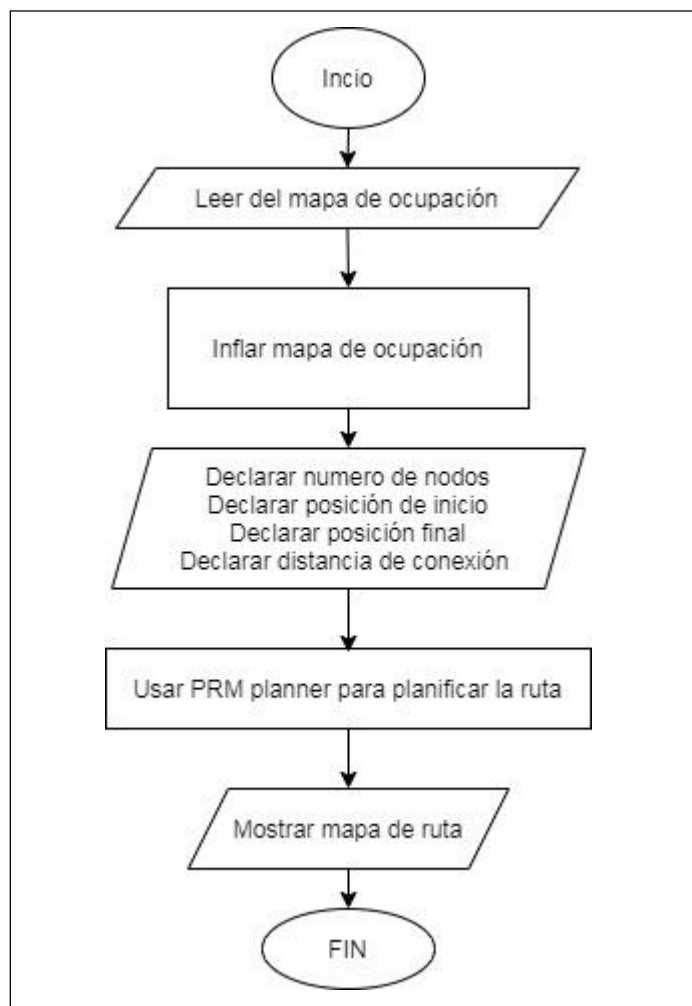


Gráfico 2-2. Algoritmo de uso del planificador de rutas PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

2.7.3 Uso del planificador A* híbrido

De la misma manera que el algoritmo PRM es necesario empezar desde la definición del mapa, el algoritmo de A* híbrido trabaja a partir de crear un objeto “stateValidator” validador de estado de un mapa previamente binarizado, este objeto genera una propiedad “ValidationDistance” que permite validar los estados a lo largo de la ruta. El planificador A* híbrido busca obstáculos en el mapa de ocupación interpolando las primitivas de movimiento y la expansión analítica según la propiedad ValidationDistance (MathWorks, 2020).

Como datos de inicio para iniciar el cálculo de la ruta con el algoritmo A* híbrido se indican las coordenadas X, Y y ϕ inicial y final de acuerdo al mapa propuesto. Además, se indica el radio de giro del robot y las distancias de movimiento de la primitivas que permitirá trazar semicircunferencias (por defecto cinco), generadas hacia adelante y hacia atrás con nodos finales que trazan las posibles rutas, el diagrama del proceso se muestra en el gráfico 3-2, al final se selecciona la ruta más corta como se muestra en la figura 14-2.

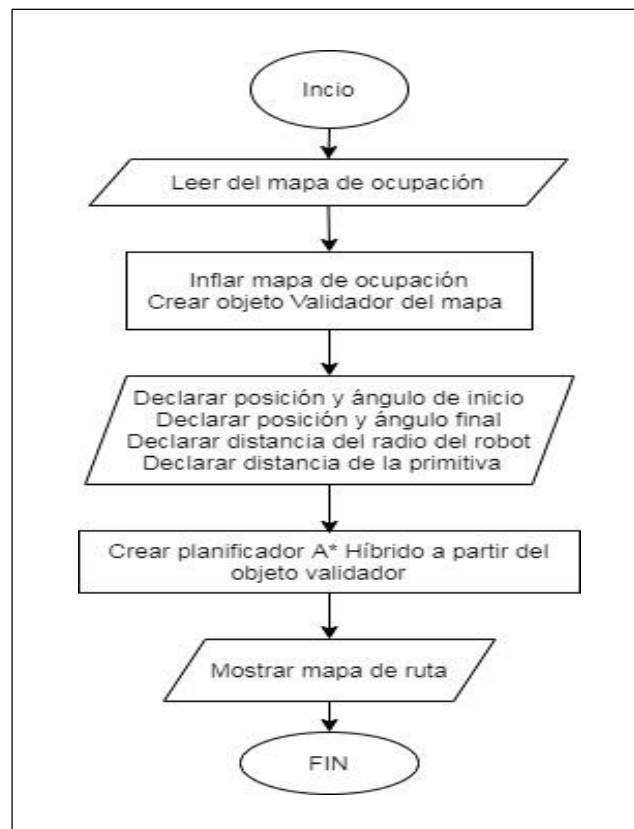


Gráfico 3-2. Algoritmo del planificador de rutas A* híbrido.

Realizado por: Pilamala, H; Hilaño, J, 2020

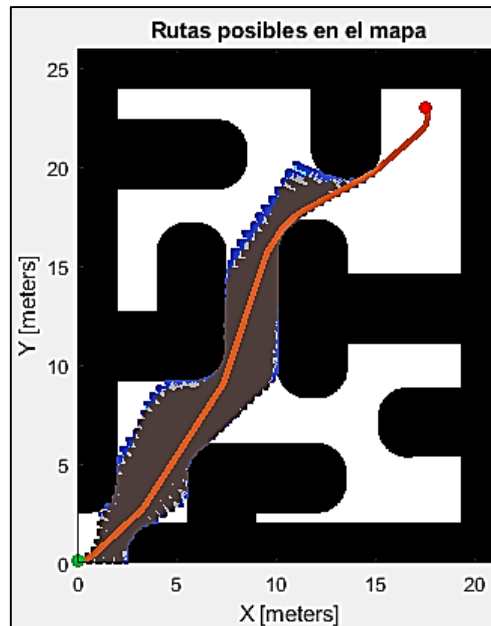


Figura 14-2: Ruta con planificador A* híbrido.

Realizado por: Pilamala, H; Hilaño, J, 2020

2.8 Simulación del sistema usando Matlab

2.8.1 Importación del diseño 3D de robot móvil a Matlab

El diseño 3D realizado en SolidWorks es usado en la simulación del sistema, el proceso de exportación hacia MATLAB implica obtener los archivos de ensamble en formato. stl. al ser un ensamble es necesario obtener los archivos. stl de cada parte.

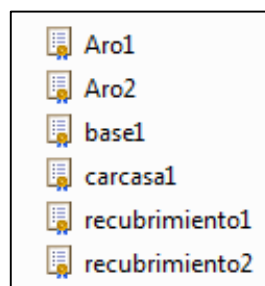


Figura 15-2: Archivos STL.

Realizado por: Pilamala, H; Hilaño, J, 2020

Es necesario definir previamente el sistema métrico del diseño y que su sistema de coordenadas sea centrado en el punto de control del diseño.

2.8.2 Desarrollo de funciones para simular movimiento del diseño 3D

En Matlab se crea un archivo .mat para guardar cada parte .stl en Matlab se crea dos funciones, la primera para guardar los parámetros de vértice y apariencia de cada archivo, la segunda función se encarga de graficar el diseño de acuerdo con una matriz de rotación en el eje Z (68).

$$R_z = \begin{pmatrix} \cos(\varphi) & -\text{sen}(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (70)$$

La función encargada de guardar los parámetros del robot (figura 16-2), carga previamente los archivos .stl almacenados en el contenedor. Mat.

```
function diferencialParametros
load 'robotPatch.mat' aroi aroii base carcasa recubrimientoii recubrimientoii
global Uniciclo;

%*****
Uniciclo.baseVertices=base.vertices';
Uniciclo.baseFaces=base.faces;
```

Figura 16-2: Función de parámetros de las partes .stl.

Realizado por: Pilamala, H; Hilaño, J, 2020

En la segunda función cada vértice de las figuras es multiplicado con la matriz de rotación y sus componentes en x, y, z son escaladas y se suma una variable de movimiento en cada eje para su actualización en la simulación (figura 17-2).

```
function Mobile_Graph=diferencialPlot(dx,dy,angz,scale)
global Uniciclo;

% Matriz de rotación z

Rz=[ cos(angz) -sin(angz) 0; sin(angz) cos(angz) 0; 0 0 1];

robotPatch = Rz*Uniciclo.baseVertices;
robotPatch(1,:)=robotPatch(1,:)*scale+dx+0.2;
robotPatch(2,:)=robotPatch(2,:)*scale+dy-0.4;
robotPatch(3,:)=robotPatch(3,:)*scale;

Mobile_Graph(1) = patch('Faces',Uniciclo.baseFaces,'Vertices',robotPatch,'FaceColor',[0,0.3,0.4],'EdgeColor','none')
```

Figura 17-2: Función de graficadora de las partes .stl.

Realizado por: Pilamala, H; Hilaño, J, 2020

El modelo del prototipo real se lo utiliza en la simulación para recorrer las trayectorias trazadas para el mapa seleccionado, el modelo del gráfico 4-2

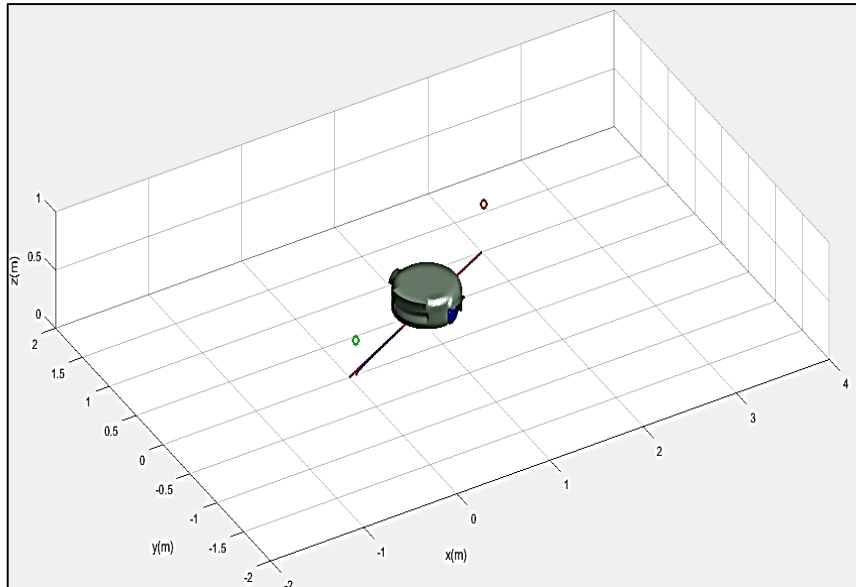


Gráfico 4-2. Modelo simulado en Matlab.

Realizado por: Pilamala, H; Hilaño, J, 2020

2.8.3 Simulación de seguimiento de trayectoria

La simulación comienza al cargar el archivo del mapa de un interior, este mapa se lo binariza para generar un mapa de ocupación dando espacios libres valores de 0 y espacios ocupados valores de 1 y se lo escala para obtener la medida del mapa real. A continuación, se ingresa los datos de localización inicial y la localización final a la que debe llegar el robot, con estos datos se aplican los algoritmos planificadores ya sean el algoritmo PRM o algoritmo A* híbrido, como resultado de estos planificadores obtenemos una trayectoria de referencia expresa en valores de posición y orientación x, y, φ que debe seguir el robot para llegar al punto final deseado, para comenzar el ciclo de simulación se ingresa los datos de tiempo de simulación y distancia del punto desplazado del robot, con estos datos se inicia el ciclo de simulación comenzando por obtener los errores entre las posiciones de referencia y las posiciones calculadas, estos datos de error son necesarios para que entren en el control de seguimiento de trayectoria con el fin de obtener las velocidad lineal y angular total del robot que a continuación pasaran a las ecuaciones 15,16,17 para estimar la posición y orientación del robot a lo largo de su trayectoria. Estos valores de posición y orientación entran a un ciclo para generar movimiento de un robot móvil mediante la función `diferencialPlot` que permite visualizar este movimiento en 3D.

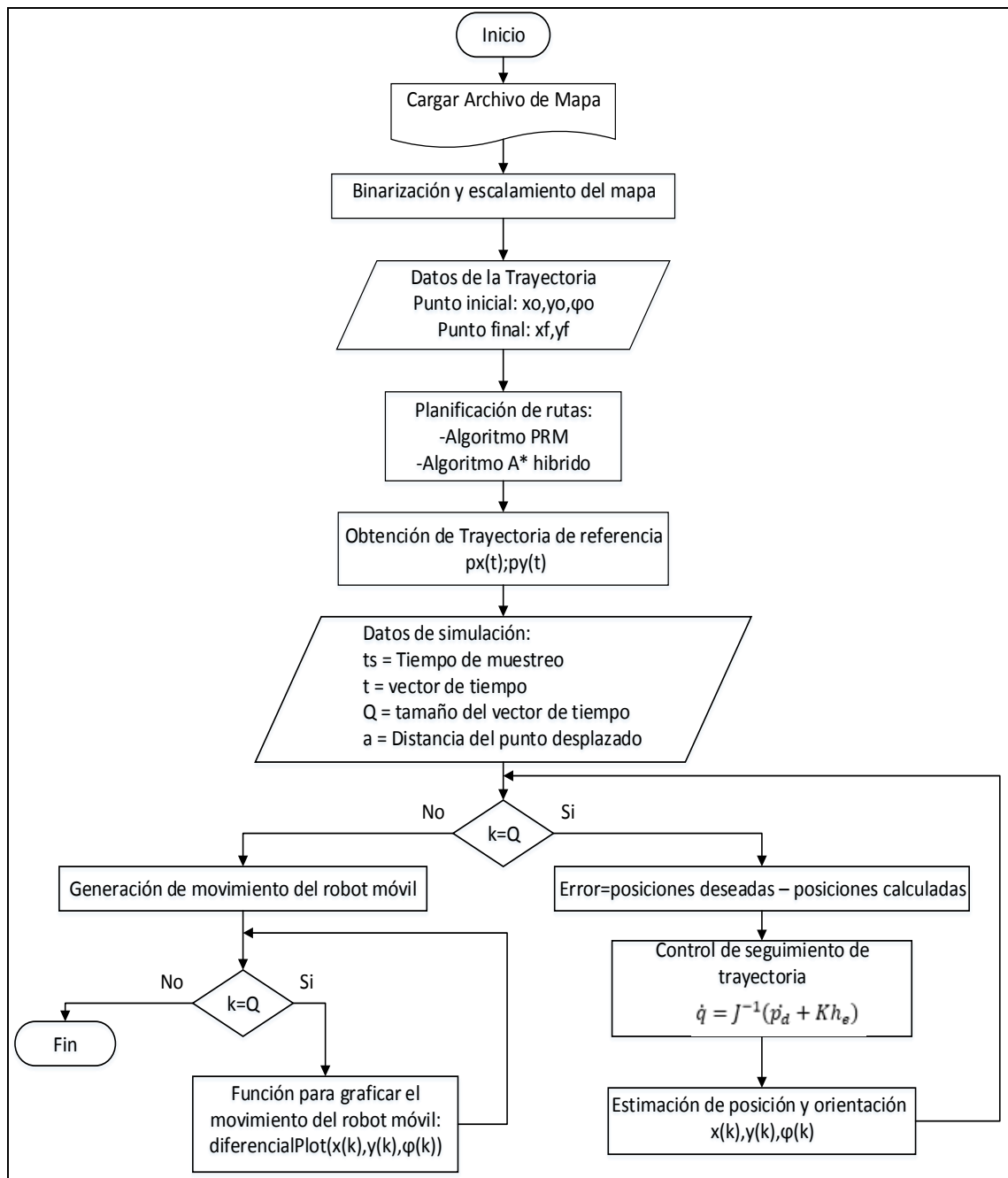


Gráfico 5-2. Diagrama de flujo de simulación

Realizado por: Pilamala, H; Hilaño, J, 2020

En las figuras 18-2 y 19-2 se muestran la generación de una trayectoria mediante el Algoritmo de planificación de ruta PRM y el algoritmo A* híbrido respectivamente.

Localización inicial: $x= 0.1 \text{ m}$ $y=0.1\text{m}$ $\varphi=90^\circ$

Localización final: $x= 1.75 \text{ m}$ $y=2.3 \text{ m}$

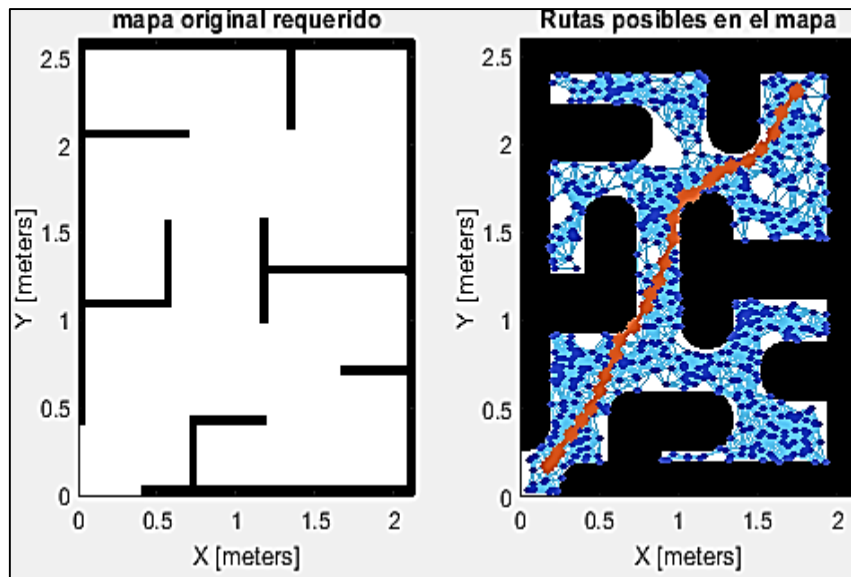


Figura 18-2: Trayectoria generada mediante el Algoritmo PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

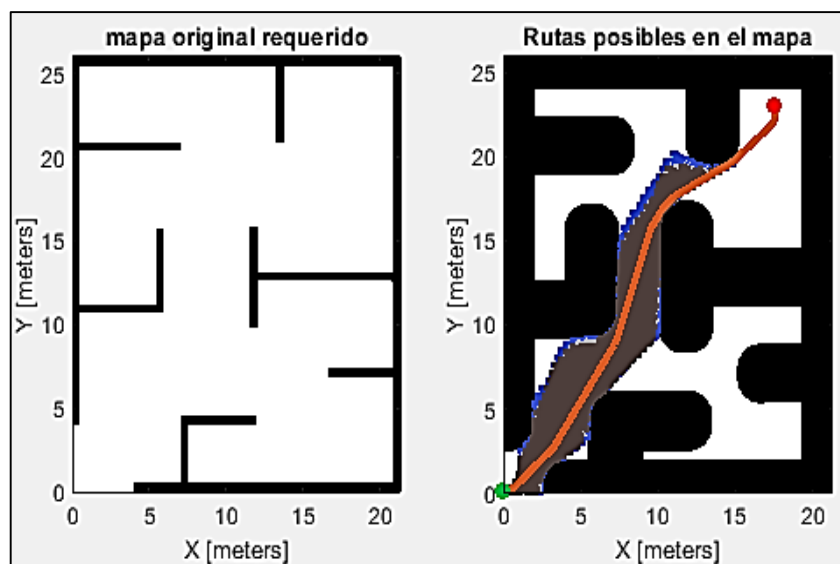


Figura 19-2: Trayectoria generada mediante el Algoritmo A* híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

En los gráficos 6-2 y 7-2 se muestra la simulación del desplazamiento del robot para seguir una trayectoria mediante el Algoritmo PRM y el algoritmo A* híbrido respectivamente. La línea roja muestra la trayectoria deseada y la línea azul la trayectoria recorrida por el robot.

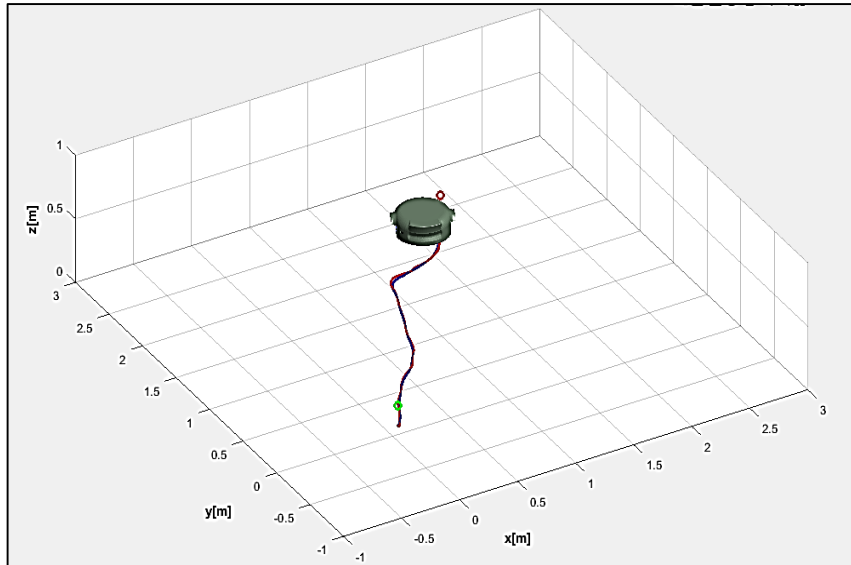


Gráfico 6-2. Desplazamiento del robot al usar el Algoritmo PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

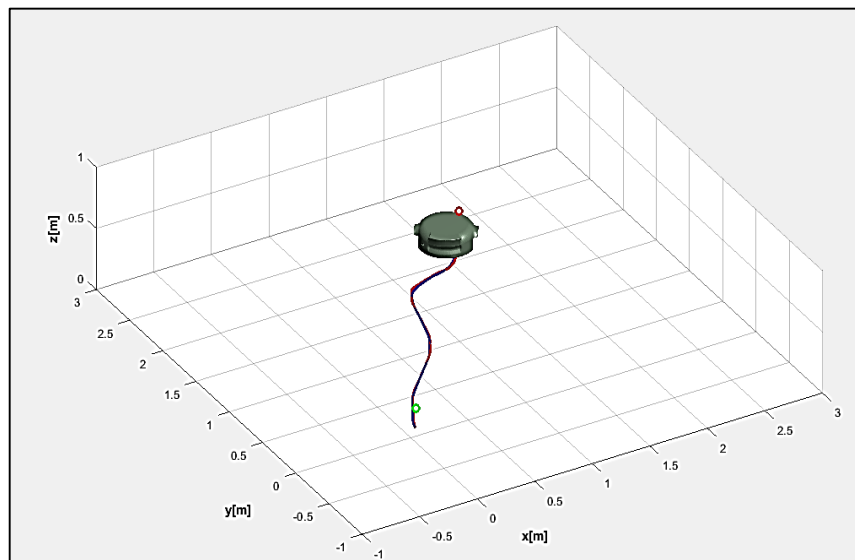


Gráfico 7-2. Desplazamiento del robot al usar el Algoritmo A* híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

2.9 Diseño del prototipo

2.9.1 Estructura mecánica

El diseño del robot móvil en configuración diferencial del presente trabajo de investigación se lo realizó de forma que no presente perturbaciones al momento de ejecutar las tareas en su plano referencial. El diseño del robot y la disposición de los elementos se lo realizó en el software SolidWorks.

Se ha definido una forma circular para el diseño del robot, tomando en cuenta que esta forma le facilitará en caso de rozamiento para no estancarse, las dimensiones que se han definido para la base del robot son las que se muestra en la figura 20-2.

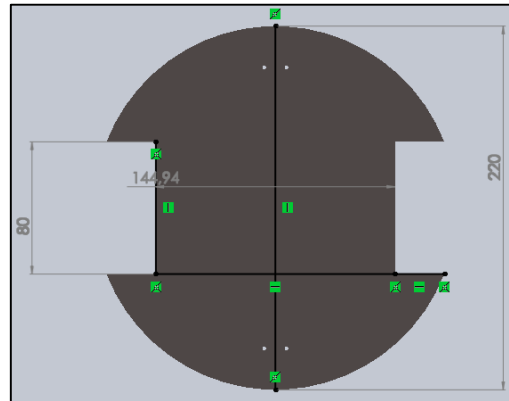


Figura 20-2: Dimensiones de la base del robot móvil

Realizado por: Pilamala, H; Hilaño, J, 2020

La base del robo tiene dos soportes metálicos que sujetan a los motores del robot, estos soportes son colocados de tal manera que la distancia y posición entre las llantas sea simétrica. Se añade una separación para la ubicación de las baterías como se muestra en la figura 21-2.

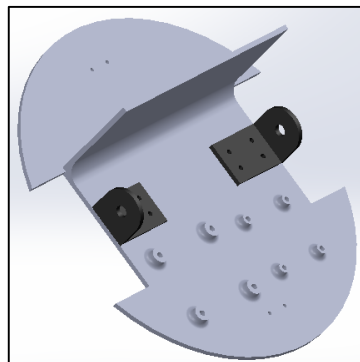


Figura 21-2: Modelo de la base del robot móvil.

Realizado por: Pilamala, H; Hilaño, J, 2020

La disposición de los elementos del robot queda de acuerdo a los requerimientos planteados para el sistema, como se muestra en la figura 22-2.

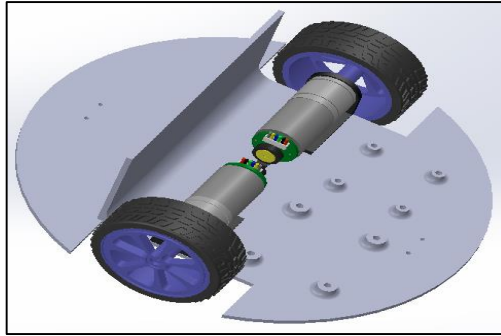


Figura 22-2: Disposición de motores en la base del robot móvil.

Realizado por: Pilamala, H; Hilaño, J, 2020

Para que el robot tenga soporte en la parte delantera y trasera se han añadido dos ruedas locas para evitar tambaleos y que el desplazamiento se facilite tal como lo muestra la figura 23-2.

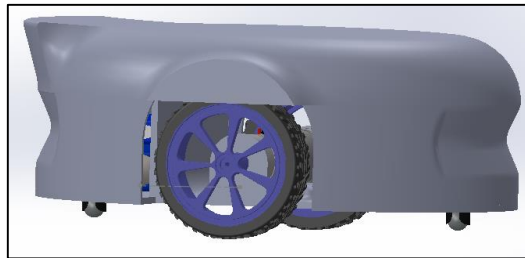


Figura 23-2: Ruedas locas para soporte.

Realizado por: Pilamala, H; Hilaño, J, 2020

La vista general del prototipo queda como lo indica la figura 24-2 donde podemos observar el diseño 3D propuesto para el prototipo de robot móvil en configuración diferencial.

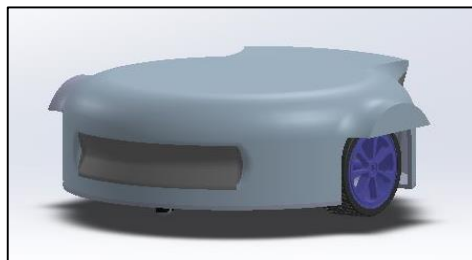


Figura 24-2: Diseño general del prototipo

Realizado por: Pilamala, H; Hilaño, J, 2020

El diseño mecánico también permite que la disposición de los elementos no desestabilice al robot, esto haciendo una correcta distribución del peso total del prototipo.

2.10 Sistema electrónico

2.10.1 Motores y encoders

Los motores seleccionados para el prototipo son motorreductores metálicos con una reducción de 1/45, se ha considerado una velocidad moderada de 130 RPM que serán controlados con señales PWM, la ventaja de estos motores es su Encoder de cuadratura integrado, el mismo usa sensores de efecto hall para detectar los pulsos por revolución (Amazon,2020).

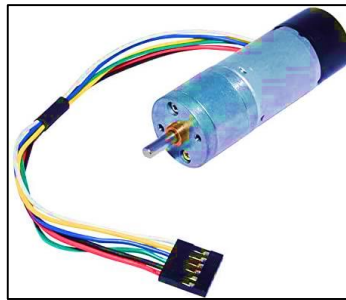


Figura 25-2: Motor dc con encoder integrado

Realizado por: Pilamala, H; Hilaño, J, 2020

Tabla 1-2: Datos técnicos del motor

Característica	Valor	Unidades
Alimentación motor	6-12	[v]
Velocidad sin carga	130	RPM
Corriente	0.15	A
Encoder	-	Cuadratura
Alimentación de encoder	5	[v]
Torque con carga nominal	2.6	lbs
Relación de engranajes	1:45	
PPR (Pulsos por revolución)	12	

Fuente: Amazon, 2020

Realizado por: Pilamala, H; Hilaño, J, 2020

2.10.2 Driver para motor

El módulo dual driver L298N, puente H, permite controlar 2 motores de corriente directa, mediante la técnica PWM. Cuenta con entradas digitales para el control de la señal PWM y de la dirección de giro de ambos motores por medio de un microcontrolador.

El módulo hace uso del circuito integrado L298N, y cuenta con un disipador, el cual le permite alcanzar una capacidad de corriente total de salida de hasta 4 amperes. Puede manejar tanto motores de mediana capacidad como de alto par (Torque), a voltajes típicamente entre 6 y 24 volts (Geek, 2013).

Este módulo puede abastecer a los motores elegidos, su rango de tensión y corriente satisface las necesidades del prototipo incluso en el peor de los casos con demanda máxima de corriente.

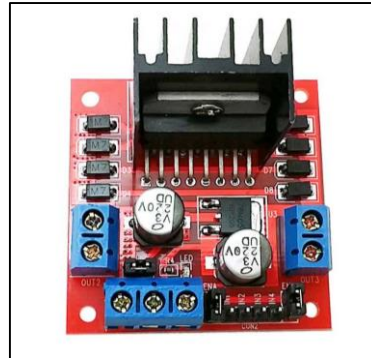


Figura 26-2: Driver L298N

Realizado por: Pilamala, H; Hilaño, J, 2020

Tabla 2-2: Datos técnicos driver L298N

Característica	Valor	Unidades
Circuito de control	L298N	
Motores a controlar	2	
Control de giro	Puente H doble	
Alimentación circuito lógico	5	[v]
Corriente máx. en circuito lógico	36	[mA]
Voltaje PWM motores	5-35	[v]
Corriente máx. para motores	4	[A]
Frecuencia de conmutación	10-25	khz
Disipación del módulo	25	W

Fuente: Geek Factory, 2013




Realizado por: Pilamala Henry & Hilaño Jorge, 2020

2.10.3 Arduino Nano

La tarjeta de desarrollo elegida para controlar el sistema fue un Arduino Nano porque tiene memoria y capacidad para realizar interrupciones digitales a altas frecuencias, además sus dimensiones físicas son compactas para el espacio designado dentro del prototipo, y lo más

destacable es que cumple con las mismas capacidades que su consecuente Arduino Uno pero a un costo menor (Arduino, 2020).

Tabla 3-2: Características de tarjetas de desarrollo

CARACTERÍSTICAS	Arduino UNO R3	Arduino Nano	Arduino Mega2560
			
Alimentación	5v	5v	5v
Io max	40ma	40ma	12ma
Frecuencia de reloj	16Mhz	16Mhz	16Mhz
Microcontrolador	ATmega328	ATmega328	ATmega2560
Flash Memory	32KB	32KB	256KB
SRAM	2KB	2KB	8KB
EEPROM	1 KB	1KB	4
USB	Si	Si	Si
Analog I/O	6	8	16
Wifi	No	No	No
UART	Si	Si	Si
PWM	6	6	15
SPI/I2C	Si	Si	Si
Interrupciones	2 fijos y 2 programable	2 fijos y 2 programable	6
Costo en el mercado	10\$	6\$	16\$

Fuente: Arduino, 2020

Realizado por: Pilamala Henry & Hilaño Jorge, 2020

2.10.4 Módulo de radio frecuencia

El módulo de comunicación seleccionado para el robot es un módulo de radio frecuencia “Holybro Telemetry Radio” de la figura 27-2, que tiene alcances de más de 300 metros dependiendo el tipo de antena, este módulo es de código libre y excelente misiones de planificación, en el caso del presente estudio resulta muy eficiente para la localización en interiores (Getfpv, 2020).



Figura 27-2: Holybro Telemetry Radio

Realizado por: Pilamala, H; Hilaño, J, 2020

Las características del módulo se especifican en la tabla 4-2.

Tabla 4-2: Características de tarjetas de desarrollo

Especificación.	Característica.
Frecuencia	915Mhz
Alimentación	5v
Corriente de transmisión	100mA at 20dBm
Corriente de recepción	25mA
Interfaz serial	3.3V UART
Dimensiones	26 x 53 x 10.7mm
Puerto	Micro USB
Conector	JST-GH de 4 posiciones
Conector aéreo	RP-SMA

Fuente: Getfpy, 2020

Realizado por: Pilamala Henry & Hilaño Jorge, 2020

2.10.5 Diseño de placa principal electrónica

El esquema general de conexión electrónica está representado en la figura 28-2, el centro de conexión de los equipos es el Arduino Nano, que recibe los datos de los encoders y del módulo de radio frecuencia. La etapa de regulación alimenta a todo el sistema con voltajes de 12 para los motores, con 5 voltios para los encoders, arduino nano y módulo de comunicación.

El driver L298N controla la velocidad de los motores con PWM a partir del control efectuado por el Arduino, los mandos de control para realizar la ruta son enviados desde la PC hacia el Arduino por tramas y a la vez cada trama realizada por el robot es regresado con datos reales hacia Matlab.

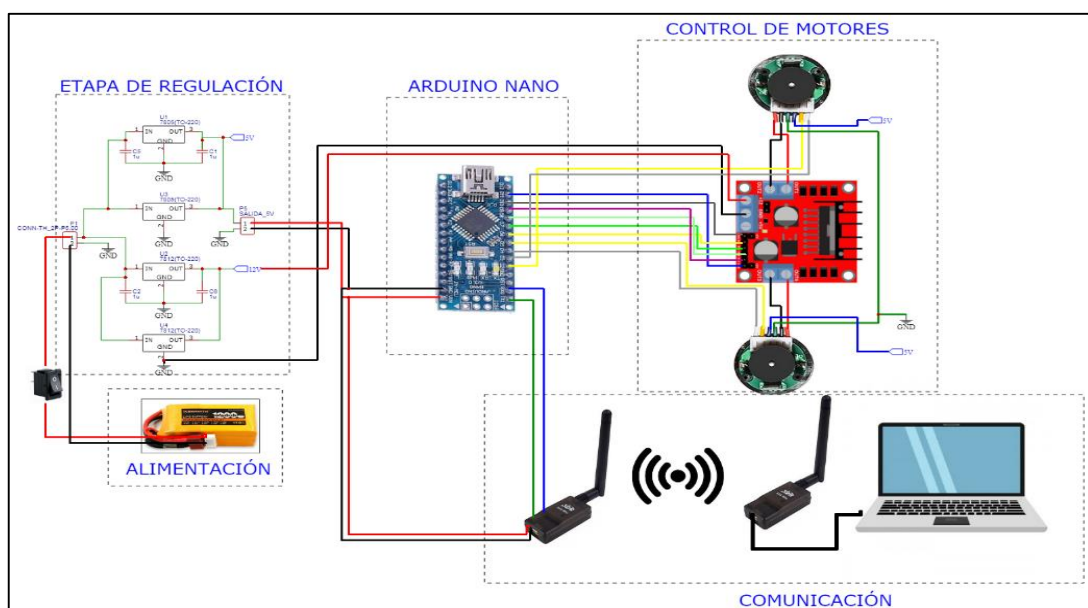


Figura 28-2: Esquema general de conexiones

Realizado por: Pilamala, H; Hilaño, J, 2020

2.10.6 Diseño PCB

El diseño general de la placa electrónica se lo realiza en el software libre EasyEDA, que da buenas prestaciones para diseño electrónico, le presente diseño de la figura 29-2 presenta un espacio designado para regulación, tarjeta de desarrollo Arduino nano, controlador L298N y módulo de telemetría, además de pines y espadines designado para la conexión de encoders y un tester de carga de la batería.

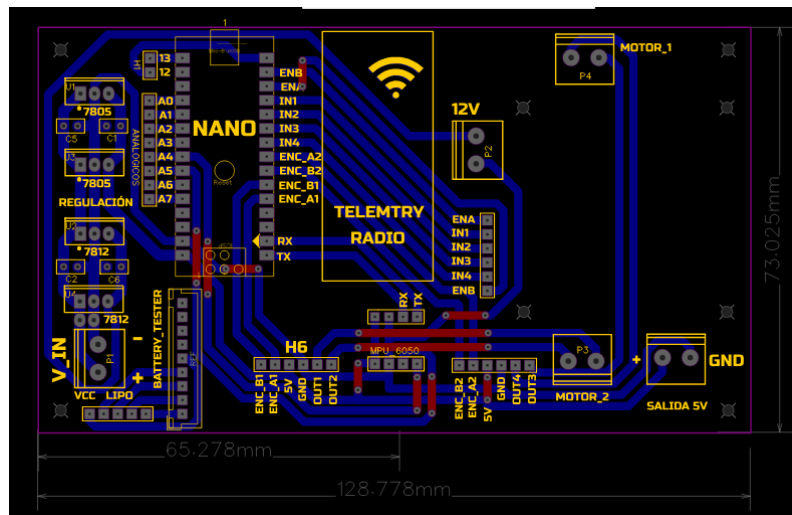


Figura 29-2: Diseño de PCB y disposición de elementos.

Realizado por: Pilamala, H; Hilaño, J, 2020

El software EasyEDA también tiene la opción de presentar una vista 3D del diseño PCB que se realiza, para tener una idea de cómo será la vista y espacios de los elementos reales, como se ve en la figura 30-2.

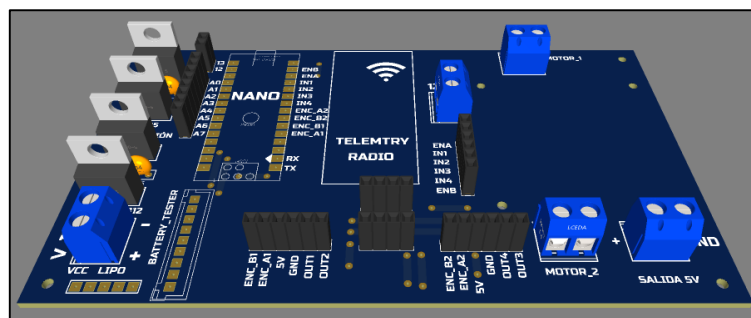


Figura 30-2: Vista3D del diseño en PCB.

Realizado por: Pilamala, H; Hilaño, J, 2020

2.10.7 Cálculo de batería

Para la elección de batería se dimensionó con respecto a los picos máximos de corriente que demanda cada elemento presente en el sistema de manera que se evite daños por sobre corriente y resguardando la vida útil de la batería.

Tabla 5-2: Elementos de consumo de corriente del robot

Elementos de consumo de [I] del robot			
Característica	Carga nominal	Carga máxima	Unidades
Motor1	150	1600	[mA]
Motor2	150	1600	[mA]
Driver L298N	36	36	[mA]
Arduino Nano	40	40	[mA]
LM7805(5v) x2	500	2000	[mA]
LM7812 (12v) x2	500	2000	[mA]
Holybro Telemetry Radio	25	100	[mA]
TOTAL	1401	7376	[mA]

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

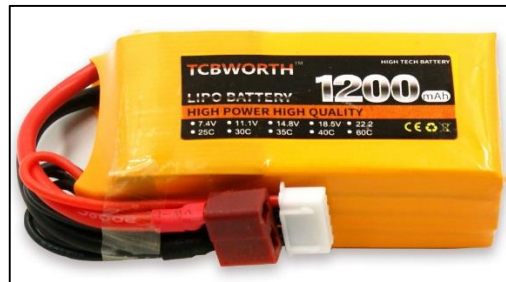


Figura 31-2: Batería Lipo 4 celdas

Realizado por: Pilamala, H; Hilaño, J, 2020

Batería lipo con cuatro celdas de 3.7v cada uno, tomando en cuenta que se requiere máximo 30 minutos de trabajo constante.

Corriente máx. que puede dar la batería:

$$Pico\ maximo\ de\ I = 1200mah * 35$$

$$Pico\ maximo\ de\ I = 42A$$

- Duración estimada carga nominal:

$$T.\ estimado\ de\ duracion = \frac{1200mAH * 60min}{1401.2mAH}$$

$$T.\ estimado\ de\ duracion = 51,39\ min$$

- Duración estimada carga máxima:

$$T. \text{ estimado de duracion} = \frac{1200\text{mAH} * 60\text{min}}{7376\text{mAH}}$$

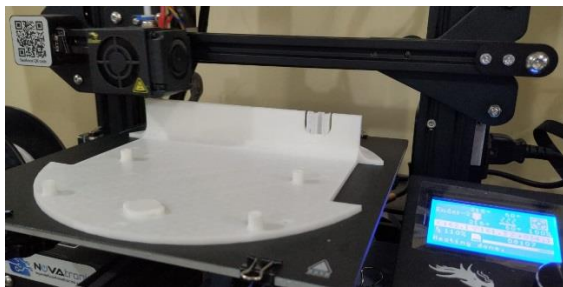
$$T. \text{ estimado de duracion} = 9,76 \text{ min}$$

2.11 Construcción del prototipo de robot móvil.

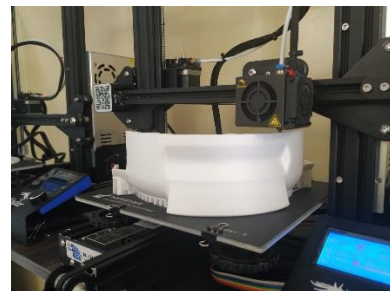
En este apartado se muestra el procedimiento de construcción, y la metodología para cada caso. Se inicia con la impresión 3D y su posterior ensamblaje, posteriormente se realiza la manufactura de la placa general electrónica en baquelita, procediendo con la soldadura de sus elementos, al final se muestra el diseño completo del robot móvil en configuración diferencial.

2.11.1 Impresión 3D del prototipo de robot móvil.

El diseño del robot móvil en configuración diferencial es impreso en PLA, al ser un prototipo de investigación se opta por este material por su rigidez y durabilidad como se muestra en la figura 32-2.



a)



b)

Figura 32-2: a) Impresión de la base, b) Impresión de la estructura externa

Realizado por: Pilamala, H; Hilaño, J, 2020

2.11.2 Desarrollo de placa electrónica

El desarrollo de la placa electrónica en una lámina de baquelita se lo puede constatar en la figura 33-2, donde a un lado se realiza las pistas del circuito y al otro una presentación de serigrafía de los elementos (figura 34-2).

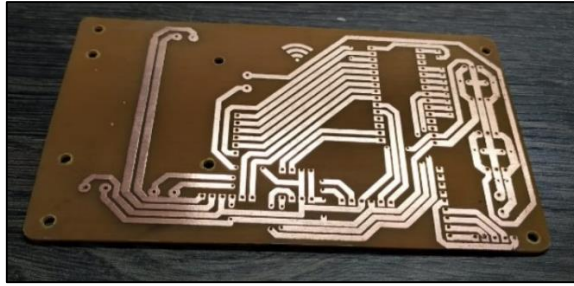


Figura 33-2: Placa electrónica del sistema.

Realizado por: Pilamala, H; Hilaño, J, 2020

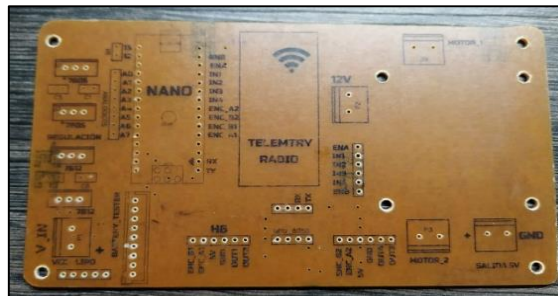


Figura 34-2: Serigrafía de Placa electrónica del sistema.

Realizado por: Pilamala, H; Hilaño, J, 2020

La disposición de los elementos en la placa electrónica se visualiza en la figura 35-2.

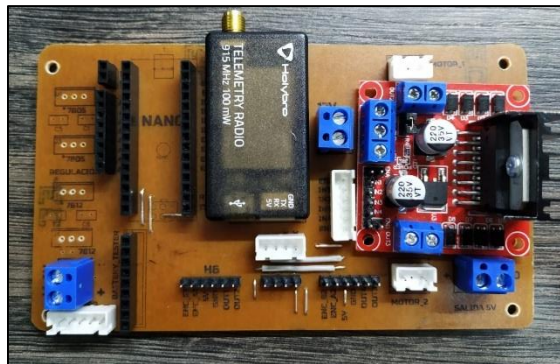


Figura 35-2: Elementos de placa electrónica del sistema.

Realizado por: Pilamala, H; Hilaño, J, 2020

2.11.3 Acoplamiento de los elementos constitutivos del prototipo

El robot móvil diseñado para la localización en interiores debe cumplir con los requerimientos establecidos para lo cual se ha procurado que tanto su peso este balanceado como que su estructura mecánica no presente tantos errores sistemáticos que afecten a su movimiento como se ve en la figura 36-2.



Figura 36-2: Estructura externa del robot.

Realizado por: Pilamala, H; Hilaño, J, 2020

El modelo del robot con todas sus partes incorporadas la podemos ver en la figura 37-2(a), y su forma en funcionamiento en la figura 37-2 (b).

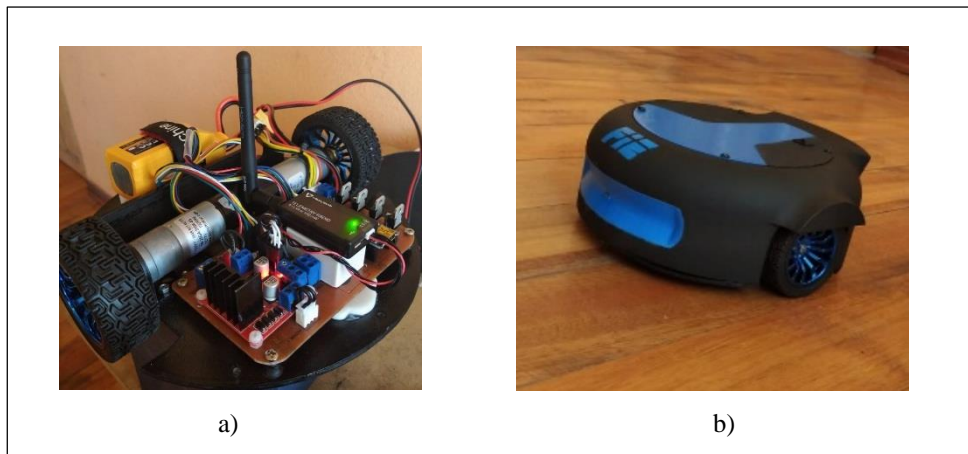


Figura 37-2: a) disposición interna de los elementos, b) Vista general del robot

Realizado por: Pilamala, H; Hilaño, J, 2020

2.12 Lectura Encoders de cuadratura

Los motores utilizados poseen encoders de cuadratura, los mismos constan de dos sensores de efecto hall encargados de dar las señales digitales al pasar una banda magnética acoplada al eje externo del motor, estos sensores se encuentran desfasados 90 grados eléctricos y generan una secuencia de on-off que permite saber su sentido de giro, la velocidad y el número de giros.

Los flancos de subida y bajada de las dos señales podemos verlas en la figura 38-2, los mismos que generan cuatro posibles combinaciones, 00,01,10,11 asumiendo a la señal A como el bit más significativo. Como se denota en la figura la señal A de amarillo y la señal B de azul están desfasadas 90°, la secuencia de detección de giro en sentido horario es 10,11,01,00.

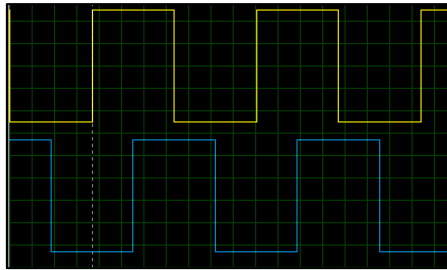


Figura 38-2: Señal digital del Encoder en sentido horario

Realizado por: Pilamala, H; Hilaño, J, 2020

Para el giro en sentido antihorario las combinaciones quedan de la siguiente forma 00,01,11,10 según la figura 39-2.

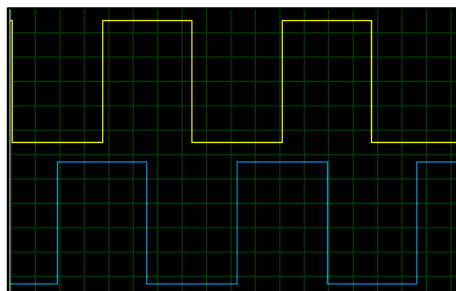


Figura 39-2: Señal digital del Encoder en sentido anti-horario

Realizado por: Pilamala, H; Hilaño, J, 2020

Existen combinaciones de acuerdo a su estado anterior que nunca se van a dar (00,11), (11,00), (01,10), (10,01) y se las marca como E. En la tabla 6-2 se muestra todas las combinaciones posibles en que los encoders cambian de estado.

Tabla 6-2: Estados anterior vs actual.

ESTADO	ANTERIOR	0	1	2	3
ACTUAL	AB	00	01	10	11
0	00	0	Horario	AntiHorario	E
1	01	AntiHorario	0	E	Horario
2	10	Horario	E	0	AntiHorario
3	11	E	AntiHorario	Horario	0

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

Se toma en consideración para los cálculos de odometría la resolución del encoder, PPR o pulsos por revolución y se utiliza una precisión cuádruple, es decir se toma en consideración flancos de subida y bajada en los dos canales. Mediante la ecuación 71 y los datos del fabricante podemos calcular la resolución:

$$Re = N_p * s * r \quad (71)$$

Donde:

N_p : número de pulsos por revolución del eje del motor

s : número de estados generado por los canales AB

r : relación de reducción de la caja reductora

$$Re = 12 * 4 * 45$$

$$Re = 2160$$

Para la lectura con arduino se usan interrupciones para la detección de los flancos, el procedimiento se detalla el diagrama de flujo del gráfico 8-2.

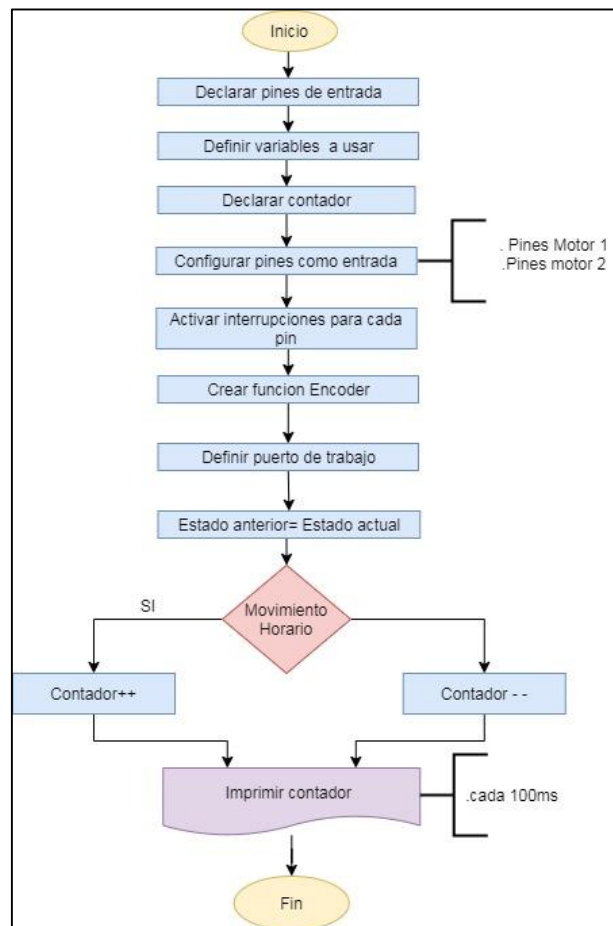


Gráfico 8-2. Proceso de lectura del encoder en arduino

Realizado por: Pilamala, H; Hilaño, J, 2020

Se utiliza los puertos de interrupción del arduino nano para las lecturas de los flancos de los canales A y B del encoder.

La resolución ayuda al cálculo de la velocidad angular y la velocidad lineal del sistema de acuerdo con las siguientes ecuaciones que se basan en las ecuaciones 18 y 20, se multiplica por mil para expresar en segundos ya que t está en milisegundos:

$$\omega = \frac{2\pi * 1000 * N}{t * Re} \quad (72)$$

$$v = \frac{D * \pi * 100}{t * Re} \quad (73)$$

Donde:

ω : velocidad angular

v : velocidad lineal

D : diámetro de las llantas

Re : resolución de encoder

N : numero de pulsos cada tiempo t

t : tiempo de muestreo

2.13 Control de velocidad en motores

Las velocidades angulares que se obtienen a partir de las ecuaciones 52 y 53 pasan a ser consignas las cuales son enviadas al robot, para que los motores del robot lleguen a estas consignas de velocidad se debe tener en su programación un control PID por lo que es necesario conocer la función de transferencia de los motores para hallar las constantes del controlador PID, como los motores son los mismos se analizara uno de ellos.

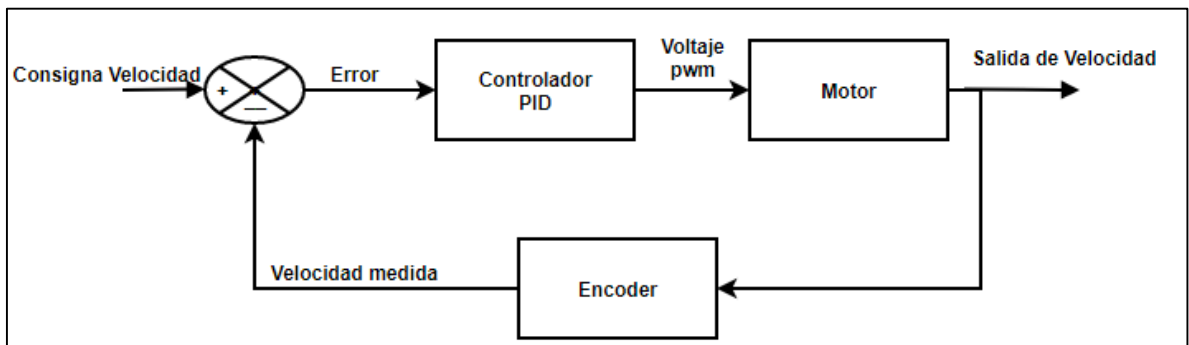


Figura 40-2: Diagrama de bloques para un control PID de velocidad de un motor

Realizado por: Pilamala, H; Hilaño, J, 2020

La herramienta System Identification de Matlab permite estimar el modelo de un proceso al cargar datos de las entradas y salidas del sistema las cuales se pueden obtener al hacer pruebas en lazo abierto. En el gráfico 9-2 se muestra la respuesta en lazo abierto del motor ante una entrada tipo escalón, esta entrada es de velocidad angular en rad/s.

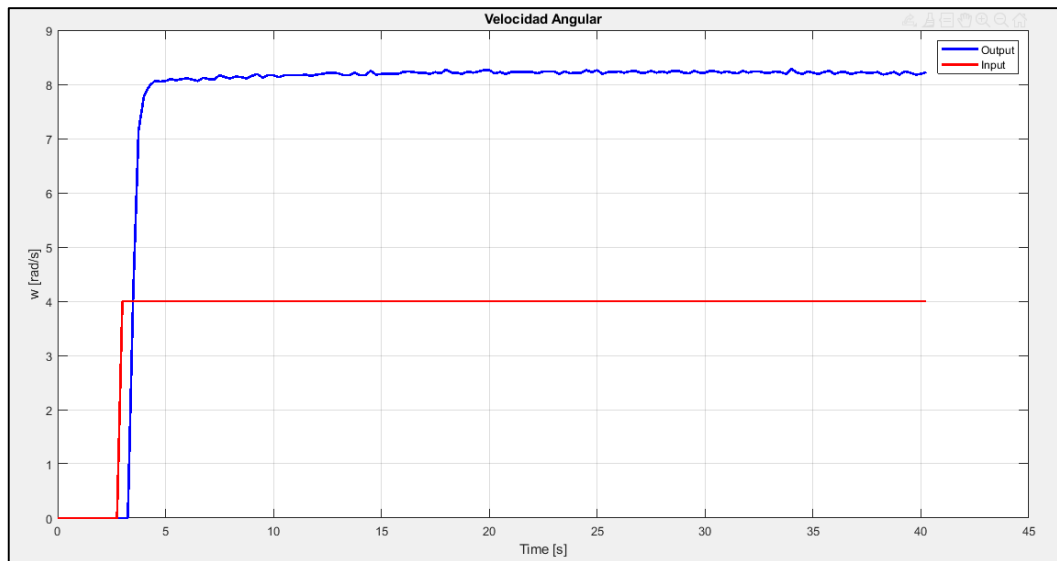


Gráfico 9-2. Evolución de la velocidad angular total del robot en el tiempo.

Realizado por: Pilamala, H; Hilaño, J, 2020

La figura 41-2 nos muestra la estimación del modelo de proceso del motor en una función de transferencia de primer orden entregando como datos la constante de ganancia estática K y la constante de tiempo Tp1.

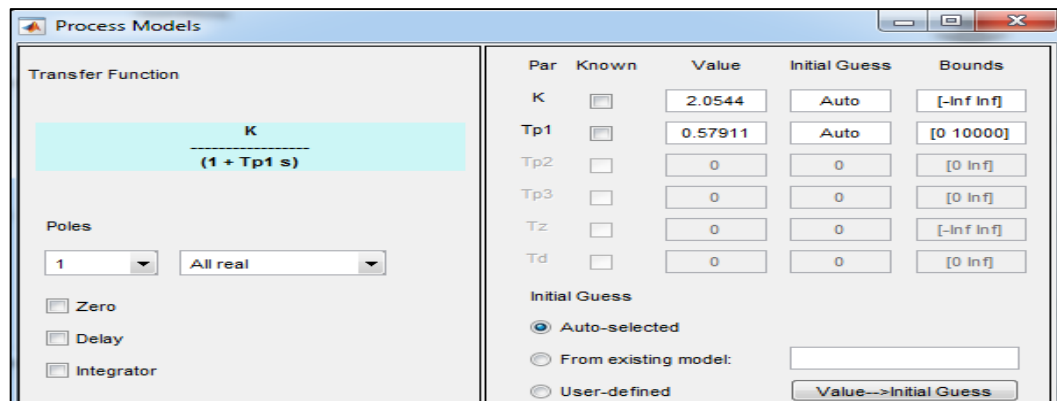


Figura 41-2: Estimación del modelo de proceso del motor

Realizado por: Pilamala, H; Hilaño, J, 2020

La función de transferencia del motor será la siguiente:

$$H_m = \frac{2.0544}{1 + 0.57911s} \quad (74)$$

En el gráfico 10-2 se muestra la estimación de la respuesta en lazo abierto del motor. La línea negra representa la respuesta original y la línea roja la respuesta estimada.

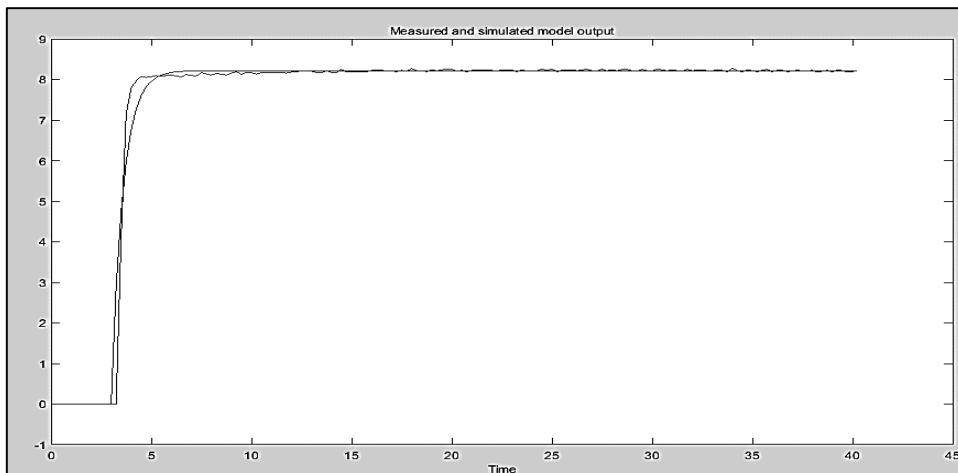


Gráfico 10-2. Estimación de la respuesta en lazo abierto del motor.

Realizado por: Pilamala, H; Hilaño, J, 2020

Una vez obtenida la función de transferencia del motor se procede a subirla en la herramienta PID Tuner de Matlab la cual analiza la respuesta en lazo cerrado obteniendo las constantes del controlador PID como se observa en el gráfico 11-2. De acuerdo a la figura 42-2 el controlador que se debe aplicar es un PI ya que solo se tiene las dos constantes k_p y k_i .

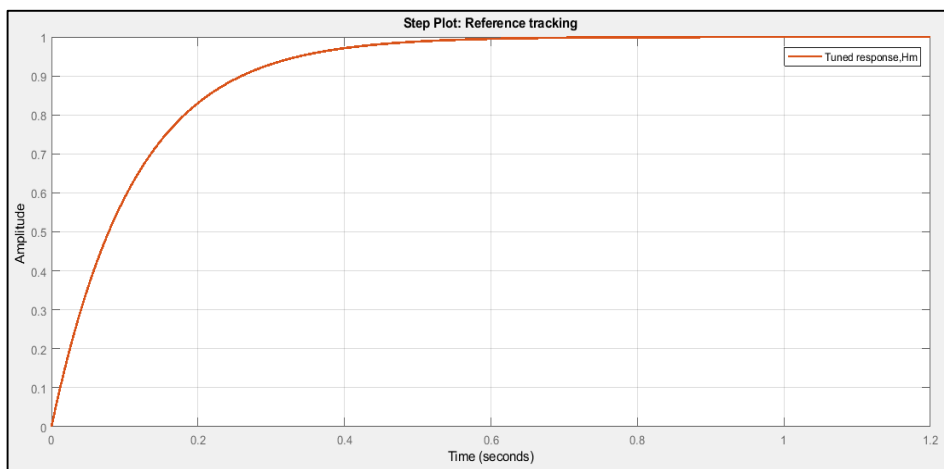


Gráfico 11-2. Respuesta en Lazo cerrado del motor.

Realizado por: Pilamala, H; Hilaño, J, 2020

Controller Parameters	
	Tuned
K_p	2.5003
K_i	4.3174
K_d	0
T_f	n/a

Figura 42-2: Parámetros del Controlador PID

Realizado por: Pilamala, H; Hilaño, J, 2020

Al hallar las constantes del controlador PI se procede a implementar este control en Arduino, en el gráfico 12-2 se muestra el diagrama de flujo del funcionamiento del control PID de velocidad dentro de la programación en Arduino, El control en sistemas digitales se lo hace de manera discreta, un control PID discreto tiene la siguiente ecuación (Arellano, 2015, p.37).

$$U(k) = k_p * e(t) + k_i * \frac{T}{2} * (e(k) + e(k - 1)) + k_d * \frac{(e(k) - e(k - 1))}{T} \quad (75)$$

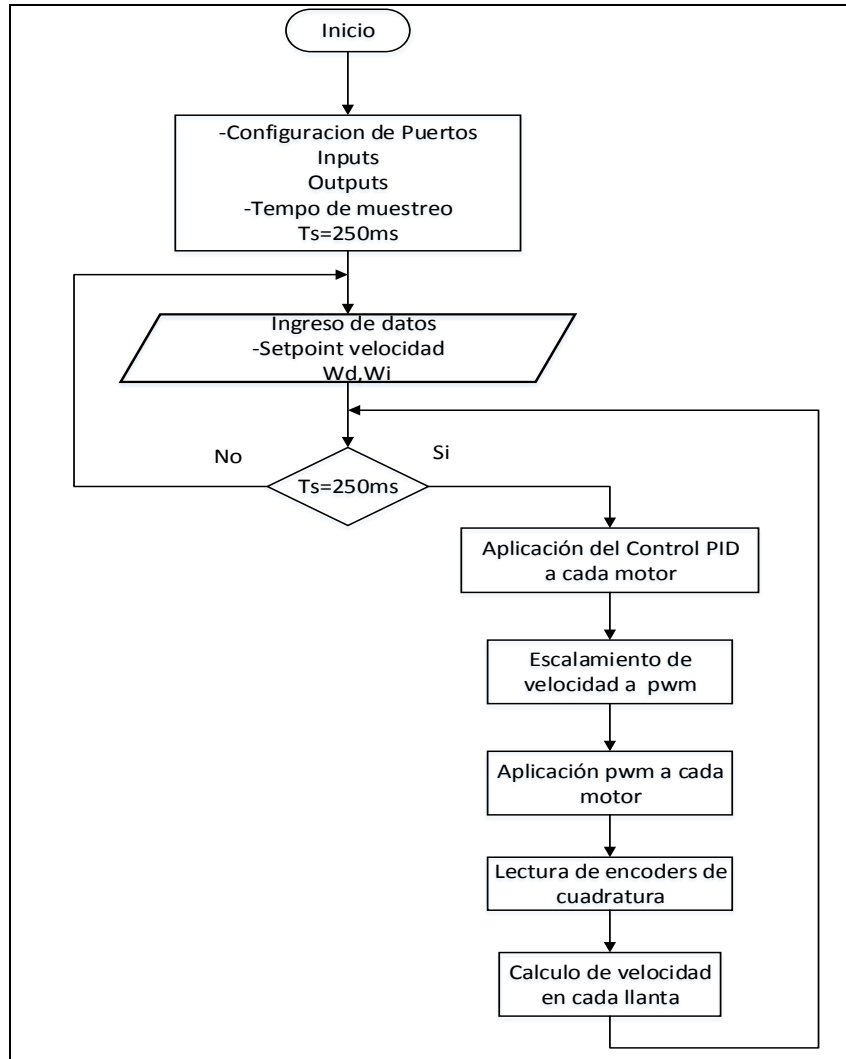


Gráfico 12-2. Diagrama de flujo de Control PID de motores

Realizado por: Pilamala, H; Hilaño, J, 2020

La programación del control PID de velocidad en Arduino comienza con la declaración de cada pin como entradas o salidas, se lee la consigna de velocidad y se entra a un bloque de condición para asegurar el tiempo de muestreo en que se tomara los datos de los encoders, una vez cumplido la condición del tiempo de muestreo se aplica el control PID discreto, este control se basa en encontrar el error entre la consiga y la salida que se obtiene por cada tiempo de muestreo. De

acuerdo con la ecuación 75 para la parte proporcional se trabaja con el error actual, en la parte integral se debe sumar el error actual con el error pasado y para la parte derivativa se debe restar el error actual del error pasado. Aplicado el controlador obtenemos una salida de control con la cual se obtiene los valores de pwm los cuales se aplicarán a cada motor. Para cerrar el lazo se retroalimentan la salida al medir la velocidad que tiene cada motor mediante los encoders, esta velocidad se volverá a comparar con la consigna repitiéndose de nuevo el ciclo para obtener los errores y aplicar el controlador con el fin de llegar a la consigna deseada.

2.14 Odometría en Arduino

El cálculo de odometría dentro de Arduino se lleva a cabo con los datos del conteo de pulsos por revolución en los encoders magnéticos, una vez obtenido estos datos se procede con el cálculo de los desplazamientos realizados por el robot en cada llanta, en su centro y su desplazamiento angular para posteriormente calcular la posición y orientación basándose en las ecuaciones de la odometría 9, 10, 11, los datos calculados se acumulan en cada tiempo de muestreo para obtener los valores totales. El gráfico 13-2 muestra el diagrama de flujo del proceso de la implantación de la odometría en Arduino.

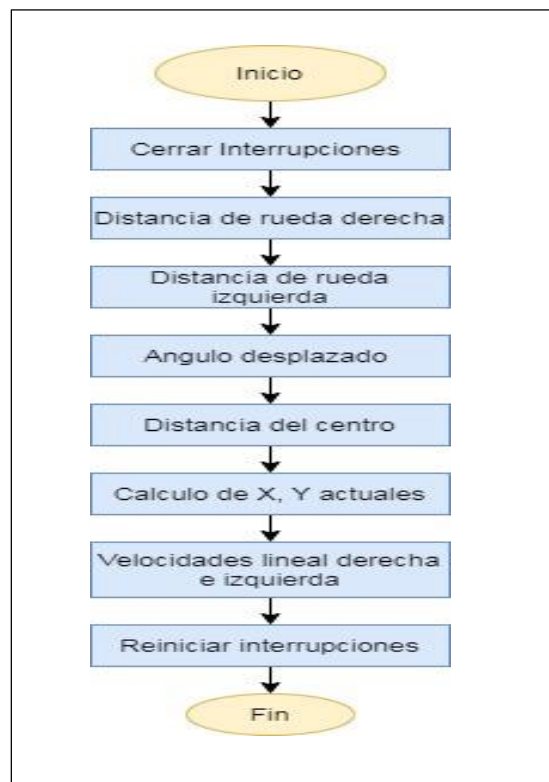


Gráfico 13-2. Lectura de posición y orientación en Arduino

Realizado por: Pilamala, H; Hilaño, J, 2020

2.15 Comunicación inalámbrica R.F. entre Matlab y Arduino

La comunicación entre Matlab y Arduino se realiza a través de los módulos RF de telemetría que trabajan a una frecuencia de 915 Mhz, la velocidad asignada de comunicación serial es 57600 baudios, los datos que se transmiten corresponden a velocidades de referencia, velocidades reales obtenidas por el modelo físico, y posiciones estimadas mediante odometría.

Para la comunicación es necesario dos módulos RF para realizar una comunicación Half-duplex, envío y recepción de datos para cada unidad (PC y Arduino), (Sen et al., 2009, p.15). El procedimiento de envío de datos y recepción se muestra en el diagrama de flujo del gráfico 14-2.

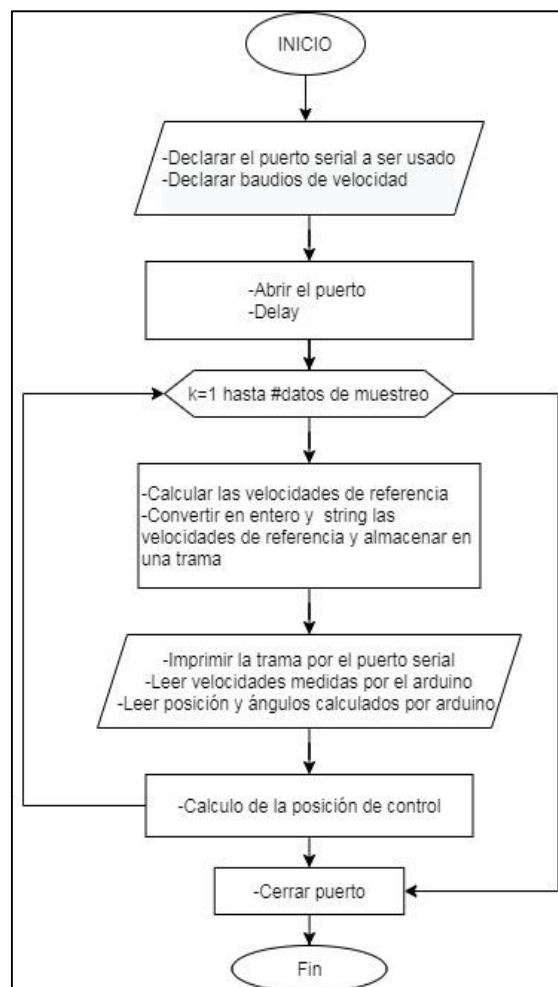


Gráfico 14-2. Diagrama de flujo de comunicación del sistema.

Realizado por: Pilamala, H; Hilaño, J, 2020

CAPÍTULO III

3. VALIDACIÓN DEL PROTOTIPO

En el presente capítulo se detalla las pruebas y los resultados obtenidos del robot móvil en interiores para diferentes trayectorias que le permiten llegar a un punto especificado dentro de un mapa conocido con obstáculos. De igual manera se presenta una comparación de rutas realizadas, pero con diferente planificador. Para la obtención de los resultados se hace uso de análisis estadísticos con el fin de validar la eficiencia del prototipo.

3.1 Ambiente de pruebas

El mapa diseñado para realizar las pruebas se ha implementado sobre un tablero MDF plano sin fricción de dimensiones 2.16x2.60m como se observa en la figura 1-3, para considerar el diámetro del robot se infla el mapa como se muestra en la figura 12-2. Las pruebas consisten en que el robot parta desde la posición (0.20,0.20) m hacia tres diferentes puntos como se describe en la tabla 1-3.



Figura 1-3: Ambiente de pruebas

Realizado por: Pilamala, H; Hilaño, J, 2020

Tabla 1-3: Puntos de prueba en el mapa.

N° Prueba	X [m]	Y [m]
Prueba 1	1.90	0.24
Prueba 2	1,75	2.3
Prueba 3	0.35	1,70

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

3.2 Evaluación de la funcionalidad prototipo

3.2.1 Pruebas de control de velocidad

A continuación, se evaluará que cada motor llegue a una velocidad de referencia al aplicar el controlador PI y las respectivas constantes obtenidas en el capítulo anterior.

Se evaluará el sistema para las siguientes velocidades:

- $\omega_r = \omega_l = 6 \text{ rad/s}$

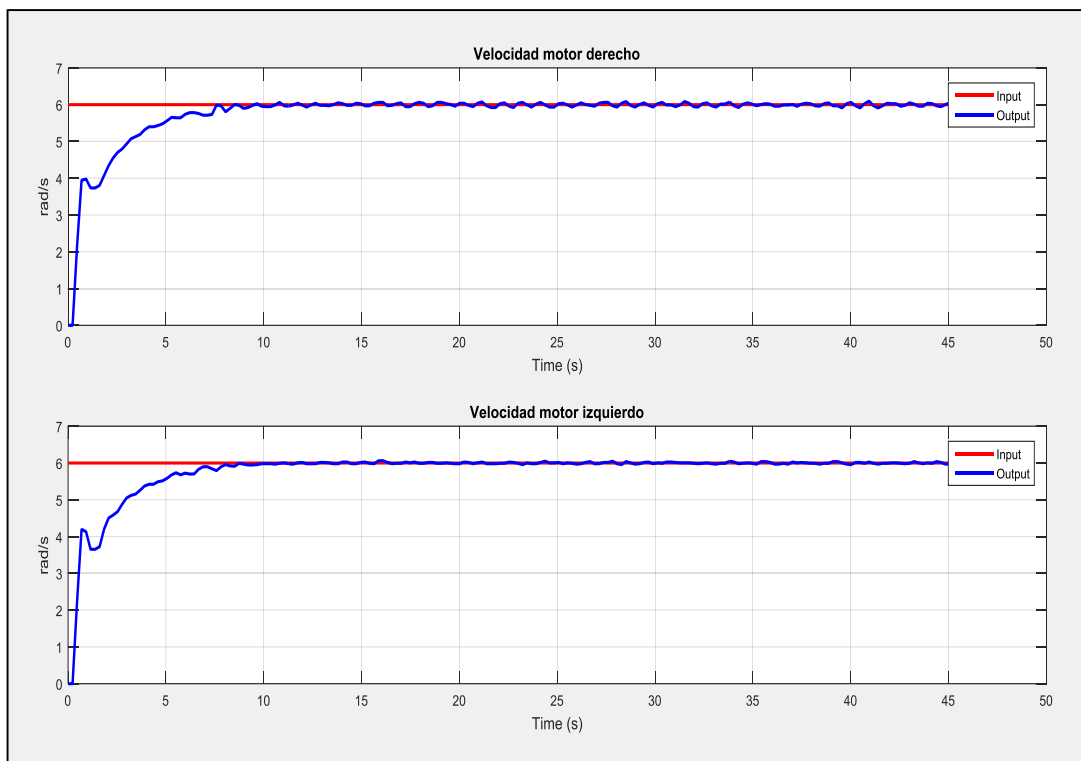


Gráfico 1-3. Respuesta de velocidad $w_r = w_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

El gráfico 1-3 muestra cómo la velocidad en cada motor llega a la consigna de entrada de velocidad, al ser la misma velocidad en ambos motores el robot debe ir en línea recta tal como se muestra en el gráfico 2-3 en donde se observa una simulación del movimiento real que ha hecho el robot.

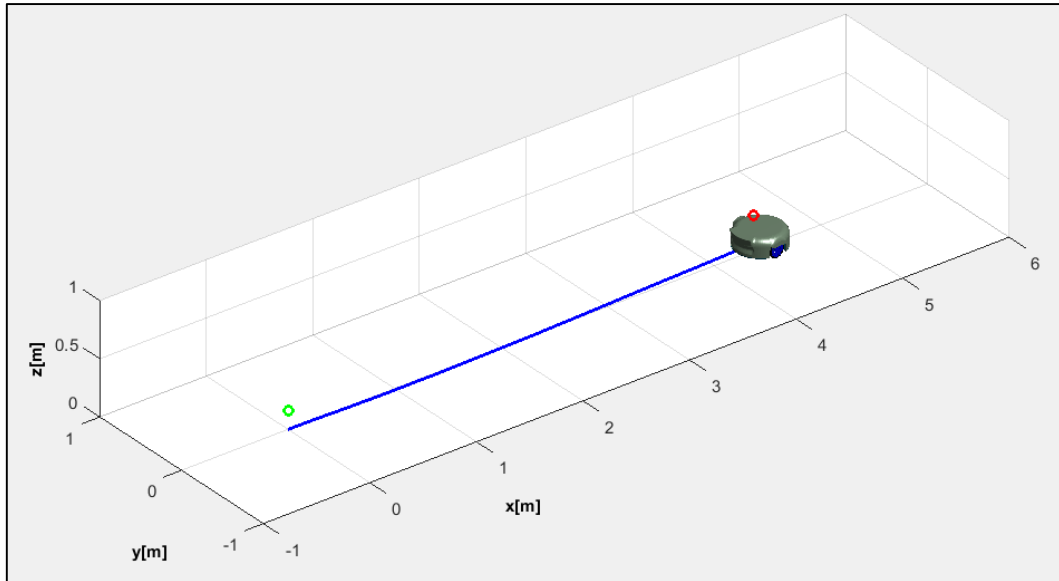


Gráfico 2-3. Trayectoria del robot móvil $w_r = w_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

Para cumplir una trayectoria deseada los motores del robot móvil deberán moverse a diferentes velocidades por lo que se ejecuta una prueba en que a cada motor se le dan entradas diferentes de velocidad.

- $\omega_r = 7.74 \text{ rad/s}$
- $\omega_l = 4.29 \text{ rad/s}$

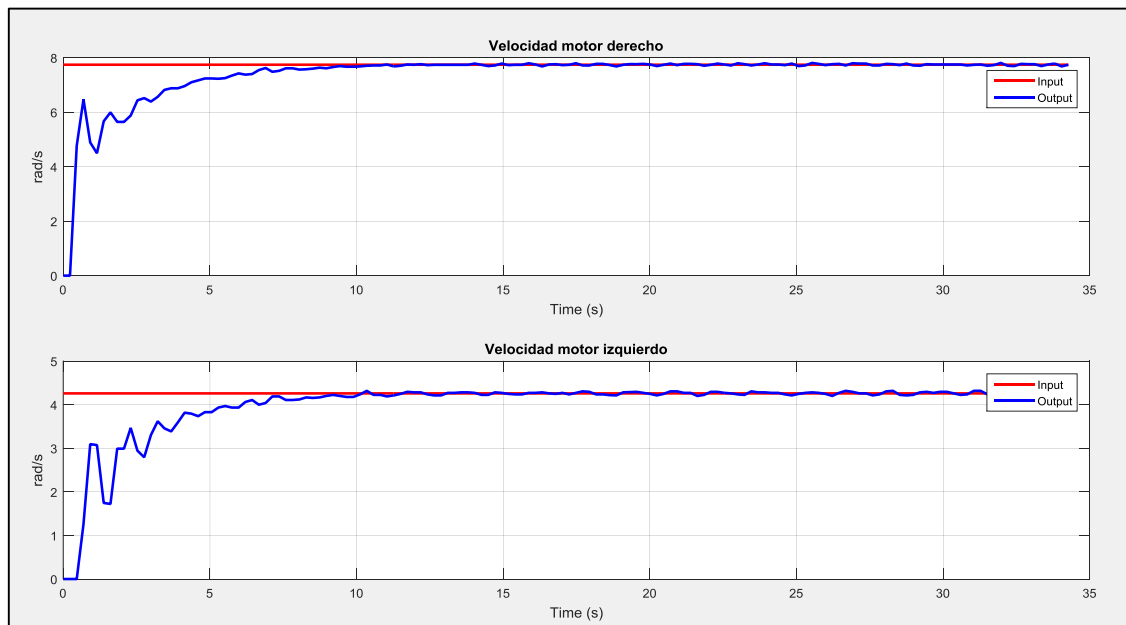


Gráfico 3-3. Respuesta de velocidad $w_r > w_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

De acuerdo con el gráfico 3-3 cada motor lleva a su consigna de entrada de velocidad ratificando que el controlador está funcionando y el sistema se estabiliza. La velocidad del motor derecho es

mayor que la del motor izquierdo por lo que el robot dará un giro hacia la izquierda generando una trayectoria circular en forma anti horaria como se muestra en el gráfico 4-3.

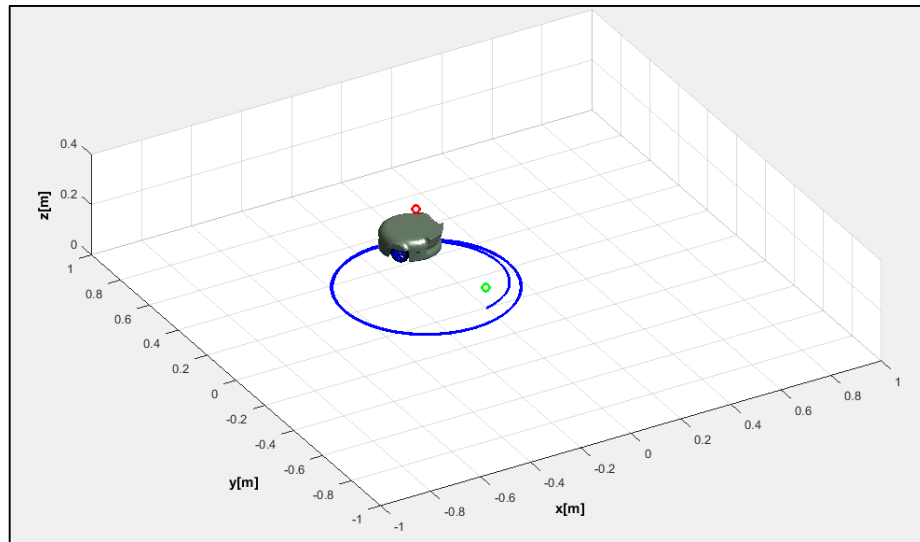


Gráfico 4-3. Trayectoria del robot móvil cuando $\omega_r > \omega_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

- $\omega_r = 4.29 \text{ rad/s}$
- $\omega_l = 7.74 \text{ rad/s}$

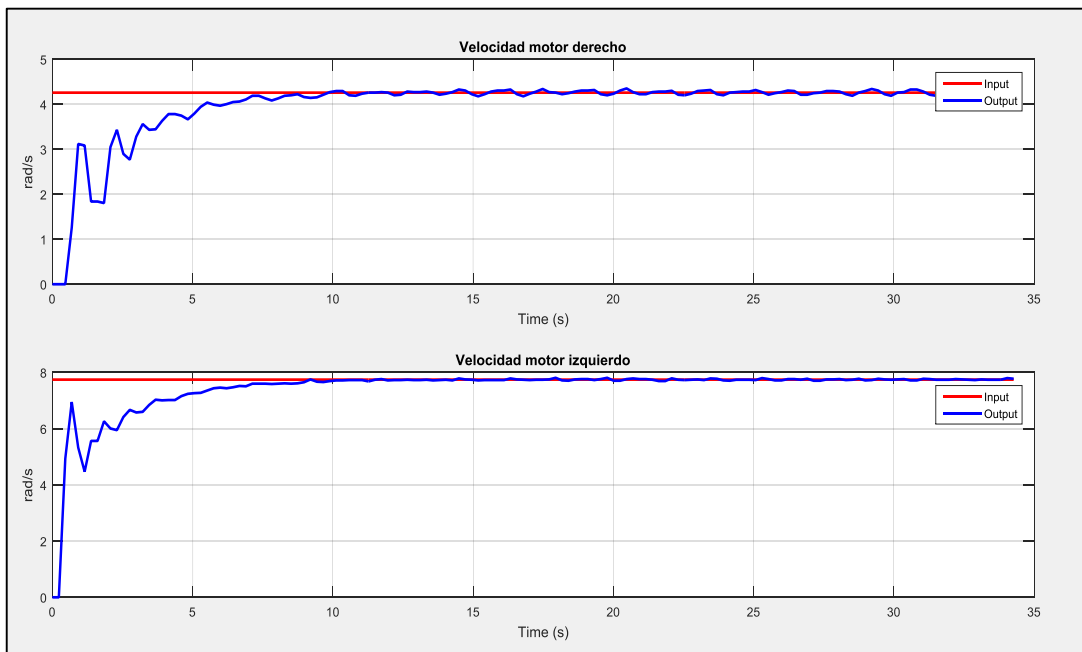


Gráfico 5-3. Respuesta de velocidad $\omega_r < \omega_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

De acuerdo con el gráfico 5-3. cada motor llega a su consigna por lo que el controlador está funcionando correctamente. La velocidad del motor izquierdo es mayor que la del motor derecho

por lo que el robot dará un giro hacia la derecha generando una trayectoria circular solo que esta vez será en forma horaria como se muestra en el gráfico 6-3

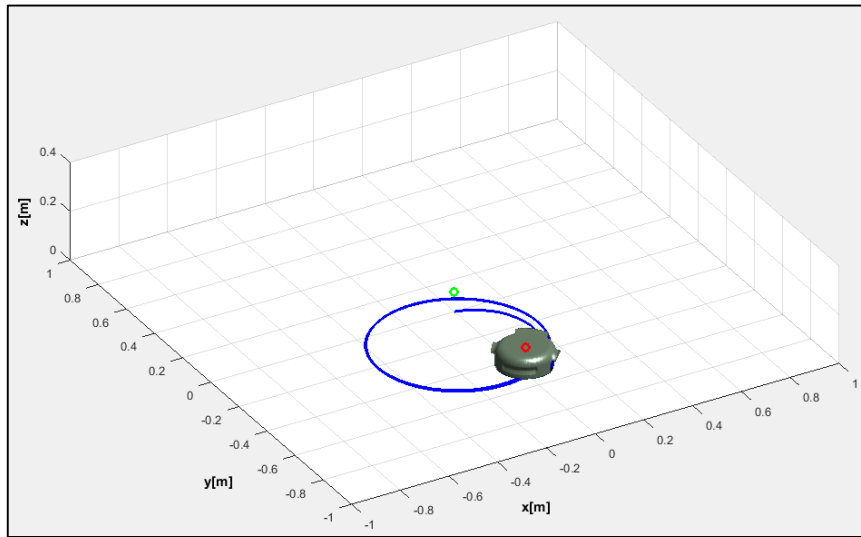


Gráfico 6-3. Trayectoria del robot móvil $w_r < w_l$

Realizado por: Pilamala, H; Hilaño, J, 2020

De acuerdo a las gráficas 1-3, 3-3 y 5-3 el tiempo aproximado de establecimiento de la velocidad hacia su consigna es de 8 segundos.

3.2.2 Pruebas de posición con odometría

Se realizó la comparación entre las medias de las posiciones calculados por odometría con las posiciones medidas con un instrumento patrón o flexómetro y estos a la vez con las posiciones deseadas para verificar su precisión, se realizó pruebas paramétricas t-student tanto para coordenadas en X e Y. Para las pruebas se posicionó al prototipo en tres coordenadas diferentes por 10 ocasiones, en la tabla 2-3 se detalla los resultados obtenidos.

Tabla 2-3: Mediciones de posicionamiento.

N° Prueba	Posición deseada		Posición medida con equipo patrón		Posición medida con odometría	
	X_D	Y_D	X_P	Y_P	X_E	Y_E
1	0,6	0,6	0,6099	0,607	0,6017	0,6036
2	0,6	0,6	0,609	0,605	0,6038	0,5951
3	0,6	0,6	0,606	0,609	0,5953	0,6045
4	0,6	0,6	0,599	0,609	0,6012	0,6036
5	0,6	0,6	0,607	0,61	0,5996	0,6057
6	0,6	0,6	0,606	0,611	0,5994	0,6083
7	0,6	0,6	0,612	0,609	0,6084	0,6041

8	0,6	0,6	0,611	0,609	0,6084	0,6041
9	0,6	0,6	0,607	0,612	0,6012	0,6083
10	0,6	0,6	0,604	0,598	0,5996	0,6057
11	1,2	1,2	1,21	1,203	1,2073	1,1972
12	1,2	1,2	1,209	1,212	1,2007	1,2095
13	1,2	1,2	1,206	1,207	1,1909	1,1931
14	1,2	1,2	1,205	1,206	1,1947	1,2041
15	1,2	1,2	1,201	1,203	1,1938	1,1919
16	1,2	1,2	1,207	1,204	1,199	1,1967
17	1,2	1,2	1,206	1,198	1,1908	1,1901
18	1,2	1,2	1,203	1,201	1,195	1,191
19	1,2	1,2	1,204	1,21	1,1958	1,2069
20	1,2	1,2	1,206	1,208	1,1929	1,1972
21	1,8	1,8	1,807	1,809	1,7899	1,7901
22	1,8	1,8	1,808	1,809	1,7906	1,7901
23	1,8	1,8	1,806	1,812	1,799	1,8072
24	1,8	1,8	1,809	1,81	1,8139	1,7995
25	1,8	1,8	1,809	1,81	1,7901	1,8005
26	1,8	1,8	1,813	1,814	1,8026	1,8056
27	1,8	1,8	1,826	1,807	1,8152	1,791
28	1,8	1,8	1,818	1,82	1,8052	1,8114
29	1,8	1,8	1,817	1,823	1,8052	1,8201
30	1,8	1,8	1,816	1,818	1,8009	1,807

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

La primera prueba se refiere a la comparación entre las posiciones calculadas con odometría y las posiciones medidas con el instrumento patrón en referencia al eje X.

- Se define una hipótesis nula y otra alternativa

$$H_0 : \mu = \mu_0$$

$$H_A : \mu \neq \mu_0$$

Ho: La media de las posiciones calculadas con la odometría es igual a la media de las posiciones medidas con el instrumento patrón en el eje X.

Ha: La media de las posiciones calculadas con la odometría es diferente a la media de las posiciones medidas con el instrumento patrón en el eje X.

- Se define de un nivel de significancia del 5%, es decir $\alpha = 0.05$, y un nivel de confiabilidad del 95%.
- Si $t\text{-estadístico} > t\text{-crítico}$ se rechaza Ho.

Tabla 3-3: Prueba t-student X odometría vs X patrón

Prueba t para dos muestras suponiendo varianzas desiguales		
	X_E	X_P
Media	1.1997367	1.2085633
Varianza	0.24807376	0.2507091
Observaciones	30	30
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	0,06845443	
P(T<=t) una cola	0,47282967	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,94565935	
Valor crítico de t (dos colas)	2,00171748	

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

A partir del análisis estadístico realizado con los datos muestrales se puede concluir que con un nivel de confianza del 95% que la media de las posiciones calculadas con la odometría es igual a la media de las posiciones medidas con el instrumento patrón en el eje X por lo que se acepta la hipótesis nula y se rechaza la hipótesis alternativa.

La segunda prueba se refiere a la comparación entre las posiciones calculadas con odometría y las posiciones medidas con el instrumento patrón en referencia al eje Y.

- Se define una hipótesis nula y otra alternativa

$$H_0 : \mu = \mu_0$$

$$H_A : \mu \neq \mu_0$$

Ho: La media de las posiciones calculadas con la odometría es igual a la media de las posiciones medidas con el instrumento patrón en el eje Y.

Ha: La media de las posiciones calculadas con la odometría es diferente a la media de las posiciones medidas con el instrumento patrón en el eje Y.

- Se define de un nivel de significancia del 5%, es decir $\alpha=0.05$, y un nivel de confiabilidad del 95%.
- Si $t\text{-estadístico} > t\text{-crítico}$ se rechaza Ho

Tabla 4-3: Prueba t-student Y odometría vs Y patrón

Prueba t para dos muestras suponiendo varianzas desiguales		
	Y_E	Y_P
Media	1,20144	1.20876667
Varianza	0,24748488	0,25049991
Observaciones	30	30
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	0,05686683	
P(T<=t) una cola	0,47742337	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,95484673	
Valor crítico de t (dos colas)	2,00171748	

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

A partir del análisis estadístico realizado con los datos muestrales se puede concluir que con un nivel de confianza del 95% que la media de las posiciones calculadas con la odometría es igual a la media de las posiciones medidas con el instrumento patrón en el eje Y por lo que se acepta la hipótesis nula y se rechaza la hipótesis alternativa.

Con estas dos primeras pruebas se puede asegurar que las posiciones calculadas con odometría concuerdan con las posiciones medidas con el instrumento patrón.

La tercera prueba se refiere a la comparación entre las posiciones medidas con el instrumento patrón con las posiciones deseadas en referencia al eje X:

- Se define una hipótesis nula y otra alternativa

$$H_0 : \mu = \mu_0$$

$$H_A : \mu \neq \mu_0$$

Ho: La media de las posiciones medidas con el instrumento patrón es igual a la media de las posiciones deseadas en el eje X.

Ha: La media de las posiciones medidas con el instrumento patrón es diferente a la media de las posiciones deseadas en el eje X.

- Se define de un nivel de significancia del 5%, es decir $\alpha = 0.05$, y un nivel de confiabilidad del 95%.
- Si t-estadístico > t-crítico se rechaza Ho.

Tabla 5-3: Prueba t-student X patrón vs X deseado.

Prueba t para dos muestras suponiendo varianzas desiguales		
	X_P	X_D
Media	1,20856333	1,2
Varianza	0,25070914	0,24827596
Observaciones	30	30
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	-0,06639872	
P(T<=t) una cola	0,47364437	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,94728874	
Valor crítico de t (dos colas)	2,00171748	

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

A partir del análisis estadístico realizado con los datos muestrales se puede concluir que con un nivel de confianza del 95% que la media de las posiciones medidas con el instrumento patrón es igual a la media de las posiciones deseadas en el eje X por lo que se acepta la hipótesis nula y se rechaza la hipótesis alternativa.

La cuarta prueba se refiere a la comparación entre las posiciones medidas con el instrumento patrón con las posiciones deseadas en referencia al eje Y:

- Se define una hipótesis nula y otra alternativa

$$H_0 : \mu = \mu_0$$

$$H_A : \mu \neq \mu_0$$

Ho: La media de las posiciones medidas con un instrumento patrón es igual a la media de las posiciones deseadas en el eje Y.

Ha: La media de las posiciones medidas con un instrumento patrón es diferente a la media de las posiciones deseadas en el eje Y.

- Se define de un nivel de significancia del 5%, es decir $\alpha=0.05$, y un nivel de confiabilidad del 95%.
- Si t-estadístico > t-crítico se rechaza Ho

Tabla 6-3: Prueba t-student Y patrón vs Y deseado

Prueba t para dos muestras suponiendo varianzas desiguales		
	Y_P	Y_D
Media	1,20876667	1,2
Varianza	0,25049991	0,24827586
Observaciones	30	30
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	-0,06798959	
P(T<=t) una cola	0,47301388	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,94602777	
Valor crítico de t (dos colas)	2,00171748	

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

A partir del análisis estadístico realizado con los datos muestrales se puede concluir que con un nivel de confianza del 95% que la media de las posiciones medidas con el instrumento patrón es igual a la media de las posiciones deseadas en el eje Y por lo que se acepta la hipótesis nula y se rechaza la hipótesis alternativa.

Como se puede observar en el gráfico 4-3. y 5-3. las medidas de los valores calculados con respecto a los valores reales deseados no difieren significativamente, pero es evidente que al incrementar la distancia de traslado se nota variaciones entre las medidas.

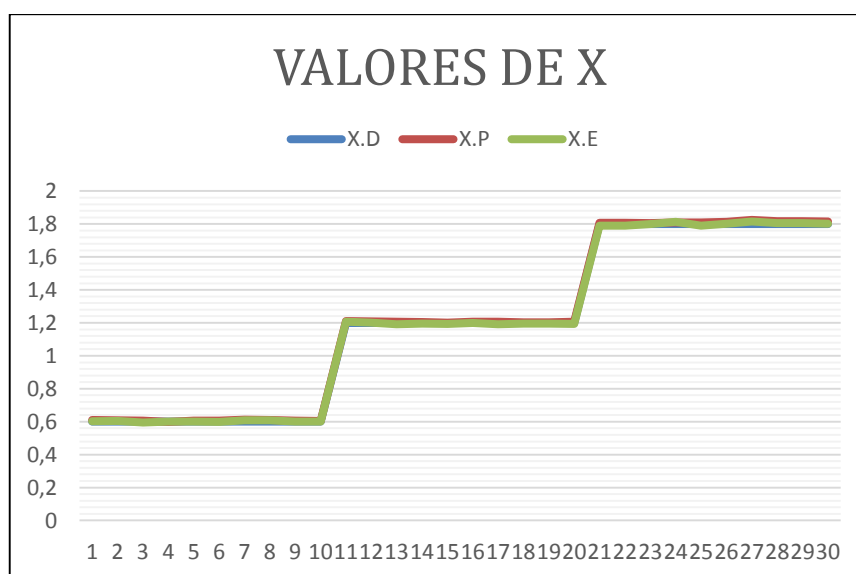


Gráfico 7-3. Medidas de posición en el eje X.

Realizado por: Pilamala, H; Hilaño, J, 2020

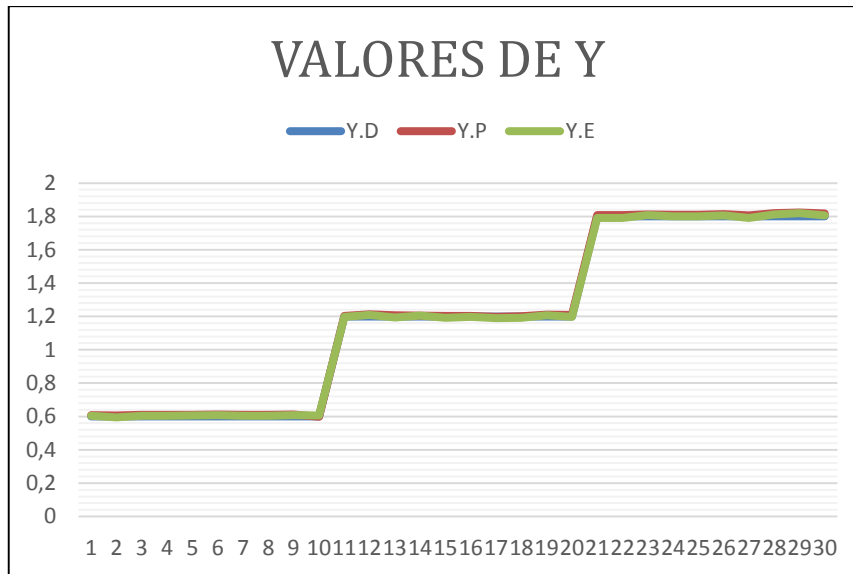


Gráfico 8-3. Medidas de posición en el eje Y.

Realizado por: Pilamala, H; Hilaño, J, 2020

3.2.3 Prueba de trayectoria en mapa con planificador PRM

La prueba realizada con respecto al primer punto (1.9, 0.24) arroja los datos de la tabla 7-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 7-3: Posiciones de la trayectoria 1 con algoritmo PRM

ALGORITMO PRM							
PLANIFICADOR		ODOMETRÍA		ERROR ABSOLUTO [m]		ERROR RELATIVO %	
X [m]	Y [m]	X _m [m]	Y _m [m]	E _x	E _y	Er _x	Er _y
0,2000	0,2000	0,2000	0,2000	0,0000	0,0000	0,0000	0,0000
0,1986	0,1940	0,2063	0,2010	0,0077	0,0070	3,8867	3,5970
0,2786	0,2787	0,2714	0,2817	0,0072	0,0029	2,5684	1,0498
0,3061	0,3193	0,3030	0,3119	0,0032	0,0074	1,0406	2,3210
0,3762	0,4023	0,3833	0,3911	0,0071	0,0112	1,8770	2,7740
0,4538	0,5128	0,4616	0,5047	0,0078	0,0080	1,7227	1,5673
0,5577	0,6058	0,5696	0,6164	0,0119	0,0106	2,1251	1,7475
0,6889	0,6388	0,7019	0,6407	0,0130	0,0019	1,8835	0,2964
0,7182	0,6327	0,7298	0,6356	0,0116	0,0029	1,6192	0,4572
0,8554	0,6341	0,8452	0,6248	0,0102	0,0094	1,1878	1,4802
0,9551	0,6439	0,9421	0,6291	0,0130	0,0148	1,3598	2,2958
1,0850	0,6790	1,0728	0,6655	0,0122	0,0135	1,1243	1,9874
1,2323	0,6537	1,2456	0,6501	0,0133	0,0036	1,0769	0,5478

1,2962	0,6226	1,3182	0,6104	0,0219	0,0123	1,6903	1,9739
1,3948	0,5774	1,4217	0,5646	0,0269	0,0128	1,9252	2,2145
1,4980	0,4943	1,4812	0,4892	0,0168	0,0051	1,1222	1,0283
1,6090	0,4362	1,6354	0,4147	0,0264	0,0215	1,6406	4,9325
1,7087	0,4194	1,7354	0,4083	0,0267	0,0111	1,5622	2,6578
1,7686	0,3832	1,7868	0,3724	0,0182	0,0107	1,0286	2,8034
1,8753	0,2819	1,8510	0,2884	0,0243	0,0065	1,2942	2,2927
1,9000	0,2400	1,8925	0,2389	0,0075	0,0011	0,3973	0,4677
			Promedio	0,0137	0,0083	1,53	1,83

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,0137 m
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,0083 m
- El promedio del error relativo % con respecto a los datos en el eje X es 1,53%
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,83%

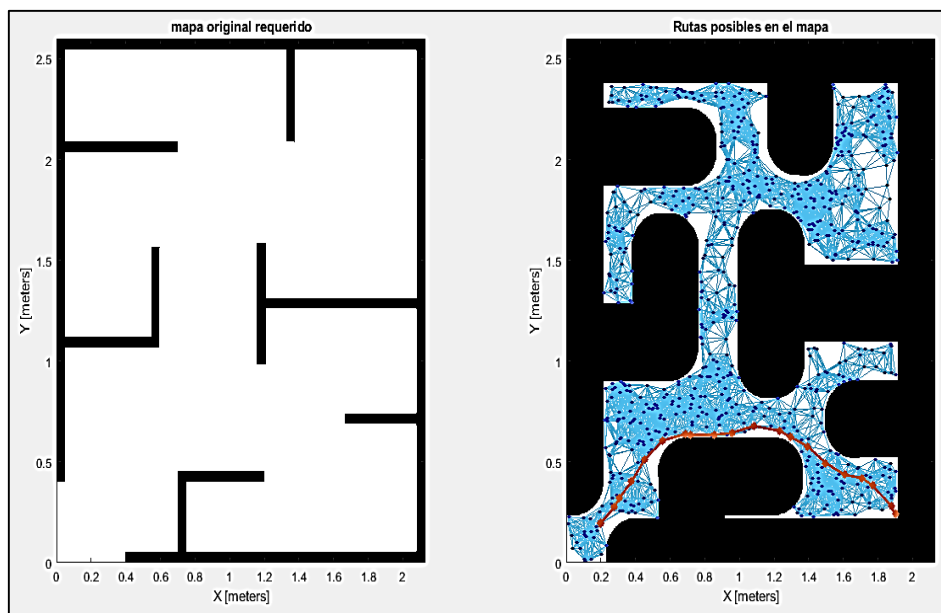


Figura 2-3: Trayectoria generada al punto 1 con planificador PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

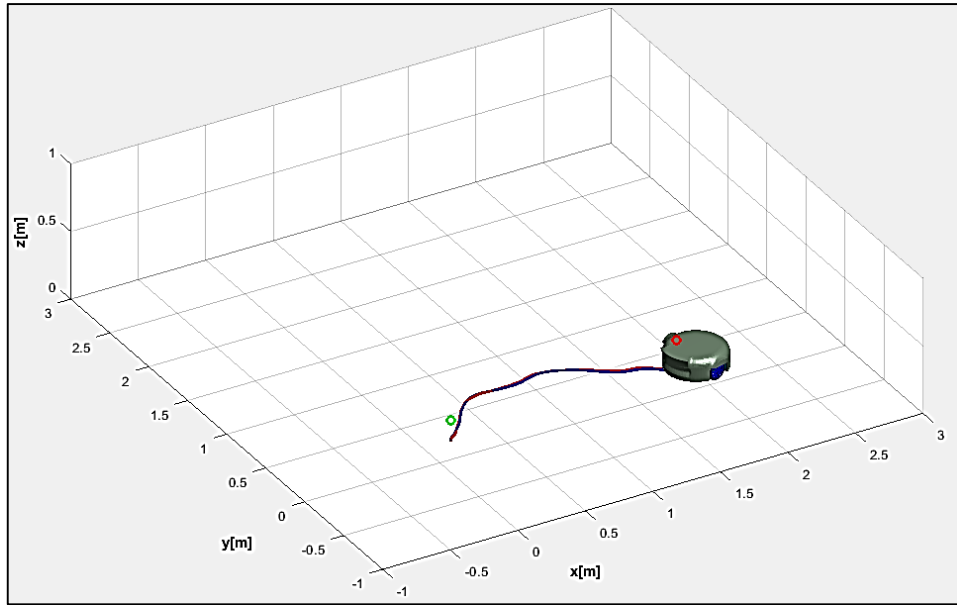


Gráfico 9-3. Trayectoria recorrida al punto 1 con planificador PRM.

Realizado por: Pilamala, H; Hilaño, J, 2020

La prueba realizada con respecto al segundo punto (1.75, 2.3) m arroja los datos de la tabla 8-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 8-3: Posiciones de la trayectoria 2 con algoritmo PRM

ALGORITMO PRM							
PLANIFICADOR		ODOMETRÍA		ERROR ABSOLUTO [m]		ERROR RELATIVO %	
X [m]	Y [m]	X _m [m]	Y _m [m]	E _x	E _y	Er _x	Er _y
0,2000	0,2000	0,2000	0,2000	0,0000	0,0000	0,0000	0,0000
0,2500	0,3528	0,2585	0,3446	0,0085	0,0083	3,4031	2,3449
0,3100	0,4768	0,3163	0,4710	0,0063	0,0058	2,0340	1,2193
0,3700	0,5687	0,3746	0,5600	0,0046	0,0087	1,2529	1,5342
0,4300	0,6388	0,4328	0,6324	0,0028	0,0064	0,6470	1,0001
0,4900	0,6971	0,4966	0,6728	0,0066	0,0242	1,3369	3,4780
0,5500	0,7536	0,5639	0,7755	0,0139	0,0219	2,5262	2,9053
0,6100	0,8184	0,6087	0,8039	0,0013	0,0145	0,2159	1,7694
0,6700	0,9017	0,6805	0,9105	0,0105	0,0088	1,5600	0,9801
0,7300	0,9953	0,7449	1,0231	0,0149	0,0278	2,0364	2,7935
0,7900	1,1025	0,7979	1,1266	0,0079	0,0241	1,0062	2,1840
0,8500	1,2723	0,8506	1,2962	0,0006	0,0240	0,0670	1,8835
0,9100	1,4909	0,9020	1,4731	0,0080	0,0179	0,8743	1,1986
0,9700	1,6170	0,9553	1,6342	0,0147	0,0172	1,5148	1,0618

1,0300	1,6802	1,0576	1,6984	0,0276	0,0182	2,6808	1,0834
1,0900	1,7279	1,1186	1,7400	0,0286	0,0121	2,6200	0,7029
1,1500	1,7668	1,1574	1,7524	0,0074	0,0144	0,6445	0,8162
1,2100	1,7986	1,2386	1,7812	0,0286	0,0175	2,3605	0,9719
1,2700	1,8270	1,2605	1,8018	0,0095	0,0252	0,7501	1,3777
1,3300	1,8554	1,3425	1,8409	0,0125	0,0144	0,9366	0,7784
1,3900	1,8875	1,3709	1,8703	0,0191	0,0172	1,3775	0,9111
1,4500	1,9270	1,4694	1,9310	0,0194	0,0040	1,3409	0,2087
1,5100	1,9813	1,5323	2,0085	0,0223	0,0272	1,4743	1,3730
1,5700	2,0812	1,5918	2,1290	0,0218	0,0477	1,3891	2,2929
1,6300	2,2008	1,6514	2,2483	0,0214	0,0475	1,3134	2,1564
1,6900	2,3000	1,7069	2,3244	0,0169	0,0244	0,9996	1,0597
1,7500	2,3000	1,7633	2,3279	0,0133	0,0279	0,7598	1,2151
			Promedio	0,0129	0,0188	1,38	1,46

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,0129 m
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,0188 m
- El promedio del error relativo % con respecto a los datos en el eje X es 1,38%
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,46%

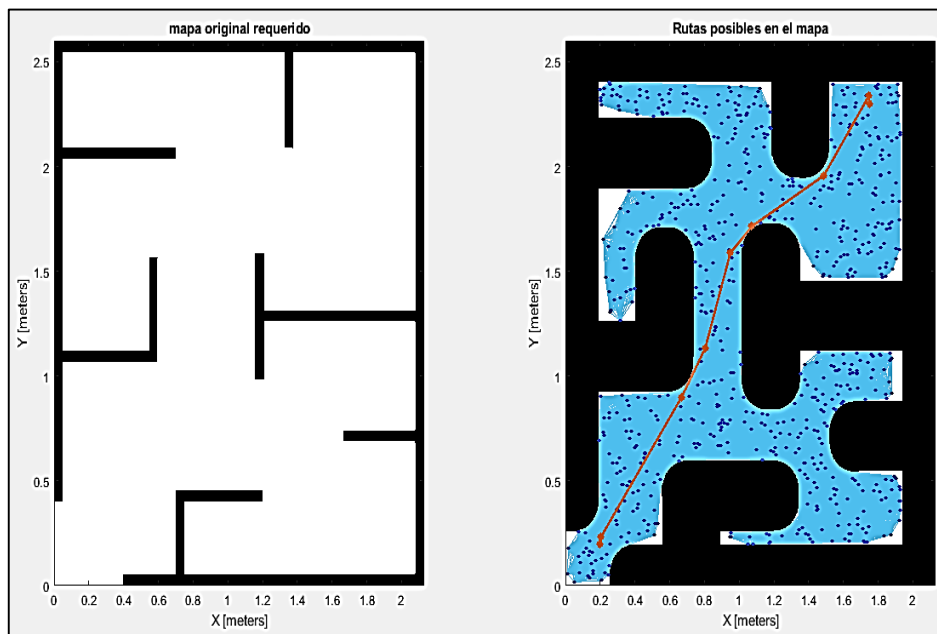


Figura 3-3: Trayectoria generada al punto 2 con planificador PRM

Elaborado por: Pilamala, H; Hilaño, J, 2020

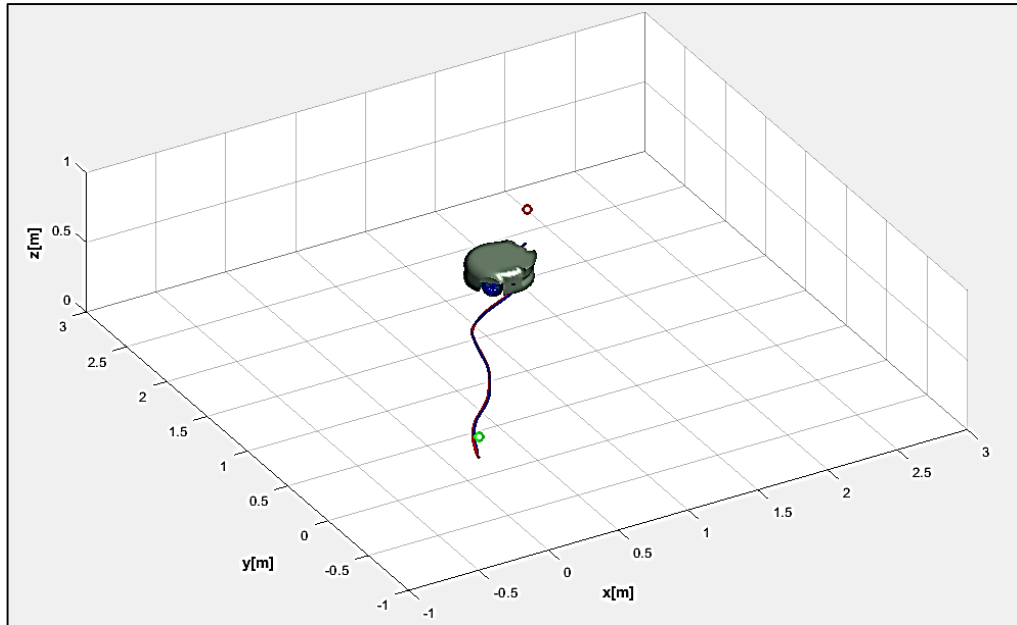


Gráfico 10-3. Trayectoria recorrida al punto 2 con planificador PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

La prueba realizada con respecto al tercer punto (0.35, 1.70) arroja los datos de la tabla 9-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 9-3: Posiciones de la trayectoria 3 con algoritmo PRM

ALGORITMO PRM							
PLANIFICADOR		ODOMETRÍA		ERROR ABSOLUTO [m]		ERROR RELATIVO %	
X [m]	Y [m]	X _m [m]	Y _m [m]	E _x	E _y	Er _x	Er _y
0,2000	0,2000	0,2000	0,2000	0,0000	0,0000	0,0000	0,0000
0,2400	0,2630	0,2409	0,2607	0,0009	0,0023	0,3589	0,8680
0,2900	0,3337	0,2944	0,3297	0,0044	0,0040	1,5172	1,1879
0,3400	0,3979	0,3470	0,3921	0,0070	0,0058	2,0549	1,4493
0,3900	0,4581	0,3956	0,4625	0,0056	0,0044	1,4359	0,9625
0,4400	0,5169	0,4501	0,5255	0,0101	0,0086	2,2955	1,6628
0,4900	0,5769	0,4989	0,5812	0,0089	0,0044	1,8163	0,7551
0,5400	0,6405	0,5480	0,6590	0,0080	0,0185	1,4778	2,8869
0,5900	0,7104	0,6018	0,7257	0,0117	0,0153	1,9915	2,1562
0,6400	0,7890	0,6509	0,8107	0,0109	0,0216	1,7107	2,7423
0,6900	0,8790	0,7077	0,9174	0,0177	0,0384	2,5632	4,3720
0,7400	0,9820	0,7563	1,0291	0,0163	0,0470	2,2053	4,7899
0,7700	1,2390	0,7590	1,2094	0,0110	0,0296	1,4300	2,3908
0,7379	1,7155	0,7434	1,7379	0,0055	0,0223	0,7518	1,3025
0,6979	1,7292	0,7094	1,7822	0,0115	0,0530	1,6523	3,0675

0,6579	1,7376	0,6404	1,8018	0,0175	0,0642	2,6574	3,6976
0,6179	1,7418	0,6197	1,7977	0,0018	0,0559	0,2983	3,2075
0,5679	1,7430	0,5548	1,7692	0,0131	0,0262	2,3057	1,5016
0,5179	1,7339	0,5081	1,7441	0,0098	0,0102	1,8905	0,5862
0,4679	1,7167	0,4596	1,7190	0,0082	0,0023	1,7591	0,1347
0,4179	1,7003	0,4081	1,6961	0,0098	0,0042	2,3361	0,2449
0,3500	1,7000	0,3567	1,6895	0,0067	0,0105	1,9284	0,6156
			Promedio	0,0089	0,0204	1,66	1,84

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,0089 m
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,0204 m
- El promedio del error relativo % con respecto a los datos en el eje X es 1,66%
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,84%

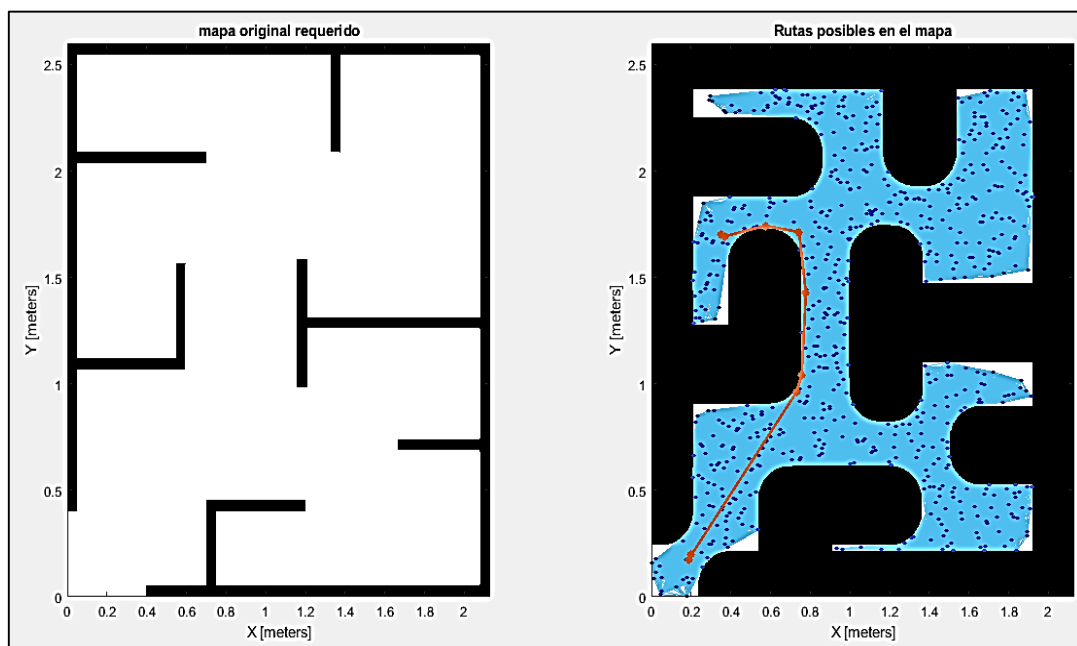


Figura 4-3: Trayectoria generada al punto 3 con planificador PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

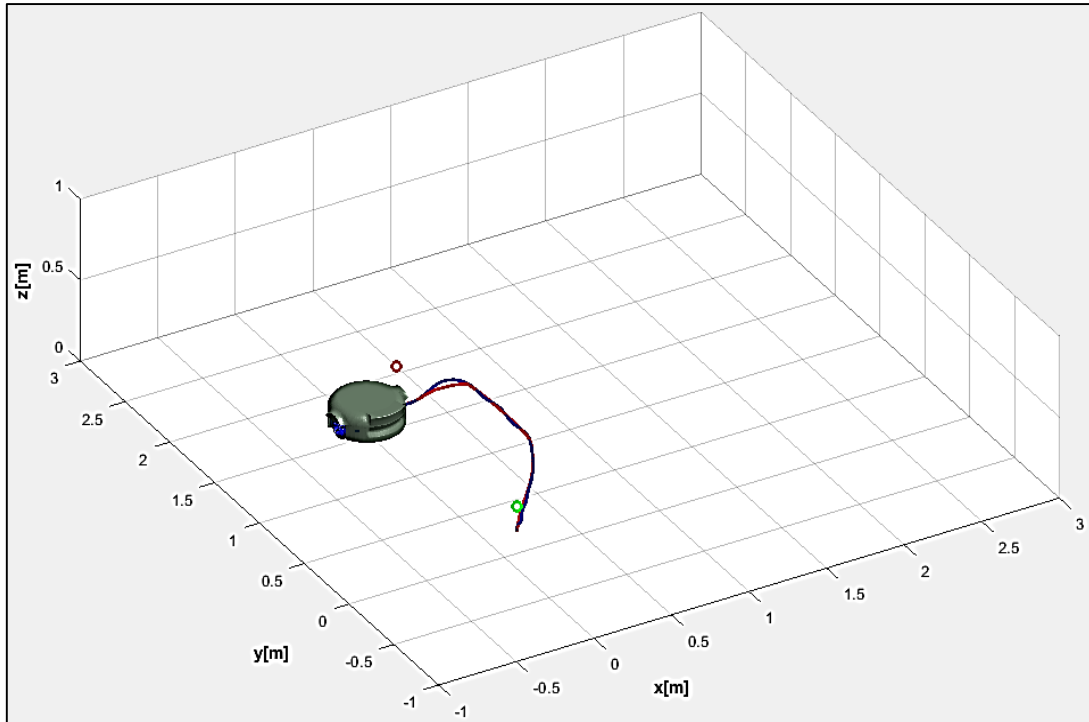


Gráfico 11-3. Trayectoria recorrida al punto 3 con planificador PRM

Realizado por: Pilamala, H; Hilaño, J, 2020

3.2.4 Prueba de trayectoria en mapa con planificador A* Híbrido

La prueba realizada con respecto al primer punto (1.9, 0.24) arroja los datos de la tabla 10-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 10-3: Posición y orientación de la trayectoria 1 con Algoritmo A* Híbrido

ALGORITMO A * HÍBRIDO											
PLANIFICADOR			ODOMETRÍA			ERROR ABSOLUTO			ERROR RELATIVO %		
X [m]	Y [m]	φ [°]	X_m [m]	Y_m [m]	φ_m [°]	E_x	E_y	E_φ	Er_x	Er_y	Er_φ
0,2000	0,2000	45,0000	0,2000	0,2000	45,0000	0,00	0,00	0,00	0,00	0,00	0,00
0,2707	0,2707	45,0000	0,2769	0,2760	44,1434	0,01	0,01	0,86	2,29	1,94	1,94
0,3414	0,3414	45,0000	0,3507	0,3476	44,1698	0,01	0,01	0,83	2,72	1,80	1,88
0,4121	0,4121	45,0000	0,4198	0,4198	44,2877	0,01	0,01	0,71	1,86	1,86	1,61
0,4869	0,4781	30,6761	0,4953	0,4804	31,0674	0,01	0,00	0,39	1,74	0,48	1,26
0,5729	0,5291	30,6761	0,5950	0,5394	30,9874	0,02	0,01	0,31	3,87	1,95	1,00
0,6589	0,5801	30,6761	0,6526	0,5712	31,2476	0,01	0,01	0,57	0,95	1,54	1,83
0,7449	0,6311	30,6761	0,7511	0,6264	30,1425	0,01	0,00	0,53	0,83	0,74	1,77
0,8309	0,6822	30,6761	0,8495	0,6864	31,8746	0,02	0,00	1,20	2,23	0,63	3,76
0,8739	0,7077	30,6761	0,8866	0,6985	30,9856	0,01	0,01	0,31	1,45	1,29	1,00
0,9697	0,7111	-26,5742	0,9893	0,6995	-26,9876	0,02	0,01	0,41	2,02	1,62	1,53

1,0630	0,6644	-26,5742	1,0871	0,6469	-26,0146	0,02	0,02	0,56	2,27	2,62	2,15
1,1563	0,6177	-26,5742	1,1617	0,6013	-26,1872	0,01	0,02	0,39	0,46	2,65	1,48
1,2496	0,5710	-26,5742	1,2662	0,5453	-27,1098	0,02	0,03	0,54	1,33	4,52	1,98
1,3430	0,5244	-26,5742	1,3273	0,5156	-26,3289	0,02	0,01	0,25	1,17	1,66	0,93
1,4363	0,4777	-26,5742	1,4542	0,4687	-27,0014	0,02	0,01	0,43	1,25	1,88	1,58
1,5296	0,4310	-26,5742	1,5358	0,4410	-27,1041	0,01	0,01	0,53	0,41	2,31	1,95
1,6229	0,3844	-26,5742	1,6400	0,3784	-27,1537	0,02	0,01	0,58	1,05	1,55	2,13
1,7162	0,3377	-26,5742	1,7216	0,3398	-26,0955	0,01	0,00	0,48	0,31	0,63	1,83
1,8095	0,2910	-26,5742	1,8187	0,2871	-26,7280	0,01	0,00	0,15	0,51	1,34	0,58
1,9000	0,2400	-45,0000	1,9184	0,2321	-44,8416	0,02	0,01	0,16	0,97	3,30	0,35
					Promedio	0,01	0,01	0,48	1,41	1,73	1,55

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,01 m
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,01 m
- El promedio del error absoluto con respecto a los datos de φ es $0,48^\circ$
- El promedio del error relativo % con respecto a los datos en el eje X es 1,41%
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,73%
- El promedio del error relativo % con respecto a los datos de φ es 1,55%

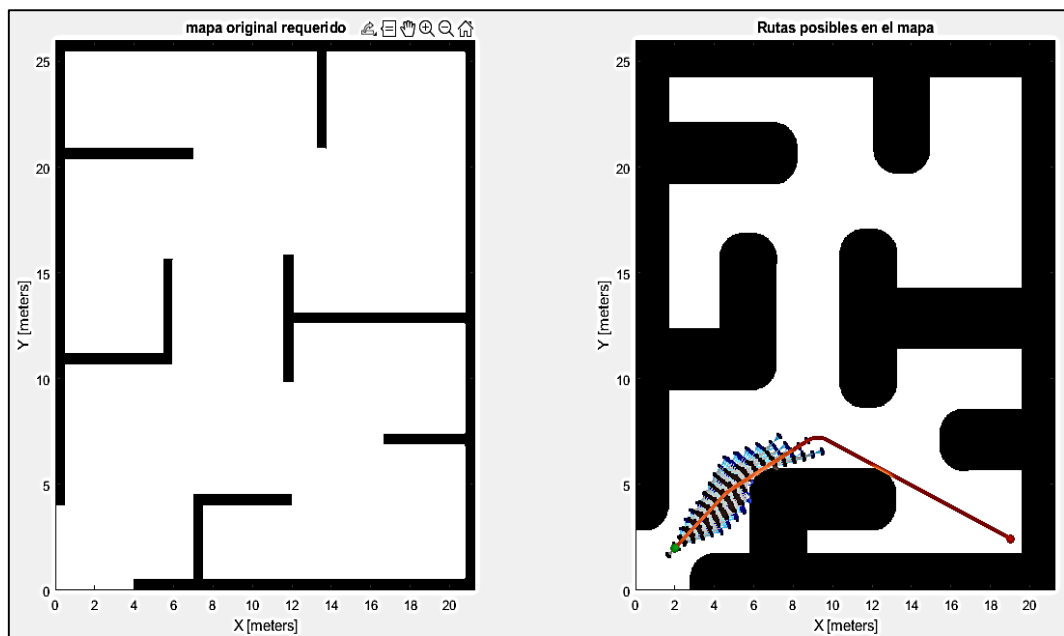


Figura 5-3: Trayectoria generada al punto 1 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

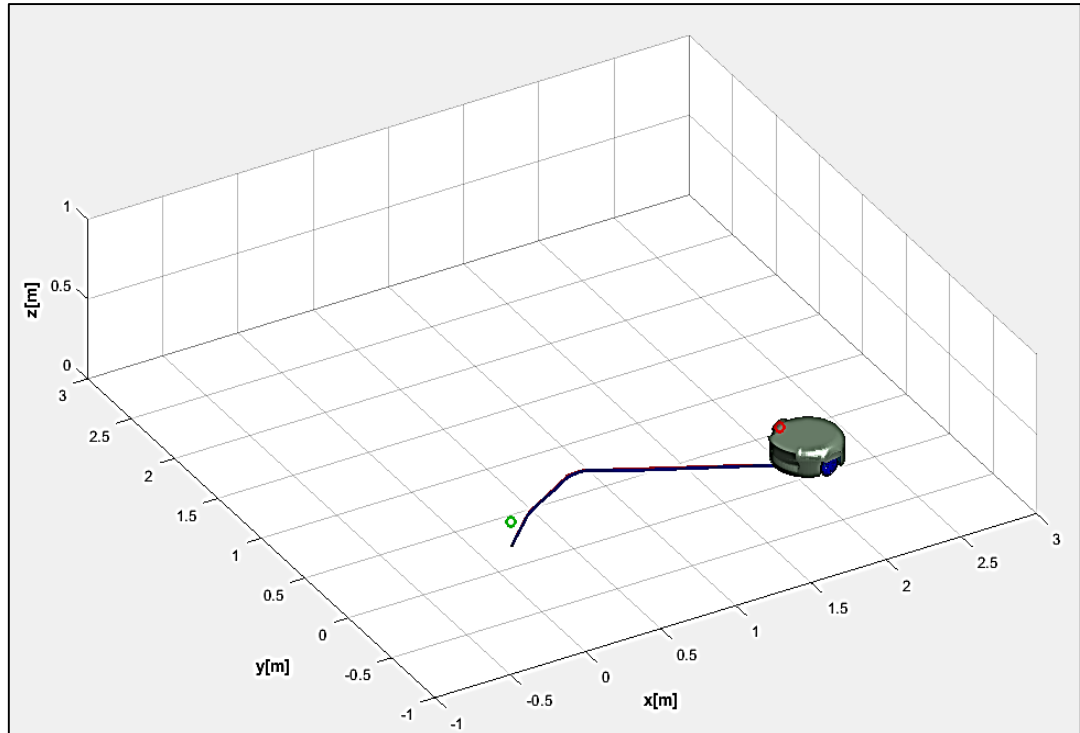


Gráfico 12-3. Trayectoria recorrida al punto 1 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

La prueba realizada con respecto al segundo punto (1.75, 2.3) arroja los datos de la tabla 11-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 11-3: Posición y orientación de la trayectoria 2 con Algoritmo A*híbrido

ALGORITMO A * HÍBRIDO											
PLANIFICADOR			ODOMETRÍA			ERROR ABSOLUTO			ERROR RELATIVO %		
X [m]	Y [m]	ϕ [°]	X_m [m]	Y_m [m]	ϕ_m [°]	E_x	E_y	E_ϕ	Er_x	Er_y	Er_ϕ
0,2000	0,2000	90,0000	0,2000	0,2000	90,0000	0,00	0,00	0,00	0,00	0,00	0,00
0,2362	0,2918	61,3521	0,2393	0,2821	60,5602	0,00	0,01	0,79	1,32	3,32	1,29
0,2842	0,3796	61,3521	0,2936	0,3677	60,0874	0,01	0,01	1,26	3,34	3,12	2,06
0,3321	0,4673	61,3521	0,3349	0,4540	60,5147	0,00	0,01	0,84	0,84	2,84	1,36
0,3800	0,5551	61,3521	0,3795	0,5451	61,9874	0,00	0,01	0,64	0,13	1,80	1,04
0,4280	0,6429	61,3521	0,4198	0,6227	61,6982	0,01	0,02	0,35	1,92	3,14	0,56
0,4912	0,7199	47,0282	0,5041	0,7308	47,5708	0,01	0,01	0,54	2,61	1,51	1,15
0,5594	0,7931	47,0282	0,5690	0,7924	47,4852	0,01	0,00	0,46	1,71	0,09	0,97
0,6276	0,8662	47,0282	0,6334	0,8602	48,5981	0,01	0,01	1,57	0,94	0,69	3,34
0,6957	0,9394	47,0282	0,7027	0,9404	48,1472	0,01	0,00	1,12	1,00	0,11	2,38
0,7489	1,0237	61,3521	0,7550	1,0328	62,5962	0,01	0,01	1,24	0,82	0,89	2,03
0,7968	1,1115	61,3521	0,8032	1,1318	62,3345	0,01	0,02	0,98	0,80	1,83	1,60
0,8448	1,1992	61,3521	0,8576	1,2331	62,5684	0,01	0,03	1,22	1,52	2,83	1,98

0,8927	1,2870	61,3521	0,9034	1,3092	61,8724	0,01	0,02	0,52	1,19	1,73	0,85
0,9350	1,3773	75,6761	0,9411	1,3924	75,3987	0,01	0,02	0,28	0,66	1,10	0,37
0,9597	1,4742	75,6761	0,9639	1,5008	75,4522	0,00	0,03	0,22	0,44	1,81	0,30
0,9844	1,5711	75,6761	0,9959	1,6028	75,5489	0,01	0,03	0,13	1,16	2,02	0,17
1,0151	1,6659	61,3521	1,0294	1,6910	61,0871	0,01	0,03	0,27	1,41	1,51	0,43
1,0909	1,7291	32,7042	1,1199	1,7446	32,4749	0,03	0,02	0,23	2,66	0,90	0,70
1,1750	1,7832	32,7042	1,1843	1,7789	32,1204	0,01	0,00	0,58	0,79	0,24	1,79
1,2592	1,8372	32,7042	1,2396	1,8113	32,6547	0,02	0,03	0,05	1,55	1,41	0,15
1,3433	1,8912	32,7042	1,3517	1,8857	33,1214	0,01	0,01	0,42	0,63	0,29	1,28
1,4358	1,9285	18,3803	1,4147	1,9153	18,8824	0,02	0,01	0,50	1,47	0,68	2,73
1,5106	1,9926	48,3791	1,5278	2,0034	48,6987	0,02	0,01	0,32	1,14	0,54	0,66
1,5981	2,0910	48,3791	1,6043	2,1030	49,4587	0,01	0,01	1,08	0,39	0,57	2,23
1,6855	2,1894	48,3791	1,7063	2,2284	48,8954	0,02	0,04	0,52	1,23	1,78	1,07
1,7500	2,3000	90,0000	1,7573	2,3273	90,8746	0,01	0,03	0,87	0,42	1,19	0,97
					Promedio	0,01	0,02	0,63	1,19	1,40	1,24

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,01 m
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,02 m
- El promedio del error absoluto con respecto a los datos de φ es $0,63^\circ$
- El promedio del error relativo % con respecto a los datos en el eje X es 1,19%
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,40%
- El promedio del error relativo % con respecto a los datos de φ es 1,24%

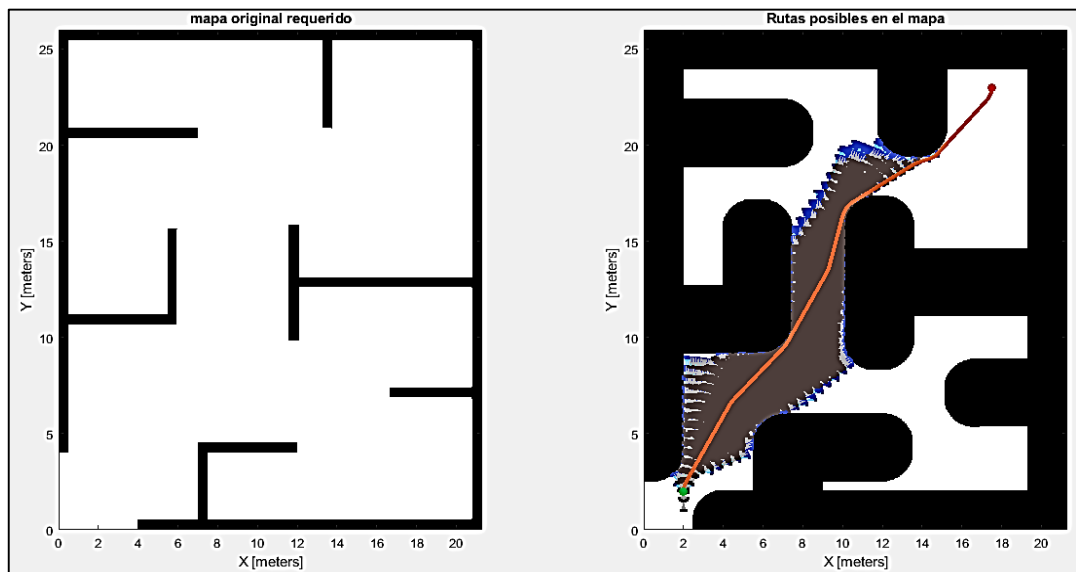


Figura 6-3: Trayectoria generada al punto 2 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

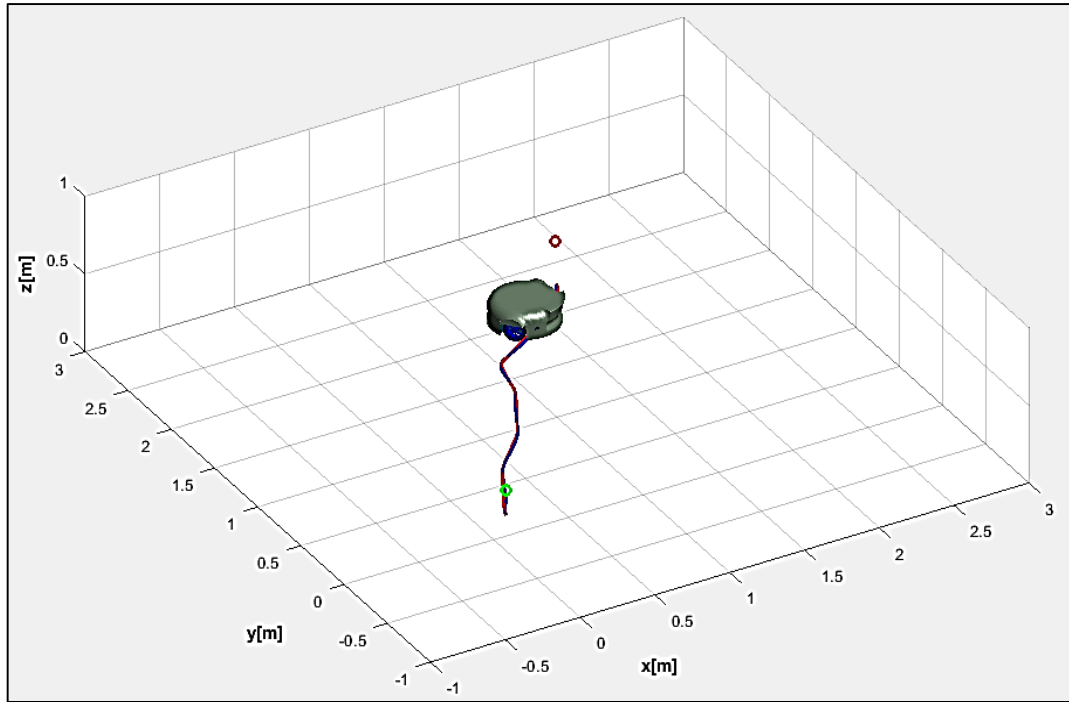


Gráfico 13-3. Trayectoria recorrida al punto 2 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

La prueba realizada con respecto al segundo punto (0.35, 1.7) arroja los datos de la tabla 12-3, del total de puntos de la trayectoria se ha elegido una muestra significativa para su análisis. Los valores de referencia dados por el planificador son comparados con los valores calculados en la odometría del sistema.

Tabla 12-3: Posición y orientación de la trayectoria 3 con algoritmo A* Híbrido.

ALGORITMO A * HÍBRIDO											
PLANIFICADOR			ODOMETRÍA			ERROR ABSOLUTO [m]			ERROR RELATIVO %		
X [m]	Y [m]	ϕ [°]	X_m [m]	Y_m [m]	ϕ_m [°]	E_x	E_y	E_ϕ	Er_x	Er_y	Er_ϕ
0,2000	0,2000	90,0000	0,2000	0,2000	90,0000	0,00	0,00	0,00	0,00	0,00	0,00
0,2362	0,2918	61,3521	0,2388	0,2802	60,1021	0,00	0,01	1,25	1,12	3,99	2,04
0,2842	0,3796	61,3521	0,2918	0,3696	61,5750	0,01	0,01	0,22	2,69	2,63	0,36
0,3321	0,4673	61,3521	0,3269	0,4452	61,2587	0,01	0,02	0,09	1,56	4,73	0,15
0,3800	0,5551	61,3521	0,3846	0,5563	60,0587	0,00	0,00	1,29	1,19	0,21	2,11
0,4280	0,6429	61,3521	0,4325	0,6324	59,8603	0,00	0,01	1,49	1,06	1,63	2,43
0,4912	0,7199	47,0282	0,4837	0,7019	46,2885	0,01	0,02	0,74	1,54	2,50	1,57
0,5594	0,7931	47,0282	0,5461	0,7669	46,7892	0,01	0,03	0,24	2,38	3,29	0,51
0,6276	0,8662	47,0282	0,6217	0,8515	48,4365	0,01	0,01	1,41	0,94	1,70	2,99
0,6957	0,9394	47,0282	0,6844	0,9184	47,8551	0,01	0,02	0,83	1,63	2,23	1,76
0,7489	1,0237	61,3521	0,7366	0,9881	62,5091	0,01	0,04	1,16	1,65	3,47	1,89
0,7611	1,1217	90,0000	0,7549	1,1309	92,6224	0,01	0,01	2,62	0,82	0,83	2,91
0,7611	1,2217	90,0000	0,7504	1,2000	92,1480	0,01	0,02	2,15	1,41	1,77	2,39

0,7611	1,3217	90,0000	0,7702	1,3144	87,4551	0,01	0,01	2,54	1,20	0,55	2,83
0,7611	1,4217	90,0000	0,7745	1,4050	88,4992	0,01	0,02	1,50	1,76	1,17	1,67
0,7611	1,5217	90,0000	0,7658	1,4967	88,7233	0,00	0,02	1,28	0,61	1,64	1,42
0,7611	1,6217	90,0000	0,7697	1,6347	91,5783	0,01	0,01	1,58	1,13	0,81	1,75
0,7611	1,6717	90,0000	0,7710	1,6581	92,2697	0,01	0,01	2,27	1,30	0,81	2,52
0,7152	1,7558	147,2958	0,7275	1,7335	146,3885	0,01	0,02	0,91	1,73	1,27	0,62
0,5942	1,7580	-165,9584	0,5804	1,7650	-167,2149	0,01	0,01	1,26	2,32	0,40	0,76
0,4723	1,7275	-165,9584	0,4556	1,7149	-164,0157	0,02	0,01	1,94	3,53	0,73	1,17
0,3500	1,7000	180,0000	0,3525	1,6885	178,0654	0,00	0,01	1,93	0,71	0,67	1,07
					Promedio	0,01	0,02	1,30	1,47	1,68	1,59

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es 0,01 m.
- El promedio del error absoluto con respecto a los datos en el eje Y es 0,02 m.
- El promedio del error absoluto con respecto a los datos de φ es $1,30^\circ$.
- El promedio del error relativo % con respecto a los datos en el eje X es 1,47%.
- El promedio del error relativo % con respecto a los datos en el eje Y es 1,68%.
- El promedio del error relativo % con respecto a los datos de φ es 1,59%.

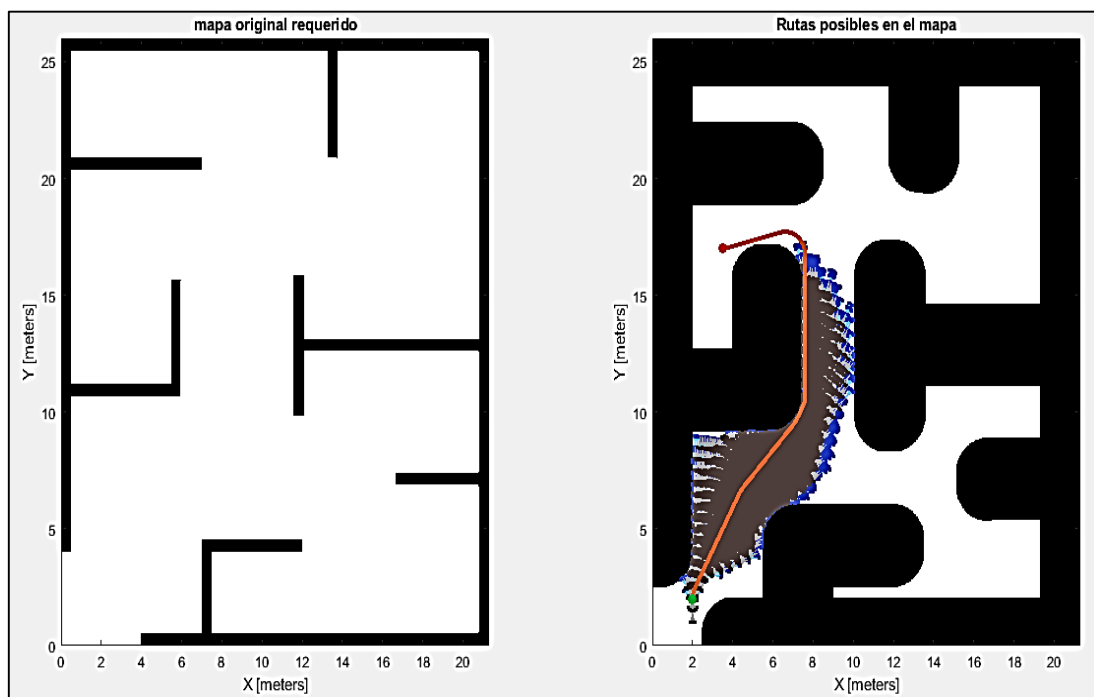


Figura 7-3: Trayectoria generada al punto 3 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

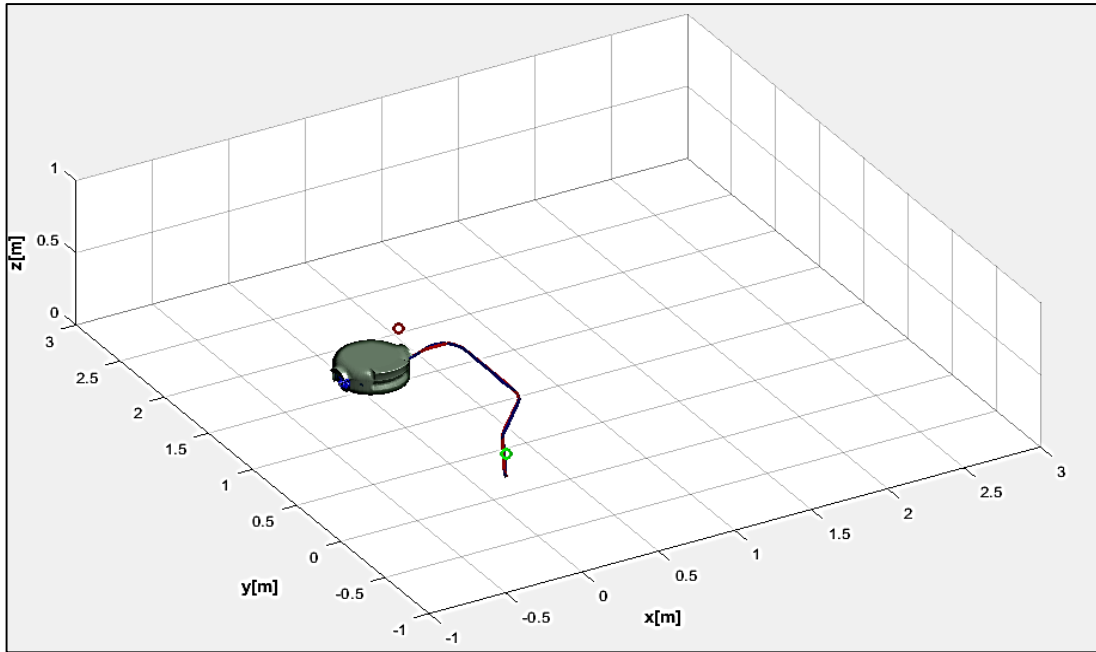


Gráfico 14-3. Trayectoria recorrida al punto 3 con planificador A* Híbrido

Realizado por: Pilamala, H; Hilaño, J, 2020

3.2.5 *Análisis de errores de posicionamiento*

En las dos secciones anteriores se ha realizado pruebas para que el robot debe llegar a tres puntos de destino como se muestra en la tabla 1-3 lo que ha generado tres trayectorias en las cuales se ha obtenido las posiciones del robot y se ha calculado su error absoluto respecto a las posiciones deseadas obtenidas mediante los algoritmos de planificación de rutas. En la tabla 13-3 se muestra el promedio de los errores de posicionamiento del robot para cada prueba realizada

Tabla 13-3: Promedio de errores absolutos de posicionamiento.

Algoritmo Planificador	Trayectoria	Promedio de errores absolutos	
		\bar{E}_x [m]	\bar{E}_y [m]
PRM	Prueba Punto 1	0.0137	0.0083
	Prueba Punto 2	0.0129	0.0188
	Prueba Punto 3	0.0089	0.0204
A* Híbrido	Prueba Punto 1	0.01	0.01
	Prueba Punto 2	0.01	0.02
	Prueba Punto 3	0.01	0.02
Promedio total de errores absolutos[m]		0.011	0.016
Promedio total de errores absolutos [cm]		1.1	1.6

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

- El promedio del error absoluto con respecto a los datos en el eje X es ± 1.1 cm.
- El promedio del error absoluto con respecto a los datos en el eje Y es ± 1.6 cm.

3.3 Pruebas de rendimiento del prototipo

3.3.1 Pruebas de duración de batería

El tiempo de funcionamiento de prototipo fue calculado teóricamente en el apartado 2.10.7 donde se determinó el consumo sin carga y a carga nominal. Las medidas fueron tomadas en el transcurso de las pruebas realizando desplazamiento continuo. De acuerdo con la tabla 14-3 se determinó la duración de batería aproximada teniendo en consideración que las baterías tipo lipo trabajan de 3 v a 4.5 v es casos extremos, es decir su carga nunca debe aproximarse a estos límites, se considera una descarga optima de alrededor de 3.5 v.

Tabla 14-3: Tiempo de descarga de batería.

Nº de Prueba	Tiempo de descarga 3.5v ± 0.15 v
1	43
2	45
3	44
4	47
5	50
PROMEDIO	45,8

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

3.3.2 Pruebas de tiempo de ejecución de trayectorias

Dentro de la programación de Matlab se ejecuta un ciclo for en la cual está el control de seguimiento de trayectoria, se envían los datos de velocidad de Matlab a Arduino y también se lee los datos de posición y orientación que Arduino envía a Matlab.

El tiempo de ejecución de cada repetición que genera el ciclo for es medido en Matlab mediante la función tic que registra el tiempo inicial en cero y la función toc que utiliza el valor registrado para calcular el tiempo transcurrido (MathWorks, 2020). En la tabla 15-3 se registran los tiempos de ejecución de las tres trayectorias recorridas por el robot hacia los puntos de destino al usar los algoritmos de planificación de rutas PRM y A* Híbrido. Se ha usado para todo este proceso una laptop hp pavilion dv6 con procesador core i3 y una memoria ram de 4gb.

Tabla 15-3: Tiempo de ejecución trayectorias

Posición deseada [m]		Tempo de ejecución [s]	
x	y	Algoritmo PRM	Algoritmo A* Híbrido
1.75	2.3	31.5142	30.2496
1.90	0.24	24.9696	22.9578
0.35	1.70	19.5462	24.9434

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

CAPÍTULO IV

4. EVALUACIÓN ECONÓMICA

4.1 Análisis de costos

Los costos del prototipo realizado se presentan en la tabla 16-3, los costos abarcan recursos, materia prima y costos de producción, teniendo en cuenta una producción por unidad de manera local.

Tabla 1-4: Costos del prototipo.

Cantidad	Componente	Valor Unitario	Valor Total
Sistema Electrónico			
1	Arduino Nano V3.0	10.00	10.00
1	L298N Driver para motor DC	5.00	5.00
2	Regulador de voltaje L7805	1.20	2.40
2	Regulador de voltaje L7812	1.20	2.40
1	Batería Lipo 4 celdas 14.8V 1500mA	36	36
2	Capacitor 1uF	0.20	0.40
2	Capacitor 0.33uF	0.20	0.40
1	Conector Tx60	3.00	3.00
1	Tester batería Lipo	5.00	5.00
1	Tira de espadines macho	0.40	0.40
1	Tira de espadines hembra	1.00	1.00
5	Cable molex	0.60	3.00
1	Switch	0.50	0.50
3	Borneras de dos pines	0.25	0.75
1	Baquelita 7x10 cm	1.25	1.25
Sistema Mecánico			
2	Motor DC 6-12 V Tsiny con encoders	25.88	51.76
1	Kit 2 llantas diámetro 65mm + 2 soportes + 2 acoples 5mm	25.00	25.00
1	Impresión 3D de la estructura	80.00	80.00
2	Rueda loca mini 3PI N20	3.00	6.00
Comunicación			
1	Módulo “Holybro Telemetry Radio”	40.00	40.00
Costos adicionales			
1	Frasco de silicona de ventana	7.00	7.00
TOTAL			\$ 281.26

Fuente: Propia

Realizado por: Pilamala, H; Hilaño, J, 2020

CONCLUSIONES

- Se cumplió con el desarrollo de un robot móvil autónomo en configuración diferencial para localización en interiores usando odometría, dentro de un mapa conocido en el que el robot mediante un planificador escoge la mejor ruta hacia un punto deseado en el mapa.
- Se investigó y diseñó una morfología para el robot de tal forma que permita cumplir con los objetivos planteados y que sea funcional en su ambiente planteado de acuerdo con sus requerimientos.
- Se analizó la estimación de la posición y orientación en la simulación mediante la integral numérica mientras que para la aplicación del sistema real se usó la estimación de la localización mediante la odometría.
- Se determinó que la alta resolución de los encoders utilizados además de su salida de doble canal mejoran el cálculo de la localización del robot mediante odometría debido a que para cada tiempo de muestreo generalmente corto en ms se sabe con exactitud el número de pulsos digitales que ha generado el encoder y por ende la distancia recorrida por cada llanta en ese lapso de tiempo.
- Se determinó que los errores no sistemáticos del prototipo afectan en menor medida dentro de su ambiente de prueba escogido sobre todo porque el tablero de madera MDF que se ha usado es plano y uniforme por lo que las llantas del robot no presentan fricción con la superficie.
- Se implementó un controlador PID de velocidad para que los motores lleguen a sus consignas de velocidad lo cual mantiene al sistema estable sin embargo el tiempo de establecimiento rodea los 8 segundos por lo que el robot móvil se demora más en llegar al punto de destino.
- Se determinó que el planificador PRM en ocasiones obtiene ángulos agudos y rectos dentro de su trayectoria lo que provoca cambios bruscos de velocidad y el incremento de la corriente con el riesgo de dañar el equipo, mientras que el planificador A* híbrido realiza curvas suaves en sus trayectorias lo que facilita el desplazamiento del robot.
- Se usó la odometría para localizar al robot en tres diferentes trayectorias y de acuerdo a la tabla 13-3 se obtuvo un promedio de error absoluto de ± 1.1 cm con respecto al posicionamiento en el eje X, y un error de ± 1.6 cm en el eje Y.
- Se analizó en las pruebas el tiempo que dura el robot en llegar a los puntos deseados, concluyendo que con el planificador A* híbrido el tiempo fue menor que con el planificador PRM debido a la continuidad y suavidad en las curvas de la trayectoria generada.

RECOMENDACIONES

- Los motores deben estar alineados en torno a un mismo eje con el fin de que no esté uno más delante que el otro ya que esto provocaría desviaciones cuando el robot se mueva en una trayectoria.
- Las llantas del prototipo deben tener una cubierta lisa y no debe ser hueca en su interior para que tengan mejor agarre con la superficie de contacto.
- El prototipo debe ser probado en un piso plano y sin fricción para que el robot no patine o resbale.
- Para la planificación de rutas se debe usar la versión Matlab R2020a y no anteriores a esta ya que esta versión cuenta con el Toolbox “Robotics System Toolbox™”.
- Se recomienda inflar el mapa con un valor cercano a la medida del ancho del robot con el fin de sobredimensionar los obstáculos y que el robot no choque con estos.
- Para futuras investigaciones se recomienda la incorporación de sensores exteroceptivos para incluir el reconocimiento del entorno y obstáculos móviles dentro de rutas de mapas conocidos.
- Se recomienda la utilización de la técnica de fusión sensorial para mejorar la estimación odométrica para robots autónomos.
- Se recomienda la creación de una interface visual para que el robot pueda ser usado por más usuarios u operadores dentro de la investigación.

GLOSARIO

Encoder: Es un dispositivo de detección que proporciona una respuesta, los encoders convierten el movimiento en una señal eléctrica que puede ser leída por algún tipo de dispositivo de control en un sistema de control de movimiento (Corona Ramírez, et al. 2014, p.78).

Cinemática: La cinemática de un robot es el estudio de los movimientos de un robot. En un análisis cinemático la posición, velocidad y aceleración de cada uno de los elementos del robot son calculados sin considerar las fuerzas que causan el movimiento (Barrientos, et al. 2007, p. 93).

Odometría: Método que se utiliza para estimar la posición relativa de un robot móvil (Arellano, 2015, p.24).

Planificador: Algoritmo que permite la implementación de métodos de planificación de rutas o trayectorias (Preciado, 2018,p. 29).

Robot Móvil: Robot con capacidad de desplazamiento en su ambiente (Baturone, 2005,p. 28).

BIBLIOGRAFÍA

AGUILERA, M. et al. "Diseño y Control de robot Móviles. Nuevo Laredo: Instituto Tecnológico de Nuevo Laredo". *Asociación Mexicana de Mecatronica A.C.*, (2007),(México) pp. 1-7

AMAZON, Encoder Motor de engranaje de metal 12 V CC de alta velocidad 130 RPM con codificador para impresoras Arduino y 3D: Home Improvement. [blog]. [Consulta: 16 diciembre 2020]. Disponible en: <https://www.amazon.com/Encoder-Gearmotor-130RPM-Arduino-Printers/dp/B07GNGQ24C>.

ANZUREZ, J. et al, "Estabilidad de Sistemas No-lineales: Sistema de Nivel de Líquidos de Dos Tanques Interconectados" ITSON [En línea], 2008, Mexico vol. 5, no. 2, pp. 7.

ARDUINO, Arduino-Compare. [blog]. [Consulta: 16 diciembre 2020]. Disponible en: <https://www.arduino.cc/en/products/compare..>

ARELLANO ZEA, Luis Alberto. Diseño e implementación de un robot móvil con Control de trayectoria mediante principios odométricos. [En línea] (Trabajo de titulación), Universidad Nacional Mayor de San Marcos, Peru, 2015, pp. 17-37. [Consulta: 1 marzo 2020]. Disponible en: <http://cybertesis.unmsm.edu.pe/handle/cybertesis/4440>.

BARRIENTOS, A., PEÑÍN, L.F. & BALAGUER, C. Fundamentos de robótica. S.l.: McGraw-Hill. 2007. ISBN 978-84-481-5636-7, pp. 7-93

BATURONE, A.O. Robótica: Manipuladores y Robots Móviles. S.l.: Marcombo. 2005. ISBN 978-84-267-1313-1, pp. 8-41.

BOHÓRQUEZ BENDEZÚ, J. & MACHUCA MINES, J. "Diseño de un sistema de control de posición de un robot móvil omnidireccional". *INVESTIGA UTP* [en línea], 2018, (Perú), p. 24. [Consulta: 4 junio 2020]. Disponible en: <http://repositorio.utp.edu.pe/handle/UTP/964>.

CHAPMAN, S. *MÁQUINAS ELÉCTRICAS*. McGraw-Hill Interamericana de España . 2012, ISBN 978-970-10-4947-1, p. 345.

CHIMARRO MAILA, Maricela Nohemí & VILLAGÓMEZ TUTIVEN, Elena Paola. Construcción de un prototipo de control de iluminación de una casa, utilizando radiofrecuencia o

mensajes GSM que incluye un sistema de alarma con módulo GSM para enviar mensajes de auxilio. [en línea] (Trabajo de Titulación). Escuela Politécnica Nacional, Ecuador. 2014. p. 15. [Consulta: 20 noviembre 2020]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/8494>.

CORONA RAMÍREZ, Leonel. et al. *Sensores y actuadores: aplicaciones con Arduino*. Ascapotzalco :Larousse - Grupo Editorial Patria, 2014. pp. 2-172.

CORTÉS, F.R. *MATLAB: aplicado a la robótica y mecatrónica*. Ciudad de México: Alfaomega, 2012. ISBN 978-607-707-357-4, p. 6.

DORF, Richard. & BISHOP, Robert. *Sistemas de control moderno*. Madrid: Pearson Educación, 2005. ISBN 978-84-205-4401-4, p. 312.

FRÍAS MOYÓN, Jefferson Ramiro. Implementación de un robot móvil controlado de forma remota, aplicado al monitoreo del edificio de la FIE en tiempo real utilizando realidad aumentada. [en línea] (Trabajo de Titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2018. pp. 21-70. [Consulta: 11 diciembre 2020]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/9212>.

GEEK FACTORY *Módulo L298N puente H driver motores*. [blog]. [Consulta: 16 diciembre 2020]. Disponible en: <https://www.geekfactory.mx/tienda/modulos-para-desarrollo/modulo-l298n-puente-h-driver-motores/>.

GETFPV. Holybro 100mW Transceiver Telemetry Radio Set V3 (915Mhz). [blog], [Consulta: 17 diciembre 2020]. Disponible en: <https://www.getfpv.com/holybro-100mw-transceiver-telemetry-radio-set-v3-915mhz.html>.

GONZÁLEZ CÁRDENAS, G. & SILVA GÓMEZ, F. "Diseño e implementación de una Tarjeta de Desarrollo con profundización en desarrollo de aplicación de Touch Sensing". *11th Latin American and Caribbean Conference for Engineering and Technology* [en línea], 2013, (Mexico). [Consulta: 11 diciembre 2020]. Disponible en: <http://www.laccei.org/LACCEI2013-Cancun/RefereedPapers/RP157.pdf>

GONZÁLEZ GAVILANES, Henry Fabricio & CARRILLO TRUJILLO, María Gabriela, Implementación de un prototipo de robot sembrador de papa en terrenos sin inclinación para pequeños productores. [en línea] (Trabajo de Titulación). Escuela Superior Politécnica de

Chimborazo, Riobamba, Ecuador. 2019. pp. 14-33. [Consulta: 11 diciembre 2020]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/13686>.

GORDILLO, F., "Estabilidad de Sistemas No Lineales Basada en la Teoría de Liapunov". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 6, no. 2, (2009), (España) pp. 5-16. ISSN 1697-7912. DOI 10.1016/S1697-7912(09)70088-3.

HART, P.E., NILSSON, N.J. & RAPHAEL, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, (1968), (United States of America) pp. 100-107. ISSN 2168-2887. DOI 10.1109/TSSC.1968.300136.

HERRERA ROMAN, Danny Francisco, & TOPA GAVILEMA, Alex Omar, Desarrollo de un algoritmo basado en visión artificial para la detección y corrección de errores de odometría aplicados para tareas de locomoción autónoma. [en línea](Trabajo de Titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2018. p. 4 [Consulta: 4 marzo 2020]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/14481>.

KUO, B.C., *Sistemas de control automático*. 7ª. Mexico: Pearson Educación, 1996, ISBN 978-968-880-723-1, p. 2.

LUENGAS CONTRERAS, Lely. et al. "Caracterización de unidades de medición inercial en estática y dinámica". *Ingenio Magno* [en línea], 2017. (Colombia) 8(1). p.95. [Consulta: 12 diciembre 2020]. Disponible en: <https://es.scribd.com/document/471618168/Dialnet-CaracterizacionDeUnidadesDeMedicionInercialIMUsEnE-6881752-1-pdf>.

MARÍN, Leonardo. et al, "Localización asistida por GPS para robots móviles en configuración Ackermann de recursos limitados". *XIV Jornadas Automática*, 2013. pp. 2.

MATHWORKS. *Crear cuadrícula de ocupación con valores binarios* [blog]. [Consulta: 15 diciembre 2020]. Disponible en: https://la.mathworks.com/help/nav/ref/binaryoccupancymap.html?searchHighlight=BinaryOccupancyMap&s_tid=srchtitle.

MATHWORKS. *Hojas de ruta probabilísticas* [blog]. [Consulta: 15 diciembre 2020]. Disponible en: <https://la.mathworks.com/help/robotics/ug/probabilistic-roadmaps-prm.html>.

MATHWORKS. *Hybrid A* path planner* [blog]. [Consulta: 16 diciembre 2020]. Disponible en: https://la.mathworks.com/help/nav/ref/plannerhybridastar.html?searchHighlight=plannerHybridAStar&s_tid=srchtitle#mw_00a594aa-d19a-4473-b21a-4dd80452070f.

MATHWORKS *Start stopwatch timer* [blog]. [Consulta: 30 diciembre 2020]. Disponible en: <https://la.mathworks.com/help/matlab/ref/tic.html>.

MECAFENIX. *Sensor inercial o Sensor IMU.* [blog]. [Consulta: 5 junio 2020]. Disponible en: <https://www.ingmecafenix.com/automatizacion/sensores/sensor-inercial/>.

MOBILE INDUSTRIAL ROBOTS *MiR200TM Un robot móvil autónomo más potente.* [blog]. [Consulta: 11 diciembre 2020]. Disponible en: <https://www.mobile-industrial-robots.com/es/solutions/robots/mir200/>.

OJEDA MORENO, Miguel Antonio.; & ZÚÑIGA, Eduardo. "Encoder magnético para mediciones angulares de alta precisión" *Pistas Educativas* [en línea], 2018, (México) 38(120), p. 790. [Consulta: 11 diciembre 2020]. ISSN 2448-847X. Disponible en: <http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/582>.

OGATA, K. *Problemas de ingeniería de control utilizando Matlab.* Madrid- España : Pearson Educación, 1998. ISBN 978-84-8322-046-7, p.1.

OGATA, K., *Ingeniería de control moderna.* 5ª ed. Madrid- España: Pearson Educación, 2003 ISBN 978-84-205-3678-1. pp. 2-7.

PAREDES VINUEZA, Marjorie Nataly. Implementación de un prototipo de WSN con nodos inteligentes para el sistema de riego aplicado a la agricultura de precisión para el CER – ESPOCH. [En línea] (Trabajo de Titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2017. pp. 52-53. [Consulta: 12 diciembre 2020]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/7956>.

PRECIADO AGUILAR, Juan David. Planificación de trayectorias de robots móviles de diferentes arquitecturas en entornos dinámicos. [en línea] (Trabajo de Titulación). Universidad Militar Nueva Granada, Bogotá, Colombia. 2018. pp. 43. [Consulta: 19 diciembre 2020]. Disponible en: <http://repository.unimilitar.edu.co/handle/10654/18006>.

REYES REYES, E. & REYES CORTES, J. "Modelado dinámico y control de un robot móvil con ruedas tipo: triciclo clásico". *Mecánica Teórica* [en línea], 2018, (México): pp. 7. [Consulta: 4 junio 2020]. ISBN 2448-5551. Disponible en: <http://somim.org.mx/memorias/memorias2018/mecteo.html>.

REYES CORTÉS, Fernando. *Robótica - control de robots manipuladores*. México: Alfaomega Grupo Editor, 2011. ISBN 978-607-707-307-9, p. 83.

ROMERO BARCOJO, A. *Encoder IHMC Public Cmaps* [blog]. [Consulta: 11 diciembre 2020]. Disponible en: <https://www.yumpu.com/es/document/view/36695151/encoder-ihmc-public-cmaps-2>.

SEDIGHI, S.et al. "Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications". *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*. (2019) pp. 570-575. DOI 10.1109/ICCAR.2019.8813752.

SEN, D. et al. *RFID for Energy & Utility Industries* Oklahoma, United States of America: PennWell Books, 2009. ISBN 978-1-59370-105-5. p. 15.

SICILIANO, Bruno; et al. *Robotics: Modelling, Planning and Control*. London : Springer Science & Business Media, 2010. ISBN 978-1-84628-641-4, pp. 210-542.

SILVA ORTIGOZA, Ramón, et al. "Modelado, simulación y construcción de un robot móvil de ruedas tipo diferencial". *Latin-American Journal of Physics Education*, vol. 4, no. 3 (2010), pp. 39.

SILVA ORTIGOZA, Ramón, et al. "Una panorámica de los robots móviles". *TELEMATIQUE*, vol. 6, no. 3(2007), pp. 1-14.

SOLIDWORKS. *My SolidWorks* [blog]. [Consulta: 12 diciembre 2020]. Disponible en: <https://my.solidworks.com/>.

TZAFESTAS, Spyros. *Introduction to Mobile Robot Control* Athens, Greece: Elsevier. ISBN 978-0-12-417103-9, pp. 31-491.

VACA PAREDES, Wilson Andrés. Robot móvil para investigación en algoritmos de planeamiento de rutas: sistema de Odometría. [En línea] (Trabajo de Titulación). Universidad

Técnica del Norte, Ibarra, Ecuador. 2018. pp. 5. [Consulta: 4 marzo 2020]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/7908>.

VARELA ALDÁS, José Luis. Tele-Operación Bilateral de un Robot Manipulador Móvil. [en línea] (Trabajo de Titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2016. pp. 15-68. [Consulta: 17 diciembre 2020]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/6090>.

VÁSCONEZ CHAGÑAY, C., Construcción e implementación de un robot móvil usando un SBC (Single- Board Computer) para aplicaciones de transporte de insumos médicos en el Centro de Salud tipo C Lizarzaburu en la ciudad de Riobamba. [en línea] (Trabajo de Titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2018. p. 11-25. [Consulta: 4 junio 2020]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/9240>.

VISIOLI, A. *Basics of PID Control*. [En línea]. London: Springer, Advances in Industrial Control, 2006. [Consulta: 21 noviembre 2020]. Disponible en: https://doi.org/10.1007/1-84628-586-0_1.

WATANABE, Y.; & YUTA, S. "Position estimation of mobile robots with internal and external sensors using uncertainty evolution technique". *IEEE International Conference on Robotics and Automation Proceedings*, vol.3, (1990) , (United States of America) pp. 2011-2016 .

ZURITA, Rafael. et al. "Sistema de localización para robots móviles de bajo costo utilizando marcas de referencia artificiales en ambientes de interiores". *XXV Congreso Argentino de Ciencias de la Computación (CACIC) (Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)* [en línea], 2019, (Argentina), pp. 1081-1090 [Consulta: 7 enero 2021]. ISBN 978-987-688-377-1. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/91412>.