



**Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el
transporte de materiales**

Cumbajin Jaguaco, Adriana Lizeth

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de
Ingeniera en Electrónica, Automatización y Control.

Ing. Orozco Brito, Luis Alberto Msc.

23 de Octubre del 2020

URKUND

Document Information

Analyzed document Proyecto titulacion Cumbajin_Adriana.pdf (D82577454)
 Submitted 10/23/2020 10:10:00 PM
 Submitted by Orozco Brito Luis Alberto
 Submitter email laorozco@espe.edu.ec
 Similarity 4%
 Analysis address laorozco.espe@analysis.orkund.com

Proof Proof

Sources included in the report

W	URL: https://docplayer.es/58079756-La-version-digital-de-esta-tesis-esta-protégida-por- ... Fetched: 11/20/2019 9:24:36 PM	 1
W	URL: http://repositorio.espe.edu.ec/bitstream/21000/246/6/T-ESPE-027504.pdf Fetched: 10/24/2020 12:44:00 AM	 3
W	URL: https://scielo.conicyt.cl/pdf/infotec/v26n2/art17.pdf Fetched: 10/24/2020 12:44:00 AM	 2
W	URL: https://www.redalyc.org/pdf/911/91126903005.pdf Fetched: 10/24/2020 12:44:00 AM	 1
SA	Universidad de las Fuerzas Armadas ESPE / Tesis_Camino_Rojas_V_12_V1.pdf Document Tesis_Camino_Rojas_V_12_V1.pdf (D60821952) Submitted by: smcamino@espe.edu.ec Receiver: wgaguilar.espe@analysis.orkund.com	 7
W	URL: https://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf Fetched: 10/24/2020 12:44:00 AM	 1
W	URL: https://docplayer.es/27815245-Trabajo-fin-de-grado-planificacion-de-rutas-optimas- ... Fetched: 3/4/2020 4:07:11 AM	 1
W	URL: http://www.frlr.utn.edu.ar/archivos/alumnos/electronica/catedras/38-sistemas-de-co ... Fetched: 10/24/2020 12:44:00 AM	 2
W	URL: http://neithan.weebly.com/uploads/5/2/8/0/52807/memoria_tfg.pdf Fetched: 10/24/2020 12:44:00 AM	 9
SA	Universidad de las Fuerzas Armadas ESPE / Tesis_Camino_Rojas.pdf Document Tesis_Camino_Rojas.pdf (D59588909) Submitted by: mirojas@espe.edu.ec Receiver: wgaguilar.espe@analysis.orkund.com	 1
W	URL: https://pdfs.semanticscholar.org/d9dc/9342f9b656d2f048afe2a258ef6aef628035.pdf Fetched: 10/24/2020 12:44:00 AM	 2
	URL: http://www.sbai2015.dca.ufrn.br/download/artigo/163	



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

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

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el transporte de materiales”** fue realizado por la señorita **Cumbajin Jaguaco Adriana Lizeth**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 23 de octubre de 2020

Ing. Orozco Brito, Luis Alberto Msc.

C. C. 1710443803



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

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

RESPONSABILIDAD DE AUTORÍA

Yo, **Cumbajin Jaguaco Adriana Lizeth** con cédula de ciudadanía n° 1721853818, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el transporte de materiales”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 23 de octubre de 2020

.....
Cumbajin Jaguaco Adriana Lizeth

C.C.. 1721853818



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

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

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Cumbajin Jaguaco Adriana Lizeth** con cédula de ciudadanía n° 1721853818, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el transporte de materiales”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 23 de octubre de 2020

.....
Cumbajin Jaguaco Adriana Lizeth

C.C.: 1721853818

Dedicatoria

A mis padres Ángel y Marcita, quienes me han ayudado a alcanzar y cumplir con este objetivo; gracias por su paciencia, cariño y esfuerzo que me han brindado en esta etapa de mi vida.

A mi hermano David, que, con su apoyo, cariño, paciencia y sus consejos me han ayudado a desarrollarme profesionalmente, gracias por estar conmigo en todo el momento difícil que te compartí.

A mi hermana, Dayana, que, con su compañía, me han ayudado a fortalecer mi carácter, gracias.

A mis queridas mascotas Kyo y lori, por siempre estar junto a mí, en todas las noches de desvelo, gracias.

Adriana Lizeth Cumbajin Jaguaco

Agradecimiento

Agradezco a la Universidad de las Fuerzas Armadas ESPE y a los profesores, por todo el esfuerzo hecho para compartir sus conocimientos. De manera especial a mi tutor de tesis Ing. Luis Orozco por toda la paciencia, los consejos y la motivación para cumplir con el objetivo de este proyecto.

A mis amigos Melissa, Brenda y José, por estar a mi lado en momentos difíciles de mi vida y de la carrera, es por esto que les quedo agradecida por las experiencias que convivimos y por las palabras de aliento que siempre me dieron.

A mi carrera, por la cual descubrí las más emocionantes experiencias académicas, añadiendo a mis grandes amigos: Álvaro, Enver, Bryan, Tefy A., Tefy M, y Janis, por su amistad y compañía en las noches de desvelo y momentos difíciles que vivimos en cada una de las materias que compartimos, les estoy agradecida por haberlos conocido.

Adriana Lizeth Cumbajin Jaguaco

Índice de Contenido

Desarrollo de un sistema de navegación basado en la plataforma Pioneer P3-DX para el transporte de materiales	1
Certificación de trabajo de titulación	3
Hoja de resultados de la herramienta Urkund	2
Autoría de responsabilidad	4
Autorización de publicación	5
Dedicatoria	6
Agradecimiento	7
Índice de Contenido	8
Índice de Tablas	13
Índice de Figuras.....	14
Resumen	20
Abstract	21
Capítulo I.....	22
Planteamiento del problema de investigación.....	22
Antecedentes	22
Justificación e importancia	24
Alcance	25
Objetivos	27
<i>Objetivo General</i>	27
<i>Objetivos Específicos</i>	27

Capítulo II.....	29
Fundamento Teórico	29
Robótica Móvil	29
Tipos de locomoción robot móvil	31
Disposición de las ruedas	32
Robots Holonómicos	32
<i>Condiciones de los Robots Holonómicos</i>	33
Robots no Holonómicos	34
<i>Condiciones de los Robots no Holonómicos</i>	34
Modelo diferencial del robot móvil	36
Ecuaciones Cinemáticas.....	40
Cinemática diferencial Directa.....	42
Cinemática diferencial Inversa.....	43
Controladores de posición para robots móviles	45
<i>Controlador PID</i>	46
<i>Controlador PID 2DoF (dos grados de libertad)</i>	47
Control de Evasión de obstáculos por Impedancia.....	50
Planificación de trayectorias	53
<i>Algoritmo A*</i>	53
<i>Algoritmo RRT</i>	55
<i>Algoritmo RRT*</i>	55
<i>Algoritmo Híbrido A*</i>	57

	10
Capítulo III.....	59
Generalidades del sistema	59
Plataforma Pioneer P3-DX	59
<i>Hardware del robot</i>	59
<i>Conexión al robot</i>	61
<i>Software del robot</i>	62
Interfaz Aria Matlab	63
Arquitectura del robot	66
Capítulo IV	68
Sistema de navegación	68
Estimación del modelo del robot móvil	68
<i>Controlador PID</i>	74
<i>Controlador PID2DoF (dos grados de libertad)</i>	79
<i>Control de evasión de obstáculos</i>	85
Planificación de trayectorias	88
<i>Búsqueda en A*</i>	89
<i>RRT (Árbol aleatorio de exploración rápida)</i>	92
<i>RRT* (Árbol aleatorio de exploración rápida optimizada)</i>	93
<i>Híbrido A*</i>	95
Algoritmos para el sistema de navegación	96
Cinemática Directa.....	97
Cinemática Inversa.....	98

	11
Generación de trayectorias	99
Sistema de control de seguimiento	103
Sistema de evasión de obstáculos	108
Capítulo V	110
Pruebas y resultados	110
Prueba de comandos para velocidades de las ruedas.	110
Validación Modelo Cinemático Directo	111
Validación Modelo Cinemático Inverso	117
Seguimiento de trayectorias	122
<i>Trayectoria Rectilínea</i>	122
<i>Trayectoria Circular</i>	129
<i>Trayectoria en Caracol con lazo</i>	135
<i>Trayectoria de Rosa de cuatro Pétalos</i>	142
Planificación de trayectorias	147
Planificación RRT.....	150
Planificación RRT*	153
Planificación Híbrido A*	155
<i>Buscador A*</i>	158
Capítulo VI	163
Conclusiones y Recomendaciones.....	163
Conclusiones	163
Recomendaciones	165

Referencias bibliografía..... 167

Índice de Tablas

Tabla 1 <i>Lista de comandos obtenidos de ARIA utilizados en Matlab.</i> -----	65
Tabla 2 <i>Trayectorias generadas por fórmulas.</i> -----	100
Tabla 3 <i>Prueba del comando arrobot_setwheelvels.</i> -----	111
Tabla 4 <i>Modelo cinemático directo: Movimiento traslacional.</i> -----	112
Tabla 5 <i>Modelo cinemático directo: Movimiento rotacional.</i> -----	114
Tabla 6 <i>Velocidades estimadas para una distancia fija de 1 metro.</i> -----	118
Tabla 7 <i>Pruebas del controlador PID y PID2DoF en trayectoria rectilínea.</i> -----	122
Tabla 8 <i>Pruebas del controlador PID y PID2DoF en trayectoria circular.</i> -----	130
Tabla 9 <i>Pruebas del controlador PID y PID2DoF en trayectoria de caracol.</i> -----	136
Tabla 10 <i>Pruebas del controlador PID y PID2DoF en trayectoria de 4 pétalos.</i> -----	143

Índice de Figuras

Figura 1 <i>Plataformas Pioneers.</i>	23
Figura 2 <i>Plataforma robótica Pioneer P3-DX con carga para trasportar.</i>	26
Figura 3 <i>Posibles trayectorias que podría seguir el robot móvil.</i>	29
Figura 4 <i>Sistema de control de un robot móvil con diferentes niveles jerárquicos</i>	30
Figura 5 <i>Tipos de robots móviles.</i>	32
Figura 6 <i>Tipos de locomoción de un robot móvil.</i>	33
Figura 7 <i>Giros del robot diferencial.</i>	34
Figura 8 <i>Desplazamiento del robot no holonómico.</i>	36
Figura 9 <i>Variables físicas del robot diferencial.</i>	37
Figura 10 <i>Representación entre los dos sistemas.</i>	38
Figura 11 <i>Error de posición y orientación del robot.</i>	40
Figura 12 <i>Diagrama de bloques dela cinemática diferencial directa</i>	43
Figura 13 <i>Diagrama de bloques del modelo cinemático diferencial inverso</i>	45
Figura 14 <i>Diagrama de bloques 2DoF PID.</i>	48
Figura 15 <i>Diagrama de bloques de un controlador PI-PD.</i>	49
Figura 16 <i>Representación de las fuerzas ficticias en el método de la impedancia</i>	50
Figura 17 <i>Posiciones del robot móvil para evasión de obstáculos</i>	52
Figura 18 <i>Planificación de la trayectoria A*</i>	54
Figura 19 <i>Planificación de trayectoria RRT</i>	55
Figura 20 <i>Planificación de trayectorias RRT*</i>	57
Figura 21 <i>Diferencia entre el Algoritmo A* e Híbrido A*.</i>	58
Figura 22 <i>Cambio de baterías al Pioneer</i>	60
Figura 23 <i>Placa de control del Pioneer P3-DX</i>	62
Figura 24 <i>Paquete para la interfaz Aria-Matlab</i>	64
Figura 25 <i>Archivos de depuración para los comandos de Aria.</i>	64

Figura 26 <i>Forma de conexión cliente-servidor</i>	67
Figura 27 <i>Modelo cinemático diferencial inverso del robot Pioneer P3-DX</i>	68
Figura 28 <i>Entradas y salidas de la planta</i>	69
Figura 29 <i>Señales de entrada del robot Pioneer</i>	70
Figura 30 <i>Señales de salida del robot Pioneer</i>	70
Figura 31 <i>Importar datos a la app de 'ident' de Matlab</i>	71
Figura 32 <i>Comportamiento de la señal entrada X de salida X en la función "ident"</i>	72
Figura 33 <i>Comportamiento de la señal entrada Y de salida Y en la función "ident"</i>	72
Figura 34 <i>Selección de la función de transferencia de segundo orden</i>	73
Figura 35 <i>Funciones de transferencia desacopladas</i>	73
Figura 36 <i>Estimación de la planta del robot móvil en la salida X</i>	73
Figura 37 <i>Estimación de la planta del robot móvil en la salida Y</i>	74
Figura 38 <i>Esquema de control PID</i>	75
Figura 39 <i>Esquema de diseño del controlador</i>	76
Figura 40 <i>Respuesta al escalón unitario a la función de transferencia en lazo cerrado</i>	79
Figura 41 <i>Estructura PID 2DoF, basado en un controlador PI-PD para el robot móvil</i>	79
Figura 42 <i>Esquema de diseño del controlador</i>	80
Figura 43 <i>Esquema del controlador PI</i>	81
Figura 44 <i>Esquema del controlador PD</i>	82
Figura 45 <i>Ganancia k_d por el método del lugar geométrico</i>	83
Figura 46 <i>Respuesta al escalón unitario a la función de transferencia en lazo cerrado</i>	85
Figura 47 <i>Diagrama de control de evasión</i>	86
Figura 48 <i>Distribución de la distancia de mínima de evasión para cada sensor</i>	86
Figura 49 <i>Distancia XY con respecto a la ganancia k</i>	87
Figura 50 <i>Diagrama de control de evasión y seguimiento de trayectorias</i>	88
Figura 51 <i>Cuadrícula de ocupación del mapa</i>	89

Figura 52 <i>Número de direcciones de cada nodo</i>	90
Figura 53 <i>Diagrama de flujo del algoritmo A*</i>	91
Figura 54 <i>Conexión del punto de inicio y objetivo con el algoritmo A*</i>	92
Figura 55 <i>Esquema de entradas y salidas para el algoritmo RRT</i>	93
Figura 56 <i>Diagrama de conexión del algoritmo RRT</i>	93
Figura 57 <i>Esquema de entradas y salidas para el algoritmo RRT*</i>	94
Figura 58 <i>Diagrama del algoritmo RRT*</i>	95
Figura 59 <i>Esquema de entradas y salidas para el algoritmo híbrido A*</i>	96
Figura 60. <i>Diagrama del algoritmo Híbrido A*</i>	96
Figura 61 <i>Diagrama de flujo propuesto para la cinemática diferencial directa</i>	98
Figura 62 <i>Diagrama de flujo propuesto para la implementación en el robot móvil</i>	99
Figura 63 <i>Diagrama de flujo de la selección del tipo de trayectorias</i>	101
Figura 64 <i>Diagrama de flujo de las subrutinas de los tipos de trayectos</i>	101
Figura 65 <i>Interfaz para ingresar la distancia o el radio, y el tiempo del trayecto</i>	102
Figura 66 <i>Planificador de trayectorias</i>	103
Figura 67 <i>Diagrama de flujo para el controlador PID</i>	105
Figura 68 <i>Diagrama de flujo para el controlador PID 2DoF</i>	107
Figura 69 <i>Diagrama de flujo de la incorporación del control de evasión</i>	108
Figura 70 <i>Diagrama de flujo del método de evasión</i>	109
Figura 71 <i>Prueba N°1: Desplazamiento traslacional $t = 5$ s</i>	113
Figura 72 <i>Prueba N°2: Desplazamiento traslacional $t = 10$ s</i>	113
Figura 73 <i>Prueba N°3: Desplazamiento traslacional $t = 15$ s</i>	113
Figura 74 <i>Prueba N°4: Desplazamiento rotacional vs Tiempo</i>	115
Figura 75 <i>Prueba N°5: Desplazamiento rotacional vs Tiempo</i>	115
Figura 76 <i>Prueba N°6: Desplazamiento rotacional vs Tiempo</i>	115
Figura 77 <i>Prueba N°7: Desplazamiento rotacional vs Tiempo</i>	116

Figura 78 Prueba N°8: Desplazamiento rotacional vs Tiempo	116
Figura 79 Prueba N°9: Desplazamiento rotacional vs Tiempo.	117
Figura 80 Prueba N°10: Desplazamiento al punto (3,0) m a un tiempo $t=30s$	119
Figura 81 Prueba N°11: Desplazamiento al punto (3,2) m a un tiempo $t=30s$	119
Figura 82 Prueba N°12: Desplazamiento al punto (3,-2) m a un tiempo $t=30s$	119
Figura 83 Prueba N°14: Trayectoria circular a un tiempo $t = 40s$	120
Figura 84 Error de posición de la Prueba N°14.....	120
Figura 85 Prueba N°15: Trayectoria rosa de 4 pétalos a un tiempo $t=40s$	121
Figura 86 Error de posición de la Prueba N°15.	121
Figura 87. Prueba N°1 PID:Desplazamiento al punto (3,0) m a un tiempo $t=30s$	123
Figura 88 Prueba N°1 I PID: Error de posición vs Tiempo.	124
Figura 89 Prueba N°2 PID: Desplazamiento al punto (3,2)m a un tiempo $t=30s$	124
Figura 90 Prueba N°2 PID: Error de posición vs Tiempo	125
Figura 91 Prueba N°2 PID:Desplazamiento al punto (3,-2) m a un tiempo $t=30s$	125
Figura 92 Prueba N°3 PID2DoF: Error de posición vs Tiempo.....	126
Figura 93 Prueba N°4 PID2DoF: Desplazamiento al punto (3,0)m a un tiempo $t=30s$	127
Figura 94 Prueba N°4 PID2DoF: Error de posición vs Tiempo.....	127
Figura 95 Prueba N°5 PID2DoF: Desplazamiento al punto (3,2)m a un tiempo $t=30s$	128
Figura 96 Prueba N°5 PID2DoF: Error de posición vs Tiempo.....	128
Figura 97 Prueba N°6 PID2DoF: Desplazamiento al punto (3,-2) m a un tiempo $t=30s$..	129
Figura 98 Prueba N°6 PID2DoF: Error de posición vs Tiempo.....	129
Figura 99 Trayectoria circular de radio 1 metro.....	130
Figura 100 Prueba N°7 PID: Trayectoria circular en el tiempo $t=40s$	131
Figura 101 Prueba N°7 PID: Error de posición vs Tiempo.	131
Figura 102 Prueba N°8 PID: Trayectoria circular en el tiempo $t=60s$	132
Figura 103 Prueba N°8 PID2: Error de posición vs Tiempo	132

Figura 104 Prueba N°9 PI2DoF: Trayectoria circular en el tiempo $t=40s$	133
Figura 105 Prueba N°9 PI2DoF: Error de posición vs Tiempo.	134
Figura 106 Prueba N°10 PID2DoF: Trayectoria circular en el tiempo $t=60s$	135
Figura 107 Prueba N°10 PID2DoF: Error de posición vs Tiempo.	135
Figura 108 Trayectoria en caracol con lazo de radio 1 metro.	136
Figura 109 Prueba N°11 PID: Trayectoria caracol en el tiempo $t=40s$	137
Figura 110 Prueba N°11 PID: Error de posición vs Tiempo.	138
Figura 111 Prueba N°12 PID: Trayectoria del caracol en el tiempo $t=60s$	139
Figura 112 Prueba N°12 PID: Error de posición vs Tiempo.	139
Figura 113 Prueba N°13 PID2DoF: Trayectoria del caracol en el tiempo $t=40s$	140
Figura 114 Prueba N°13 PID2DoF: Error de posición vs Tiempo.	140
Figura 115 Prueba N°14 PID2DoF: Trayectoria del caracol en el tiempo $t=60s$	141
Figura 116 Prueba N°14 PID2DoF: Error de posición vs Tiempo.	141
Figura 117 Trayectoria en caracol con lazo de radio 1 metro.	142
Figura 118 Prueba N°15 PID: Trayectoria en el tiempo $t=40s$	144
Figura 119 Prueba N°15 PID: Error de posición vs Tiempo.	144
Figura 120 Prueba N°16 PID: Trayectoria en el tiempo $t=40s$	145
Figura 121 Prueba N°16 PID: Error de posición vs Tiempo.	145
Figura 122 Prueba N°17 PID2DoF: Trayectoria en el tiempo $t=40s$	146
Figura 123 Prueba N°17 PID2DoF: Error de posición vs Tiempo.	146
Figura 124 Prueba N°18 PID2DoF: Trayectoria en el tiempo $t=60s$	147
Figura 125 Prueba N°18 PID2DoF: Error de posición vs Tiempo.	147
Figura 126 Entorno de trabajo.	148
Figura 127 Mapa de ocupación del espacio de trabajo en Matlab.	149
Figura 128 Obstáculos fuera del mapa de ocupación.	149
Figura 129 Aumento de grosor al mapa de ocupación.	150

Figura 130 Ruta RRT en el patio del laboratorio	151
Figura 131 Obtención de los puntos de la ruta determinado por el algoritmo RRT.	151
Figura 132 Trayectoria realizada por el móvil por la ruta extraída del algoritmo	152
Figura 133 Error de posición vs Tiempo de la ruta RRT.	152
Figura 134 Ruta RRT* en el patio del laboratorio	153
Figura 135 Obtención de los puntos de la ruta determinado por el algoritmo RRT*.....	154
Figura 136 Trayectoria realizada por el móvil por la ruta extraída del algoritmo RRT*.	154
Figura 137 Error de posición vs Tiempo de la ruta RRT*	155
Figura 138 Ruta RRT en el patio del laboratorio.	156
Figura 139 Obtención de los puntos de la ruta del algoritmo A* Híbrido.	156
Figura 140 Trayectoria realizada por el móvil por la ruta del algoritmo Híbrido A*.. ...	157
Figura 141. Error de posición vs Tiempo de la ruta Híbrido A*.	158
Figura 142 Mapa de ocupación para el algoritmo A*	159
Figura 143 Mapa de ocupación del espacio de trabajo en Matlab	159
Figura 144 Ruta A* en el patio del laboratorio.	160
Figura 145 Representación de los puntos de la ruta determinado por el algoritmo A*.	161
Figura 146 Trayectoria realizada por el móvil por la ruta extraída del algoritmo A*.	161
Figura 147 Error de posición vs Tiempo de la ruta Híbrido A*	162

Resumen

En la industria, así como en la vida cotidiana el uso de los robots móviles es de gran importancia, en aplicaciones como la navegación en mapas que definen el entorno de trabajo, por lo cual debe tener un control suficiente para seguir una trayectoria o evadir un obstáculo en un determinado ambiente, y alcanza el objetivo deseado, para el cual móvil es programado. En consecuencia, el presente trabajo de titulación se encamina al desarrollo de un sistema de navegación utilizando el robot de entrenamiento móvil Pioneer P3-DX, para el transporte de materiales mediante la implementación de un modelo cinemático enfocado, en la utilización de las velocidades de las ruedas derecha e izquierda del móvil. En primer lugar, se realizó pruebas del modelo cinemático en el seguimiento de cuatro diferentes trayectorias (rectilínea, circular, caracol con lazo y rosa de cuatro pétalos) con dos tipos de control PID y PID2DoF, analizando el desempeño de ambos controladores. En segundo lugar, se realizó pruebas con cuatro planificaciones de trayectorias (A^* , RRT, RRT*, Híbrido A^*), utilizando un mapa de los laboratorios de la Universidad de las Fuerzas Armadas, trazando una posición inicial y final; y verificar el seguimiento y la evasión de obstáculos colocados en el mapa.

PALABRAS CLAVE:

- **ROBOT MÓVIL**
- **PLANIFICACIÓN DE TRAYECTORIA**
- **SISTEMA DE NAVEGACIÓN**

Abstract

In industry, as well as in everyday life, the use of mobile robots is of great importance, in applications such as navigation on maps that define the work environment, for which you must have sufficient control to follow a trajectory or evade an obstacle in a certain environment, and reaches the desired objective, for which the mobile is programmed. Consequently, the present degree work is aimed at the development of a navigation system using the mobile training robot Pioneer 3-DX, for the transport of materials through the implementation of a kinematic model focused on the use of the speeds of the right and left wheels of the mobile. In the first place, tests of the kinematic model were carried out in the follow-up of four different trajectories (rectilinear, circular, snail with loop and rose with four petals) with two types of PID and PID2DoF control, analyzing the performance of both controllers. Secondly, tests were carried out with four trajectory plans (A *, RRT, RRT *, Hybrid A *), using a map of the laboratories of the University of the Armed Forces, drawing an initial and final position; and verify the tracking and avoidance of obstacles placed on the map.

KEYWORDS:

- **MOBILE ROBOT**
- **TRAJECTORY PLANNING**
- **NAVIGATION SYSTEM**

Capítulo I

Planteamiento del problema de investigación

Antecedentes

Los robots móviles son utilizados principalmente para el transporte de materiales debido a su fácil movilidad en varios tipos terrenos. (Musuña, Zapata, Oñate, & Campusano, 2013) Actualmente la robótica móvil se considera un área de tecnología avanzada para la solución de problemas de alta complejidad; puesto que incorpora varias herramientas como: programación, percepción, instrumentación e inteligencia artificial, que sirven principalmente para aplicaciones en el campo industrial. (Bermúdez, 2002)

Los robots móviles operando en escenarios industriales no estructurales permiten la navegación de vehículos que deben enfrentarse a problemas de posicionamiento, identificación y evasión de obstáculos ya sea estáticos o dinámicos. Además, posee varias limitaciones como: moverse desde un punto inicial hasta un punto objetivo, ya que es un movimiento arriesgado para un robot móvil en comparación con un manipulador industrial. Por tal razón, no se espera que el robot móvil siga trayectorias o alcance su destino final con el mismo nivel de precisión y exactitud de un manipulador industrial. (Avanti, 2008)

Para un robot móvil su principal objetivo es seguir trayectorias, con o sin mapa del entorno, y posteriormente dirigir su movimiento sobre la misma de acuerdo a la información que obtiene de los sensores externos que posee tales como: ultrasónicos, visión o láser; permitiendo de esta manera desplazarse entre dos puntos cualesquiera evitando colisiones. Entonces sugiere diseñar métodos de control de trayectorias que impliquen parámetros como: posición, dirección y velocidad, asegurando la mayor independencia con el soporte de sensores externos que posea el robot. (Musuña, Zapata, Oñate, & Campusano, 2013)

En el Ecuador, ya se han desarrollado diversos prototipos de robots móviles en distintas universidades del país, ya sean éstos autónomos o radio controlados, (Jácome,

2016) con diferentes propósitos tales como: deporte, limpieza, transporte, entrenamiento, etc.

En la Universidad de las Fuerzas Armadas – ESPE, existe un acercamiento a un enfoque más industrial referente a la robótica móvil, usando principalmente los robots de investigación Pioneer, los cuales son plataformas personalizables, actualizables y lo suficientemente resistentes como para durar años de uso en el laboratorio. Los dos modelos de plataformas desarrolladas por MobileRobots mostradas en la Figura 1, que posee la institución son: el Pioneer P3-DX y el Pioneer P3-AT. En tales plataformas, se ha desarrollado ya varios algoritmos de control aplicando diferentes entornos de programación, permitiendo al estudiante conocer e interactuar con estos robots móviles y saber cómo actúa el mismo dentro de un ambiente industrial. (MobileRobots, 2011).

Figura 1

Plataformas Pioneers.



Nota. Tomado de: (Morales & Jaramillo, 2010).

Alguno de estos proyectos con el Pioneer P3-DX destaca la implementación de controladores clásicos, modernos e inteligente, tales como: control proporcional, control por métodos numéricos y control por redes neuronales realizados en la ESPE y la EPN (Santillán, 2015). Otros proyectos, realizados dentro de la institución con esta plataforma contempla la incorporación y el reemplazo de nuevos elementos con una tarjeta FPGA para el control del

de movimientos, y el desarrollo de un sistema de reconocimiento de voz inalámbrico Wireless (Pioneer, 2015) para el procesamiento de las señales del robot.

Justificación e importancia

Dentro del sector industrial, la función principal de los robots móviles es el transporte de materiales como: herramientas, materiales, y a su vez reducir tiempos de entrega, por lo cual es de vital importancia evadir obstáculos ya sean éstos dinámicos o estáticos sin el control de un piloto remoto.

Actualmente, los robots móviles en algunas industrias se utilizan para transportar varias herramientas y materiales de un lugar a otro, tantas veces conforme a su programación sin supervisión, manteniendo un servicio de al menos de 13 horas continuas, lo cual es un factor de vital importancia dentro de una empresa incluyendo control autónomo para el proceso de carga cuando el robot haya terminado sus tareas. (Vinssa, 2009)

La navegación de un robot móvil es una de las aplicaciones más significativas en el desarrollo de los sistemas de control, debido a sus aplicativos en entornos reales, siendo una de ellas el transporte de materiales, lo cual implica un control en el seguimiento de trayectorias y a su vez un control de evasión cuando se encuentre con algún obstáculo dentro del entorno de trabajo, lo cual ha sido de gran interés para algunos investigadores.

El problema consiste en definir una ruta de navegación para un robot móvil en un entorno observable (Holman, Jacinto, & Martínez, 2014), entre dos puntos específicos, determinado si el robot está libre de colisiones, además si satisface cierto criterio de desempeño en cuanto al controlador, al momento en que el robot realice las tareas de transporte (Espitia Cuchango & Sofrony Esmeral, 2012).

Por otro lado, existe ciertas limitaciones en la implementación de algoritmos de los sistemas de control, debido a la capacidad limitada del procesamiento que contiene el controlador del robot Pioneer P3-DX (Parra Andrade, 2015).

La importancia de este proyecto radica en el transporte de materiales y la navegación de la plataforma móvil. Dentro de las tareas que conlleva la navegación (Martha, Bautista, & Iruegas, 2007), se encuentra la planificación de movimientos, por lo cual se enfocará en el desarrollo de algoritmos basados en temas tales como: la evasión de obstáculos y seguimiento de trayectorias; debido a que en trabajos anteriores únicamente se ha implementado controladores con trayectorias previamente establecidas y no generadas, además realizando pruebas sin el transporte de objetos.

Además, con este proyecto se pretende utilizar actualizaciones en cuanto al software como: Matlab y Aria, con las cuales trabaja actualmente el robot; manteniendo el mismo controlador inicial con el que vino incorporado el robot Pioneer P3-DX, ya que, en proyectos anteriores realizados en la institución, se ha tenido la necesidad de reemplazar por tarjetas con mayor capacidad de procesamiento como, por ejemplo, la tarjeta de desarrollo FPGA, para poder implementar controladores modernos e inteligentes.

Todo esto permitirá a la plataforma Pioneer P3-DX, simular su comportamiento en entornos industriales, incorporando nuevas estrategias de control a partir de la percepción del entorno; lo cual puede ser útil para las empresas que busquen una solución de automatización enfocadas en robótica móvil.

Finalmente, la utilización de esta plataforma permitirá un desarrollo más profundo en el área de la robótica móvil, a los estudiantes de la Universidad de las Fuerzas Armadas-ESPE.

Alcance

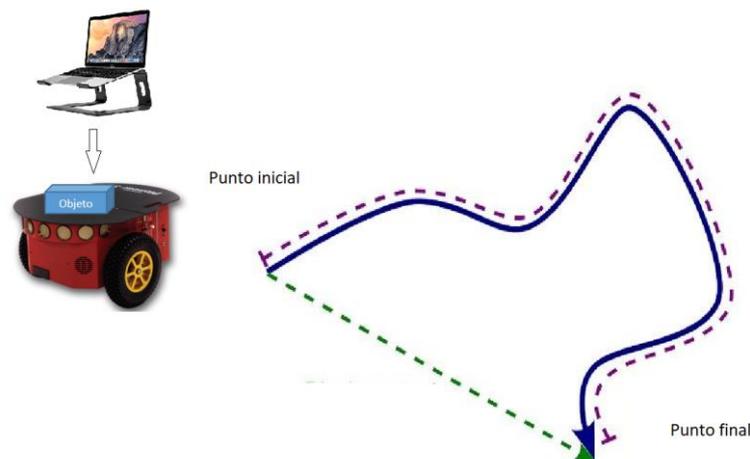
El alcance de este proyecto de investigación se enfoca principalmente en el transporte de objetos (computador personal) desde un punto inicial hasta un punto deseado.

Los objetos a transportar serán mancuernas de diferentes pesos que sumando con el peso de la computadora personal deberán estar dentro de los límites y especificaciones de la plataforma Pioneer P3-DX, las cuales se detallan a en la Figura 2:

- **Peso del objeto de carga:** Menores a 17 *kg*. (recomendado).
- **Tamaño y forma del material de carga:** Medidas de 44 *cm* x 38 *cm* x 22 *cm*.
(Morales & Jaramillo, 2010)
- **Velocidad máxima de carga:** Para cargas de hasta 17 *kg*, su velocidad ideal es de $1.2 \frac{m}{s}$.

Figura 2

Plataforma robótica Pioneer P3-DX con carga para transportar.



La computadora será sostenida mediante un soporte que se incorporará a la plataforma Pioneer, mientras que los objetos se colocarán debajo del mismo de forma manual al iniciar cada trayectoria.

El área en el cual se va a movilizar la plataforma móvil será emulada, dentro de los laboratorios y/o hangares de la institución, generando el ambiente industrial que se requiere para el transporte de materiales, para ello la superficie deberá ser plana y sin inclinaciones.

Las trayectorias por la que se movilizará la plataforma serán generadas por medio de algoritmos de planificación de movimientos, tales como: métodos RRT y A*.

Para el diseño del sistema de control de movimiento se utilizará el modelo cinemático del Pioneer 3-DX, cuyas variables a controlar son: la velocidad (traslacional y angular) y la localización (posición cartesiana y orientación). Los dos controladores a desarrollar son: PID (clásico) y espacio de estado y/o PID de dos grados de libertad. Los mismos que serán puestos a prueba en el seguimiento y evasión de obstáculos conjuntamente, comparando y analizando sus ventajas y desventajas. Aquellos algoritmos se desarrollarán en base a los datos proporcionados por los sensores incorporados del robot. (Capito & Proaño, Repositorio Digital EPN, 2011).

Para la implementación de los algoritmos se utilizará el software ARIA, que es el software de programación propia de robots Pioneer basada en C++. Además, se empleará el programa de Matlab para el desarrollo e implementación de los sistemas de control y la interfaz de visualización del robot móvil para el cumplimiento de las trayectorias, debido a que permite una mayor facilidad para la programación de los modelos matemáticos. (Martínez, 2008).

Objetivos

Objetivo General

- Diseñar e implementar un sistema de navegación mediante la integración de algoritmos de planificación y sistemas de control de movimientos, para la plataforma móvil Pioneer 3-DX.

Objetivos Específicos

- Proponer algoritmos de planificación de movimientos (generación de trayectorias), para la plataforma Pioneer 3-DX.
- Implementar un sistema de control para el seguimiento de las trayectorias previamente determinadas por la generación de trayectorias, basados en técnicas de control en el Espacio de Estado y/o PID de dos grados de libertad y PID clásico.

- Integrar algoritmos de evasión de obstáculos, para el transporte de objetos (computador personal) de un lugar a otro, en espacios que emulan ambientes industriales internos, mediante software e interfaz especializada ejecutadas en un computador personal.

Capítulo II

Fundamento Teórico

A continuación, se recopila teóricamente los conceptos y herramientas a utilizar en el proyecto de investigación; en el presente capítulo se realiza una descripción detallada del software y hardware.

Robótica Móvil

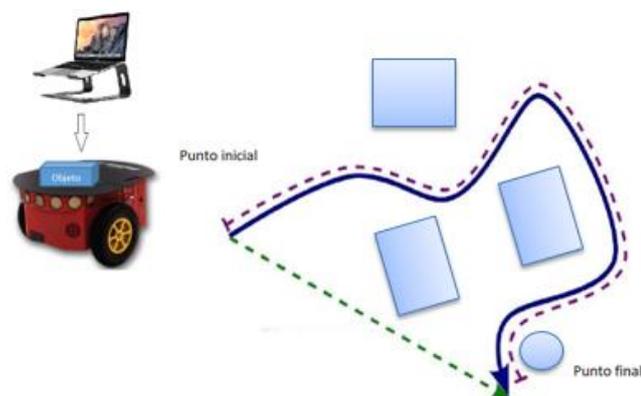
El estudio que abarca la robótica móvil, conlleva un rol importante en diversas áreas y aplicaciones dentro del sector industrial como el doméstico, tales como: militar, entreteniendo, asistencia médica, quehacer humano, seguridad, limpieza, carga, entre otras. (Ramos E. , 2010).

Debido a las diferentes aplicaciones en el estudio dentro de la robótica móvil, ésta se puede clasificar en vehículos aéreos, terrestres y acuáticos.

Los robots móviles deben enfrentarse a significativas perplejidades con respecto a la posición, ya que funcionan en grandes ambientes industriales o cotidianos. Una de esas perplejidades, es moverse de un punto inicial hasta un punto final, lo cual es una tarea muy difícil para un robot móvil en comparación a un manipulador industrial.

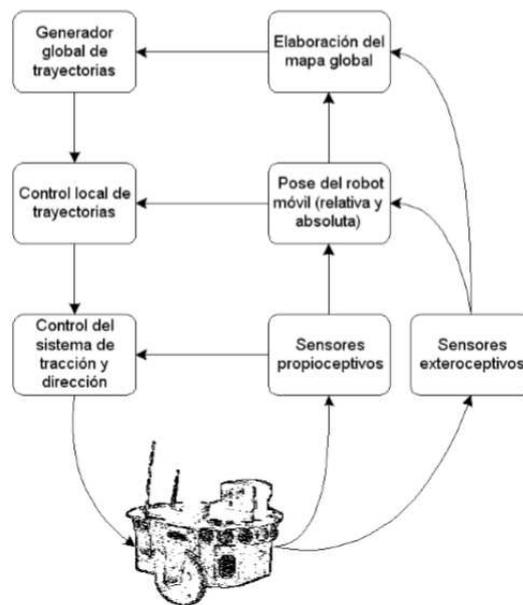
Figura 3

Posibles trayectorias que podría seguir el robot móvil.



Uno de los problemas más importantes a solucionar para un robot móvil es el seguimiento de trayectorias preestablecidas, como se indica en la Figura 3, en función al procesamiento de la información que provee el sistema de sensores externos, permitiendo al robot móvil moverse dentro de un ambiente de trabajo desde un punto inicial a uno objetivo, libre de choques. Por lo tanto, como señala (Cruz, 2008), se sugiere diseñar sistemas de control para el seguimiento de trayectorias y evasión, los cuales pueden involucrar: posición, dirección y velocidad, dependiendo de los sensores que posea el robot móvil y lograr la mayor autonomía posible.

Figura 4 Sistema de control de un robot móvil con diferentes niveles jerárquicos.



En la Figura 4 se muestra un diagrama la arquitectura general de control y las partes que lo componen. Tomado de: (Cruz, 2008).

Generador Global de Trayectorias (GGT): Dentro del esquema jerárquico ocupa el nivel alto, y se basa principalmente en su autonomía, el cual consiste en proponer coordenadas de inicio y de destino, definiendo así una trayectoria. Cuando el robot detecte un obstáculo dentro de su trayectoria, éste tiene que redefinirla mediante la información que provean los sensores externos o utilizando mapas de su ambiente de trabajo. (Cruz, 2008)

Generador Local de Trayectorias (GLT): Dentro del esquema jerárquico ocupa el nivel intermedio. Donde el robot en ocasiones toma el lugar del operador realizando funciones como: variar la velocidad, evitar obstáculos y realizar correcciones a la trayectoria dependiendo de la tarea que esté realizando. Cuando el robot no conoce el escenario de trabajo éste genera valores de referencia con, con ayuda de los sensores exteriores permitiendo al robot tomar decisiones en línea. (Cruz, 2008)

Control Local del Sistema de Tracción y Dirección (CL): Dentro del esquema jerárquico es el nivel más bajo, el cual genera acciones de control hacia el robot, es decir a sus motores, y conseguir que el robot alcance el punto objetivo. En este principalmente, los algoritmos de control se enfocan al control clásico. (Cruz, 2008)

Tipos de locomoción robot móvil

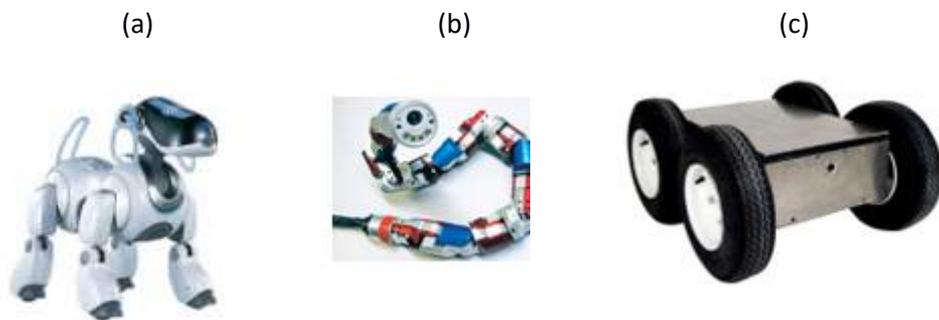
Dependiendo del tipo de entorno en el cual opera el robot móvil, se requiere la utilización de diferentes tipos de locomoción:

- **Con patas,** se caracteriza por:
 - Pueden atravesar mayor variedad de obstáculos.
 - Requieren mayor energía para realizar los movimientos.
 - Requieren un sistema de control complejo y rápido.
 - EL equilibrio puede generar problemas.
 - Se estabiliza en un tiempo determinado.
- **Bioinspirados:** serpiente, salamandra, etc.
 - Estables estáticamente.
 - Mayor coordinación.
 - Mayor estabilidad.
 - Requiere de un sistema de control complejo.
 - Pueden atravesar mayor variedad de obstáculos.
 - Requieren mayor energía para realizar los movimientos.

- **Con ruedas**, se caracteriza por:
 - Requieren mayor energía para realizar los movimientos
 - Requieren un sistema de control sencillo.
 - Implementación mecánica simple
 - El equilibrio del robot no genera problemas.
 - Mayor dificultad para atravesar obstáculos dependiendo del terreno en que opera el robot móvil.

Figura 5

Tipos de robots móviles.



En la Figura 5, se muestra: (a) Robot con patas, (b) Robot inspirados, (c) Robot con ruedas.

Disposición de las ruedas

Las combinaciones de los diversos tipos de ruedas generan una gran variedad de robots móviles que se diferencian por su grado de movilidad. A continuación, se detalla las dos configuraciones de robots móviles más comunes presentando sus principales características de diseño.

Robots Holonómicos

Son robots capaces de cambiar de lugar en diferentes direcciones aleatorias, sus principales características son:

- La posición/orientación del robot no está restringida por la velocidad.

- El robot se puede mover instantáneamente en cualquier dirección del espacio (2D / 3D).
- Son robots omnidireccionales. (Ramos O. , 2016).

Figura 6

Tipos de locomoción de un robot móvil.



En la Figura 6 se muestran diferentes robots móviles con diferente configuración.

Condiciones de los Robots Holonómicos

Para determinar si un robot es holonómico debe cumplir con dos condiciones, que se detallan a continuación:

Condición 1. La velocidad $v(t)$ es integrable si existe una trayectoria $s(t)$ que depende solamente de la posición (x, y) y orientación (θ) ; tal como se expresa en la ecuación (1):

$$s = s(x, t, \theta) \quad (1)$$

Equivalente a:

$$ds = \frac{\partial s}{\partial x} dx + \frac{\partial s}{\partial y} dy + \frac{\partial s}{\partial \theta} d\theta \quad (2)$$

La condición para que la integralidad de la trayectoria, se obtiene derivando parcialmente la ecuación (2) con respecto al eje x, y y θ , obteniendo como resultado la ecuación (3):

$$\frac{\partial^2 s}{\partial x \partial y} = \frac{\partial^2 s}{\partial y \partial x} \quad \frac{\partial^2 s}{\partial x \partial \theta} = \frac{\partial^2 s}{\partial \theta \partial x} \quad \frac{\partial^2 s}{\partial y \partial \theta} = \frac{\partial^2 s}{\partial \theta \partial y} \quad (3)$$

Condición 2. La velocidad del robot es su restricción.

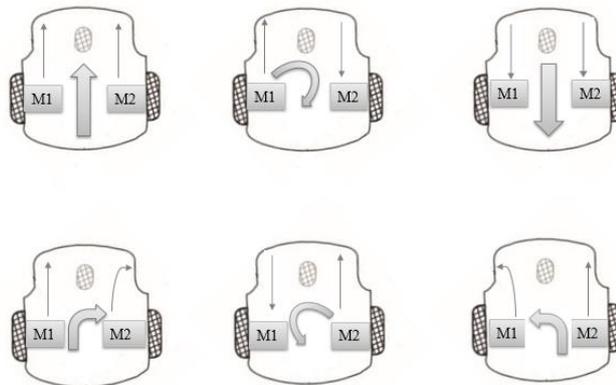
Robots no Holonómicos

El movimiento en este tipo de robots se produce únicamente por dos desplazamientos, y posee sólo dos grados de libertad, sus principales características son:

- La velocidad (no en todas las direcciones), restringe la orientación/posición del robot.
- La movilización instantánea del robot, no es posible en cada dirección del espacio, como se muestra en la Figura 7.
- Para el cálculo de la posición/orientación final del robot, se necesita de la medición de la distancia recorrida; y también se debe conocer cómo se ejecutó el movimiento en función del tiempo.
- Son robots no omnidireccionales. (Ramos O. , 2016)

Figura 7

Giros del robot diferencial.



Condiciones de los Robots no Holonómicos

Para determinar que un robot es no holonómico debe cumplir dos condiciones, que se detallan a continuación:

Condición 1. La ecuación (4), indica que las ecuaciones diferenciales no son integrables para la posición final. Se toma por ejemplo un robot móvil diferencial, cuya velocidad instantánea está dada por:

$$v(t) = \frac{ds}{dt} = \frac{dx}{dt} \cos(\theta) + \frac{dy}{dt} \sin(\theta) \quad (4)$$

Despejando el diferencial de la trayectoria se obtiene:

$$ds = dx * \cos(\theta) + dy * \sin(\theta) \quad (5)$$

Para definir si el robot es holonómico o no holonómico, se aplica la condición de integralidad de la trayectoria a la ecuación (5), obteniendo:

$$\frac{ds}{dx} = \cos(\theta), \quad \frac{ds}{dy} = \sin(\theta), \quad \frac{ds}{d\theta} = 0 \quad (6)$$

Satisfaciendo una de las tres igualdades de la condición de integralidad, derivando parcialmente las expresiones de la ecuación (6), se tiene la desigualdad de la ecuación (7) :

$$\left(\frac{\partial^2 s}{\partial x \partial \theta} = 0 \right) \neq \left(\frac{\partial^2 s}{\partial \theta \partial x} = -\sin(\theta) \right) \quad (7)$$

Por lo tanto, concluye que el robot es no holonómico.

Condición 2. La ecuación (8), indica que las distancias recorridas trazadas en las dos trayectorias S_1 y S_2 son iguales tanto, para la rueda derecha (S_{1R} y S_{2R}) como la izquierda (S_{1L} y S_{2L}), expresada en la ecuación (9) y (10). Sin embargo la ecuación (11), que expresa las posiciones finales (x_1, y_1) y (x_2, y_2) no son iguales, como se observa en la Figura 8.

(Ramos O. , 2016)

$$S_1 = S_2 \quad (8)$$

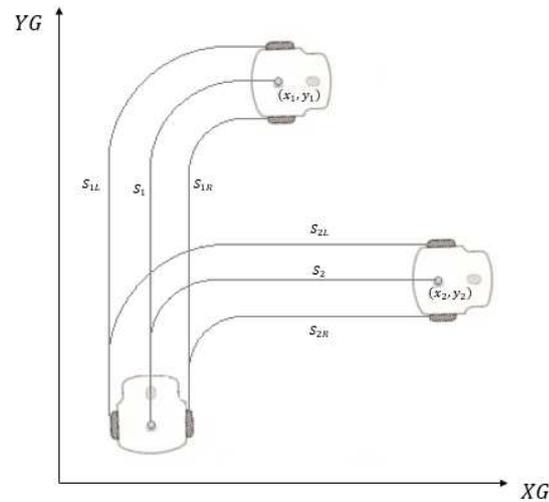
$$S_{1R} = S_{2R} \quad (9)$$

$$S_{1L} = S_{2L} \quad (10)$$

$$(x_1, y_1) \neq (x_2, y_2) \quad (11)$$

Figura 8

Desplazamiento del robot no holonómico.



Modelo diferencial del robot móvil

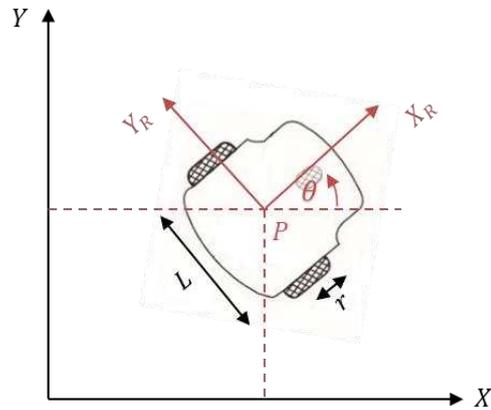
Para realizar el modelado del robot móvil se toma en cuenta ciertas consideraciones, con respecto al comportamiento del mismo, las cuales se detallan a continuación:

- El robot en consideración es un robot de accionamiento diferencial de dos ruedas, donde cada rueda se acciona independientemente. El movimiento hacia adelante se logra cuando ambos, las ruedas funcionan al mismo ritmo. El robot móvil logra girar a la derecha al conducir la rueda izquierda a una velocidad mayor que la rueda derecha y viceversa para girar a la izquierda. Este tipo de robot móvil puede girar en el acto conduciendo una rueda adelante y la segunda rueda en dirección opuesta. La tercera rueda es una rueda giratoria necesaria para la estabilidad del robot móvil. (Bensaci, Zennir, & Pomorski, 2018)
- Cada rueda individual contribuye al movimiento y al mismo tiempo, impone restricciones en movimiento del robot. Se asume que el robot se desliza en superficies planas sin rozamiento. Además, que el robot se mueve solo por las fuerzas ejercidas por el movimiento rotacional y traslacional de las ruedas.

- Se considera las dimensiones físicas del robot móvil, las medidas de interés son: el radio de las ruedas “ r ” y la distancia entre las ruedas “ L ”, ver Figura 9.

Figura 9

Variables físicas del robot diferencial.



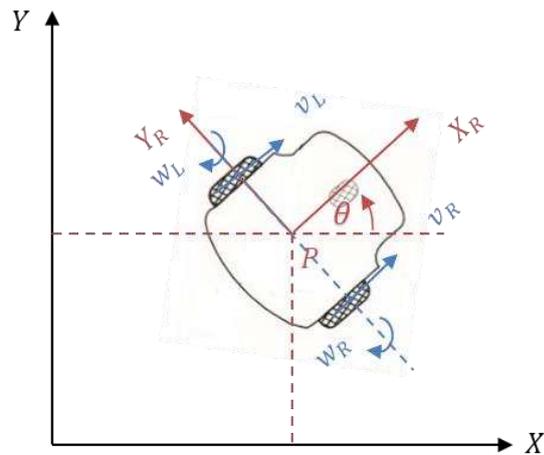
El objetivo de modelar el robot de configuración diferencial, es buscar una relación directa, de cómo influyen las velocidades de las ruedas (entradas) a los estados del sistema (X, Y, θ) . Considerando dos sistemas: el primero es el sistema de coordenadas inercial (X, Y) y el segundo el sistema del robot móvil (X_R, Y_R) , ver Figura 10. La relación entre el sistema del robot y el inercial es dada por la matriz de transformación de la ecuación (12):

$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (12)$$

La posición del robot $P(X, Y, \theta)$ se representa en coordenadas cartesianas en el sistema de coordenadas inercial.

Figura 10

Representación entre los dos sistemas.



Los comandos de movimiento real del robot son las velocidades angulares w_R y w_L de la rueda derecha e izquierda respectivamente, en lugar de las velocidades de dirección y orientación v y ω del robot. Por consiguiente, se analiza la relación entre las velocidades de las ruedas y las del robot.

Primero, se considera la contribución de la velocidad de giro de cada rueda a la velocidad de traslación en P en la dirección de $+X_R$. Si la rueda derecha gira ($v_1 = r w_R$) mientras la otra rueda no aporta nada y está parada, entonces ($v_2 = 0$); dado que P está en el punto medio entre las dos ruedas como se indica en la Figura 9, está se moverá instantáneamente con la mitad de la velocidad, como se indica en la ecuación (13):

$$v_1 = \left(\frac{1}{2}\right) r w_R. \quad (13)$$

De igual manera, en la ecuación (14) se expresa la velocidad de la rueda izquierda, con respecto al punto P :

$$v_2 = \left(\frac{1}{2}\right) r w_L. \quad (14)$$

En el robot de accionamiento diferencial, estas dos velocidades de la ecuación (13) y (14) simplemente se pueden sumar para calcular v_x , y a su vez la velocidad lineal v como se indica en la ecuación (15):

$$v = v_x = v_1 + v_2 = r \left(\frac{w_R + w_L}{2} \right) \quad (15)$$

Ahora considerando, por ejemplo, el movimiento del robot diferencial en el que cada rueda gira con la misma velocidad, pero en direcciones opuestas. El resultado es un robot giratorio estacionario, por lo tanto, v_x será cero. Mientras que al no haber un movimiento lateral en el sistema de referencia del robot, la velocidad en $v_y = 0$.

Finalmente, se calcula el componente rotacional ω del robot. Una vez más se pueden calcular las contribuciones de cada rueda de forma independiente. Considerando la rueda derecha (se nombra como rueda 1), el giro hacia adelante de esta rueda da como resultado un giro rotacional en el sentido de las agujas del reloj en el punto P . Tomando en cuenta que si la rueda 1 gira sola, el robot girará alrededor de la rueda 2. La velocidad de rotación w_1 en P se puede calcular porque la rueda se mueve instantáneamente a lo largo del arco de un círculo de radio L , como se indica en la ecuación (16):

$$w_1 = \frac{r}{L} w_R. \quad (16)$$

El mismo cálculo se aplica a la rueda izquierda (se nombra como rueda 2), con la excepción de que el giro hacia adelante da como resultado una rotación en el sentido opuesto a las agujas del reloj en el punto P , como se indica en la ecuación (17):

$$w_2 = -\frac{r}{L} w_L. \quad (17)$$

Al igual la velocidad lineal del robot, la velocidad angular de la ecuación (18), se obtiene sumando las velocidades de las ecuaciones (16) y (17):

$$\omega = w_1 + w_2 = r \left(\frac{w_R - w_L}{L} \right) \quad (18)$$

Donde r es el radio de las ruedas y L es la distancia entre las ruedas. (Kumar & Majumdar, 2014).

Cambiando las velocidades angulares (w_R, w_L) de las ruedas por las velocidades lineales (v_R, v_L), y expresándolas con respecto a las velocidades del robot (v, ω) en forma matricial, se tiene la ecuación (19):

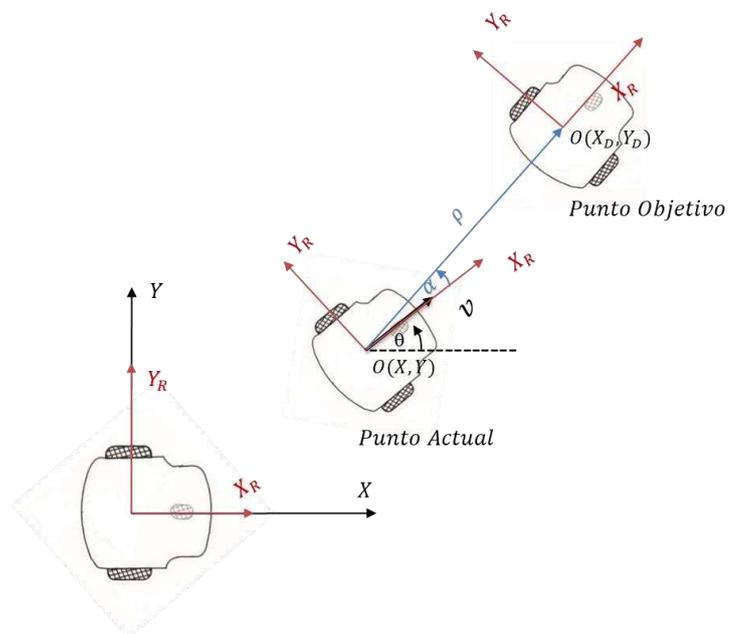
$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \cdot \begin{bmatrix} v_R \\ v_L \end{bmatrix} \quad (19)$$

Ecuaciones Cinemáticas

Suponiendo que el robot está en una posición arbitraria $P(X, Y, \theta)$ y la distancia entre su posición actual y una posición deseada $O(X_D, Y_D)$; definido con referencia al sistema inercial como se indica en la Figura 11. (Siegwrt & Nourbakhsh, 2004),

Figura 11

Error de posición y orientación del robot.



Nota. Tomado de : (Malu & Majumdar, 2014).

Donde:

ρ : es el error de posición del robot con respecto al sistema inercial.

α : es el error de orientación del robot con respecto al sistema inercial.

θ : ángulo de orientación del robot con respecto al sistema inercial.

X_D : punto objetivo en el eje x del sistema inercial.

Y_D : punto objetivo en el eje y del sistema inercial.

X : punto en el eje x del sistema inercial.

Y : punto en el eje y del sistema inercial.

X_R : es el eje x del sistema del robot.

Y_R : es el eje y del sistema del robot.

El sistema de coordenadas del robot se rige por la acción combinada de la velocidad lineal v y la velocidad angular ω . Usando suposiciones geométricas, las cuales se resumen en las ecuaciones (20), (21) y (22):

$$\dot{X} = v \cos(\theta) \quad (20)$$

$$\dot{Y} = v \sin(\theta) \quad (21)$$

$$\dot{\theta} = \omega \quad (22)$$

Donde $v \cos(\theta)$ y $v \sin(\theta)$ son los componentes de v a lo largo del eje X y Y , en el sistema inercial, que se obtienen al rotar el sistema inercial con respecto al sistema del robot, mediante la utilización de la matriz de rotación.

Según (Malu & Majumdar, 2014), esa misma posición $P(X, Y, \theta)$ del robot también puede representarse en términos de coordenadas polares con respecto al sistema inercial, involucrando un error de distancia $\rho > 0$. Donde, ρ es la distancia entre dos sistemas de coordenadas, como se indica en la ecuación (23):

$$\rho = \sqrt{(X_D - X)^2 + (Y_D - Y)^2} \quad (23)$$

Ahora, como se observa en la Figura 10, el ángulo α se encuentra medido entre el eje X_R de la referencia relativa al robot. Expresado con respecto al sistema inercial se tiene que, ver ecuación (24):

$$\alpha = \text{atan} \left(\frac{Y_D - Y}{X_D - X} \right) - \theta \quad (24)$$

El objetivo del planteamiento de las ecuaciones ρ y α , es utilizarlas como velocidades del robot, pero calculadas con respecto al sistema del robot, debido a que se encuentran rotadas con respecto al sistema inercial.

Para el cálculo de la velocidad lineal y angular del robot, se utiliza el sistema de coordenadas del robot como entradas las velocidades de los ejes del robot, como se indica en la ecuación (25) y (26).

$$v = \sqrt{(\dot{X}_R)^2 + (\dot{Y}_R)^2} \quad (25)$$

$$\omega = \text{atan}\left(\frac{\dot{Y}_R}{\dot{X}_R}\right) \quad (26)$$

Cinemática diferencial Directa

Para definir la cinemática diferencial directa que interesa para esta aplicación, se reemplazan las velocidades lineales de las ruedas que se obtuvieron anteriormente, en las ecuaciones que representan la dinámica del movimiento del robot de cada eje, como se indica en las expresiones (27), (28) y (29).

$$\dot{X} = \left(\frac{v_R + v_L}{2}\right) \cos(\theta) \quad (27)$$

$$\dot{Y} = \left(\frac{v_R + v_L}{2}\right) \sin(\theta) \quad (28)$$

$$\dot{\theta} = \omega = \left(\frac{v_R - v_L}{L}\right) \quad (29)$$

Luego, de obtener la cinemática diferencial directa, éste se utilizará para desarrollar el sistema de control para el seguimiento de la trayectoria, mediante simulaciones y comprobar el comportamiento del modelo. Para ello, es necesario obtener la posición y orientación del robot móvil, integrando las ecuaciones (30), (31) y (32).

$$X = \int \left(\frac{v_R + v_L}{2}\right) \cos(\theta) \quad (30)$$

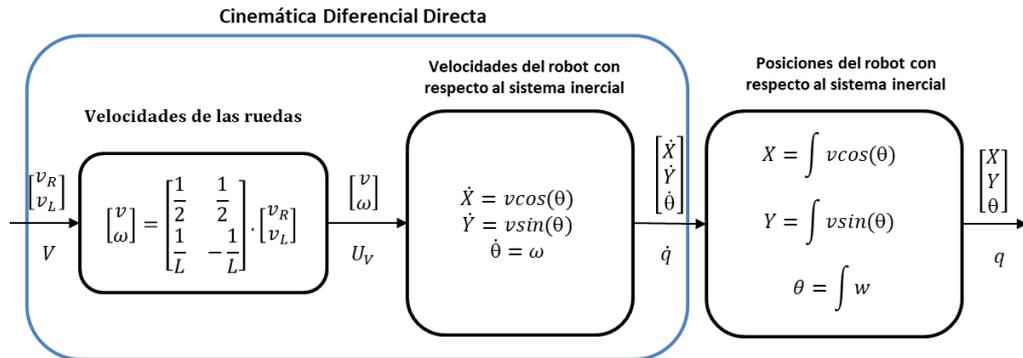
$$Y = \int \left(\frac{v_R + v_L}{2}\right) \sin(\theta) \quad (31)$$

$$\theta = \int \left(\frac{v_R - v_L}{L} \right) \quad (32)$$

En la Figura 12 se plantea el diagrama de bloques propuesto para la cinemática diferencial directa, incorporando las fórmulas mencionadas anteriormente.

Figura 12

Diagrama de bloques de la cinemática diferencial directa.



Donde:

V : es el vector de velocidades de entrada de las ruedas izquierda y derecha.

U_V : es el vector de velocidades lineal y angular del robot móvil, expresadas en el sistema del robot.

\dot{q} : es el vector de las velocidades del robot expresado en el sistema inercial.

q : es el vector de las posiciones del robot móvil con respecto al sistema inercial.

Cinemática diferencial Inversa

Para lograr un movimiento controlado por el robot, es necesario alterar las señales de entrada de la planta. Como se definió anteriormente las variables de entrada del robot son las velocidades de las ruedas, por lo tanto, hay que encontrar una representación matemática partiendo de la velocidad lineal y angular para determinar dichas velocidades.

Para encontrar la expresión matemática, se parte de las velocidades de los ejes en el sistema inercial \dot{X} y \dot{Y} . Ahora, como se mencionó anteriormente las velocidades del robot a utilizar (v, ω) , vienen dadas por las ecuaciones (25) y (26), las cuales están definidas en el sistema del robot, por lo tanto, se realiza un cambio de sistemas de coordenadas mediante

una matriz de rotación para encontrar las velocidades del robot, como se indica en la ecuación (33).

$$\begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \quad (33)$$

Para obtener el ángulo de orientación, se parte de un valor inicial $\theta = 0$, cuyo primer valor en el tiempo t_0 , servirá para calcular la velocidad angular en ese mismo instante, luego se estimará la orientación de los demás intervalos mediante la ecuación $\dot{\theta} = \omega$, hasta el tiempo de trayectoria ingresado.

Luego, de obtener las velocidades en el sistema del robot, éstas se reemplazan en las ecuaciones (25), y (26) para calcular la velocidad lineal v y la velocidad angular ω .

Finalmente, se utiliza la ecuación matricial de las velocidades de las ruedas de la ecuación (19), pero teniendo como entradas las velocidades del robot para calcular las velocidades lineales de las ruedas v_L y v_R , como se muestra en la ecuación (34).

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{L} \\ \frac{1}{2} & -\frac{1}{L} \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (34)$$

Una vez definida, la cinemática diferencial que describe el comportamiento del desplazamiento del robot, se plantea una solución para cumplir el objetivo de este trabajo de investigación, la cual consiste en definir una trayectoria determinada y dividirla en diferentes puntos por los cuales el robot se va a desplazar; al conseguir las coordenadas se pone en conocimiento la posición actual y la posición objetiva, que el robot deberá seguir.

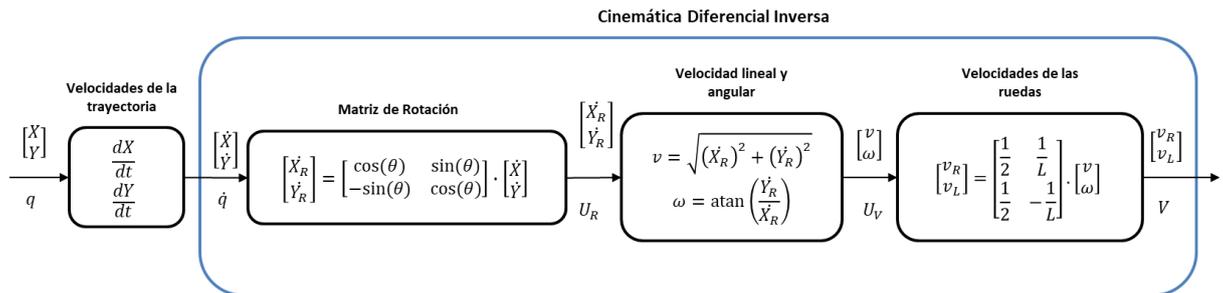
Para ingresar una trayectoria a la cinemática inversa, ésta se deberá definir en posiciones tanto para el eje X y Y , los cuales se deriva con respecto al tiempo, para obtener las velocidades de cada eje, y poder determinar las velocidades de las ruedas, como se muestra en la ecuaciones (35) y (36).

$$\dot{X} = \frac{dX}{dt} \quad (35)$$

$$\dot{Y} = \frac{dY}{dt} \quad (36)$$

Figura 13

Diagrama de bloques del modelo cinemático diferencial inverso.



Donde:

q : es el vector de entradas de las posiciones en el eje X y Y .

\dot{q} : es el vector de las velocidades del robot con respecto al sistema inercial.

U_R : es el vector de las velocidades del robot con respecto al sistema del robot.

U_V : velocidad lineal y angular del robot en el sistema del robot.

V : es el vector de las velocidades de las ruedas del robot móvil.

En la Figura 13, se plantea el diagrama de bloques para la cinemática diferencial inversa, incorporando las fórmulas mencionadas.

Controladores de posición para robots móviles

En este trabajo de investigación se pretende utilizar dos métodos para el control de posición en: PID y PID 2DoF. Considerando trayectorias previamente establecidas.

El objetivo de estos controles a implementar es que el robot móvil establezca movimientos de forma autónoma de trayectorias preestablecidas, formulando leyes de control que permitan estabilizar al robot sobre un punto de trabajo. Una solución para el problema de seguimiento de trayectorias se sugiere que el error entre la posición actual y la posición objetiva tienda a 0. (Valencia, Montoya, & Ríos, 2009).

En tales situaciones, hay grandes probabilidades de que el robot se desvíe de su trayectoria programada debido a parámetros no sistemáticos, en su mayoría se introducen en la trayectoria de los robots, tales como obstáculos o condiciones irregulares del piso; llegando a la conclusión de que la velocidad del robot móvil no se mantiene, por lo que es necesario implementar un sistema de control adecuado que permita controlar la velocidad en diferentes circunstancias que se introducen en el medio del robot.

Controlador PID

Los controladores PID son ampliamente utilizados en la industria robótica para controlar el funcionamiento de los robots móviles.

Respuesta Proporcional

El término proporcional produce un valor de salida que es proporcional al valor de error actual $e(t)$. La respuesta proporcional se ajusta multiplicando el error por una constante llamada ganancia proporcional (K_p), provocando una reducción en el error.

(Agarwal, Mahtab, Bandyopadhyay, & Gupta, 2013)

La respuesta proporcional está dada por la ecuación (37):

$$P_{out} = K_p \cdot e(t) \quad (37)$$

Donde:

K_p : es la ganancia proporcional.

$e(t)$: es el error.

P_{out} : es el término proporcional.

t : es el instante de tiempo que se calcula el error.

Respuesta Integral

El término integral suma el error en el tiempo. El resultado es aún el menor valor de error causará que el componente integral aumente lentamente. El error acumulado se multiplica por la ganancia integral (T_i). (Agarwal, Mahtab, Bandyopadhyay, & Gupta, 2013)

La respuesta integral está dada por la ecuación (38):

$$I_{out} = \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau \quad (38)$$

Donde:

T_i : es la ganancia integral.

$e(t)$: es el error.

I_{out} : es el término integral.

t : es el instante de tiempo que se calcula el error.

τ : variable de integración, que toma valores de 0 hasta el tiempo t .

Respuesta Derivativa

La respuesta derivativa se calcula determinando la pendiente del error a lo largo del tiempo y multiplicando esta tasa de cambio por la ganancia derivada (T_D). (Agarwal, Mahtab, Bandyopadhyay, & Gupta, 2013) La respuesta derivada viene dada ecuación (39):

$$D_{out} = T_D \cdot \frac{d(e(t))}{dt} \quad (39)$$

Donde:

T_D : es la ganancia derivativa.

$e(t)$: es el error.

D_{out} : es el término derivativo.

t : es el instante de tiempo que se calcula el error.

La salida del controlador PID $u(t)$, combinado las ecuaciones (12), (13) y (14), da como resultado la expresión de la ecuación (40):

$$u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t) + T_d \frac{de(t)}{dt} \right\} \quad (40)$$

Controlador PID 2DoF (dos grados de libertad)

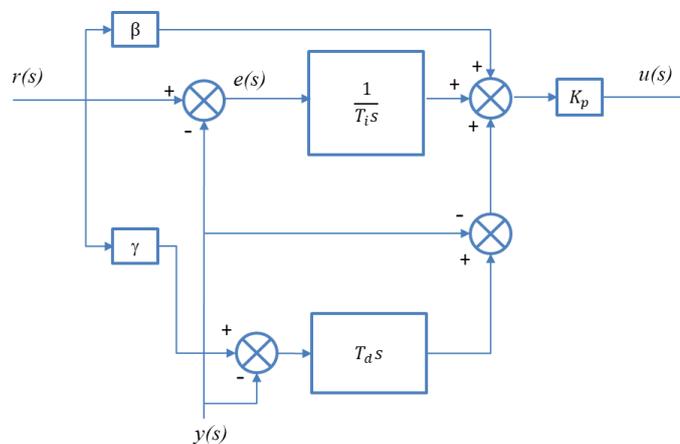
Al usar la ley de control de la ecuación (40), resulta claro que un cambio en escalón de la señal de referencia originará un impulso en la señal de control, situación que afecta a la señal de salida.

Por tal razón la acción derivativa es poco aplicada sobre la señal de referencia. Otra posibilidad es que la acción proporcional sea aplicada tan sólo sobre una fracción de la señal de referencia. Esto se denomina asignación de peso al punto de ajuste.

Un controlador PID dado por la ecuación (40) se convierte entonces un algoritmo de control derivado integral proporcional de 2DoF (dos grados de libertad), cuyo esquema se muestra en la Figura 14 y su señal de salida en la ecuación (41):

Figura 14

Diagrama de bloques 2DoF PID.



Nota. Tomado de : (Alfaro & Vilanova, 2016)

$$u(t) = K_p \left\{ (\beta r(t) - y(t)) + \frac{1}{T_i} \int_0^t (r(t) - y(t)) + T_d (\gamma r(t) - y(t)) \right\} \quad (41)$$

$$u(s) = K_p \left\{ (\beta r(s) - y(s)) + \frac{1}{T_i s} (r(s) - y(s)) + T_d s (\gamma r(s) - y(s)) \right\} \quad (42)$$

Donde:

$e(s)$: es la señal del error.

$r(s)$: es la señal de referencia del sistema.

$y(s)$: es la señal de salida del sistema

K_p : es la ganancia del controlador.

T_i : la constante de tiempo integral.

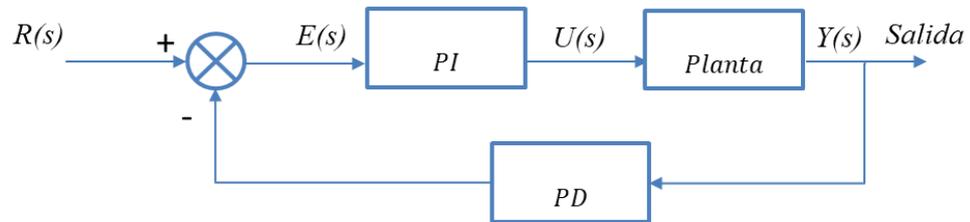
T_d : la constante de tiempo derivada.

β y γ : los pesos de punto de ajuste, y α la constante de filtro derivada.

El controlador de la ecuación (42), puede ser realizado bajo la forma de un controlador PI-PD como se muestra en la Figura 15.

Figura 15

Diagrama de bloques de un controlador PI-PD.



Nota. Tomado de : (Alfaro & Vilanova, 2016)

Donde:

$$PI: k_p' + \frac{k_i'}{s}$$

$$PD: 1 + k_d's$$

La relación de entrada-salida para el controlador es:

$$u(s) = k_p' r(s) + \frac{k_i}{s} [r(s) - y(s)] - (k_p' + k_d' k_i') y(s) + k_p' k_d' s y(s) \quad (43)$$

Operando algebraicamente sobre la ecuación (17) y comparando los coeficientes con la ecuación (43) se deducen las siguientes equivalencias:

$$Kp = kp' + k_d' k_i' \quad (44)$$

$$T_i = \frac{kp' + k_d' k_i'}{k_i'} \quad (45)$$

$$T_d = \frac{kp' k_d'}{kp' + k_d' k_i'} \quad (46)$$

$$\beta = \frac{kp'}{kp' + k_d' k_i'} \quad (47)$$

$$\gamma = 0 \quad (48)$$

Los valores obtenidos de las ecuaciones (44), (45), (46), (47) y (48) se utilizan para la forma de la ecuación (43).

La estructura definida por la Figura 10 es ajustable de manera más sencilla, ya que los parámetros K , T_i y T_d pueden ser determinados en primer término a partir de un controlador PI y PD. Una vez hecho esto, se puede ajustar la señal de salida, seleccionando los valores de los pesos con β y γ . (Cova, 2005)

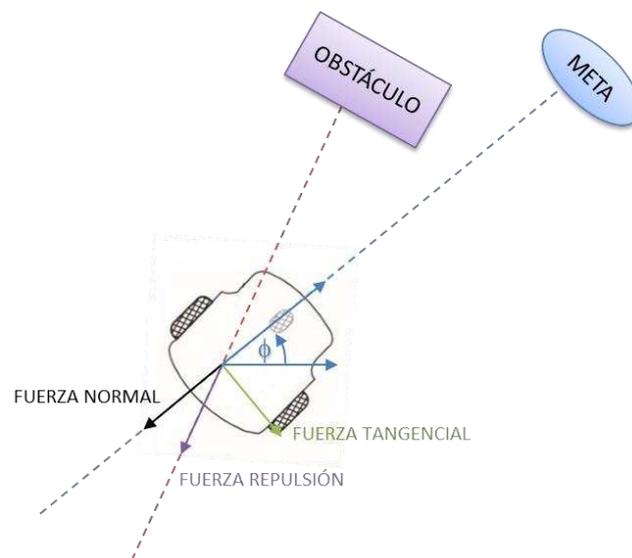
Control de Evasión de obstáculos por Impedancia

El control que se utilizará para la evasión de obstáculos está basado en la impedancia, el cual es un método muy utilizado en la evasión, y usa el concepto de impedancia generalizada o extendida para caracterizar la relación entre un robot móvil que se mueve hacia un obstáculo y una fuerza de repulsión ficticia (Ferreira, Pereira, Vassallo, Filho, & Filho, 2008).

En la evasión de obstáculos las fuerzas que se involucran son: la fuerza de repulsión, normal y tangencial, las mismas que aparecen representadas en la Figura 16:

Figura 16

Representación de las fuerzas ficticias en el método de la impedancia.



En este trabajo de investigación, se considera una trayectoria definida en el tiempo, por lo que es necesario realizar una modificación a este método.

De acuerdo a la teoría aplicada por (Capito, L., & Proaño, P., 2011): “Esta variación consiste en tomar el concepto de fuerza tangencial y transformarlo en un ángulo de rotación, y tomar el principio de variación de magnitud de la fuerza de repulsión para calcular el valor del ángulo de desviación”. Para ello es necesario, establecer dos variables para determinar el ángulo de desviación tales como:

Distancia de inicio de evasión d_{div} , se constituye como la distancia con la que el robot comienza a evadir el obstáculo.

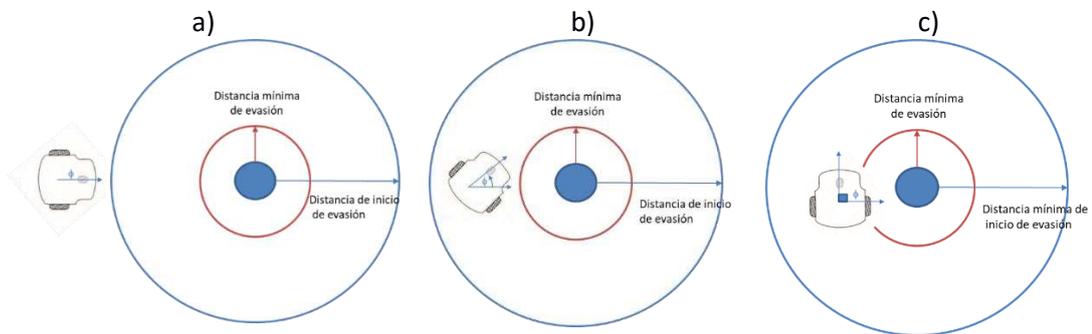
- Distancia mínima de evasión d_{min} , se constituye como la distancia que debe haber entre el obstáculo y el robot para poder evadir.

Cuya función dentro del método de evasión es el siguiente:

- Cuando el robot se encuentra a mayor distancia de la de inicio de evasión d_{div} , el ángulo de desviación tiene un valor de 0° , como se indica en la Figura 17 a).
- Cuando el robot se encuentra a menor distancia de la de inicio de evasión d_{div} , se comienza a calcular el ángulo de desviación provocando la rotación en sentido positivo o negativo, dependiendo de la posición del obstáculo, como se indica en la Figura 17 b).
- Cuando el robot rebasa el valor de la distancia mínima de evasión d_{min} , el ángulo de desviación tiene un valor de 90° , como se indica en la Figura 17 c).

Figura 17

Posiciones del robot móvil para evasión de obstáculos.



De esta manera, se obtiene un ángulo de desviación con valores de 0° hasta $\pm 90^\circ$, y que cambia su módulo dependiendo de la distancia a la cual se encuentra el robot del obstáculo, y la posición del obstáculo con respecto al robot.

Para calcular el ángulo de desviación se utiliza la fórmula (49):

$$\phi = \begin{cases} 0^\circ & , \text{ si } d(S) > d_{div} \\ \text{signo}(\sigma) \tan^{-1} \left(\frac{k}{|d(S) - d_{min}|} \right) & , \text{ si } d(S) \leq d_{div} \end{cases} \quad (49)$$

Donde:

$d(S)$: distancia que existe entre el obstáculo y el robot, detectada por el sensor.

d_{min} : distancia mínima para la evasión del obstáculo.

d_{div} : distancia de inicio de evasión del obstáculo.

ϕ : ángulo de desviación.

σ : variable que genera la orientación de giro.

k : constante que calibra la rapidez de giro del ángulo de desviación que tiende a $\pm 90^\circ$.

Luego, se aplica la matriz de rotación inversa de la ecuación (50) utilizando el ángulo de desviación calculada, utilizando las velocidades del robot con respecto al sistema inercial \dot{X} y \dot{Y} como entradas, calculando nuevas velocidades para cambiar la posición del robot cuando éste detecte un obstáculo.

$$\begin{bmatrix} h\dot{X} \\ h\dot{Y} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \quad (50)$$

Posteriormente se añade una variable β , que se encarga de regular la velocidad con la que el robot evade el obstáculo, la cual no se realiza a la velocidad actual, pues existe una gran posibilidad de colisión, si no se realiza este cambio, como se indica en la ecuación (51) :

$$\beta = \begin{cases} 0 & , \text{ si } d(S) > d_{div} \\]0,1[& , \text{ si } d(S) \leq d_{div} \end{cases} \quad (51)$$

Finalmente, en la ecuación (52) se incorpora todos los parámetros añadidos a la matriz de rotación con el ángulo de desviación para el cálculo de las nuevas velocidades:

$$\begin{bmatrix} h\dot{X} \\ h\dot{Y} \end{bmatrix} = \begin{cases} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} & , \text{ si } d(S) > d_{div} \\ \beta \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} & , \text{ si } d(S) \leq d_{div} \end{cases} \quad (52)$$

Planificación de trayectorias

Para este trabajo de investigación se utilizarán cuatro métodos para planificación de trayectorias: A*, RRT, RRT* e híbrido A*.

Algoritmo A*

Para el desarrollo del algoritmo de búsqueda A*, se toma como referencia, el algoritmo interpretado por (Ueland, 2019), el cual utiliza la estructura del objeto de estudio como información extra para encontrar de forma más rápida su objetivo final, con una ruta de menor costo mediante la utilización de una función heurística, y así la encontrar el punto inicial al final.

La función heurística que utiliza el algoritmo A* se define por la ecuación (53) :

$$f(n) = h'(n) + g(n) \quad (53)$$

Donde:

$f(n)$: coste mínimo de la ruta que pasa por n .

n : es el nodo a evaluar desde su actual hasta su final.

$h'(n)$: representa el valor heurístico del nodo a evaluar desde el actual n , hasta el final.

$g(n)$: el costo de movimiento para pasar del punto de partida a una celda dada en la cuadrícula, siguiendo el camino generado para llegar.

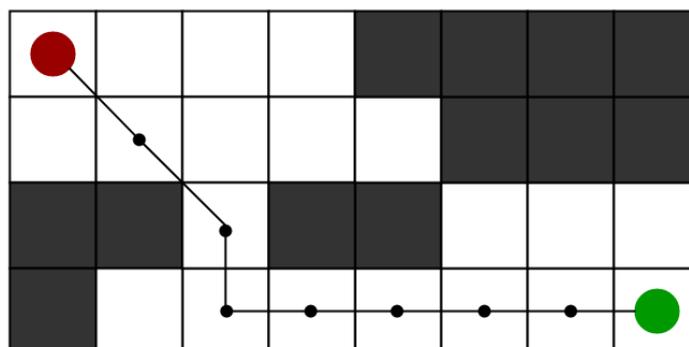
El algoritmo también utiliza estructuras de datos auxiliares que se denomina “abiertos” y “cerrados”, el primero implementada como una cola de prioridad para los valores de $f(n)$ de cada nodo, y la segunda, el lugar donde se guarda la información de los datos que han sido evaluados.

Luego, el algoritmo realiza la expansión del primer nodo que esté en los datos “abiertos” como “hijos” del mismo, y en caso que no sea el nodo, el objetivo, se calcula la $f(n)$ de todos sus hijos como se indica la Figura 18, insertándolos en los datos “abiertos”, y trasladando el nodo analizado a los datos “cerrados”.

Finalmente, A* equilibra los valores de $h(n)$ y $g(n)$ a medida que avanza desde el punto de partida hasta la meta. Cada vez que pasa por el bucle principal, examina el vértice que tiene el más bajo $f(n) = g(n) + h(n)$. (Swift, 2017).

Figura 18

Planificación de la trayectoria A.*



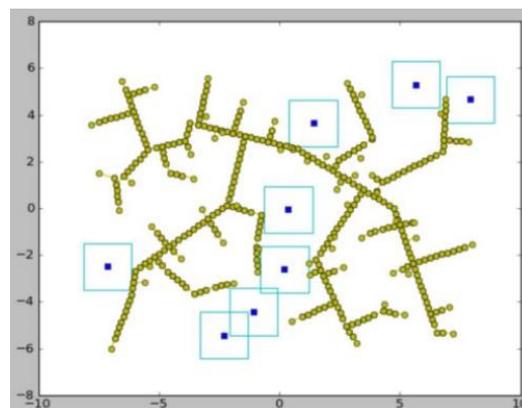
En la ilustración se observa que una vez que superamos el obstáculo, el algoritmo prioriza el nodo con la menor $f(n)$ y la mejor posibilidad de llegar al final. Tomado de:

Algoritmo RRT

El algoritmo RRT se basa en la generación de nodos aleatorios y conectar al nodo disponible más cercano, desde un punto inicial a punto deseado. Cada nodo que se crea, debe verificarse que se encuentre fuera de un obstáculo. Además, la conexión del nodo principal al nodo vecino más cercano también debe evitar los obstáculos. (López, Brav, Cuesta, & Ollero, 2006) El algoritmo finaliza cuando se genera un nodo dentro de la región objetivo o se alcanza un límite. Es necesario calcular el vector que forma la distancia más corta entre el nuevo vértice y el borde más cercano. En el punto de intersección, se agrega un nuevo nodo al borde y se conecta al vértice generado aleatoriamente. Alternativamente, el vértice se puede unir al nodo más cercano conectado un enlace de nodos discretizados a él. El algoritmo RRT produce gráficos muy cúbicos como se muestra en la Figura 19, debido a que los nodos están unidos a su nodo más cercano. La naturaleza estructural de las conexiones dificulta la probabilidad de encontrar una ruta óptima. (Chin, 2019)

Figura 19

Planificación de trayectoria RRT.



Nota. Tomando de: (Chin, 2019).

Algoritmo RRT*

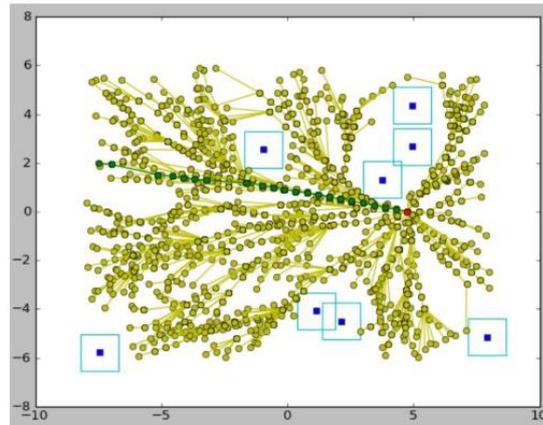
El algoritmo RRT * es una versión optimizada del algoritmo RRT. Cuando el número de nodos se aproxima al infinito, el algoritmo RRT * entregará la ruta más corta posible hacia el objetivo. Si bien no es factible, esta afirmación sugiere que el algoritmo funciona para desarrollar un camino más corto. El principio básico de RRT * es el mismo que RRT, pero dos características son clave al algoritmo dan como resultado resultados significativamente diferentes.

En primer lugar, el algoritmo RRT * registra la distancia que ha recorrido cada nodo en relación con su nodo principal, esto se conoce como el coste del nodo. Después de encontrar el nodo más cercano en el espacio de búsqueda, se examinan los nodos vecinos en un radio fijo desde el nuevo nodo. Si se encuentra un nodo con un coste más barato que el proximal, el nodo más económico reemplaza al nodo proximal. El efecto de esta característica se puede ver con la adición de ramas en forma de abanico en la estructura del árbol. Se elimina la estructura cúbica de RRT.

La segunda diferencia que agrega al algoritmo RRT * es la conexión del árbol. Después de que un nodo se ha conectado al nodo más económico, los nodos vecinos se examinan nuevamente y se verifica, si volver a conectarse al nuevo nodo agregado, hará que su costo disminuya. Si el coste disminuye, el nodo vecino se vuelve a conectar al nodo recién agregado. Esta característica hace que el camino sea más suave como se ilustra en la Figura 20. (Chin, 2019)

Figura 20

Planificación de trayectorias RRT.*



Nota. Tomado de: (Chin, 2019).

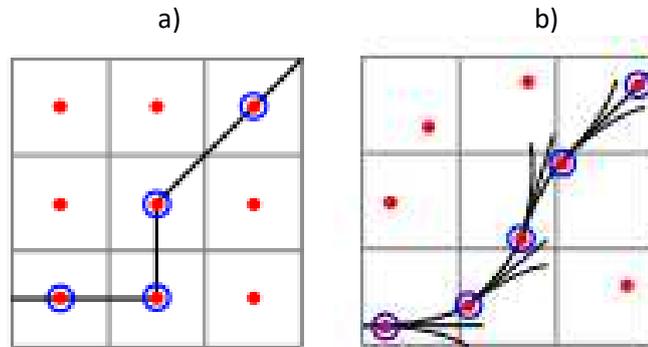
Algoritmo Híbrido A*

Al igual que en el algoritmo A* convencional, el algoritmo híbrido A* requiere que el espacio de búsqueda (x, y, θ) esté discretizado, pero a diferencia del A* que asocia costos con centros de celdas y solo recorre estados que corresponden a centros de celdas de cuadrícula, el algoritmo híbrido A* asocia un estado continuo con cada celda y la puntuación de la celda es el costo de su estado continuo asociado como se indica en Figura 21. (Dolgov, Thrun, Montemerlo, & Diebel, 2008). El algoritmo híbrido A* es capaz de tener en cuenta la naturaleza continua del espacio de búsqueda que representa el mundo real. Esto se logra mediante el uso de un conjunto de primitivas de movimiento precalculadas (los cuales codifican una trayectoria específica del móvil) para determinar estados accesibles y, por lo tanto, poder construir un árbol de búsqueda en línea. La posición continua, a la que llega una primitiva de movimiento se almacena junto con la posición discreta en la celda correspondiente. Éstas primitivas están definidas mediante arcos, cuya longitud debe ser lo suficientemente larga como para que el móvil, sobresalga de la celda del espacio discretizado. La curvatura está limitada por el ángulo de la dirección máxima, cuyo cambio

de rumbo debe ser múltiplo del tamaño del paso de la discretización de la dimensión del espacio buscado. (Petereit, Emter, & Frey)

Figura 21

Diferencia entre el Algoritmo A e Híbrido A*.*



Nota. Tomado de: (Dolgov, Thrun, Montemerlo, & Diebel, 2008).

El algoritmo híbrido A*, no garantiza que encuentre la solución de costo mínimo, debido a su fusión de estados de coordenadas continuas que ocupan la misma celda en el espacio discreto. Sin embargo, si garantiza que la ruta resultante sea manejable (en lugar de ser lineal por partes como en el caso del algoritmo estándar A*).

Capítulo III

Generalidades del sistema

Plataforma Pioneer P3-DX

Un robot Pioneer es una plataforma robótica cuya finalidad es la investigación, así como también para realizar aplicaciones tanto en el ámbito educativo como el cotidiano. (Flores & Proaño, 2013). En este caso, se toma a la plataforma Pioneer P3-DX, debido a su fácil configuración y modelamiento matemático.

Hardware del robot

El Pioneer P3-DX tiene capacidad para 3 baterías de 12 voltios los cuales pueden entregar una potencia de 252 vatios si se encuentran cargadas totalmente. Tiene unos motores potentes y llantas de 19 cm de diámetro con las que puede alcanzar velocidades de 1.2 m/s y llevar cargas de hasta 17 Kg. (Morales & Jaramillo, 2010)

Actualmente, el laboratorio de robótica cuenta con tres Pioneer P3-DX, de los cuales dos se habilitaron debido a que manejan el entorno de programación que se está trabajando y se puede utilizar para la realización de diversas pruebas, su reincorporación se realizó mediante el cambio de sus baterías antiguas por unas nuevas de marca Famma de 12v/33ah (voltios/amperio-hora) mostrado en la Figura 22. Únicamente, se usó dos baterías que fueron suficientes para lograr que el Pioneer se mueva sin problemas. Además, se realizó una serie de pruebas para determinar el tiempo de uso, llegando a la conclusión de que tiene una autonomía de hasta 6 horas por 4 días sin cargar.

Figura 22

Cambio de baterías al Pioneer.



El P3-DX es de código abierto y completamente programable. Además, se le pueden incorporar varios dispositivos y accesorios extra para deberes para las que no fue previamente programado desde un principio, tales como: sensores ultrasónicos láser, parachoques, visión, cámara, y brújula, que pueden usarse para ejecutar aplicaciones avanzadas como la navegación.

A continuación se detalla las especificaciones técnicas del robot (Manual, 2006).

Especificaciones:

- Dimensiones: 455x381x237mm
- Velocidad máxima: 1.2m/s
- Velocidad de giro: 300°/s
- Peso: 9kg
- Carga máxima: 17kg
- Voltaje: 12V
- Autonomía: 8-10h
- Tiempo de carga: 12h o 2.4h (cargador especial).

Fuselaje

- Carcasa de aluminio de 1.6mm
- 2 Ruedas de goma de 190 mm de diámetro.
- Rueda trasera giratoria de 62mm.

Controlador a bordo

- Cobra EBX-12 Versalogic.
- Controlador diferencial de conducción.
- Tarjeta de red Ethernet con antena Wi-Fi.

Software

- S.O. Debian (ARCOS)
- Pioneer SDK

Accesorios y sensores

- Bumpers frontales para colisiones
- Array frontal de 8 sensores Sonar
- Láser SICK LMS-500 190º 75Hz 0-80m
- Brújula
- Cámara Canon VC-C50i 3-ejes 460p Zoom óptico 26x
- Giroscopio

Alimentación

- Arreglo de 3 baterías 12V 2.5A.

Conexión al robot

Antes de encender el robot, se recomienda siempre seguir el siguiente procedimiento para evitar problemas en la conexión:

- Poner en ON el interruptor de la alimentación.
- Conectar del computador personal al puerto serial mediante el cable serial –USB como se indica en la Figura 23.

navegación y localización láser, MOGS para el uso de antenas GPS y SONARNL, usado especialmente con los detectores de ultrasonidos de corto alcance que incorporan la mayoría de robots móviles.

MobileSim: es un simulador con el que se pueden realizar las pruebas de los programas realizados antes de ejecutarlas en el robot físico. Incluye una interfaz en la que se puede incorporar un mapa conocido con extensión (. map) y en el que se puede seleccionar el modelo del robot deseado, como si se tratase del real. Dichos modelos incluyen los sensores tales como: láseres, sonars, bumpers, etc., con el fin de obtener una simulación casi real. (Martínez D. J., 2014)

Mapper3: es una aplicación que permite crear diferentes entornos añadiendo prohibidas, zonas descarga, obstáculos y otras opciones. Esta aplicación se guarda en un archivo con extensión (. map, .JPEG), e incorporase en el simulador MobileSim para la simulación del robot con un entorno más realista.

Matlab: es el programa con el cual se va realizar la programación de los diferentes tipos de sistemas de control, además de la cinemática diferencial del robot y generación de trayectorias. Todo esto se puede lograr mediante la ejecución de una serie de librerías que pueden extraer funciones del ARIA, para lograr el movimiento del robot sin la necesidad programar en el entorno del ARIA.

Interfaz Aria Matlab

Los paquetes de ARIA-Matlab se le pueden encontrar en la siguiente dirección web: <https://github.com/reed-adept/aria-matlab> ya que es un software libre.

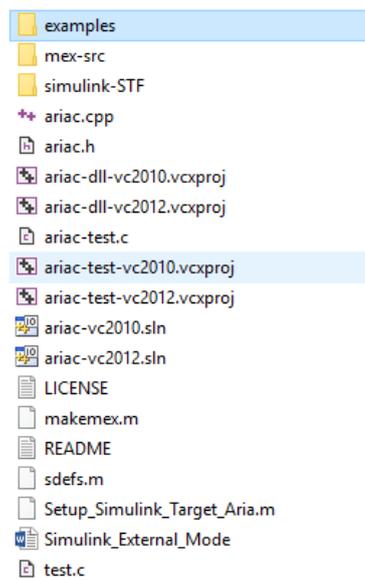
Para la instalación debe tener los siguientes programas:

- ARIA 2.7.5.2 o posterior.
- Visual Studio 2010
- Matlab 2012b hasta el actual 2019b
- Windows 7, 8 o 10

La interfaz ARIA-Matlab se logra mediante una colección de archivos de origen C de extensión *.cpp* (en la carpeta *mex-src*) mostrado en la Figura 24 que contienen los comandos para el movimiento del robot; que al compilar el programa *makemex.m* de Matlab ejecuta la interfaz MEX (subrutinas enlazada dinámicamente ejecutada en el entorno Matlab).

Figura 24

Paquete para la interfaz Aria-Matlab.



En la Figura 25, se observa los archivos creados con extensión *.mexw64* (archivo de funciones compilado de Matlab) y *.mexw64.pdb* (archivo de depuración que no expone su algoritmo de código fuente) que se utilizan para el uso de los en Matlab.

Figura 25

Archivos de depuración para los comandos de Aria.

 arrobot_connect.mexw64	5/12/2019 18:37	Archivo MEXW64
 arrobot_connect.mexw64.pdb	5/12/2019 18:37	Program Debug Database

A continuación se enlista los comandos disponibles de ARIA que se pueden ejecutar en Matlab (Hedges, 2013):

Tabla 1

Lista de comandos obtenidos de ARIA utilizados en Matlab.

Funciones Aria -Matlab	
COMANDO	DESCRIPCIÓN
arrobot_connect	Conecta al robot Pioneer
arrobot_disconnect	Desconecta al robot Pioneer
arrobot_getx	Posición X del robot móvil (<i>mm</i>).
arrobot_gety	Posición y del robot móvil (<i>mm</i>).
arrobot_getth	Orientación (theta) del robot móvil (<i>grados</i>).
arrobot_setvel	Enviar comando para establecer la velocidad de avance / retroceso del robot móvil (<i>mm/seg</i>).
arrobot_setrotvel	Enviar comando para establecer la velocidad de rotación del robot móvil (<i>grados / seg</i>), en sentido antihorario +.
arrobot_setlatvel	Enviar comando para establecer la velocidad lateral del robot móvil (<i>mm / seg</i>), hacia la derecha +.
arrobot_setwheelvels	Enviar comando para configurar las velocidades de la rueda izquierda, derecha por separado (<i>mm / seg</i>), anula vel, rotvel.
arrobot_getvel	Obtener la velocidad actual de avance / retroceso del robot móvil (<i>mm / seg</i>).
arrobot_getrotvel	Obtener la velocidad de rotación actual del robot móvil (<i>grados / seg</i>), en sentido antihorario +.
arrobot_stop	Enviar comando para detener el movimiento del robot.
arrobot_setdeltaheading	Solicitar un desplazamiento de rumbo para activar.
arrobot_getlatvel	Velocidad lateral actual del robot móvil (<i>mm / seg</i>).
arrobot_move	Solicitar una distancia de traducción (adelante / atrás) para conducir el robot.
arrobot_resetpos	Restablecer la estimación de pose odométrica del robot móvil a 0,0,0 (no mueve el robot en la simulación).
arrobot_setpose	Cambiar estimación de pose de robot almacenada.
arrobot_getbatteryvoltage	Voltaje de la batería del robot móvil.
arrobot_length	Longitud del robot móvil (<i>mm</i>).
arrobot_width	Ancho del robot móvil (<i>mm</i>).

CONTINÚA



arrobot_radius	Radio (máximo) del robot móvil (<i>mm</i>).
arrobot_getnumsonar	Número de sensores de sonda en robot móvil.
arrobot_getsonarrange	Obtener lectura de rango de un sensor de sonda (indexado a 0) del robot móvil (<i>mm</i>).
arrobot_getsonar	Obtener lecturas de alcance de todos los sonares frontales en el robot móvil (<i>mm</i>).
arrobot_isstalled	¿Están detenidos los motores?
arrobot_isleftstalled	¿El motor izquierdo está parado?
arrobot_isrightstalled	¿Está el motor derecho parado?
arrobot_getnumfrontbumpers	Número de parachoques delanteros en robot móvil.
arrobot_getnumrearbumpers	Número de parachoques traseros en robot móvil.
arrobot_getfrontbumper	Obtener lectura de rango de un parachoques delantero (indexado en 0) del robot móvil.
arrobot_getrearbumper	Obtenga la lectura de rango de un parachoques delantero (indexado en 0) del robot móvil.
arrobot_getdigin	Obtener el estado actual de las entradas digitales (solo Pioneer).
arrobot_setdigout	Establecer estado de salidas digitales (solo Pioneer).
arrobot_motorsenabled	¿Están habilitados los motores?
arrobot_enable_motors	Enviar comando para habilitar o volver a habilitar motores.
arrobot_disable_motors	Enviar comando para deshabilitar motores.

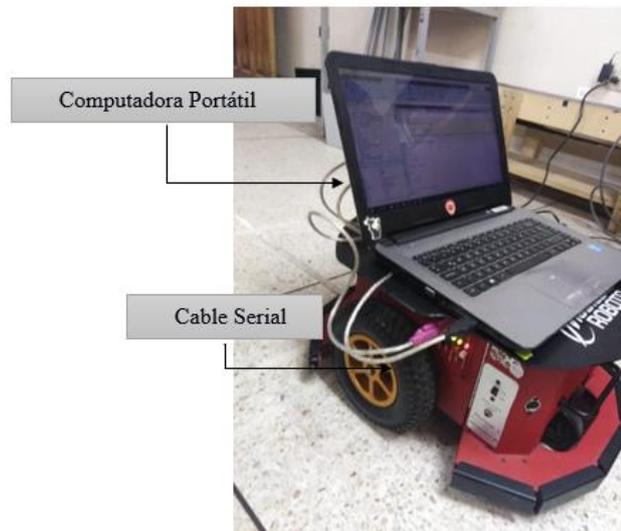
De la tabla 1, se utiliza la función `arrobot_setwhelvels`, para mandar velocidades independientes al robot Pioneer P3-DX, y las funciones `arrobot_getx`, `arrobot_gety` y `arrobot_getth`, para poder estimar su posición odométrica.

Arquitectura del robot

La plataforma Pioneer P3-DX utiliza una arquitectura cliente-servidor, donde el servidor es el Pioneer y el cliente es una computadora portátil conectada a través de un cable serial USB como se observa en la Figura 26, el cual se encarga de transmitir las instrucciones a la plataforma móvil.

Figura 26

Forma de conexión cliente-servidor.



El conector SERIAL, contiene un indicador de datos entrantes y salientes LED (RX y TX, respectivamente), donde éste interactúa con el microcontrolador ARCOS desde una computadora portátil para el control del modo cliente-servidor. El puerto SERIAL es compartido internamente por el puerto serie HOST, al que se conecta la computadora portátil o un dispositivo Ethernet-SERIAL. Se puede usar el conector SERIAL o HOST para la comunicación del modo cliente-servidor.

Capítulo IV

Sistema de navegación

En este capítulo se pretende incorporar el modelo diferencial cinemático directo e inverso, cuyo objetivo es el uso de las velocidades de las ruedas del robot para la movilización, integrando las estrategias de control para el seguimiento de trayectorias y posteriormente la evasión de obstáculos, los cuales se probarán en dos tipos de generación de trayectorias: planificación de trayectorias y fórmulas matemáticas. Esta implementación se realizará en el robot Pioneer P3-DX, que se encuentra disponible en las instalaciones de la Universidad de las Fuerzas Armadas ESPE.

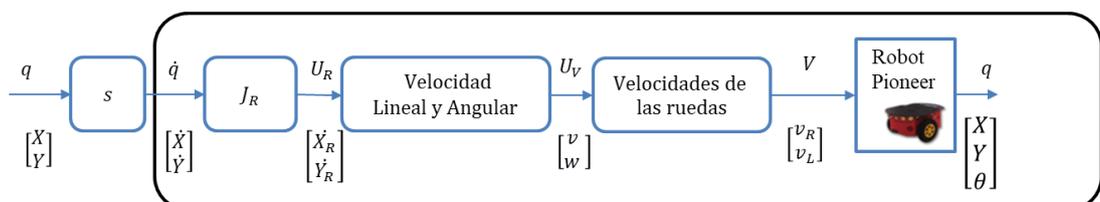
Estimación del modelo del robot móvil

Para el diseño de los diferentes sistemas de control es necesario estimar la planta en primer lugar, y poder obtener los diferentes parámetros que se utilizará para el control del seguimiento de trayectorias.

Para ello, se utiliza la cinemática diferencial inversa del robot mostrada en la Figura 13, debido a que se puede calcular las velocidades de las ruedas que se envían al robot, en función de las posiciones en el eje X y Y que describen una trayectoria específica, como se muestra en la Figura 27. Por lo tanto, también es necesario considerar las dimensiones físicas del robot Pioneer P3-DX, las cuales son: $r = 0.095m$ y $L = 0.275m$.

Figura 27

Modelo cinemático diferencial inverso del robot Pioneer P3-DX.



Donde:

J_R : matriz de rotación.

$$\text{Velocidades de las ruedas: } \begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{0.275} \\ \frac{1}{2} & -\frac{1}{0.275} \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix},$$

Una vez obtenida la estructura del modelo, se utiliza la identificación paramétrica para obtener una descripción externa del sistema, en forma de funciones de transferencia en el dominio del tiempo, a partir de los datos de entrada-salida medidos. Como datos de entrada tenemos las velocidades del robot con respecto al sistema inercial $[\dot{X}, \dot{Y}]$, las cuales se obtendrán derivando las posiciones que se ingresan a la planta, y como salidas la posición final del robot $[X, Y]$, obtenidas de la odometría del robot Pioneer, ver Figura 28. Luego, este sistema aproximado se utilizará en aplicaciones de control para la obtención de sus parámetros.

Figura 28

Entradas y salidas de la planta.



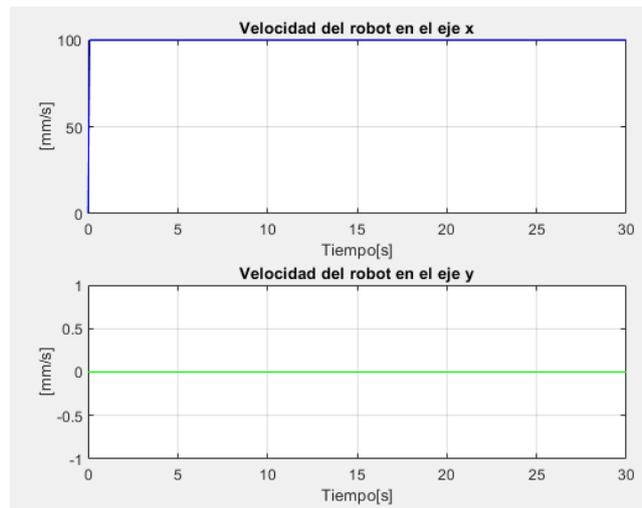
Partiendo de la interfaz de Matlab se envió al robot físico una determinada velocidad, mediante el ingreso de un tiempo y una trayectoria deseada sin aplicar ningún tipo de control. El tiempo de trayectoria que se ingresó fue de 30 segundos, con una distancia de 3000 mm, por lo tanto, aplicando las fórmula de la velocidad media, $vm = \frac{\Delta X}{\Delta t}$; se calcula que la velocidad en el eje X es de $100 \frac{mm}{s}$, la misma que se puede interpretar como la velocidad del robot.

Antes de realizar la identificación del sistema se procede en primer lugar, a determinar el comportamiento de las señales de entradas $[\dot{X}, \dot{Y}]$; y salidas del robot Pioneer $[X, Y]$. Ésta proceso se realiza con el fin de comparar la respuesta del modelo real con el modelo que se va a estimar.

En la Figura 29, se observa que la velocidad del robot con respecto al sistema inercial en el eje X es de $100 \frac{mm}{s}$. Debido a que únicamente se realizó un movimiento traslacional, y no uno lateral, por lo tanto la velocidad en el eje Y es 0.

Figura 29

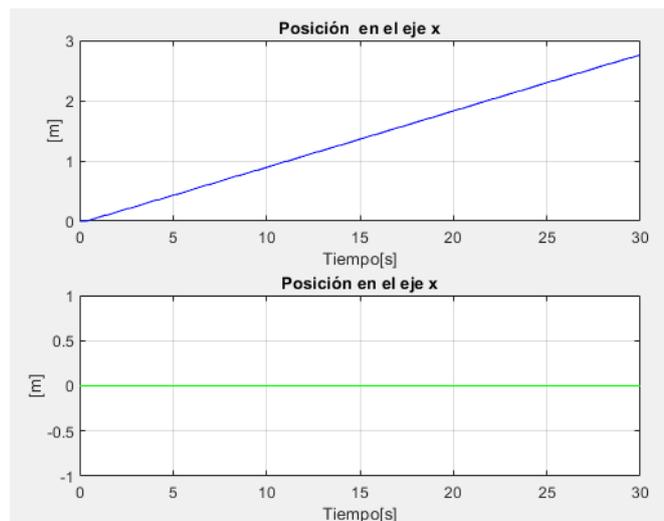
Señales de entrada del robot Pioneer.



En la Figura 30, que la posición final del robot en el eje X es de 3 m, mientras que en el eje Y es 0, ya que solo hay movimiento traslacional.

Figura 30

Señales de salida del robot Pioneer.



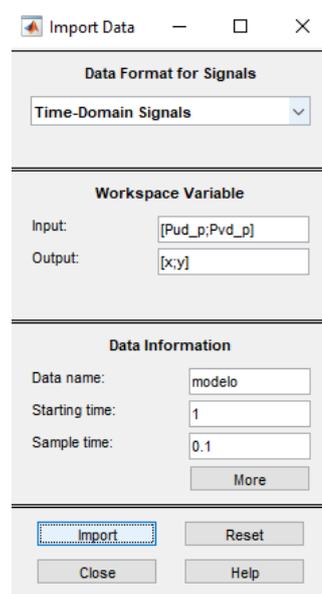
Para estimar la planta se utilizó el toolbox de Identificación de Sistemas de Matlab.

La herramienta que permite describir la dinámica de un sistema a partir de datos temporales

es la función “*ident*”. Debido a que la planta es un sistema MIMO (múltiples entradas, múltiples salidas), en el parámetro de entrada se colocó las velocidades del robot con respecto al sistema inercial $[\dot{X}, \dot{Y}]$ (en el programa se nombraron como Pud_p a \dot{X} , y Pvd_p a \dot{Y}), obteniendo a la salida del sistema la posición del robot $[X, Y]$ como se indica en la Figura 31, con un tiempo de muestreo de 0.1 segundos.

Figura 31

Importar datos a la app de 'ident' de Matlab.

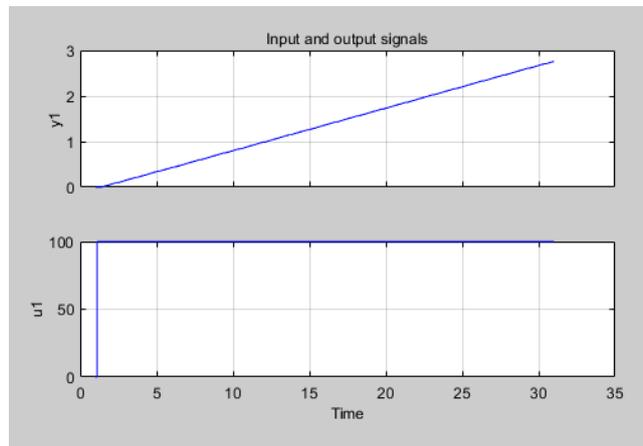


Una vez ingresado los parámetros de la planta se verifica que el modelo ingresado en la función “*ident*”, tenga las mismas características de las señales de entrada y salida que el robot físico.

En la Figura 32 se observa que la señal entrada (nombrada por la herramienta u1), se mantiene a un valor de $100 \frac{mm}{s}$, y la salida X (nombrada por la herramienta como y1) alcanza la distancia de 3 m en el tiempo de los 30 segundos.

Figura 32

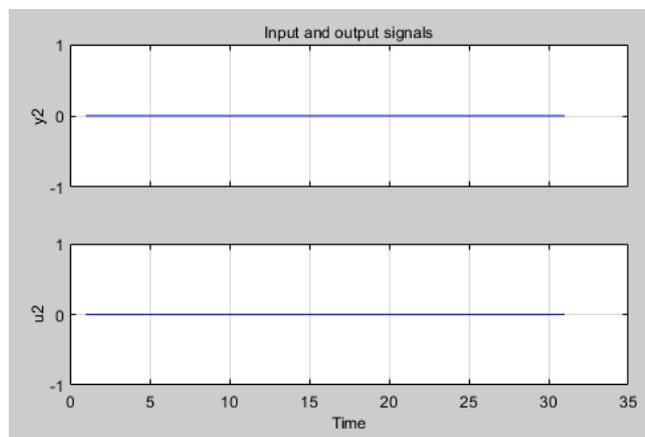
Comportamiento de la señal entrada \dot{X} de salida X en la función "ident".



En la Figura 33 se observa que la señal entrada \dot{Y} (nombrada por la herramienta $u2$) y la salida Y (nombrada por la herramienta como $y2$) se mantiene a un valor de 0, ya que no existe movimiento lateral.

Figura 33

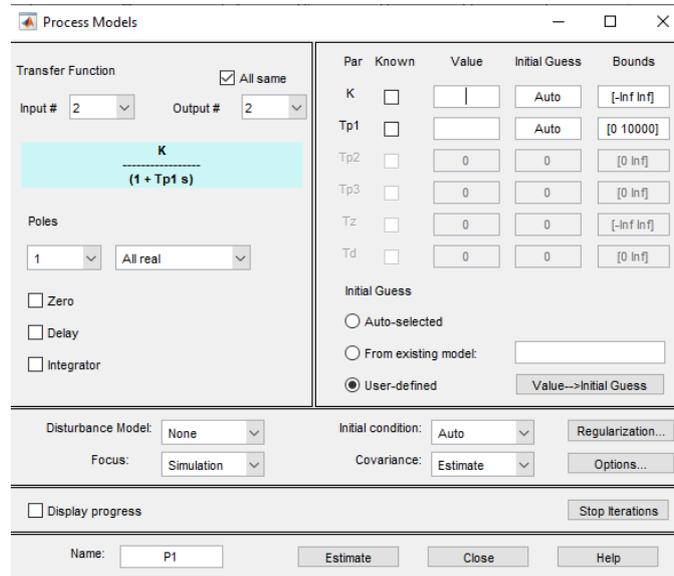
Comportamiento de la señal entrada \dot{Y} de salida Y en la función "ident".



Luego, a través del método de selección "Process Models" de la función de "ident" de Matlab indicada en la Figura 34, se escogió el tipo de función de transferencia que representará la planta estimada. La función de transferencia escogida es de segundo orden, debido a su simplicidad y fácil análisis en el desarrollo de sistemas de control.

Figura 34

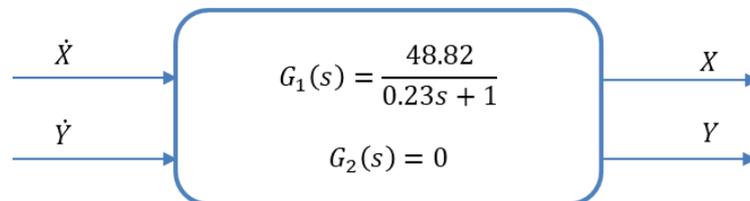
Selección de la función de transferencia de segundo orden.



En la Figura 35, se presenta las funciones de transferencia desacopladas, generadas para cada entrada y salida de la planta.

Figura 35

Funciones de transferencia desacopladas.



Las función de transferencia de $G_1(s)$ indica que hay movimiento únicamente en el eje X , mientras que en $G_2(s)$, no existe movimiento en el eje Y . Por lo tanto, el diseño del controlador se realizará con respecto a las función de transferencia $G_1(s)$.

A continuación, en la Figura 36 y 37 se muestra las señales de salida del modelo estimado, en el cual se observa que existe un error de estimación menor al 0.05 % con respecto al modelo real.

Figura 36

Estimación de la planta del robot móvil en la salida X .

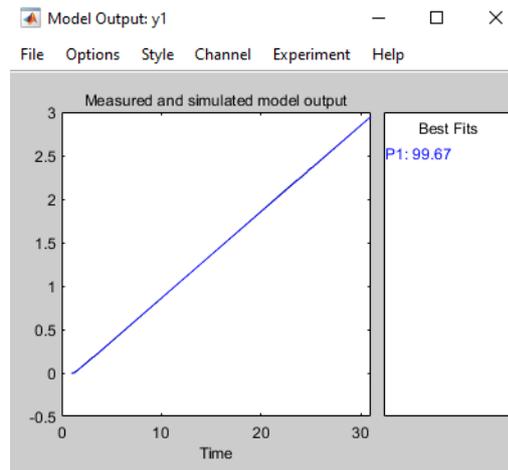
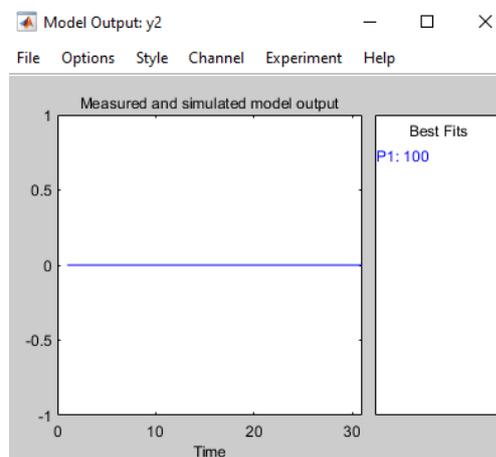


Figura 37

Estimación de la planta del robot móvil en la salida Y.



Diseño de controladores

Debido a que el robot no cuenta con sensores para medir velocidad, se utilizará el sistema odométrico que cuenta el robot Pioneer para poder estimar la posición en la que se encuentra el robot, y compararlas con las posiciones deseadas, realizando un control de posición, mediante el control de las velocidades. Para el desarrollo de los controladores se toma en consideración las funciones de transferencia obtenidas en la estimación de la planta.

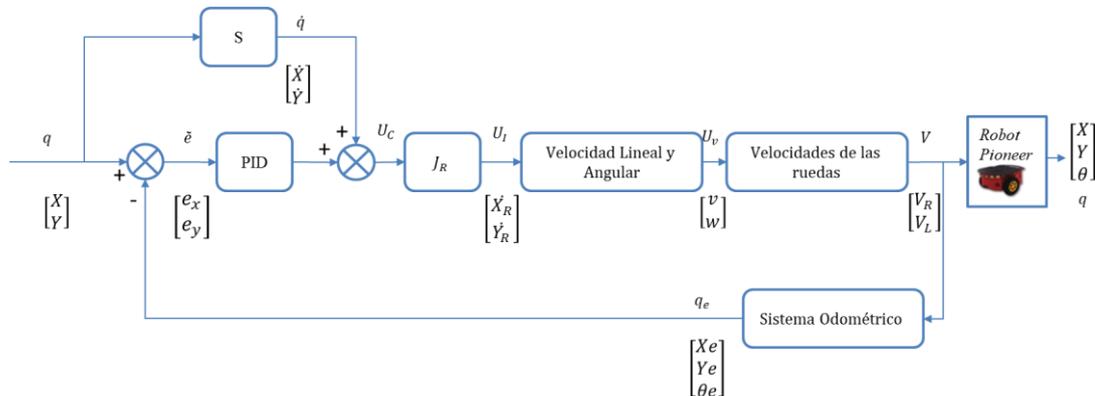
Controlador PID

Para el control de posición sobre una trayectoria del robot móvil se implementó un controlador PID, que permite direccionar de forma adecuada la plataforma hacia los puntos de la trayectoria planificada y seguirla, esto se logra mediante el control de la velocidad de la

rueda derecha e izquierda que se manda al robot móvil, en la Figura 38 se observa el esquema de control establecido.

Figura 38

Esquema de control PID.



Donde:

q_e : posición odométrico del robot.

\tilde{e} : es el vector del error del eje x y y (diferencia entre la posición de referencia y la posición odométrica del robot, ecuación (54).

$$\tilde{e} = q_{ref} - q_e \quad (54)$$

U_c : señal de salida del controlador.

La ley de control propuesta se expresa por la ecuación (55), ésta se aplica a las velocidades de los ejes \dot{X}_R y \dot{Y}_R del sistema del robot, haciendo que las variables vayan asintóticamente a cero, las cuales están directamente relacionadas con las velocidades lineal y angular v y w , y posteriormente con las velocidad de las ruedas, reduciendo de esta forma el error de posición.

$$U_c = \left(\dot{q} + K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t) + T_d \frac{de(t)}{dt} \right\} \right) \quad (55)$$

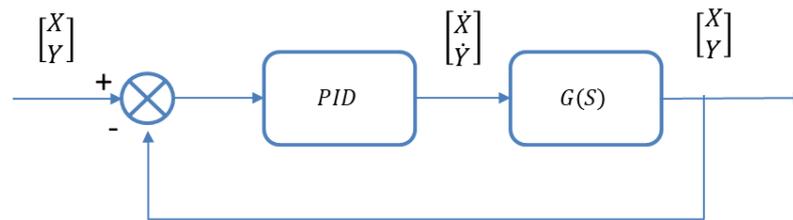
Cálculo de los Parámetros del Controlador PID

El modelo obtenido en la estimación de la planta tiene como entradas las velocidades del robot con respecto al sistema inercial $[\dot{X}, \dot{Y}]$, pero para poder obtener éstas

velocidades es necesario derivar posiciones, por lo tanto se debe acoplar el sistema de tal forma que tenga como entradas esas posiciones $[X, Y]$, y como salidas las posiciones del robot $[X, Y]$, agregando el controlador y la función del modelo estimado al mismo modelo, como se indica en la Figura 39.

Figura 39

Esquema de diseño del controlador.



Mediante el método de ubicación de polos se procede a diseñar un controlador para el modelo obtenido de la planta, haciendo uso de la forma general del controlador PID presentado en la ecuación (56) y la función de transferencia de la ecuación $G1(s)$, la cual es la misma para la rueda derecha e izquierda.

$$C(s) = kp \left(1 + \frac{1}{ki s} + kds \right) \quad (56)$$

Obteniendo la función de transferencia de lazo cerrado del sistema de control propuesto, se tiene la expresión (57):

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad (57)$$

$$H(s) = \frac{kd \cdot ki \cdot kp \cdot s^2 + ki \cdot kp \cdot s + kp}{s^2 \left(5.24 \times 10^{-3} ki + kd \cdot ki \cdot kp \right) + s(0.02 ki + kp \cdot ki) + kp} \quad (58)$$

Donde el polinomio característico del sistema de control, es el denominador de la expresión (58):

$$p(s) = s^2 \left(5.24 \times 10^{-3} ki + kd \cdot ki \cdot kp \right) + s(0.02 ki + kp ki) + kp \quad (59)$$

Por lo tanto, se desea que el polinomio característico de (59) tenga la forma estándar de una función de transferencia de segundo orden, como se indica en la ecuación (60):

$$p(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (60)$$

Donde:

ω_n : es la frecuencia natural no amortiguada (rad/s).

ζ : es la constante de amortiguamiento relativo del sistema.

Tomando como primera condición se propone un sobre impulso de $M_p = 0\%$.

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \quad (61)$$

Resolviendo la ecuación (61), se tiene que el factor de amortiguamiento es:

$$\zeta = 1$$

Como segunda condición de diseño se propone que el tiempo de establecimiento del sistema sea de 5 segundos con una tolerancia de 2%.

$$t_s = \frac{4}{\zeta\omega_n} \quad (62)$$

Para hallar la frecuencia natural ω_n se despeja de la ecuación (62) y se reemplaza los valores obtenidos y propuestos para el diseño de control.

$$\omega_n = \frac{4}{\zeta t_s} = \frac{4}{1 \times 4} = 1 \text{ rad/s}$$

Reemplazando ω_n y ζ en el polinomio característico deseado de la ecuación (47):

$$p(s) = s^2 + s \cdot 1.6 + 0.64 \quad (63)$$

Igualando los coeficientes de las ecuaciones (47) y ecuaciones (63):

$$5.24 \times 10^{-3} k_i + k_d \cdot k_i \cdot k = 1 \quad (64)$$

$$0.02 k_i + k_p \cdot k_i = 1.6 \quad (65)$$

$$k_p = 0.64 \quad (66)$$

Despejando las ecuaciones (64), (65) y (66):

$$k_p = 0.64$$

$$k_i = 2.42$$

$$k_d = 0.65$$

A continuación, se verifica los valores de las ganancias obtenidas en el análisis del controlador PID, graficando la respuesta a un escalón unitario, sin embargo, debido a que en el diseño no se consideró el bloque de la función derivativa del sistema de control de la Figura 38 en el esquema de control de la Figura 39, se realiza una comparación de las respuestas de las funciones de transferencia en lazo cerrado, analizando el efecto que tendría al añadir o quitar el bloque derivativo.

En la Figura 40, se observa la respuesta a un escalón unitario de dos funciones de transferencia en lazo cerrado:

La primera función (llamaremos F1) se basa en el esquema de la Figura 39, considerando únicamente la función de transferencia del bloque PID, la cual contiene los términos proporcional, integral y derivativo del controlador, ésta se multiplica a la función de transferencia de la planta G_1 , obteniendo un estado estacionario de amplitud 1, en lazo cerrado.

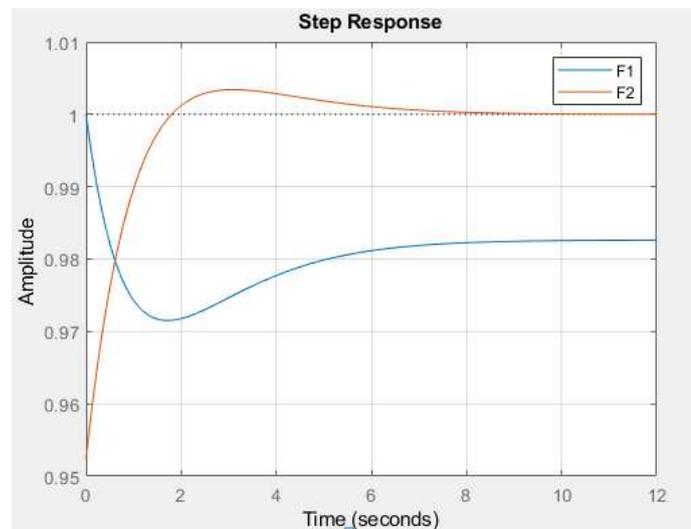
En la segunda función (llamaremos F2) se añade el bloque de la función derivativa, el cual contiene las velocidades del robot con respecto al sistema inercial $[\dot{X}, \dot{Y}]$, como se indicó en la Figura 38, cuya función se suma al bloque de la función PID, y posteriormente se multiplica a la función de transferencia de la planta G_1 , en lazo cerrado, obteniendo un estado estacionario de amplitud 0.97.

Al comparar las dos respuestas se observa que, al añadir la función derivativa, hay una disminución en el valor de la amplitud en el estado estacionario de 0.03, por lo que no

implica un gran cambio en la respuesta al escalón unitario con respecto a la función de transferencia sin la función derivativa, por tal razón, las ganancias calculadas en el diseño del controlador PID, se puedan usar en la implementación del robot físico, considerando el bloque de derivación.

Figura 40

Respuesta al escalón unitario a la función de transferencia en lazo cerrado.

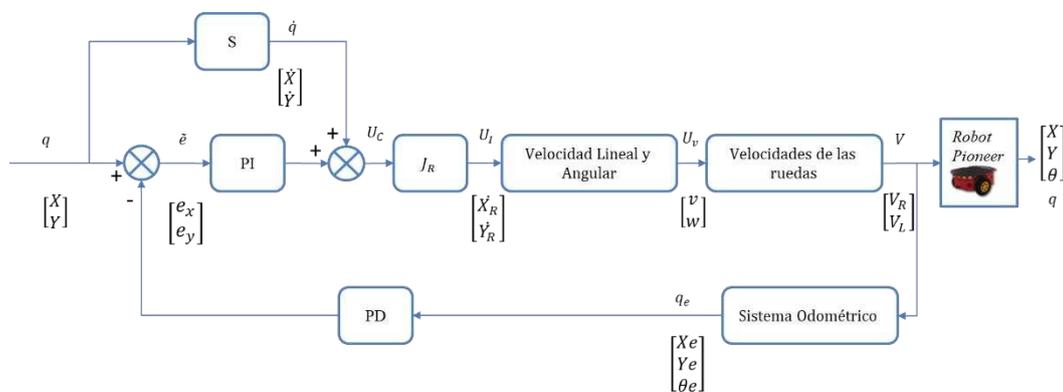


Controlador PID2DoF (dos grados de libertad)

Aplicando el controlador PID 2DoF al robot móvil, se propone el diagrama del sistema de control mostrada en la Figura 41:

Figura 41

Estructura PID 2DoF, basado en un controlador PI-PD para el robot móvil.



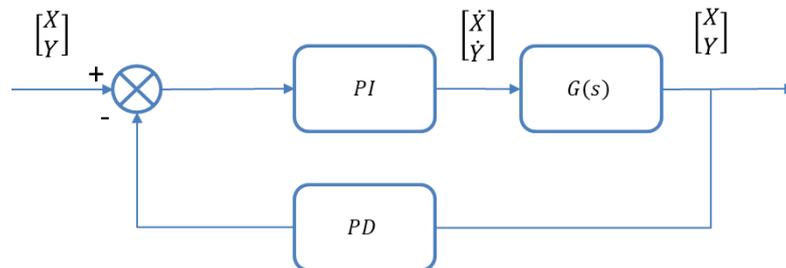
La ley de control aplicada para el sistema de control de dos grados de libertad es igual al de la ecuación (55), pero tomando en cuenta los pesos que se aginan a la parte proporcional y derivativa del error.

$$Uc = \left(\dot{q} + K_p \left\{ (\beta r(t) - y(t)) + \frac{1}{T_i} \int_0^t (r(t) - y(t)) + T_d (\gamma r(t) - y(t)) \right\} \right)$$

Para calcular los parámetros del PID 2DoF, se utiliza el diagrama de bloques de la Figura 42, acoplado la función de transferencia para las entradas de posición con respecto al sistema inercial.

Figura 42

Esquema de diseño del controlador.



El método de cálculo de los parámetros del controlador PID2DoF, sugiere que se calcule, los parámetros de los controladores PI y PD por separado, los cuales se utilizarán en las fórmulas que relacionan las ganancias. Al igual que el controlador PID, se utilizará el método por ubicación de polos para determinar los parámetros en el PI y PD con la función de transferencia de la Figura 36.

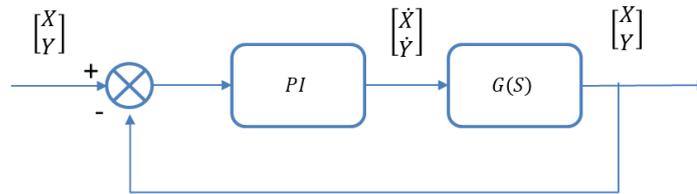
Para el controlador PI:

Para los parámetros del controlador PI, se utiliza las mismas condiciones que el controlador PID, con: tiempo de establecimiento $t_s = 5$ segundos y $MP = 0.07\%$.

Obteniendo los siguientes resultados:

Figura 43

Esquema del controlador PI.



La función de transferencia para el controlador está dada por la ecuación (67):

$$C(s) = k_p \left(\frac{s k_i + 1}{s k_i} \right) \quad (67)$$

A partir del lazo de control mostrado en la Figura 43, se obtiene la función de transferencia en el lazo cerrado representado por las ecuaciones (68) y (69):

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad (68)$$

$$H(s) = \frac{k_p + k_i k_p s}{s^2(4.71 \times 10^{-3} k_i) + s(0.02 k_i + k_i k_p) + k_p} \quad (69)$$

Posicionando un polo en $p_s = -0.8$, los polinomios característicos correspondientes son:

$$P_1(s) = s^2(4.71 \times 10^{-3} k_i) + s(0.02 k_i + k_i k_p) + k_p \quad (70)$$

$$P_2(s) = (s + 0.64)^2 = s^2 + 1.6s + 0.64 \quad (71)$$

Igualando los coeficientes de las ecuaciones (70) y (71) se tiene que los parámetros del controlador PI:

$$k_p = 0.64$$

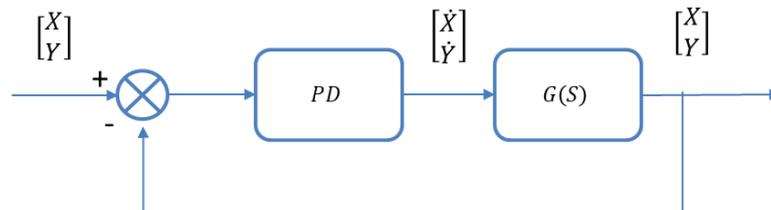
$$k_i = 2.42$$

Para el controlador PD:

Para los parámetros del controlador PD, se realiza el cálculo para el polo $ps = -0.8$, determinado con las mismas condiciones que el controlador PID. Obteniendo los siguientes resultados:

Figura 44

Esquema del controlador PD.



La función de transferencia para el controlador está dada por la ecuación (72):

$$C(s) = kp s + kd \quad (72)$$

Para hallar la constante kp , primero se obtiene la constante de error K de la ecuación (73) en estado estable, enviando una señal de entrada escalón unitario a la función de transferencia de la rueda derecha e izquierda.

$$K = \lim_{s \rightarrow 0} G(s) = \lim_{s \rightarrow 0} \frac{48.82}{0.23 s + 1} = 48.82 \quad (73)$$

$$ess = \frac{1}{1 + K} = 0.02 \quad (74)$$

Para asegurar de que se mantenga el error en estado estable de la ecuación (74) en función de una entrada escalón, incluyendo las especificaciones del controlador; se requiere obtener el valor para kp en base a la función de transferencia en el lazo abierto de la Figura 44, representado por las ecuaciones (75) y (76).

$$GA(s) = G(s) C(s) \quad (75)$$

$$GA(s) = \frac{kp + kd \cdot s}{s(4.71 \times 10^{-3}) + 0.02} \quad (76)$$

$$K = \lim_{s \rightarrow 0} GA(s) = \frac{kp + kd \cdot s}{s(4.71 \times 10^{-3}) + 0.02} = \frac{kp}{0.02} \quad (77)$$

Reemplazando el valor K en la ecuación (77) se tiene que:

$$kp = 0.98$$

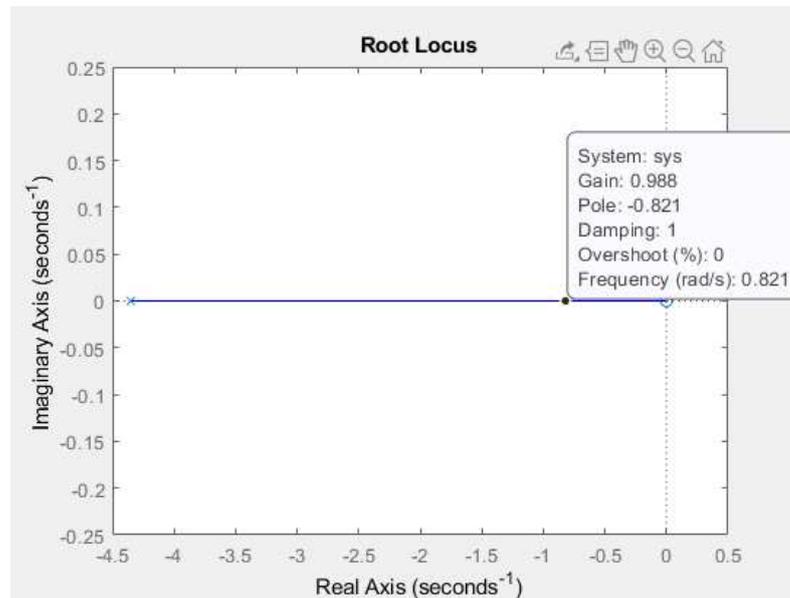
Aplicamos el método del lugar geométrico de las raíces a la ecuación característica de la función de transferencia de lazo cerrado de las expresiones (78) y (79), para obtener el valor de kd manteniendo constante el valor de $kp = 0.96$.

$$H(s) = \frac{G(s) * C(s)}{1 + G(s) * C(s)} \quad (78)$$

$$H(s) = \frac{kp + kd \cdot s}{s(kd + 4.71e^{-4}) + (0.02 + kp)} \quad (79)$$

Figura 45

Ganancia kd por el método del lugar geométrico.



En la Figura 45 se observa que para un polo $ps = -0.8$ el valor de la ganancia $kd = 0.98$.

Cálculo de los Parámetro del PID 2DoF.

Reemplazando los valores de k_p y k_i del controlador PI y el valor de k_d del controlador PD, en las ecuaciones (44), (45), (46), (47) y (48) del capítulo 2, se obtiene los parámetros del controlador PID 2DoF:

$$K_p = 2.99 \quad (80)$$

$$T_i = 1.25 \quad (81)$$

$$T_d = 0.20 \quad (82)$$

$$\beta = 0.21 \quad (83)$$

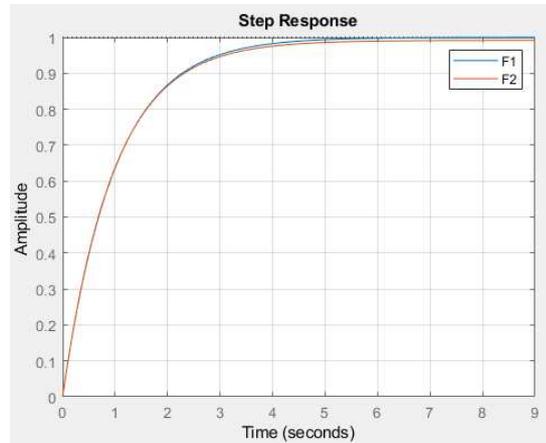
$$\gamma = 0 \quad (84)$$

A continuación, se verifica los valores de las ganancias obtenidas en el análisis del controlador PID2DoF, graficando la respuesta a un escalón unitario; sin embargo, al igual que el controlador PID, en el diseño tampoco se consideró el bloque de la función derivativa del sistema de control de la Figura 41 en el esquema de control de la Figura 42, por lo tanto, se realiza el mismo proceso para comparar las respuestas de las funciones de transferencia en lazo cerrado, añadiendo y quitando el bloque derivativo.

En la Figura 46, se observa que la respuesta a un escalón unitario de las funciones de transferencia en lazo cerrado: sin la función derivativa (F1) y con la función derivativa (F2) son similares, por tal razón, las ganancias calculadas en el diseño del controlador PID2DoF, se puedan usar en la implementación en el robot físico, considerando el bloque de derivación.

Figura 46

Respuesta al escalón unitario a la función de transferencia en lazo cerrado.

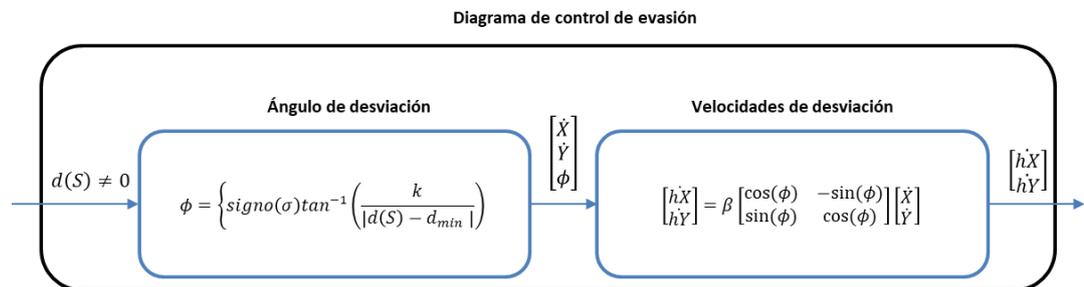


Control de evasión de obstáculos

Para aplicar este método propuesto en la sección de control de evasión de obstáculos en el capítulo 2, se utilizan los 8 sensores ultrasónicos provenientes del robot Pioneer, para definir los parámetros de las ecuaciones (49), (51) y (52). En la Figura 47, se presenta el esquema del control de evasión bajo la condición, de que existe un obstáculo presente, es decir $d(s) \neq 0$. Cuando se calcula el ángulo de desviación ϕ , éste se utilizará como ángulo en la matriz de rotación para obtener las nuevas velocidades del robot con respecto al sistema inercial $(h\dot{X}, h\dot{Y})$, utilizando como entradas a la matriz las velocidades (\dot{X}, \dot{Y}) de la trayectoria preestablecida. De esta forma, se cambia la posición del robot cuando los sensores detecten algún obstáculo; y éste a su vez trata de regresar a su posición previamente definida cuando el obstáculo este fuera del rango de los sensores.

Figura 47

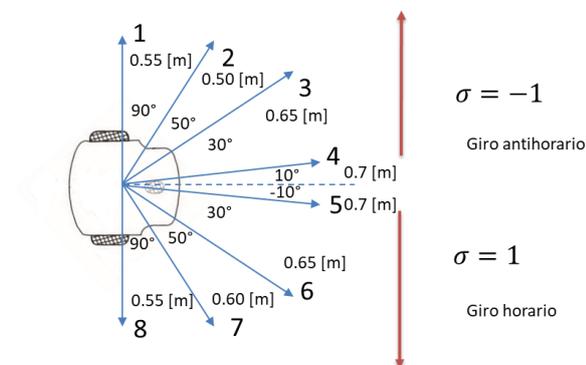
Diagrama de control de evasión.



Para comenzar a evadir, se define una distancia de inicio de evasión d_{div} , la cual se utiliza como un indicativo para que el robot inicie el giro, cuando el valor de $d(S)$ sea mayor o igual al valor de d_{div} . Para ello, se estableció diferentes valores de d_{div} y σ en cada sensor, tomando en cuenta dimensiones mayores al tamaño del robot, como se muestra en la Figura 48.

Figura 48

Distribución de la distancia de mínima de evasión para cada sensor.



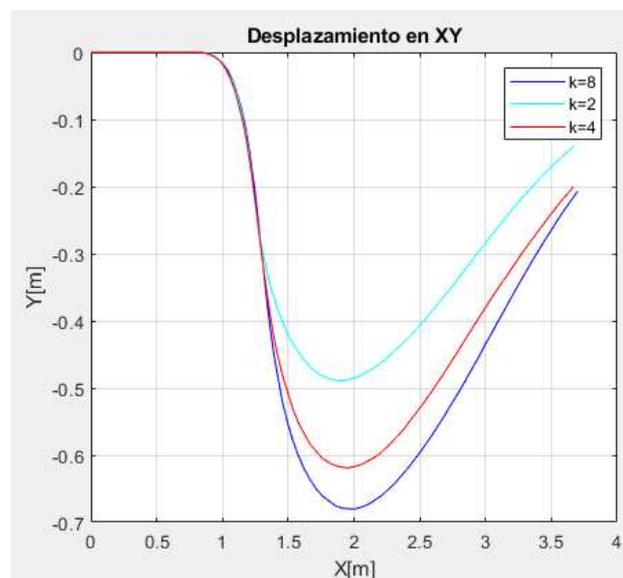
Para escoger la distancia mínima de evasión $d_{min} = 0.45 \text{ m}$, se puso en consideración la dimensión del robot, ya que necesita un espacio prudente para que pueda girar y evadir sin rozamiento.

También es importante seleccionar el valor de la ganancia k , ya que determina la rapidez, con la que el ángulo de desviación alcanza el valor de 90° , por lo tanto, se realizó varias pruebas con diferentes valores, utilizando el valor de $\beta = 4.5$, cuyo valor se utilizó para disminuir la velocidad, con la cual el robot regresa a su posición preestablecida.

En la Figura 49, se observa la curva de desviación para los valores de $k = 2, 4$ y 8 , seleccionando el valor de $k = 8$, ya que se obtiene mayor desviación cuando el robot realiza el giro.

Figura 49

Distancia XY con respecto a la ganancia k .



Para la incorporación del sistema de evasión se utilizará el control PID por facilidades de diseño, ya que maneja solo dos señales de control con respecto a las velocidades del robot eje X y Y , a diferencia del control PID2DoF que maneja 6 señales. En la Figura 50, se propone el sistema de control para la evasión:

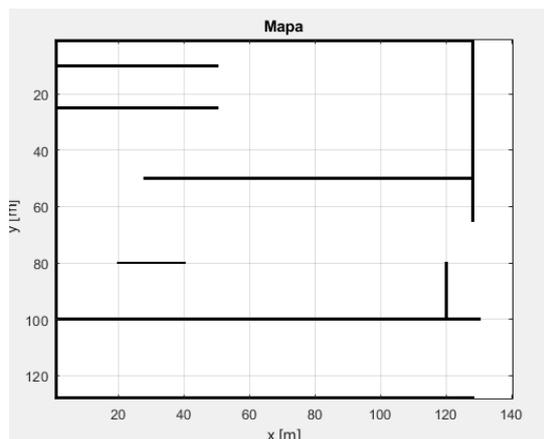
A continuación, se detalla el funcionamiento y los parámetros que se deben colocar dentro de los algoritmos para encontrar la ruta más óptima en un mapa específico.

Búsqueda en A*

Para el funcionamiento de la búsqueda en A*, se necesita de cuatro entradas: una cuadrícula de ocupación (una matriz de objetivos), el punto de inicio, el punto objetivo y la distancia de conexión preferida. El mapa se configura como una matriz de ceros y unos, donde los ceros indican los espacios en blanco, y los unos los obstáculos que puede tener el punto inicial hasta llegar al punto objetivo, como se indica en la Figura 51.

Figura 51

Cuadrícula de ocupación del mapa.



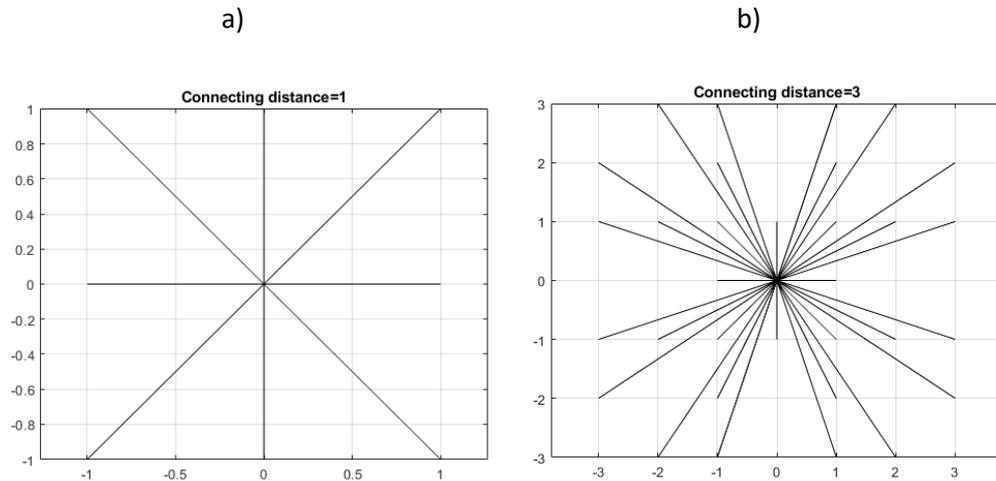
Para el ingreso de los puntos de inicio y objetivo se escogen coordenadas de números enteros que estén dentro del mapa, en caso de que los puntos estén sobre los obstáculos o no consten dentro del grafo, el algoritmo enviará un mensaje advirtiéndolo que no se puede determinar una ruta.

La distancia de conexión en el algoritmo determina las conexiones de cada nodo a las celdas vecinas. Esto significa que el algoritmo no está restringido a 4 u 8 direcciones (que suele ser el caso en otras implementaciones). El número de conexiones está dado por la fórmula 2^n , donde n es el número de direcciones de cada nodo, siendo 8 el número de direcciones (tanto verticales, horizontales y diagonales) mínimas para la conexión entre los nodos, cuando se coloque una distancia de conexión igual a 1. Así, cuando se coloque una

distancia de conexión igual a 2, se obtendrá 16 direcciones, y 32 direcciones si se definen 3 como se indica en las Figuras 52 a) y b).

Figura 52

Número de direcciones de cada nodo.

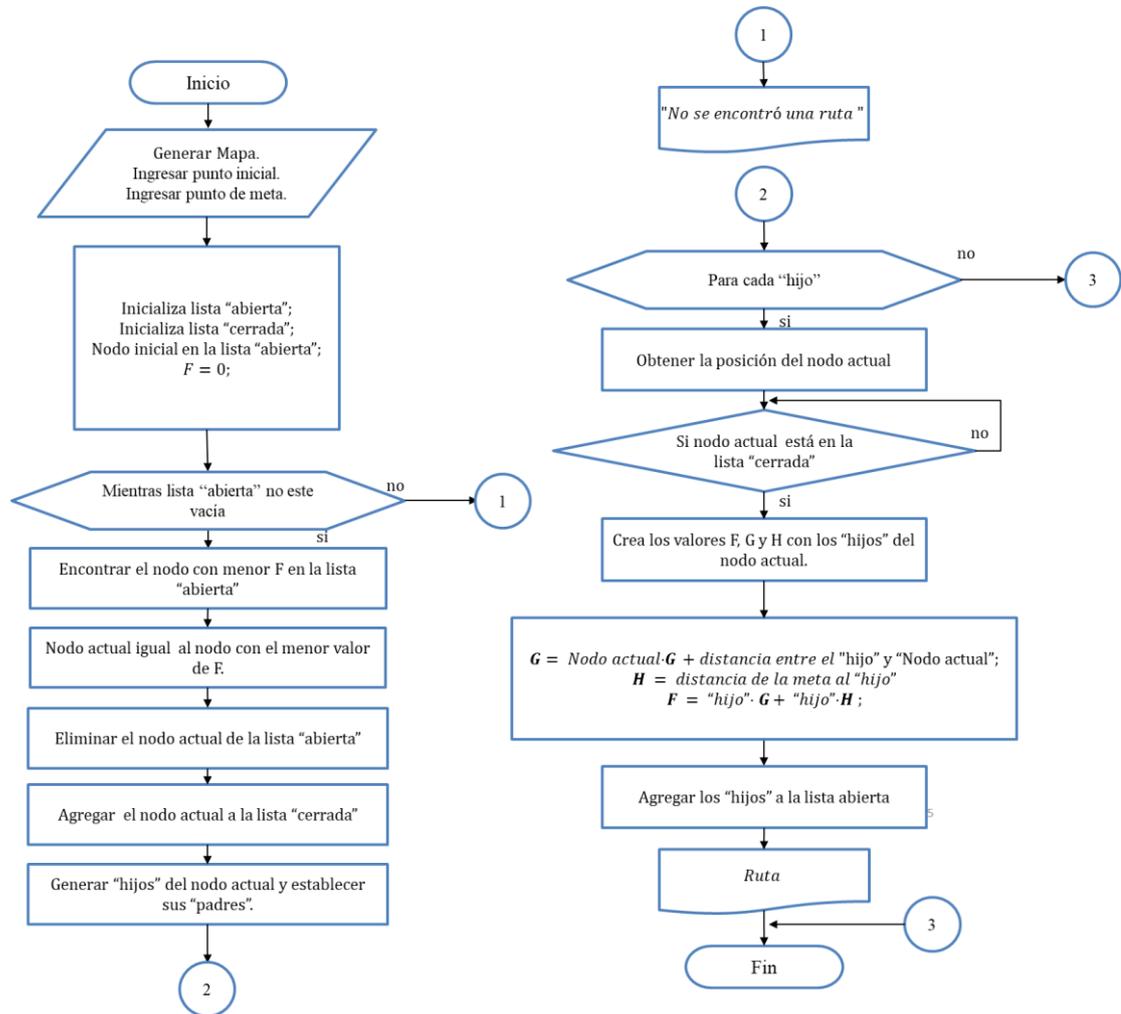


Nota. En las ilustraciones se muestran los siguiente: a) Si la distancia de conexión es igual a 1 se tiene 8 direcciones para la conexión entre nodos. b) Si la distancia de conexión es igual 3 se tiene 32 direcciones para la conexión entre los nodos.

Finalmente, con los cuatro parámetros ingresados se ejecuta el algoritmo A* previamente configurando las listas de datos auxiliares, donde se colocarán el valor de coste, tanto nodos como de los hijos de éste, con la ayuda de la función heurística y obteniendo la mejor ruta para alcanzar el objetivo. Esta ruta se dividirá en puntos tanto para el eje X y Y en metros, los cuales se utilizará como entradas al modelo diferencial inverso incorporando los sistemas de control y evasión propuesto en el esquema de la Figura 49. A continuación, en la Figura 53 se presenta el diagrama de flujo del algoritmo implementado para obtener la ruta de menor costo $f(n)$.

Figura 53

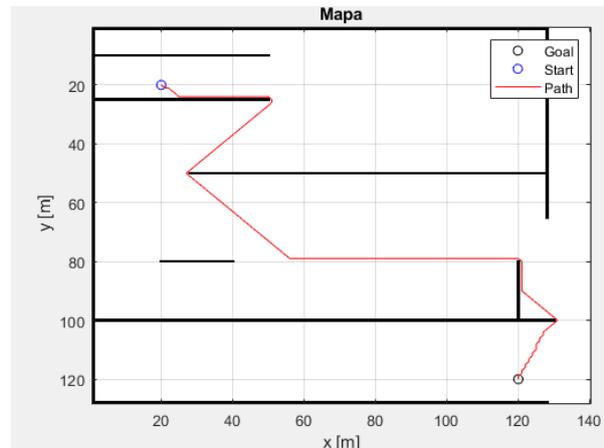
Diagrama de flujo del algoritmo A*.



En la Figura 54 se presenta la funcionalidad del algoritmo A* en un mapa de 140×120 m, creados con 0 y 1, donde se colocan los puntos de inicio y objetivo. En la ilustración se muestra la conexión del punto de inicio (20,20) m a un punto objetivo de (120,120) m, con una distancia de conexión igual a 3 (32 direcciones).

Figura 54

Conexión del punto de inicio y objetivo con el algoritmo A*.



RRT (Árbol aleatorio de exploración rápida)

El algoritmo de árbol aleatorio de exploración rápida (RRT) funciona de la siguiente manera:

Primero, se crea un mapa conocido en escala de grises trazando las posiciones de inicio y objetivo, estos puntos pueden ser valores no enteros.

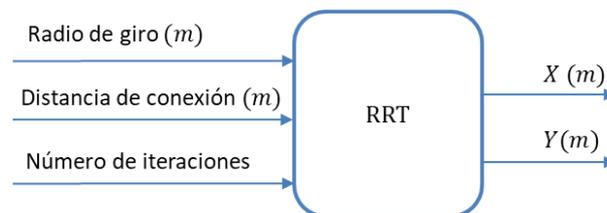
Luego, se define el espacio de búsqueda del móvil especificando los límites del mapa, y definiendo el radio de las curvas de Dubins (generación de un grupo de curvas aleatorias dentro de un espacio) cuya longitud determina la distancia de conexión entre cada nodo. El radio de giro se definió a 0.5 m , tomando en cuenta una longitud mayor al tamaño del robot móvil, y le permita al mismo realizar giros cerrados.

Después para planificación de la ruta, el algoritmo RRT muestrea estados aleatorios dentro del espacio de búsqueda e intenta conectar una ruta entre el punto inicial y los estados cercanos en el árbol generado por los nodos, conduciendo de esta forma al robot móvil hacia el punto objetivo dentro de los límites del mapa. Para definir la distancia máxima de conexión entre los nodos, se escoge un valor de 0.9 m y 3000 iteraciones para el muestreo. Después, que el tiempo de evaluación del algoritmo termine, se imprime el árbol con la ruta en términos de $(X, Y)\text{ m}$, como se indica en la Figura 55. Esta ruta generada por

el algoritmo RRT, se utilizará como datos de entradas al modelo cinemático diferencial inverso con los sistemas de control de evasión y seguimiento incorporados de la Figura 49.

Figura 55

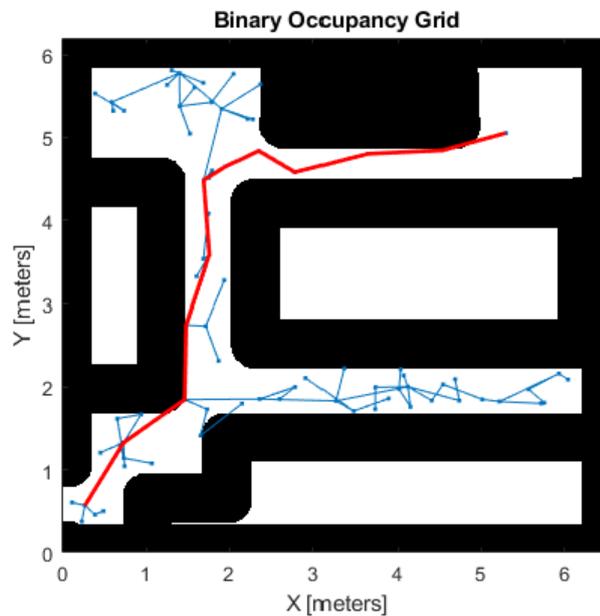
Esquema de entradas y salidas para el algoritmo RRT.



En la Figura 56, se observa un ejemplo de la generación de estados y su conexión entre los mismos, en su espacio de búsqueda hasta llegar a la meta desde su punto inicial.

Figura 56

Diagrama de conexión del algoritmo RRT.



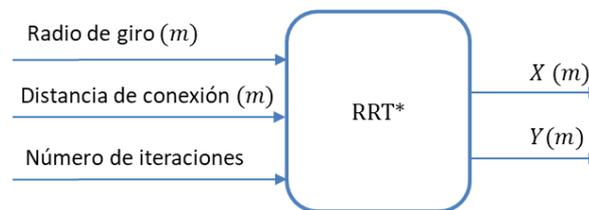
RRT* (Árbol aleatorio de exploración rápida optimizada)

Como se había mencionado en el capítulo 2 de la planificación de trayectorias, el algoritmo RRT* es una optimización del RRT, cuyo funcionamiento tiene el mismo principio. Primero, el algoritmo requiere de un mapa conocido trazando la posición inicial y final,

después se define el espacio de búsqueda del robot móvil especificando los límites del mapa, y los tres parámetros para la generación de la ruta: el radio de las curvas para la conexión entre los nodos generados, la distancia de conexión y el número de iteraciones, cuyos valores son: 0.5 m, 0.9 m y 3000 respectivamente, imprimiendo una ruta en (X, Y) m, al terminar el tiempo de ejecución, como se indica en la Figura 57. Esta ruta generada por el algoritmo RRT* al igual que las dos planificaciones anteriores, se utilizará como datos de entradas al modelo cinemático diferencial inverso con los sistemas de control de evasión y seguimiento incorporados de la Figura 49.

Figura 57

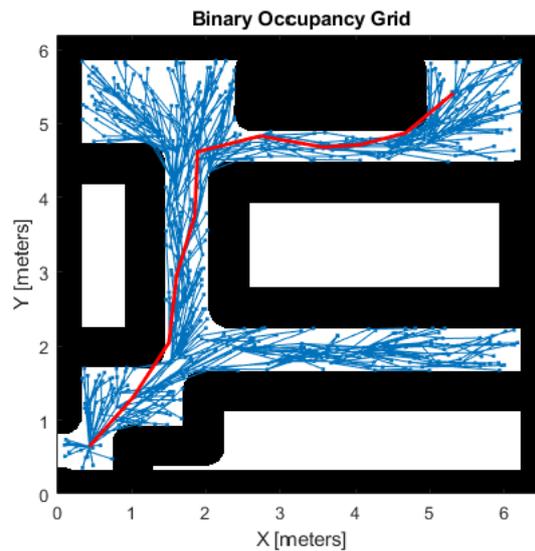
Esquema de entradas y salidas para el algoritmo RRT.*



En la Figura 58, se observa un ejemplo de la generación RRT*, donde existe más estados expandiéndose dentro del mapa, por lo tanto, hay más nodos para la conexión de la ruta, obteniendo un camino con un menor número de ángulos en comparación al RRT, por tal razón, el seguimiento de ésta trayectoria será más eficiente.

Figura 58

Diagrama del algoritmo RRT*.

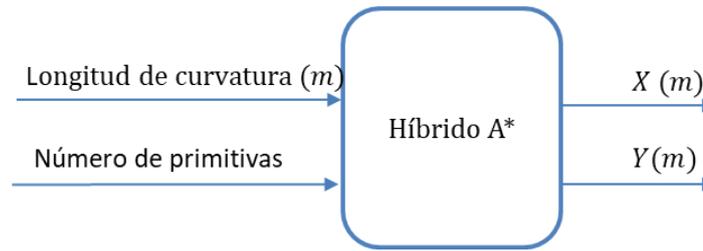


Híbrido A*

Para el funcionamiento del planificador Híbrido A*, al igual que el algoritmo RRT*, necesita de un mapa conocido en escala de grises para trazar las posiciones de inicio y punto final, después se define el espacio de búsqueda del robot especificando los límites del mapa, y los dos parámetros para la generación de la ruta: el radio de las curvas para la conexión y el número de primitivas de movimiento (conexiones de cada nodo a las celdas vecinas) del nodo, cuyos valores son: 0.5 m y 8 respectivamente, finalmente al terminar la ejecución el algoritmo, se interpola las primitivas y se imprime la ruta en $(X, Y)m$, como se indica en la Figura 59. Esta ruta generada por el algoritmo Híbrido A* al igual que las planificaciones anteriores, se utilizará como datos de entradas al modelo cinemático diferencial inverso con los sistemas de control de evasión y seguimiento incorporados de la Figura 49.

Figura 59

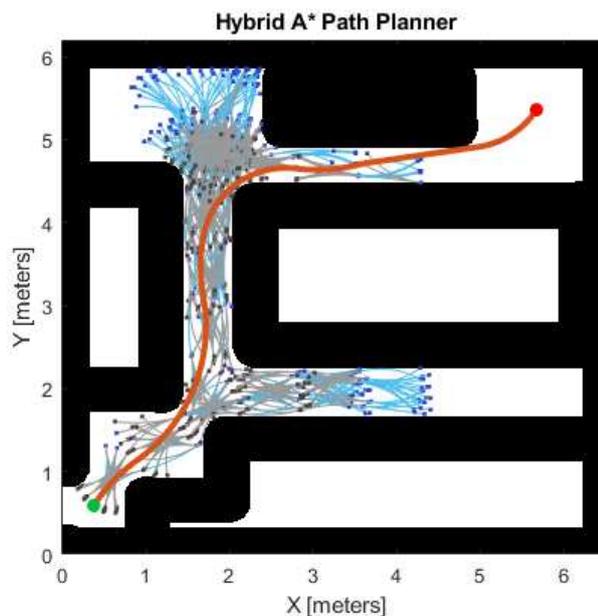
Esquema de entradas y salidas para el algoritmo híbrido A.*



En la Figura 60, se presenta un ejemplo de algoritmo con los valores de los parámetros mencionados anteriormente, donde se evidencia que la ruta obtenida tiene trayectorias curvilíneas, lo que facilita el seguimiento del robot en este tipo de planificación .

Figura 60.

Diagrama del algoritmo Híbrido A.*



Algoritmos para el sistema de navegación

A continuación, se presenta el diagrama de flujo de los algoritmos para la implementación del sistema de navegación en el robot móvil Pioneer P3-DX, el cual incluye la cinemática diferencial, el generador de trayectorias, sistema de control de seguimiento y evasión de obstáculos.

Cinemática Directa

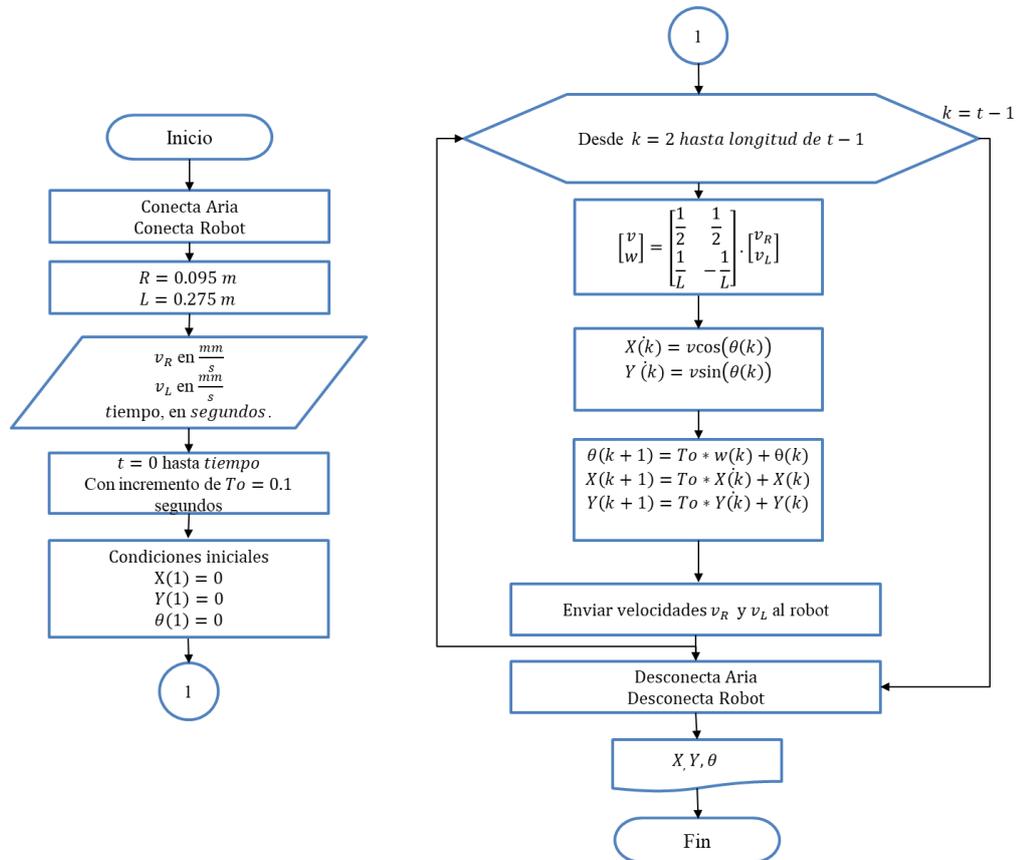
Inicialmente se realiza la conexión al robot móvil utilizando el comando *arrobot_connect*, y a su vez configurando el Aria para poder conectar la computadora personal al simulador o al robot físico, mediante el comando *aria_init* (inicializar aria). Luego se configura las restricciones del robot (Radio del robot r y la distancia a entre las ruedas L), y las condiciones iniciales de las posiciones del robot en el sistema inercial. Después se ingresa por teclado las velocidades lineales de entrada para las ruedas derecha e izquierda y el tiempo de trayectoria (tiempo de duración que se va a mover el robot móvil). Las velocidades del robot se envían mediante el comando *arrobot_setwheelvels*.

El cálculo de las posiciones se realiza utilizando las fórmulas establecidas en el diagrama de las Figura 12. Estas posiciones se calculan en función del tiempo de trayectoria ingresado, discretizando en un vector de tiempo t con un periodo de muestro de 0.1 segundos, y guardando cada posición en un vector k de longitud t , dentro de un bucle.

Finalmente, cuando se termina el ciclo se detiene y se desconecta el robot usando los comandos *arrobot_stop* (parar el robot) y *arrobot_disconnect* (desconexión al robot) imprimiendo la posición final. En la Figura 61, se plantea el diagrama de flujo propuesto del programa para la implementación de la cinemática directa del robot móvil.

Figura 61

Diagrama de flujo propuesto para la cinemática diferencial directa.



Dependiendo de la velocidad que se mande al robot, éste puede trazar trayectorias rectilíneas o circulares, cumpliendo las siguientes condiciones:

- Trayectoria rectilínea: velocidades iguales en las ruedas.
- Trayectoria circular: velocidades diferentes en las ruedas.

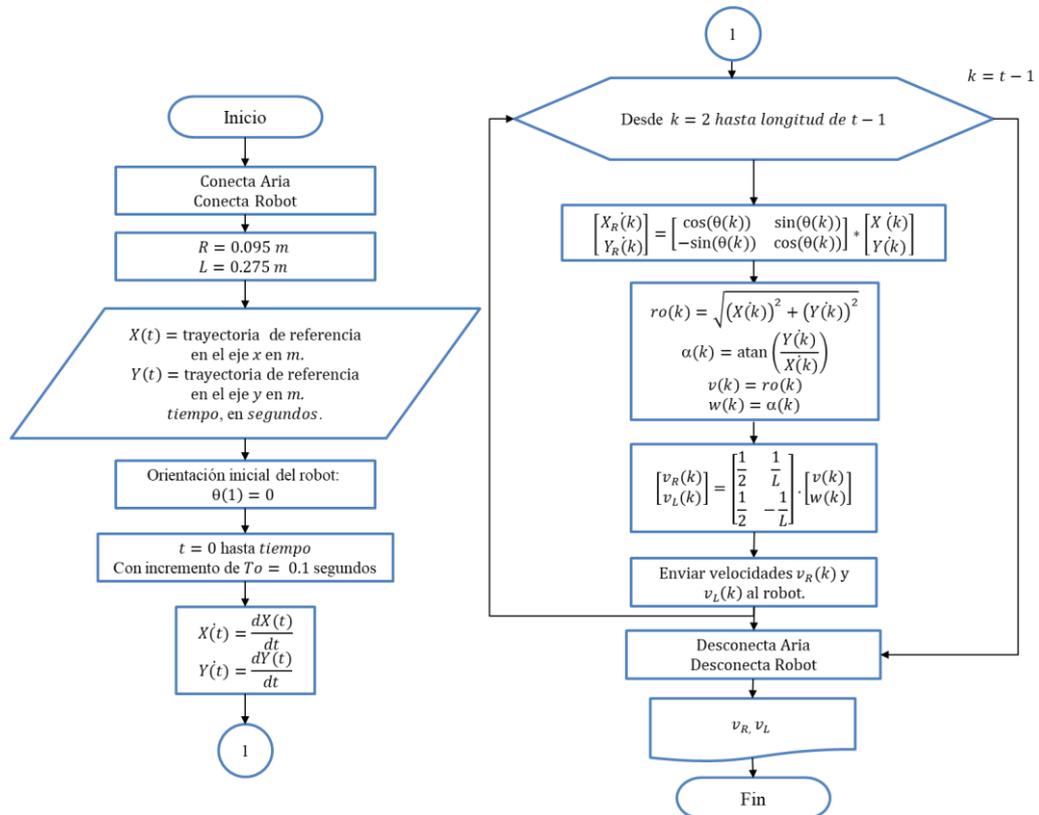
Cinemática Inversa

Para la implementación del programa de la cinemática diferencial inversa en el robot móvil, se utiliza el mismo proceso de inicialización, conexión y configuración para el robot que la cinemática directa.

Inicialmente, se ingresa por teclado las posiciones en los eje X y Y en el sistema inercial y el tiempo de trayectoria. El proceso comienza con una orientación inicial de $\theta = 0$. Al igual que la cinemática directa se plantea las fórmulas y ecuaciones de forma discreta, de acuerdo al diagrama de bloques de la Figura 13.

Figura 62

Diagrama de flujo propuesto para la implementación en el robot móvil.



En la Figura 62, se plantea el diagrama de flujo propuesto del programa para la implementación de la cinemática diferencial del robot móvil.

Generación de trayectorias

Para el seguimiento de trayectorias específicas se propone dos formas: fórmulas matemáticas y planificación de trayectorias, las cuales se utilizarán como entradas de para la cinemática inversa incorporando los sistemas de control y evasión propuestos en las secciones de diseño de la Figura (38) y (41). Las trayectorias deben definirse tanto para el eje X como en Y en el sistema inercial.

Fórmulas Matemáticas

El primer método consiste en formar figuras geométricas mediante fórmulas matemáticas expresadas en coordenadas polares sin el uso de un mapa, como se indica en la tabla 2.

Tabla 2

Trayectorias generadas por fórmulas.

Tipo de trayectoria	Fórmula Matemática	Figura
Trayectoria Rectilínea	$X(t) = distancia * logitud\left(0, \frac{t}{tiempo}\right)$ $Y(t) = distancia * logitud(0,0)$	
Trayectoria circular	$X(t) = radio - radio * sin\left(\frac{2\pi}{tiempo} * t\right)$ $Y(t) = radio * cos\left(\frac{2\pi}{tiempo} * t\right)$	
Trayectoria en forma de caracol con lazo	$r = \left(\frac{1}{2}\right) * cos\left(2 * \frac{2\pi}{tiempo} * t\right)$ $X(t) = radior * r * sin\left(\frac{2\pi}{tiempo} * t\right)$ $Y(t) = radio * r * cos\left(\frac{2\pi}{tiempo} * t\right)$	
Trayectoria en forma de cuatro pétalos	$r = \left(\frac{1}{2}\right) * sin\left(2 * \frac{2\pi}{tiempo} * t\right)$ $X(t) = radior * r * sin\left(\frac{2\pi}{tiempo} * t\right)$ $Y(t) = radio * r * cos\left(\frac{2\pi}{tiempo} * t\right)$	

Donde:

$X(t)$: posiciones a desplazarse en el eje X con respecto al sistema inercial.

$Y(t)$: posiciones a desplazarse en el eje Y con respecto al sistema inercial.

t : es el vector tiempo con periodo de muestreo de 0.1 segundos con longitud del tiempo de trayectoria ingresado por consola.

$tiempo$: tiempo de trayectoria a recorrer del robot en segundos.

$longitud(i, j)$: vector fila con longitud t .

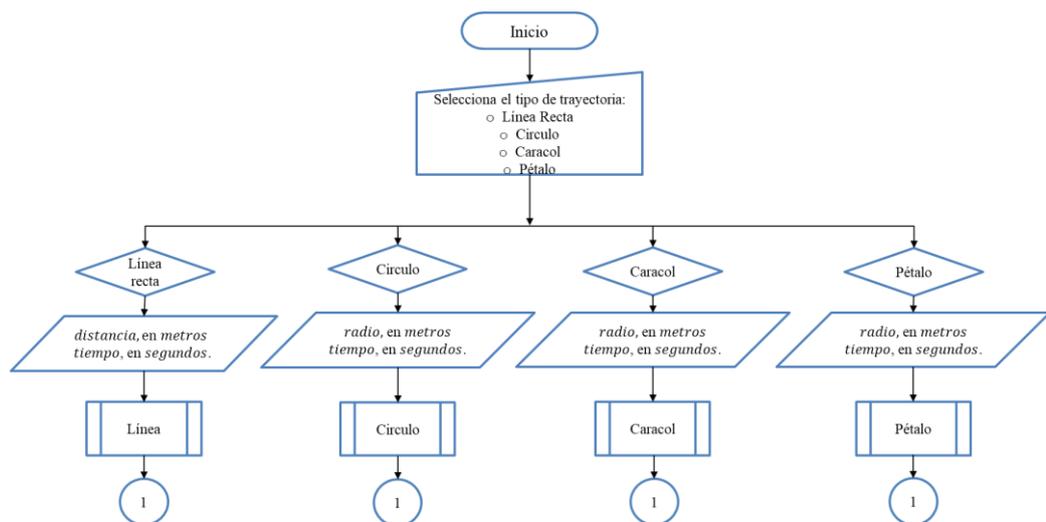
$distancia$: longitud de desplazamiento del robot en metros ingresado por consola.

$radio$: de la trayectoria de la circunferencia en metros

En la Figura 63, se presenta el diagrama de flujo de las trayectorias expresadas por las fórmulas matemáticas que se incorporan antes de iniciar con la conexión del robot.

Figura 63

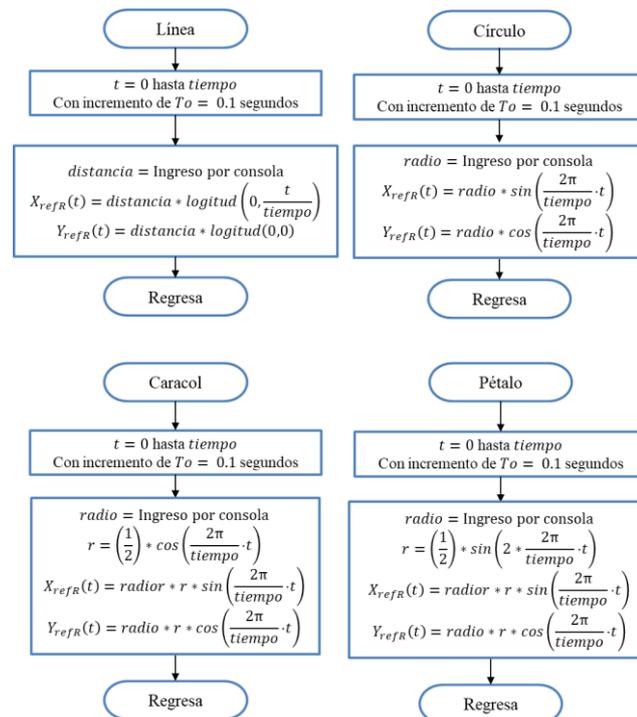
Diagrama de flujo de la selección del tipo de trayectorias.



Estas trayectorias se programaron en funciones separadas al programa principal, las cuales son llamadas mediante subrutinas como se muestra en la Figura 64, especificando la distancia o el radio con el tiempo de recorrido.

Figura 64

Diagrama de flujo de las subrutinas de los tipos de trayectos.



Cada trayectoria se ingresa por la consola de Matlab definiendo dos parámetros: la distancia o el radio que se desea recorrer, y el tiempo que se demora en realizar el trayecto como se indica en la Figura 65.

Figura 65

Interfaz para ingresar la distancia o el radio, y el tiempo del trayecto.

```

Escoger el numero de la figura
1. linea
2. circulo
3. corazon
4. petalo
Ingresar numero: 4
Ingreso de parametros para el seguimiento de la trayectoria
Tiempo de trayectria en segundos:20
Radio en metros:1

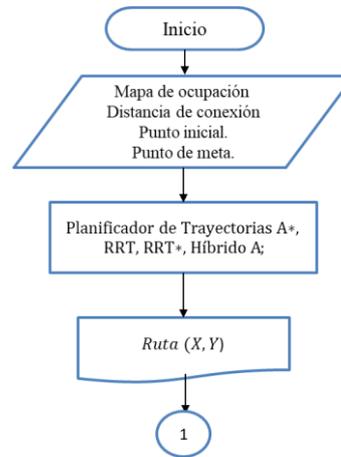
```

Planificador de trayectorias

El segundo método consiste en el uso del planificador de trayectorias mencionado en las secciones anteriores utilizando un mapa específico. Para el generador de trayectorias de la Figura 66, la ruta obtenida por los algoritmos, deben expresarse en coordenadas de X y Y . Luego, estas rutas se utilizarán como entradas a la cinemática diferencial inversa al igual que las fórmulas matemáticas para el seguimiento.

Figura 66

Planificador de trayectorias.



Sistema de control de seguimiento

A continuación, se presentan los diagramas de flujo de la incorporación de los sistemas de control al modelo cinemático diferencial inverso para el seguimiento de trayectorias:

Control PID

Después de haber ingresado las trayectorias del eje X y Y con respecto al sistema inercia, provenientes de los diagramas de flujo de las Figura (64) y (66), el robot realiza el mismo proceso de configuración para la conexión al robot móvil como se indicó en el diagrama de flujo propuesto para la cinemática inversa y directa. Luego, se deriva las posiciones, obteniendo las velocidades del robot con respecto al sistema inercial.

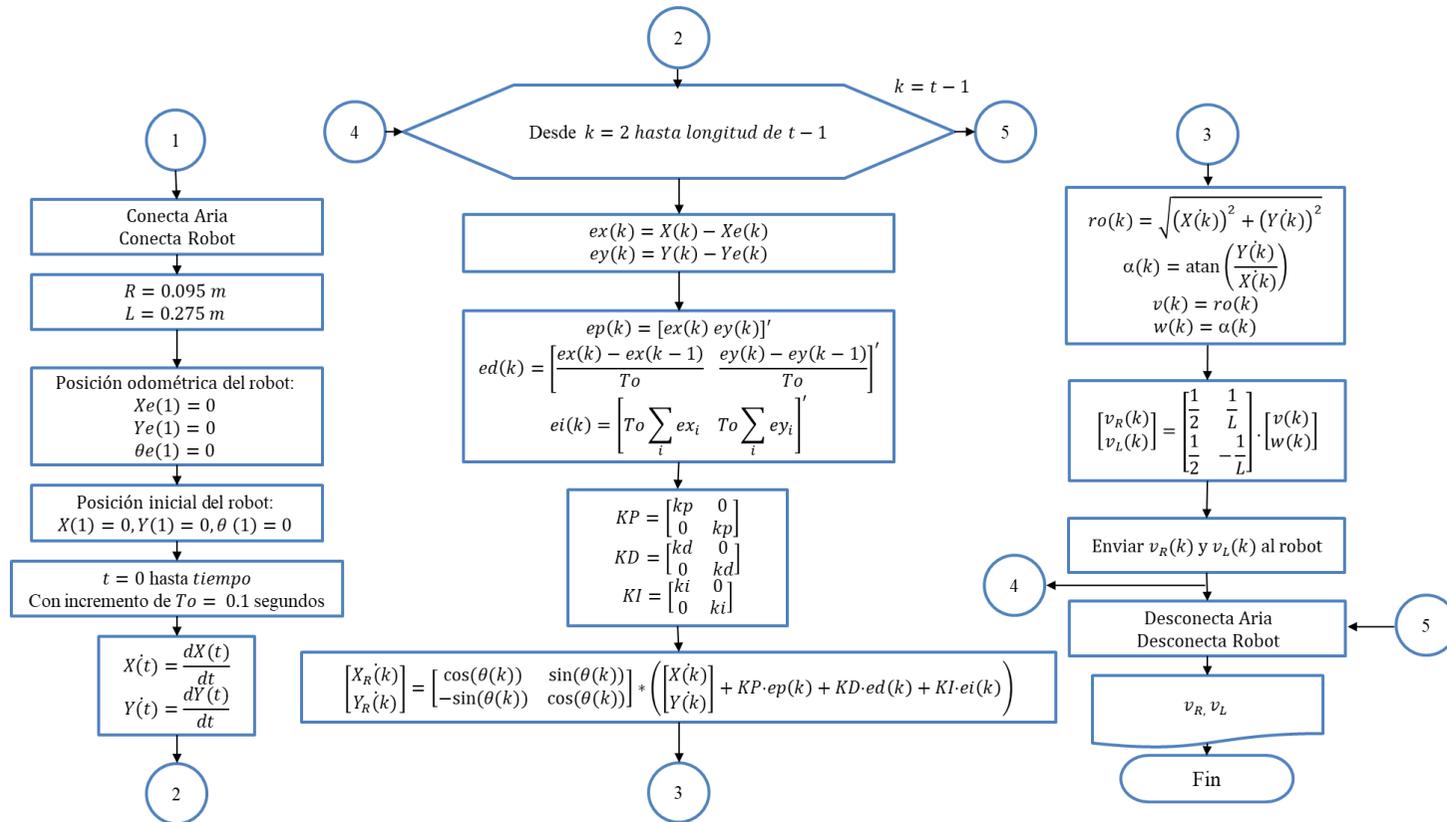
Posteriormente dentro de un bucle, se añade las señales de error obtenidas al comparar las posiciones de entradas con las odométricas proporcionadas por el Pioneer, inicializadas en 0, ya que éstos valores se almacenarán dentro de un vector en cada instante de tiempo.

Finalmente, se aplica la ley de control de la ecuación (55), la cual consiste en sumar las velocidades del robot con respecto al sistema inercial \dot{X} y \dot{Y} , a los términos proporcional, derivativo e integral del controlador PID, utilizando los valores de las ganancias del PID calculadas en la sección de diseño, estos valores se pueden utilizar en el algoritmo ya que se

comprobó al incluir el bloque de la función derivativo en el diseño del controlador no afecta a la salida del robot. El cálculo de las velocidad de las ruedas v_R y v_L , se obtiene utilizando las fórmulas de cinemática diferencial. A continuación, se presenta diagrama de flujo del algoritmo del control PID, ver Figura 67.

Figura 67

Diagrama de flujo para el controlador PID.



Control PID 2DoF

El diagrama de flujo del programa para el seguimiento de trayectorias para el controlador PID 2DoF, se realiza con las mismas trayectorias propuestas para el controlador PID.

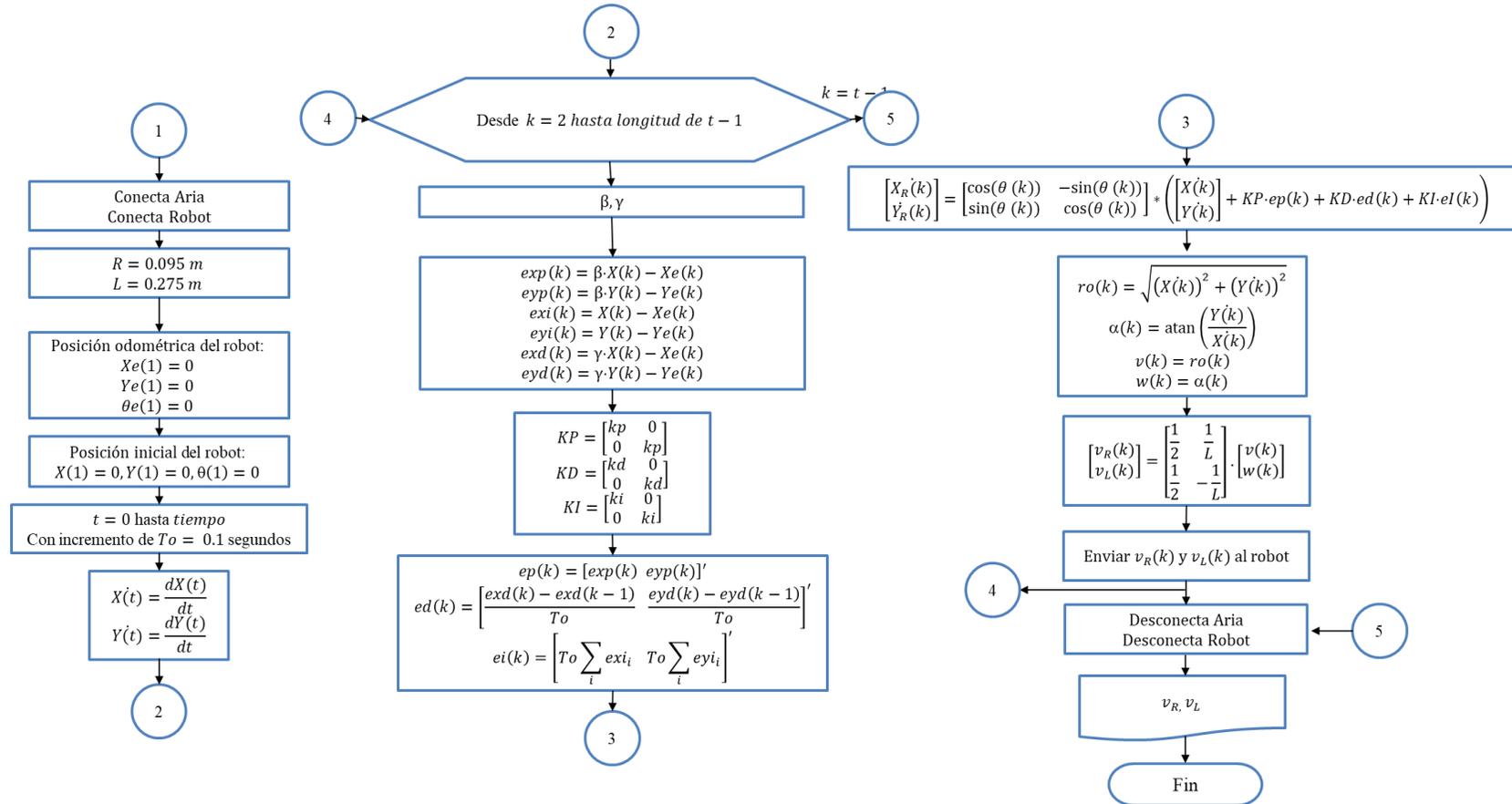
El esquema de programación para éste controlador es similar al controlador PID, cuya variación radica principalmente en las señales de error, debido a la incorporación de los pesos γ y β en las señales de referencia X y Y , que posteriormente se compara con las odométricas.

La ley de control de la ecuación (55) se aplica a las velocidades \dot{X} y \dot{Y} al igual que el controlador PID, colocando los valores de las ganancias del controlador PID 2DoF obtenidas anteriormente en el diseño, como se indica en el diagrama de flujo de la Figura 68.

Cabe recalcar que el uso de las ganancias en el algoritmo de control es posible, ya que se comprobó que el bloque derivativo no altera la salida del robot.

Figura 68

Diagrama de flujo para el controlador PID 2DoF.



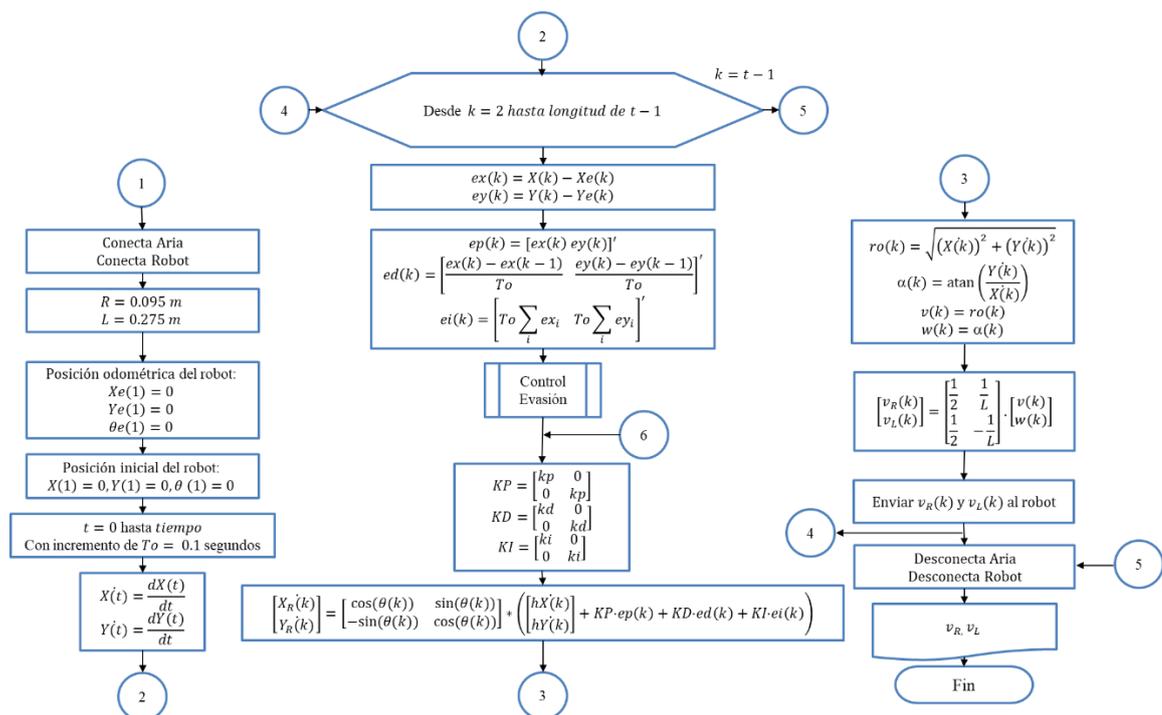
Sistema de evasión de obstáculos

Para la incorporación del sistema de evasión se utilizará el control PID por facilidades de diseño, ya que maneja solo dos señales de control con respecto al eje X y Y , a diferencia del control PID2DoF que maneja seis.

En la Figura 69, se observa la incorporación del sistema de evasión de obstáculos, el cual tiene como entrada las velocidades del robot con respecto al sistema inercial para el cálculo de las nuevas velocidades para la evasión de obstáculos, incorporando la subrutina de evasión representada en la Figura 70. Las velocidades obtenidas de la subrutina regresan nuevamente, para el cálculo de las velocidades del robot y posteriormente las velocidades de las ruedas con el control PID para el seguimiento de trayectorias.

Figura 69

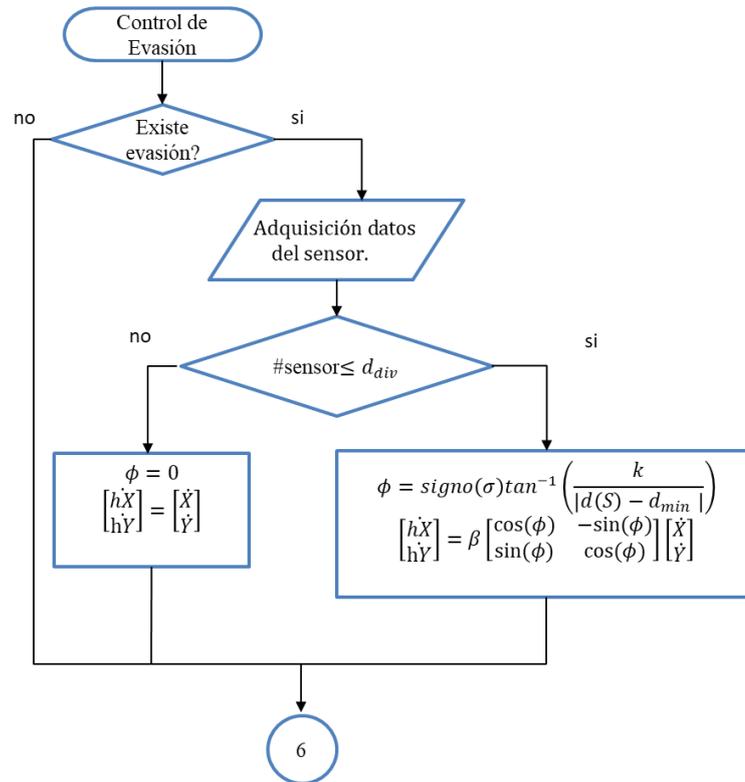
Diagrama de flujo de la incorporación del control de evasión.



La subrutina de control para la evasión de obstáculo se basa principalmente, cuando cualquiera de los 8 sensores detecta una distancia menor a la distancia de inicio de evasión, calculando el ángulo de desviación y finalmente las velocidades para el giro del robot ante la presencia del obstáculo.

Figura 70

Diagrama de flujo del método de evasión.



Capítulo V

Pruebas y resultados

En este capítulo, se procede a realizar las pruebas correspondientes a la validación del modelo diferencial propuesto. Posteriormente, se realiza pruebas para el seguimiento de trayectorias utilizando las fórmulas matemáticas y comprobar el funcionamiento de los dos controladores PID y PID2DoF. Luego, se utilizará las cuatro generaciones de trayectorias para verificar la funcionalidad del sistema de navegación integrando el seguimiento y la evasión de obstáculos.

Prueba de comandos para velocidades de las ruedas.

El objetivo de ésta prueba, fue determinar la exactitud del comando `arrobot_setwheelveles` de Aria, al enviar una velocidad hacia las ruedas del robot Pioneer. Como el robot Pioneer no cuenta con encoders para verificar si el comando está realmente enviando dicha velocidad a las ruedas, se pretende medir distancias. Por lo tanto, se estableció una distancia fija de 1000 *m*, y se envió diferentes velocidades, calculando así el tiempo de trayectoria, que emplea el robot al llegar a esa distancia.

Por ejemplo, si el robot móvil realiza un movimiento traslacional y se envía una velocidad de $100 \frac{mm}{s}$, con un tiempo de muestro de 0.1 segundos, el tiempo que emplea para llegar a esa distancia es de 10 *s*, cuyo cálculo se obtiene mediante la formula $\Delta t = \frac{d}{v}$. La velocidad inicialmente enviada fue de $11 \frac{mm}{s}$ (velocidad mínima con la que se mueve el robot Pioneer), hasta llegar a los $150 \frac{mm}{s}$, como se observa en la tabla 3.

Posteriormente, se realizó la medición real de la distancia recorrida por el robot físico, para constatar que la velocidad enviada fue suficiente para alcanzar los 1000 *mm*.

Tabla 3

Prueba del comando arrobot_setwheelvels.

Distancia real ($d = 1000$) en mm				
N°	Velocidad lineal en $\frac{mm}{s}$	Tiempo de trayectoria $\Delta t = \frac{d}{v}$ en segundos	Distancia medida (d_m) en mm	Error $= \frac{d - d_m}{d} \cdot 100\%$
1	11	90,91	1300	-30%
2	15	66,67	1216	-21,6%
3	20	50	1100	-20%
4	25	40	789	21,1%
5	50	20	779	22,9%
6	75	13,33	910	9%
7	100	10	954	4,65%
8	125	8	894	10,6%
9	150	6,67	842	15,80%

Finalmente, se calculó el error de medición mostrado en la tabla 2, concluyendo lo siguiente:

- Existe mayor distancia recorrida en velocidades menores de $\left(100 \frac{mm}{s}\right)$, con errores de hasta el 30%, que disminuye hasta un 9%, mientras la velocidad se acerca a dicho valor.
- Existe menor distancia recorrida en velocidades mayores de $\left(100 \frac{mm}{s}\right)$, con un error del 15.8% , aumentando su error de distancia, mientras la velocidad se aleja a dicho valor.

Validación Modelo Cinemático Directo

Saber con exactitud la posición y orientación del robot móvil con respecto al sistema inercial, son las incógnitas que se tienen en un robot móvil. Por lo tanto, el objetivo de esta

pueba fue estimar la posición y orientación del robot, a partir del envío de velocidades a las ruedas al robot móvil, ocupando el algoritmo de la cinemática directa propuesto en el capítulo 3. Los datos obtenidos por la cinemática se resumen en la tabla 4 y 5; y a su vez se compararon con los medidos en el robot físico.

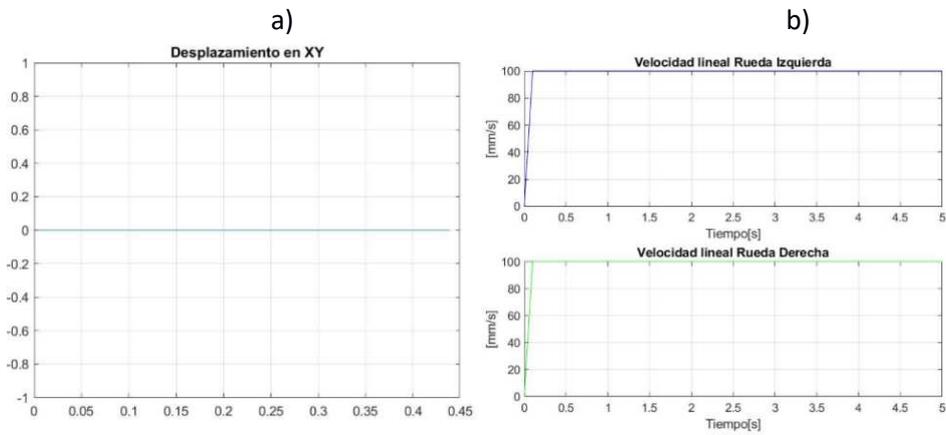
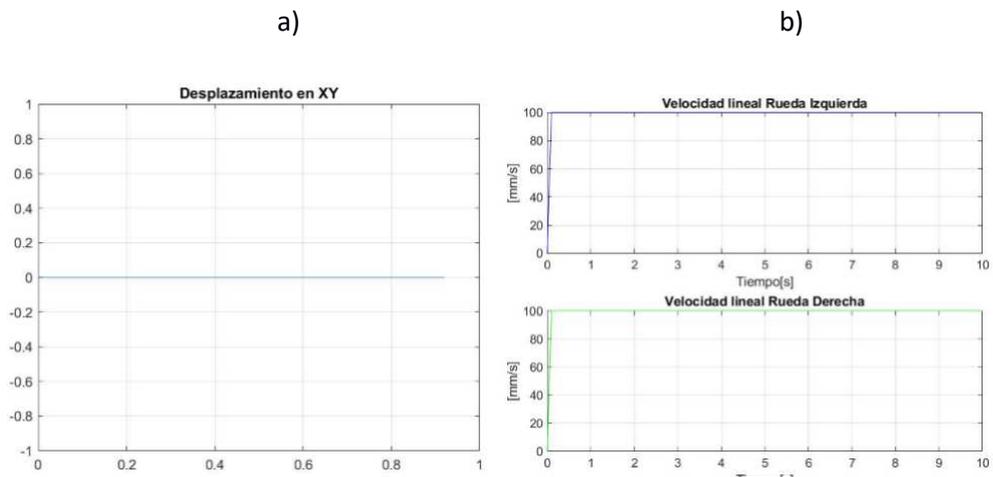
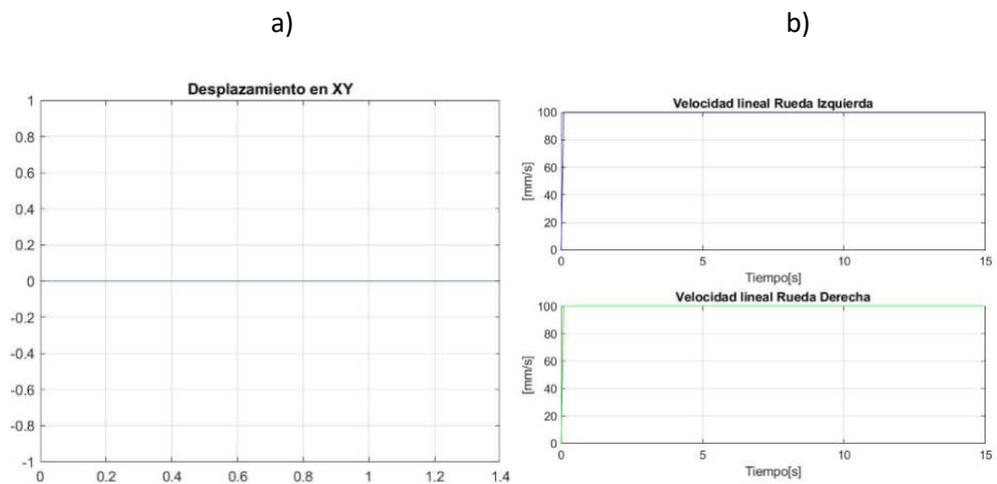
En el movimiento traslacional se realizaron desplazamientos rectilíneos de 0.5, 1 y 1.5 metros a una velocidad de $100 \frac{mm}{s}$, en tiempos de trayectorias de 5, 10 y 15 segundos, mientras que para el movimiento rotacional se envió velocidades de $50 \frac{mm}{s}$, $100 \frac{mm}{s}$, y $150 \frac{mm}{s}$ en cada rueda, durante un tiempo de trayectoria de 15 segundos.

Tabla 4

Modelo cinemático directo: Movimiento traslacional.

Movimiento Traslacional					
		Posición (m)			
		$vL = 100 \left(\frac{mm}{s} \right)$		$vR = 100 \left(\frac{mm}{s} \right)$	
N° Pruebas	Tiempo de trayectoria (segundos)	Propuesta	Medido	Error	Porcentaje
1	5	0.5	0.44	0.06	12%
2	10	1	0.929	0.08	7.7%
3	15	1.5	1.42	0.05	3.3%

En la Tabla 4, se puede observar que en el movimiento traslacional de la prueba N°1, el error porcentual tiene un valor alto, sin embargo, al aumentar la distancia de desplazamiento el error disminuye como se indica en la prueba N°3, manteniendo un error de desplazamiento menor a 0.1 metros, con respecto al medido. El error porcentual oscila entre el 12% hasta disminuir al 3.3%.

Figura 71*Prueba N°1: Desplazamiento traslacional $t = 5$ s.***Figura 72***Prueba N°2: Desplazamiento traslacional $t = 10$ s.***Figura 73***Prueba N°3: Desplazamiento traslacional $t = 15$ s.*

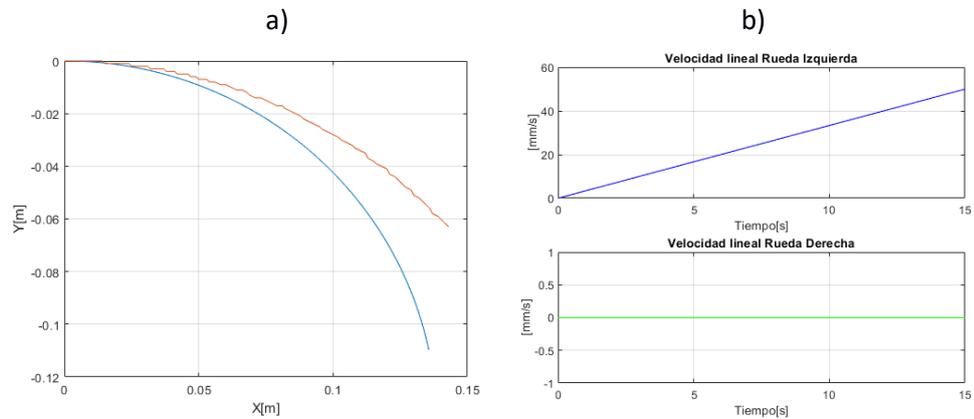
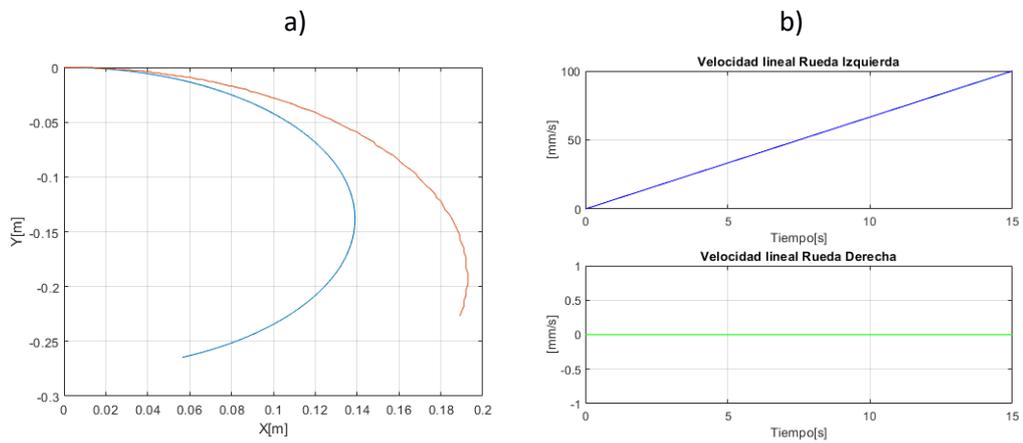
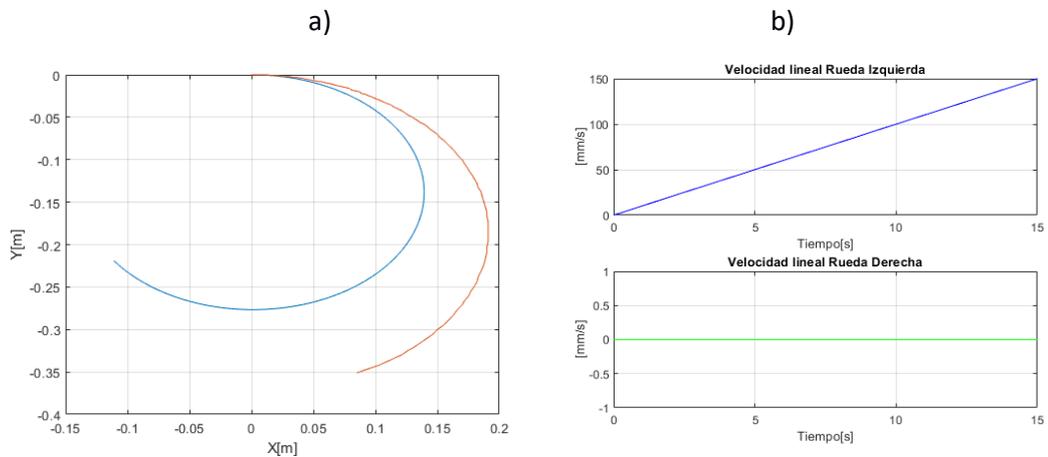
En la Figura 71 a), 72 a) y 73 a) de las pruebas N°1, 2, 3 se observa el desplazamiento realizado por el robot móvil para las distancias de 0.5, 1 y 1.5 m, en los tiempos de trayectoria de 5, 10 y 15 segundos respectivamente. El tiempo que se demora en alcanzar la velocidad de $100 \frac{mm}{s}$ es menos de 0.1 segundos como se muestra en las Figuras 71 b), 72 b) y 73 b).

Tabla 5

Modelo cinemático directo: Movimiento rotacional.

Movimiento Rotacional					
N° Pruebas	Velocidad de las ruedas (mm/s)	Orientación (<i>grados</i>)			
		Modelo	Medida	Error	Porcentaje
4	$vL = 50$ $vR = 0$	-77.61°	-48.83°	-28.71°	37.2%
5	$vL = 100$ $vR = 0$	-156.16°	-105.67°	-8.11°	32.3%
6	$vL = 150$ $vR = 0$	-232.82°	-201.35°	-4.41°	13.5%
7	$vL = 0$ $vR = 50$	77.60°	47.78°	4.1°	38.4%
8	$vL = 0$ $vR = 100$	155.22°	104.21°	7.18°	32.9%
9	$vL = 0$ $vR = 150$	232.56°	200.87°	3.42°	13.63%

En la Tabla 5, se puede observar las orientaciones estimadas y medidas del movimiento rotacional, generado por las diferentes velocidades enviadas, tanto en la rueda derecha como en la izquierda. En las pruebas se envió una velocidad de 150, 100 y $50 \frac{mm}{s}$ en una de las ruedas, mientras que en la otra una velocidad de $0 \frac{mm}{s}$, provocando de esta manera que el robot móvil gire sobre su propio eje, durante el tiempo de trayectoria de 15 segundos.

Figura 74.*Prueba N°4: Desplazamiento rotacional vs Tiempo***Figura 75***Prueba N°5: Desplazamiento rotacional vs Tiempo***Figura 76***Prueba N°6: Desplazamiento rotacional vs Tiempo*

En las Figuras 74 a), 75 a) y 76 a) de la prueba N°4, N°5 y N°6 se observa que el desplazamiento realizado por el robot móvil, es en sentido horario a las manecillas del reloj.

El error porcentual obtenido en las pruebas, se encuentra tabulados en la tabla N°5, lo cual indica que existe un error de orientación de hasta el 37.4% perteneciente a la prueba N°4, el cual disminuyó en 13.5% en la prueba N°6, mientras se aumentó la velocidad en la rueda izquierda de $50 \frac{mm}{s}$ hasta llegar a $150 \frac{mm}{s}$, durante el tiempo de trayectoria de 15 segundos. Además, se observa que la velocidad en la rueda derecha es 0 como se demuestra en las Figuras 74 b), 75 b), y 76 b).

Figura 77

Prueba N°7: Desplazamiento rotacional vs Tiempo.

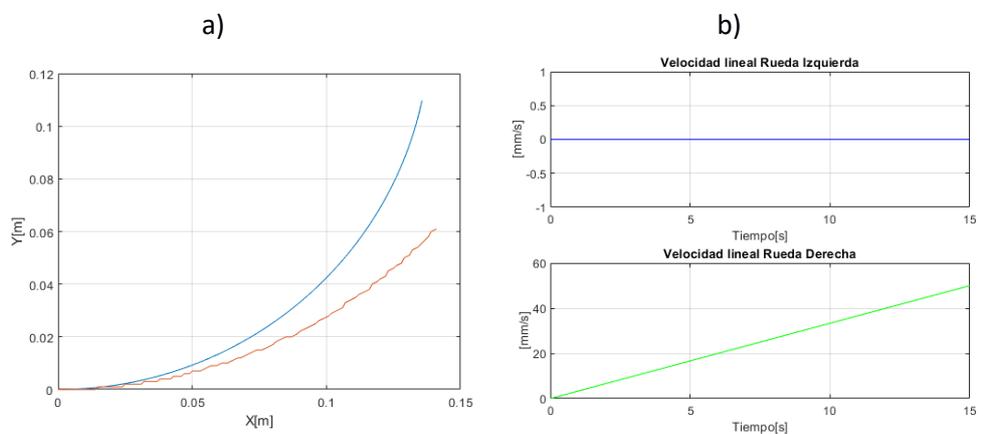


Figura 78

Prueba N°8: Desplazamiento rotacional vs Tiempo

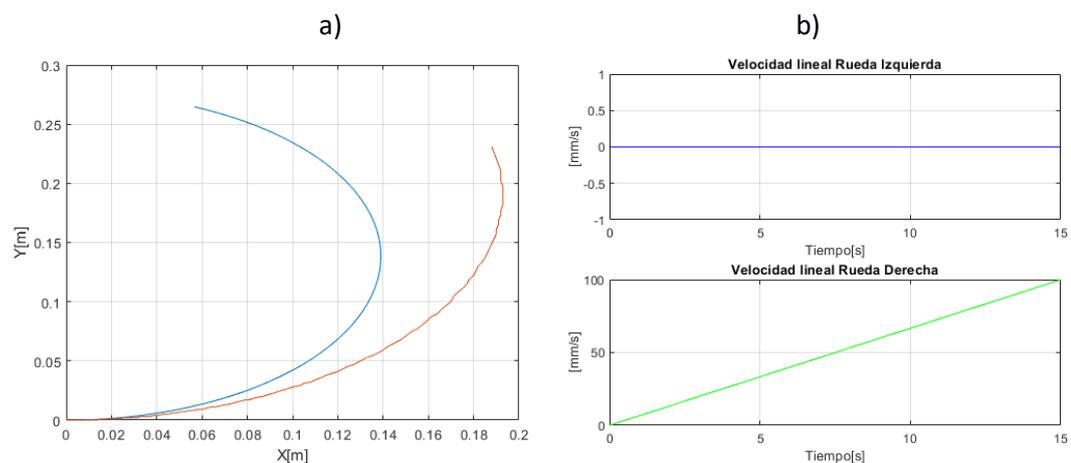
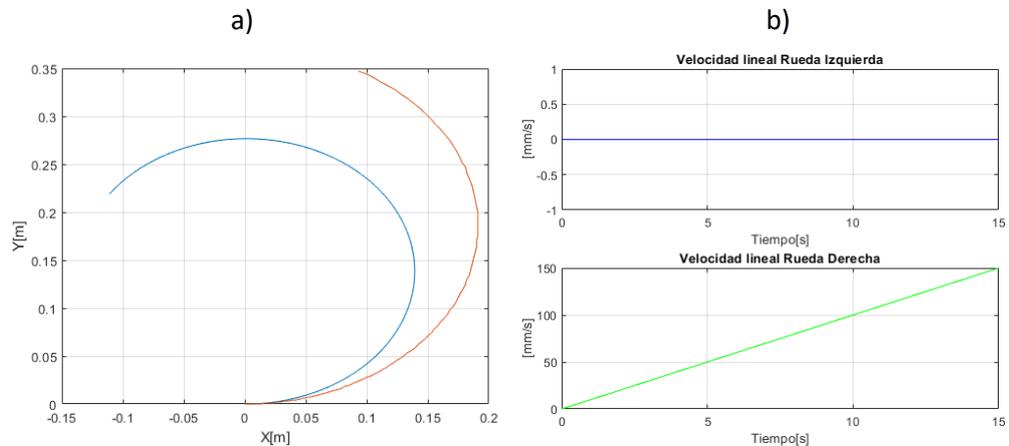


Figura 79

Prueba N°9: Desplazamiento rotacional vs Tiempo



En las Figuras 77 a), 78 a) y 79 a) de la prueba N°7, N°8 y N°9 se observa que el desplazamiento realizado por el robot móvil, es en sentido antihorario a las manecillas del reloj. El error porcentual de la orientación final al igual que las pruebas en sentido horario tienen valor alto de 38.4%, que corresponde a la prueba N°7, la cual disminuye en 13.63% mientras se aumentó la velocidad de la rueda derecha en $150 \frac{\text{mm}}{\text{s}}$, durante el tiempo de trayectoria de 15 segundos. Al igual que en la rueda derecha, se observa que la velocidad en la rueda izquierda es 0 como se demuestra en las Figuras 77 b), 78 b), y 79 b).

Validación Modelo Cinemático Inverso

Para conseguir un movimiento controlado del robot móvil, se requiere ingresar al modelo posiciones que definan una trayectoria, y calcular la velocidad con la cual, el robot se va a mover, por tal razón, el objetivo de esta prueba fue utilizar el algoritmo de la cinemática inversa propuesto en el capítulo 2. Debido a que no se puede obtener información directa de los encoders para medir la velocidad ruedas del robot, y comparar con las velocidades calculadas por la cinemática inversa, se ésta se validó mediante la medición de las posiciones en el robot físico. Las pruebas se dividieron en movimientos de traslación y rotación al igual que la cinemática directa. Para el movimiento de traslación se estableció coordenadas en $(3, 0)m$, $(2, 3)m$ y $(2, -3)m$ durante un tiempo de trayectoria de 30 segundos. En la tabla 6 se resume los datos obtenidos:

Tabla 6

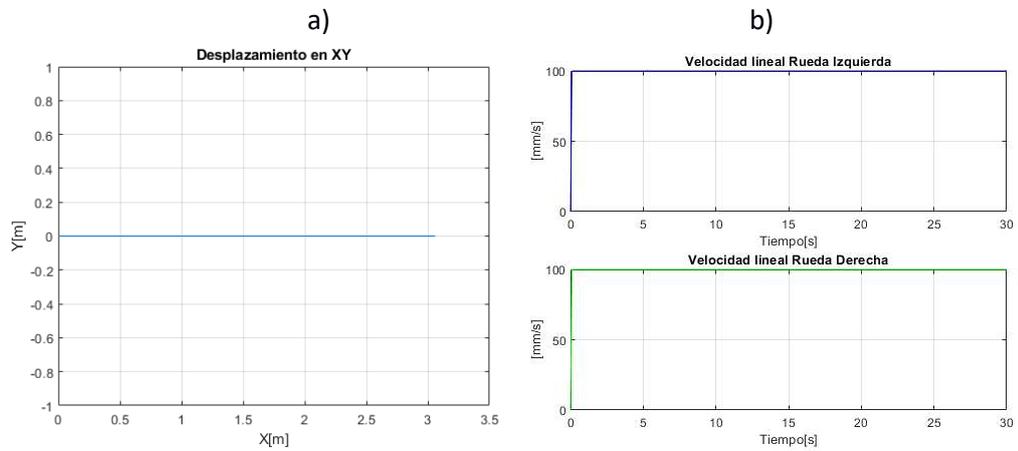
Velocidades estimadas para una distancia fija de 1 metro.

Movimiento Traslacional							
		Posición (m)					
N° Pruebas	Tiempo de trayectoria (segundos)	Propuesta (m)	Medido (m)	Error		Porcentaje	
				x	y	x	y
10	30	(3,0)	(3.21,0)	0.21	0	6.54%	0%
11	30	(3,2)	(2.9, 1,67)	0.1	0.33	3.3%	16.5%
12	30	(3,-2)	(2.92,-1.69)	0.08	0.31	2.67%	15.5%

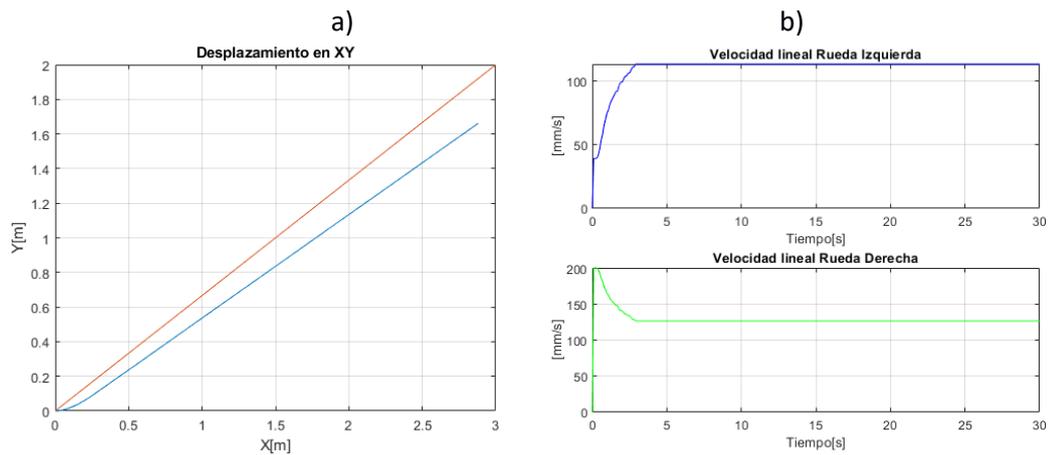
En las Figuras 80 a), 81 a) y 82 a) de la prueba N°10, N°11 y N°12 se observa la trayectoria lineal realizada por el robot móvil, cuyo error porcentual en el desplazamiento disminuye de un 6,54% hasta un 2.67 % en el eje *X*, al igual que el eje *Y* desde un valor del 16.5% hasta un 15.5%, cuando se realiza con diferentes coordenadas como se indica en la tabla 6. Éste error se produce por la exactitud que el comando `arrobot_setwhelevls` tiene, al enviar la velocidad al robot en trayectorias rectilíneas como se había mencionado anteriormente como. El tiempo que tarda el robot en alcanzar la velocidad en una trayectoria en la coordenada *X* es de 0.1 segundos, como se indica en la Figura 71 b), mientras que en trayectorias con diferentes coordenadas tardan alrededor de 4.5 segundos en estabilizarse, pero manteniendo una velocidad constante como se indica en las Figuras 72 b) y 73 b).

Figura 80

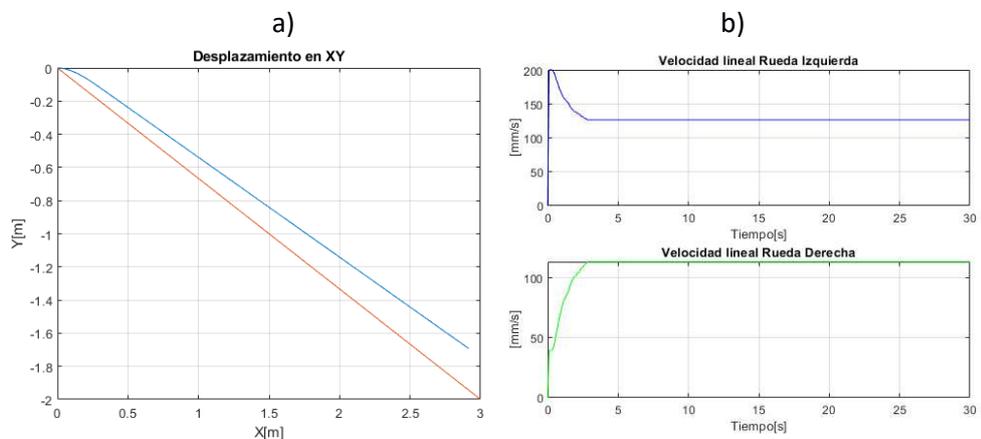
Prueba N°10: Desplazamiento al punto (3,0) m a un tiempo $t=30s$.

**Figura 81**

Prueba N°11: Desplazamiento al punto (3,2) m a un tiempo $t=30s$.

**Figura 82**

Prueba N°12: Desplazamiento al punto (3,-2) m a un tiempo $t=30s$.



Para las pruebas en movimiento rotacional se trazó una trayectoria circular y una rosa de cuatro pétalos con un radio de 1 metro, mediante el uso de las fórmulas N°2 y N°4

de la tabla 3 del capítulo 4, con tiempos de trayectoria de 40 segundos, determinando que el error que existe con respecto a la trayectoria de referencia y la trazada por el modelo.

En la Figura 83 a) de la prueba N°14 se observa que la trayectoria circular trazada por el robot móvil, tiene un error de 0.45 m como se indica en la Figura 84, debido a que existe el cambio del sistema inercial al sistema del robot, cuya posición inicia a lo largo del eje X . La velocidad se mantiene constante en $190 \frac{mm}{s}$ y $120 \frac{mm}{s}$, tanto para la rueda izquierda y derecha respectivamente como se indica en la Figura 83 b).

Figura 83

Prueba N°14: Trayectoria circular a un tiempo $t = 40s$.

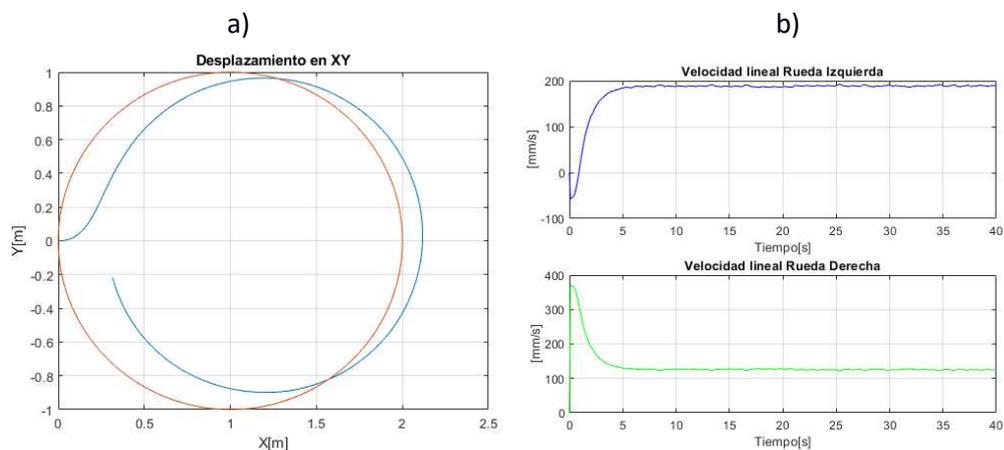
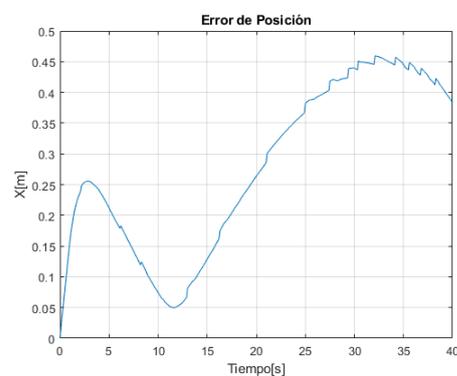


Figura 84

Error de posición de la Prueba N°14.



En la Figura 85 a) de la prueba N°15 se observa la trayectoria de una rosa de cuatro pétalos con origen $(0.0)m$ trazada por el robot móvil, con un error que oscila de $0.1 m$ a

0.6 m como se indica en la Figura 86, el cual se produce cuando el robot realiza los giros para formar los pétalos de la trayectoria deseada.

La velocidad estimada oscila desde los $300 \frac{mm}{s}$ y $25 \frac{mm}{s}$ para la rueda derecha, mientras que para la rueda izquierda oscila entre los $378.3 \frac{mm}{s}$ y $278.5 \frac{mm}{s}$, permaneciendo constante la velocidad en cada cambio de giro para realizar los pétalos de la rosa, como se indica en la Figura 85 b).

Figura 85

Prueba N°15: Trayectoria rosa de 4 pétalos a un tiempo $t = 40s$.

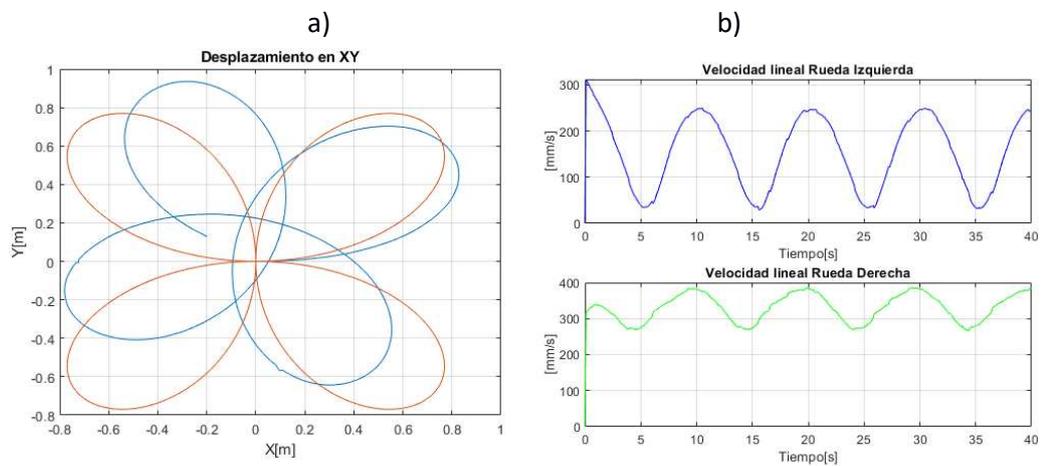
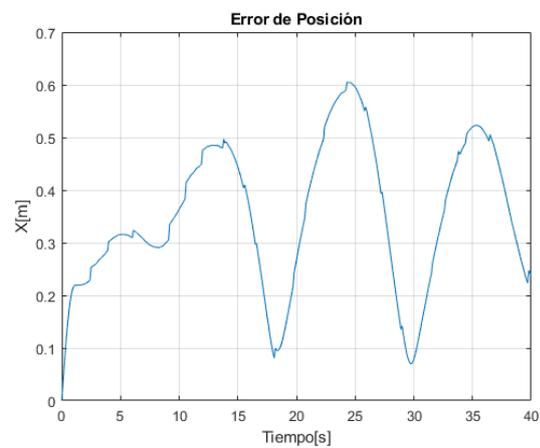


Figura 86

Error de posición de la Prueba N°15.



Seguimiento de trayectorias

Este segmento de pruebas, tuvo como objetivo validar uno de los componentes del sistema de navegación, el control, el cual se incorpora a la cinemática inversa. Para ello, se realizaron pruebas en el seguimiento de trayectorias utilizando las fórmulas de tabla 3 del capítulo 4 como datos de entrada al modelo; aplicando los dos diferentes tipos de control sugeridos PID, y PID2DoF. Además, se determinó las velocidades que son admisibles en el robot Pioneer al momento de realizar la trayectoria. Se dividió en 4 tipos de generación de trayectorias: rectilínea, circular, caracol con lazo y rosa de cuatro pétalos, cuya visualización se realizó a través de la interfaz gráfica creada en Matlab. Las gráficas presentadas para las cuatro trayectorias mencionadas a continuación, se obtuvieron del simulador MobileSim, debido a que se logró los mismos resultados que el robot físico.

Trayectoria Rectilínea

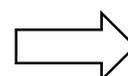
Se realizaron pruebas con tres puntos de coordenadas: $(3,0) m$, $(3,2) m$ y $(3,-2) m$ para cada controlador, con un tiempo de trayectoria de 30 segundos, tabulando el error medido y el porcentaje de error de cada prueba; graficando la trayectoria realizada por el robot móvil, así como el comportamiento de la velocidad de las ruedas ante el controlador y el error de posición con respecto al tiempo de trayectoria.

Tabla 7

Pruebas del controlador PID y PID2DoF en trayectoria rectilínea.

Movimiento Traslacional							
N° Pruebas	Control PID Tiempo de trayectoria (segundos)	Posición (m)		Error		Porcentaje	
		Propuesta (m)	Medido (m)	x	y	x	y
1	30	(3,0)	(2.99,0)	0.01	0	0.1%	0%
2	30	(3,2)	(3.99, 1.989)	0.01	0.011	0.1%	0.36%
3	30	(3,-2)	(3,-1.98)	0	0.02	0%	0.66%
Control PID2DoF		Posición (m)					

CONTINÚA



N° Pruebas	Tiempo de trayectoria (segundos)	Propuesta (m)	Medido (m)	Error			
				x	y	x	y
4	30	(3,0)	(3,0)	0.001	0	0.01%	0%
5	30	(3,2)	(3,2.004)	0	0.004	0.1%	0.13%
6	30	(3,-2)	(3,-1.996)	0	0.004	0%	0.13%

Análisis del Controlador PID en la Trayectoria Rectilínea

En la Figura 87 a) de la Prueba N°1 con control PID, se observa que el robot móvil alcanza una distancia de 2.99 m durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 7, con una velocidad constante de $109.3 \frac{mm}{s}$ y $107.4 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 87 b), cuyo error oscila entre 0.01 a 0.03 m de posición como se observa en la Figura 88.

Figura 87.

Prueba N°1 Control PID: Desplazamiento al punto (3,0) m a un tiempo t=30s

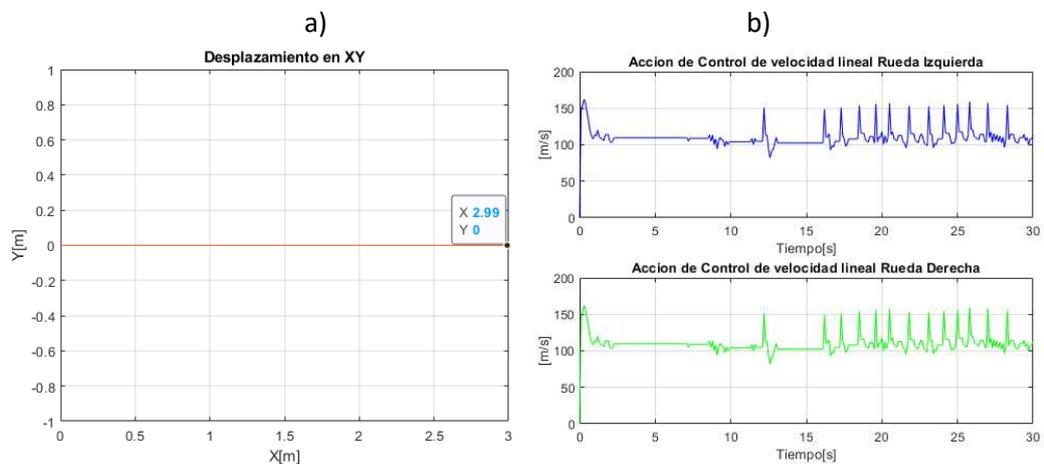
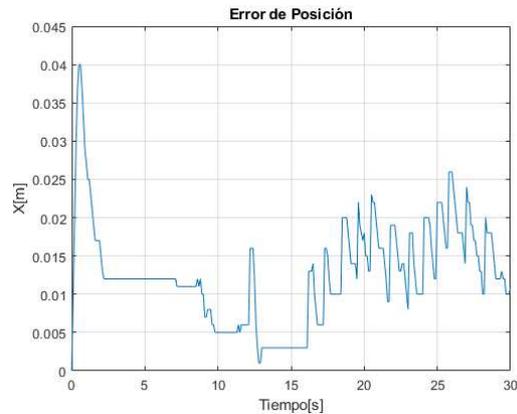


Figura 88

Prueba N°1 Control PID: Error de posición vs Tiempo.



En la Figura 89 a) de la Prueba N°2 con control PID, se observa que el robot móvil alcanza el punto (3.99, 1.989) m durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 7, con una velocidad constante de $110.45 \frac{mm}{s}$ y $125.6 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 89 b), cuyo error oscila entre 0.01 a 0.02 m de posición como se observa en la Figura 90.

Figura 89

Prueba N°2 Control PID: Desplazamiento al punto (3,2) m a un tiempo $t=30s$.

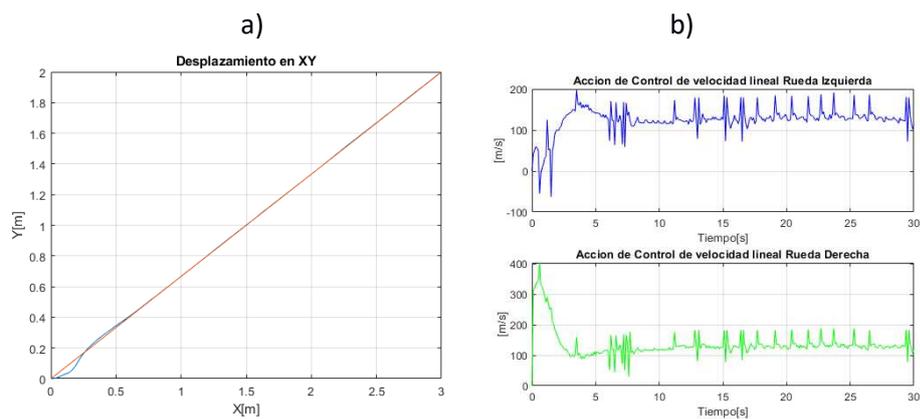
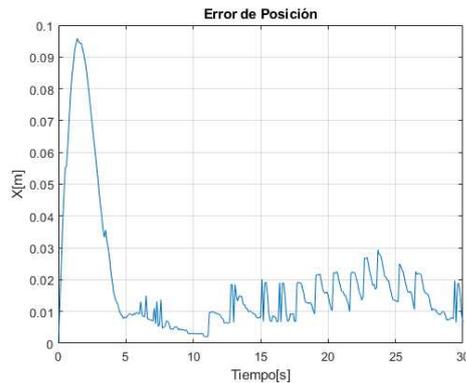


Figura 90

Prueba N°2 Control PID: Error de posición vs Tiempo.

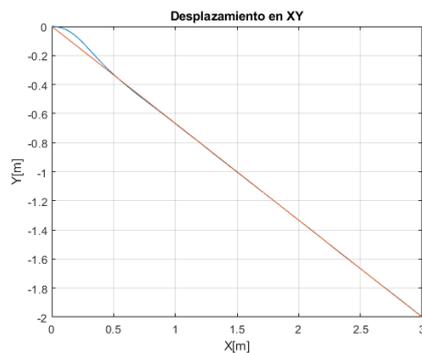


En la Figura 91 a) de la Prueba N°3 con control PID, se observa que el robot móvil alcanza al punto (3,-1.993)m durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 8, con una velocidad constante de $116.76 \frac{mm}{s}$ y $130.12 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 91 b), cuyo error oscila entre 0.005 a 0.025 m de posición como se observa en la Figura 92.

Figura 91

Prueba N°2 Control PID: Desplazamiento al punto (3,-2) m a un tiempo $t=30s$

a)



b)

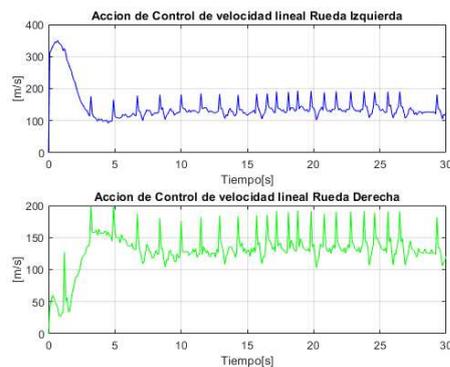
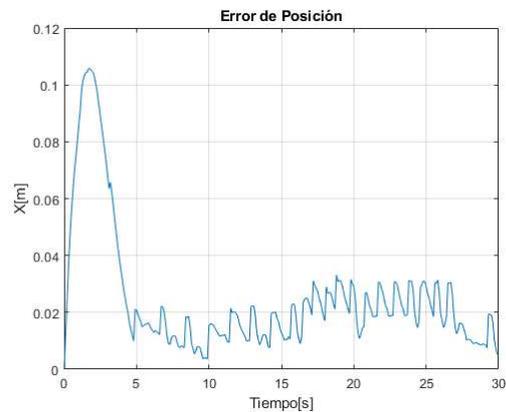


Figura 92

Prueba N°3 PID2DoF: Error de posición vs Tiempo.



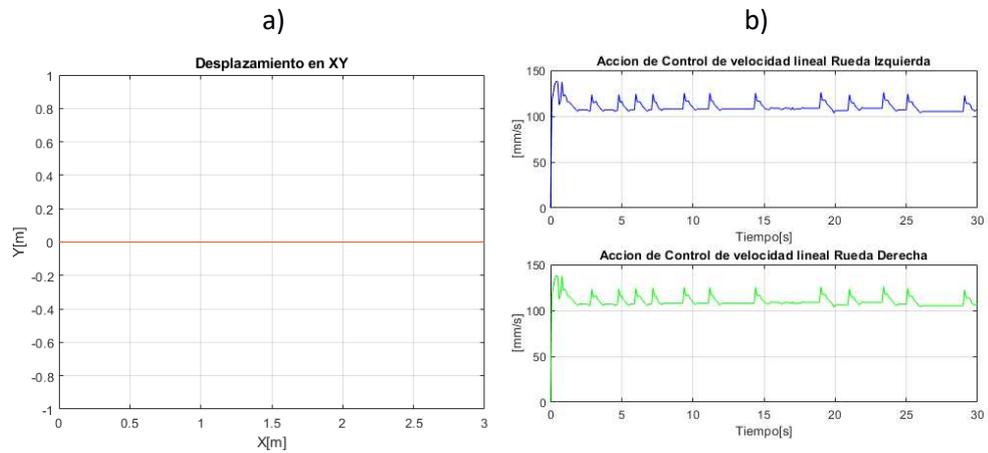
Al realizar las pruebas con el controlador PID, con un tiempo de trayectoria de 30 segundos se determinó, que existe mayor control en el eje X con valores que no superan el 0.1% de error cuando llega a la posición final, mientras que en el eje y existe mayor error con un valor de hasta 0.66%.

Análisis del Controlador PID2DoF en la Trayectoria Rectilínea

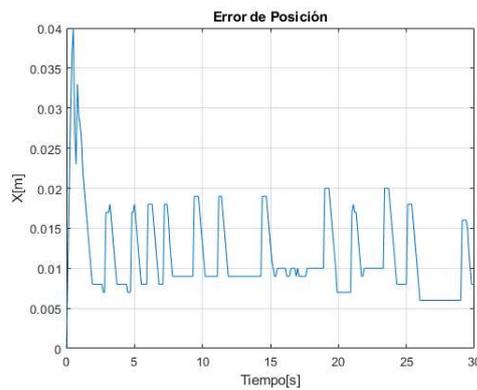
En la Figura 93 a) de la Prueba N°4 con control PID2DoF, se observa que el robot móvil alcanza una distancia de 3.001 m durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 8, con una velocidad constante de $108.33 \frac{mm}{s}$ y $109.34 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 94 b), cuyo comportamiento tiene menos oscilaciones con respecto al controlador PID, el error de posiciones oscila entre 0.007 m a 0.017 m como se observa en la Figura 94, habiendo una reducción del 0.003 m en comparación al controlador PID2DoF.

Figura 93

Prueba N°4 PID2DoF: Desplazamiento al punto (3,0)m a un tiempo $t=30s$.

**Figura 94**

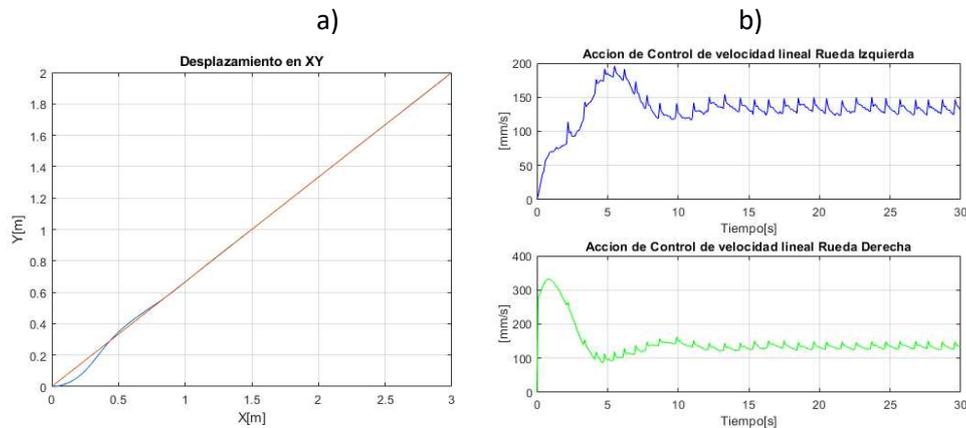
Prueba N°4 PID2DoF: Error de posición vs Tiempo.



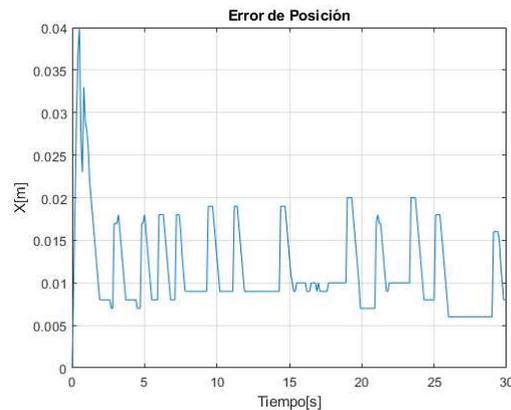
En la Figura 95 a) de la Prueba N°5 con control PID2DoF, se observa que el robot móvil alcanza al punto (3,2.004), durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 7, con una velocidad constante de $124.65 \frac{mm}{s}$ y $180.23 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 95 b), cuyo comportamiento tiene menos cambios de velocidades con respecto al controlador PID, el error de posiciones oscila entre 0.006 m a 0.019 m como se observa en la Figura 96, habiendo una reducción del 0.001 m en comparación al controlador PID2DoF.

Figura 95

Prueba N°5 PID2DoF: Desplazamiento al punto (3,2)m a un tiempo $t=30s$.

**Figura 96**

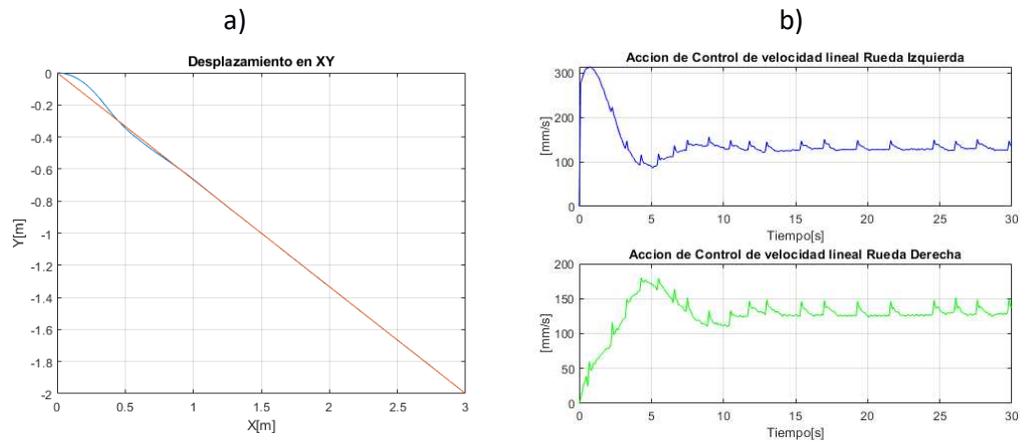
Prueba N°5 PID2DoF: Error de posición vs Tiempo.



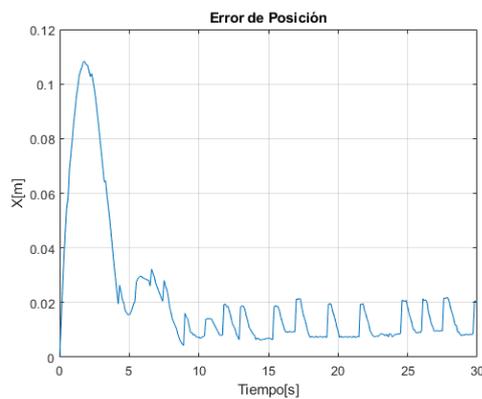
En la Figura 97 a) de la Prueba N°6 con control PID2DoF, se observa que el robot móvil alcanza al punto (2,-2.996), durante un tiempo de trayectoria de 30 segundos como se indica en la tabla 7, con una velocidad constante de $160.97 \frac{mm}{s}$ y $130.23 \frac{mm}{s}$, para la rueda derecha e izquierda respectivamente, como se muestra en la Figura 97 b), cuyo comportamiento tiene menos cambios de velocidades con respecto al controlador PID, el error de posiciones oscila entre 0.005 m a 0.02 m como se observa en la Figura 98, habiendo una reducción del 0.05 m en comparación al controlador PID2DoF.

Figura 97

Prueba N°6 PID2DoF: Desplazamiento al punto (3,-2) m a un tiempo $t=30s$.

**Figura 98**

Prueba N°6 Control PID2DoF: Error de posición vs Tiempo.



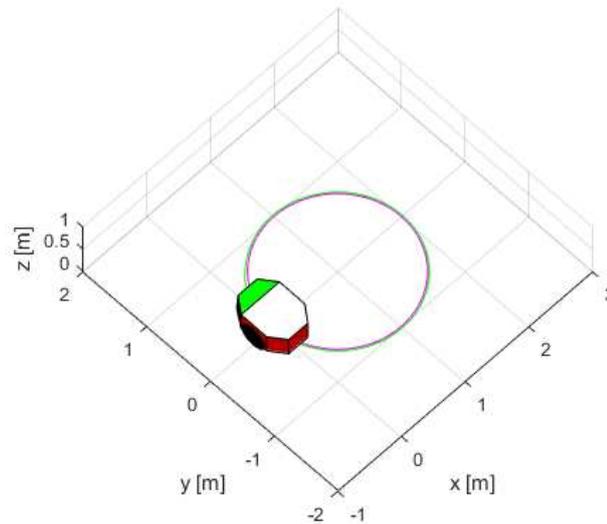
Al realizar las pruebas con el controlador PID2DoF, con un tiempo de trayectoria se observa que existe menor cambio de velocidades y una reducción de error de hasta 0.05 m con respecto al control PID.

Trayectoria Circular

Se realizaron dos pruebas con tiempos de trayectorias de 40 y 60 segundos para cada controlador, ya que al colocar un tiempo menor a 40 segundos se producen velocidades demasiado rápidas para su movilización, por lo tanto, es recomendable colocar tiempos mayores a ésta para evitar problemas en el hardware del robot móvil. El radio definido para las pruebas es de 1 metro, además se tabuló el error y el porcentaje de error de cada prueba; graficando los mismos parámetros que la trayectoria rectilínea.

Figura 99

Trayectoria circular de radio 1 metro.



El robot inicialmente partirá de la posición (0,0), por lo cual el robot iniciará su recorrido desde el extremo izquierdo con sentido anti horario hasta completar la circunferencia como se indica en la Figura 99.

Tabla 8

Pruebas del controlador PID y PID2DoF en trayectoria circular.

Circunferencia						
Tiempo de trayectoria (segundos)	Control PID			Control PID2DoF		
	N° Prueba	Radio= 1 metro		N° Prueba	Radio= 1 metro	
		Error (m)	Porcentaje		Error (m)	Porcentaje
40	7	0.045	4.5%	9	0.023	2.3%
60	8	0.02	2%	10	0.010	1.0%

Análisis del Controlador PID en la Trayectoria Circular

En la Figura 100 a) de la Prueba N°7 con control PID, se observa que el robot móvil realizó el desplazamiento circular con un error de 0.045 m durante los 40 segundos como se indica en la tabla 8, con cambios de velocidad que oscilan entre $192.7 \frac{mm}{s}$ a $265.3 \frac{mm}{s}$ en la rueda derecha y $128.4 \frac{mm}{s}$ a $224.6 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 100 b), los cuales representan cambios de $72.6 \frac{mm}{s}$ (derecha) y $96.2 \frac{mm}{s}$ (izquierda),

donde la rueda derecha necesitó más velocidad debido a que la trayectoria se realizó en sentido antihorario, y es necesario compensar la velocidad para que las dos ruedas del robot se muevan a la par a través de la trayectoria. El error de posición, como se observa en la Figura 101, se estabilizó en el tiempo de 4.5 segundos manteniendo el error de 0.045 m hasta el final de los 40, a pesar que las variaciones de velocidad fueron altas no reflejaron mayores cambios en el error, ya que dichas variaciones ocurren en instantes de 1.5 segundos.

Figura 100

Prueba N°7 PID: Trayectoria circular en el tiempo $t=40s$.

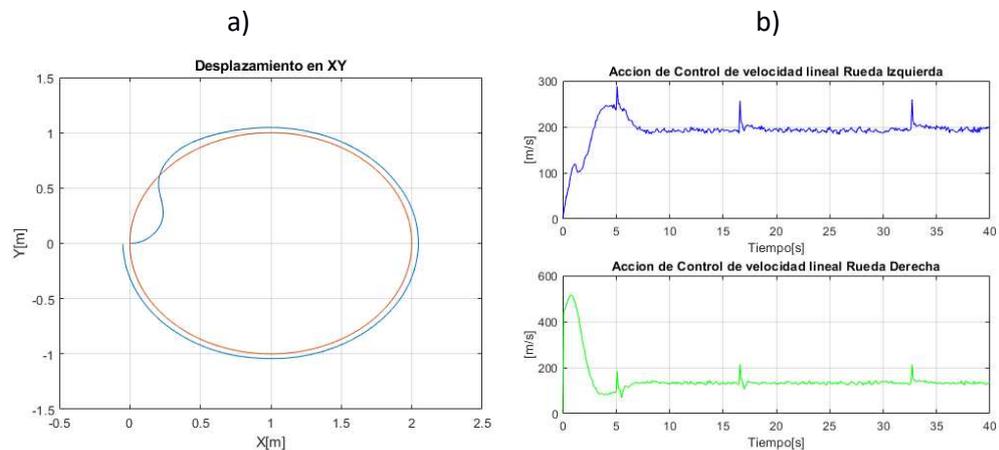
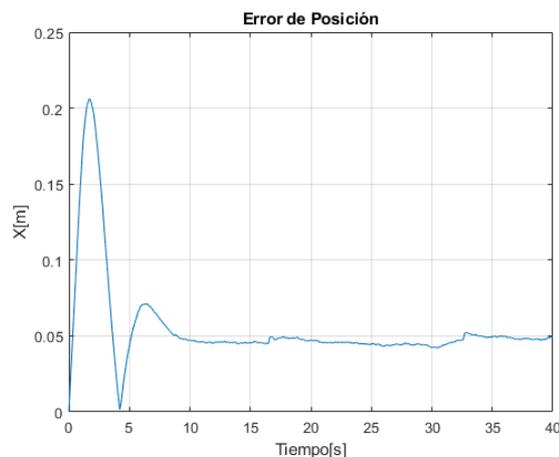


Figura 101

Prueba N°7 PID: Error de posición vs Tiempo.



En la Figura 102 a) de la Prueba N°8 con control PID, se observa que el robot móvil realizó el desplazamiento circular con un error de 0.02 m durante los 60 segundos,

disminuyendo en 0.025 m en comparación al trayecto de 40 segundos como se indica en la tabla 9, con cambios de velocidad que oscilan $128.8 \frac{mm}{s}$ a $177.2 \frac{mm}{s}$ en la rueda derecha y de $85.35 \frac{mm}{s}$ a $149.14 \frac{mm}{s}$ en la rueda izquierda, como se muestra en la Figura 102 b), los cuales representan cambios de $48.4 \frac{mm}{s}$ (derecha) y $63.79 \frac{mm}{s}$ (izquierda), donde los cambios de velocidad se redujeron y reflejan un error de posición de 0.02 m, estabilizándose a 4.5 segundos como se observa en la Figura 103.

Figura 102

Prueba N°8 PID: Trayectoria circular en el tiempo t=60s.

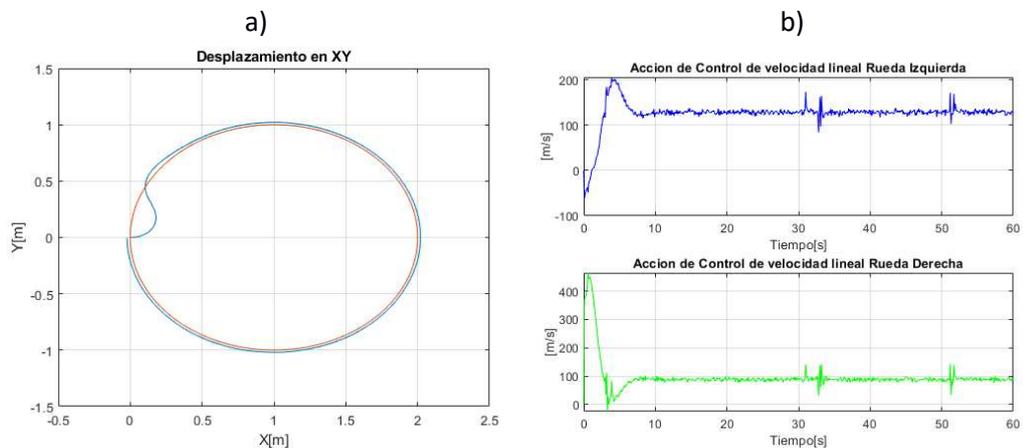
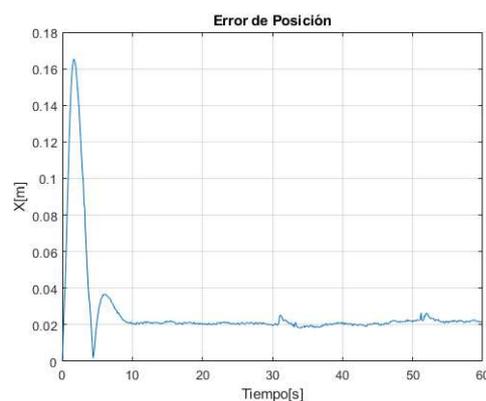


Figura 103

Prueba N°8 PID2: Error de posición vs Tiempo.



En las pruebas realizadas con el controlador PID en la trayectoria, se evidencia la disminución del error de posición hasta un valor de 0.02 m con respecto a la trayectoria

deseada mientras se aumenta el tiempo de posición y mejorando el control de la velocidad de las ruedas.

Análisis del Controlador PID2DoF en la Trayectoria Circular

En la Figura 104 a) de la Prueba N°9 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento circular con un error de 0.023 m durante los 40 segundos como se indica en la tabla 8, con cambios de velocidad que oscilan entre $125.4 \frac{mm}{s}$ a $160.9 \frac{mm}{s}$ en la rueda derecha y $133.8 \frac{mm}{s}$ a $185 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 104 b), los cuales representan cambios de $35.5 \frac{mm}{s}$ (derecha) y $51.2 \frac{mm}{s}$ (izquierda), donde se evidencia su reducción con respecto a las variaciones de velocidad de las pruebas con el controlador PID. El error de posición, como se observa en la Figura 105, se estabilizó en el tiempo de 8 segundos manteniendo el error de 0.023 m, sin embargo, existe un sobreimpulso de 0.012 m a los 4 segundos generado por la velocidad de la rueda derecha al tratar de reducir el error; al ser las variaciones de velocidad bajas no se refleja mayores cambios en el error, ya que dichas variaciones ocurren en instantes menores a 1 segundo.

Figura 104

Prueba N°9 PI2DoF: Trayectoria circular en el tiempo t=40s.

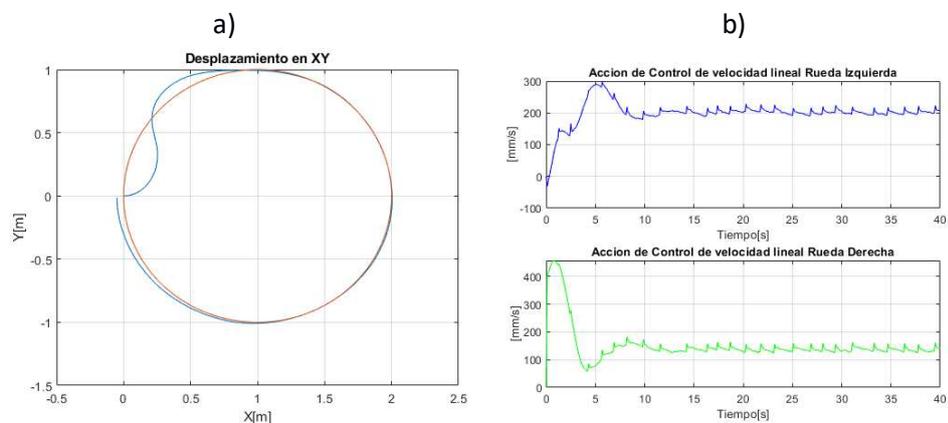
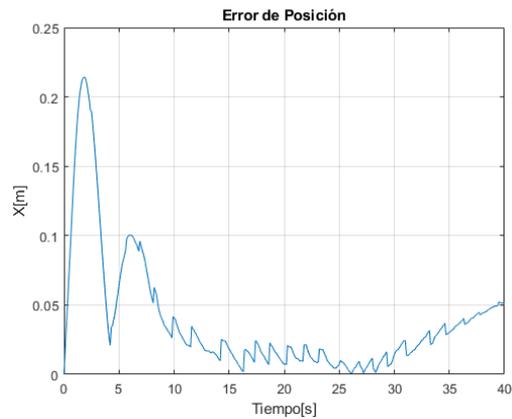


Figura 105

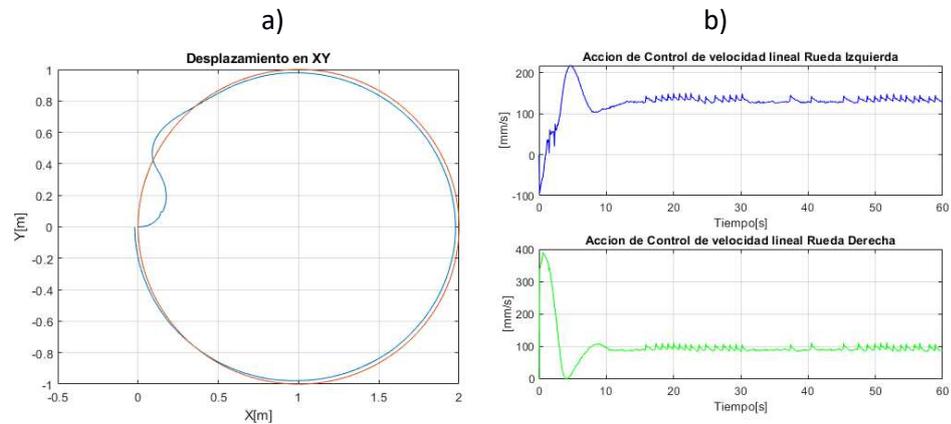
Prueba N°9 PI2DoF: Error de posición vs Tiempo.



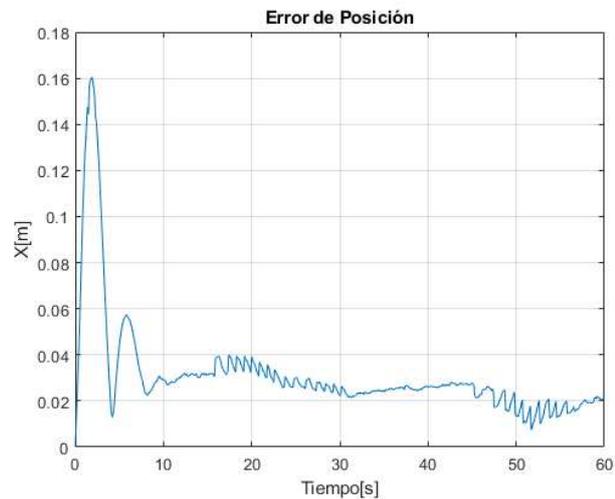
En la Figura 106 a) de la Prueba N°10 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento circular con un error de 0.011 m durante los 60 segundos, disminuyendo en 0.012 m en comparación al trayecto de 40 segundos como se indica en la tabla 8, con cambios de velocidad que oscilan entre $130.2 \frac{mm}{s}$ a $156.7 \frac{mm}{s}$ en la rueda derecha y $87.18 \frac{mm}{s}$ a $115. \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 106 b), los cuales representan cambios de $26.5 \frac{mm}{s}$ (rueda derecha) y $28.02 \frac{mm}{s}$ (rueda izquierda), donde los cambios de velocidad se redujeron aún más y reflejan un error de posición de 0.017 metros al final de los 40 segundos, estabilizándose a 8 segundos como se observa en la Figura 107, además el sobreimpulso generado por la velocidad de la rueda derecha se redujo 0.7 cm.

Figura 106

Prueba N°10 PID2DoF: Trayectoria circular en el tiempo $t=60s$.

**Figura 107**

Prueba N°10 PID2DoF: Error de posición vs Tiempo.

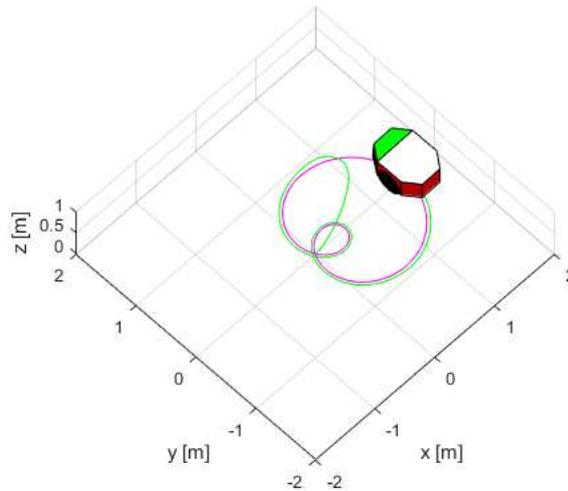


Trayectoria en Caracol con lazo

Se realizaron dos pruebas con tiempos de trayectorias de 40 y 60 segundos para cada controlador al igual que la trayectoria circular. El radio definido para las pruebas es de 1 metro, además se tabuló el error y el porcentaje de error de cada prueba; graficando los mismos parámetros que las anteriores trayectorias.

Figura 108

Trayectoria en caracol con lazo de radio 1 metro.



El robot inicialmente partirá de la posición (0,0) al igual que la trayectoria circular, por lo cual el robot iniciará su recorrido desde el centro del caracol hasta llegar al extremo derecho del mismo en sentido antihorario y completarlo como se indica en la Figura 108.

Tabla 9

Pruebas del controlador PID y PID2DoF en trayectoria de caracol.

Caracol con lazo						
Tiempo de trayectoria (segundos)	Control PID			Control PID2DoF		
	N° Prueba	Radio= 1 metro		N° Prueba	Radio= 1 metro	
		Error (m)	Porcentaje		Error (m)	Porcentaje
40	9	0.102	10.2%	11	0.064	6.4%
60	10	0.049	4.9%	12	0.027	2.7%

Análisis del Controlador PID en la Trayectoria con Caracol con lazo

En la Figura 109 a) de la Prueba N°11 con control PID, se observa que el robot móvil realizó el desplazamiento en forma de caracol, debido a la dificultad de la trayectoria se produce un error de 0.102 m durante los 40 segundos como se indica en la tabla 9, con cambios de velocidad que oscilan entre $316.1 \frac{mm}{s}$ a $360.5 \frac{mm}{s}$ en la rueda y $217.6 \frac{mm}{s}$ a $309.1 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 109 b), los cuales representan

cambios de $44.4 \frac{mm}{s}$ (derecha) y $91.2 \frac{mm}{s}$ (izquierda), donde la rueda derecha necesitó más velocidad debido a que la trayectoria se realizó en sentido antihorario, y es necesario compensar la velocidad para que las dos ruedas del robot se muevan a la par a través de la trayectoria, al igual que la trayectoria circular. Cuando el robot móvil realizó el lazo del caracol, su velocidad disminuyó durante el período de 10 a 30 segundos, hasta un valor de $14.69 \frac{mm}{s}$ (derecha) y $130.9 \frac{mm}{s}$ (izquierda) debido a que el radio del lazo es de 0.5 cm y requiere que la velocidad de la rueda izquierda sea aún más rápida que la derecha para que las ruedas del robot se desplacen de forma simultánea; disminuyendo en 0.095 m el error de posición y posteriormente regresando al valor de 0.102 m, como se observa en la Figura 110, el tiempo que tarda en estabilizarse es de 9.5 segundos. A pesar que las variaciones de velocidad fueron altas no reflejaron mayores cambios en el error, ya que dichas variaciones ocurren en instantes de 1 segundo.

Figura 109

Prueba N°11 PID: Trayectoria caracol en el tiempo $t=40s$.

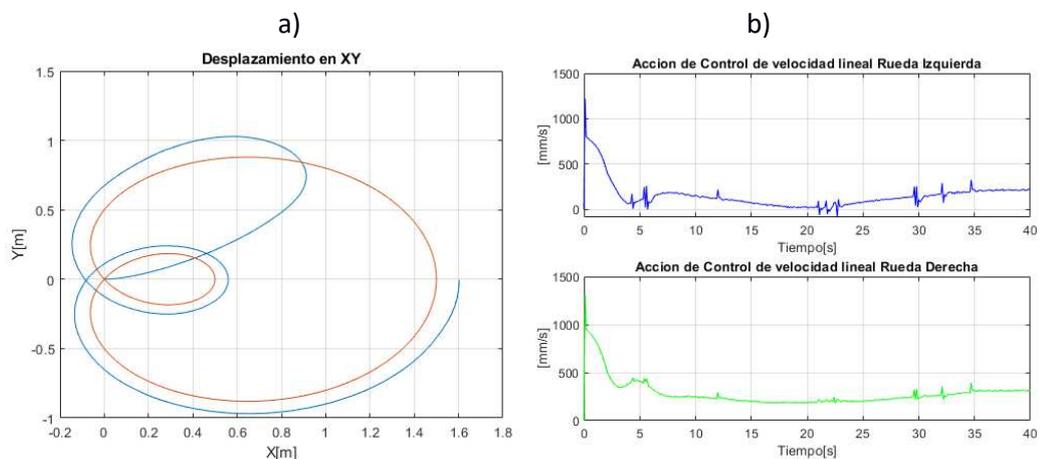
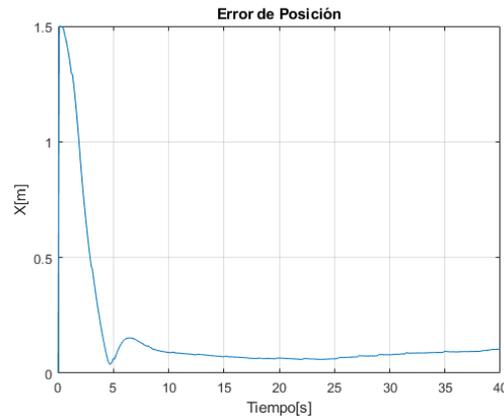


Figura 110

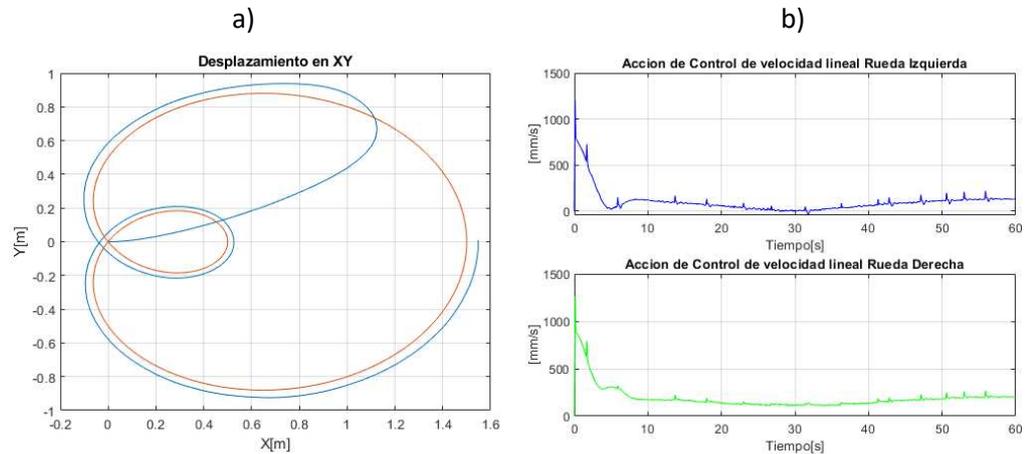
Prueba N°11 PID: Error de posición vs Tiempo.



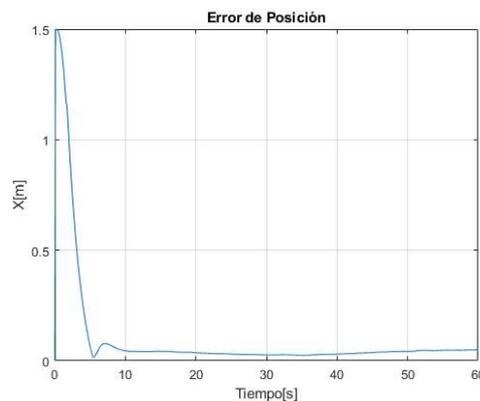
En la Figura 111 a) de la Prueba N°12 con control PID, se observa que el robot móvil realizó el desplazamiento en forma de caracol con un error de 0.049 m durante los 60 segundos, disminuyendo en 0.053 m en comparación al trayecto de 40 segundos como se indica en la tabla 9, con cambios de velocidad que oscilan entre $204 \frac{mm}{s}$ a $239.9 \frac{mm}{s}$ en la rueda derecha y $128.4 \frac{mm}{s}$ a $207.9 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 111 b), los cuales representan cambios de $35.9 \frac{mm}{s}$ (derecha) y $79.5 \frac{mm}{s}$ (izquierda). Cuando el robot móvil realizó el lazo del caracol, su velocidad disminuyó durante el período de 10 a 30 segundos al igual que la prueba anterior, hasta un valor de $8.67 \frac{mm}{s}$ (derecha) y $113.7 \frac{mm}{s}$ (izquierda), disminuyendo en 0.025 m el error de posición y posteriormente regresando al valor de 0.049 m, como se observa en la Figura 112, el tiempo que tarda en estabilizarse es de 9.5 segundos.

Figura 111

Prueba N°12 PID: Trayectoria del caracol en el tiempo $t=60s$.

**Figura 112**

Prueba N°12 Control PID: Error de posición vs Tiempo.



Análisis del Controlador PID2DoF en la Trayectoria en Caracol con lazo

En la Figura 113 a) de la Prueba N°13 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento en forma de caracol, debido a la dificultad de la trayectoria se produce un error de 0.064 m durante los 40 segundos como se indica en la tabla 9, con cambios de velocidad que oscilan entre $316.1 \frac{mm}{s}$ a $322.5 \frac{mm}{s}$ en la rueda derecha y $218 \frac{mm}{s}$ a $280.1 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 113 b), los cuales representan cambios de $6.4 \frac{mm}{s}$ (derecha) y $62.1 \frac{mm}{s}$ (izquierda), donde la rueda derecha

necesitó más velocidad debido a que la trayectoria se realizó en sentido antihorario al igual que la circular con control PID. Cuando el robot móvil realizó el lazo del caracol, su velocidad disminuyó durante el período de 10 a 30 segundos, hasta un valor de $14.69 \frac{mm}{s}$ (derecha) y $130.9 \frac{mm}{s}$ (izquierda), siendo éstas velocidades menores en comparación a la prueba anterior con el controlador PID; disminuyendo en 0.055 m el error de posición y posteriormente regresando al valor de 0.064 m como se observa en la Figura 114, el tiempo que tarda en estabilizarse el error es de 9.5 segundos.

Figura 113

Prueba N°13 PID2DoF: Trayectoria del caracol en el tiempo $t=40s$.

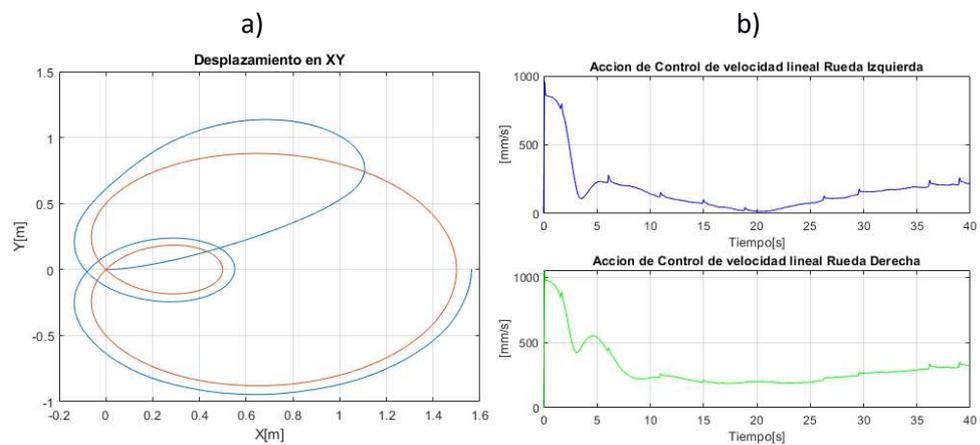
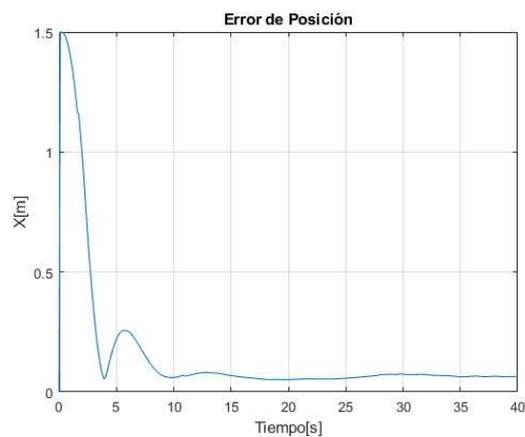


Figura 114

Prueba N°13 PID2DoF: Error de posición vs Tiempo.



En la Figura 115 a) de la Prueba N°14 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento en forma de caracol con un error de 0.027 m durante los 60

segundos, disminuyendo en 0.037 m en comparación al trayecto de 60 segundos como se indica en la tabla 9, con cambios de velocidad que oscilan entre $205 \frac{mm}{s}$ a $217.4 \frac{mm}{s}$ en la rueda derecha y $131.1 \frac{mm}{s}$ a $161 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 115 b), los cuales representan cambios de $12.4 \frac{mm}{s}$ (rueda derecha) y $29.9 \frac{mm}{s}$ (rueda izquierda). Cuando el robot móvil realizó el lazo del caracol, su velocidad disminuyó durante el período de 10 a 30 segundos al igual que la prueba anterior, hasta un valor de $2.37 \frac{mm}{s}$ (rueda derecha) y $119.30 \frac{mm}{s}$ (rueda izquierda), siendo valores más bajos que en la prueba del controlador PID, disminuyendo en 0.011 m el error de posición y posteriormente regresando al valor de 0.027 m, como se observa en la Figura 116, el tiempo que tarda en estabilizarse el error es de 9.5 segundos.

Figura 115

Prueba N°14 PID2DoF: Trayectoria del caracol en el tiempo $t=60s$.

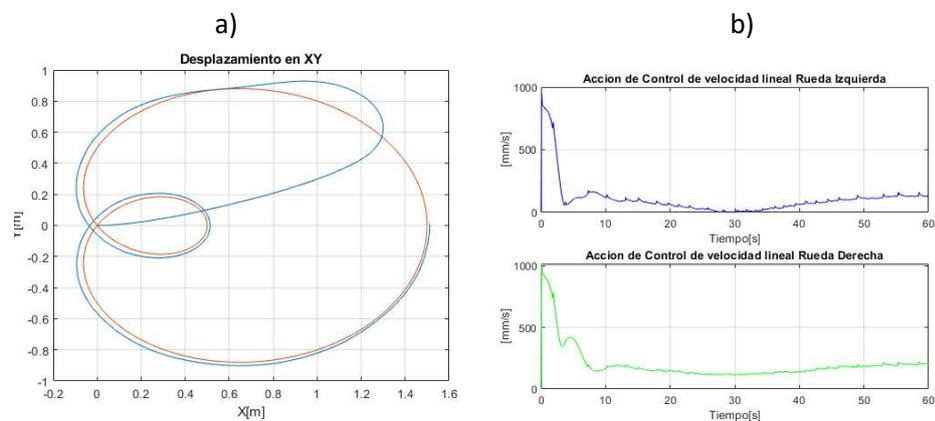
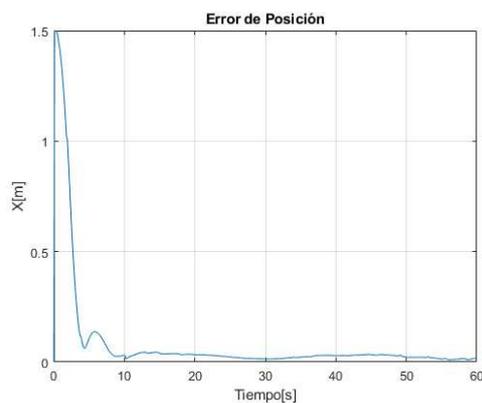


Figura 116

Prueba N°14 Control PID2DoF: Error de posición vs Tiempo.



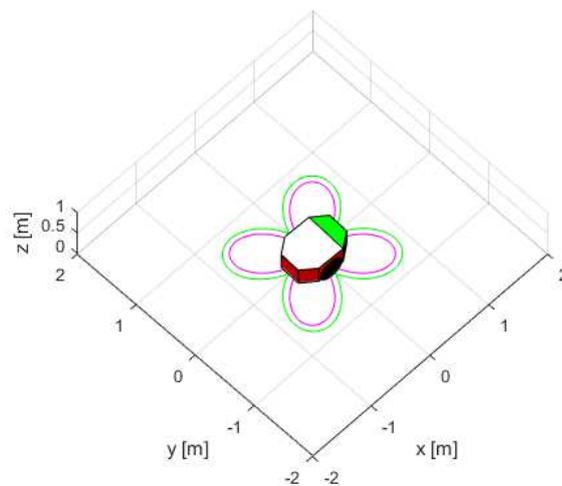
Al utilizar el controlador PID2DoF, se observa que el error disminuye notablemente, e incluso tiene mejor control de velocidad en trayectorias circulares con radios pequeños y giros angostos como en el lazo de la trayectoria del caracol, disminuyendo la velocidad a casi $0 \frac{mm}{s}$, por lo que se obtuvo menor error de posición de 0.011 m con respecto a la trayectoria deseada.

Trayectoria de Rosa de cuatro Pétalos

Se realizaron dos pruebas con tiempos de trayectorias de 40 y 60 segundos para cada controlador al igual que las demás trayectorias. El radio definido para las pruebas es de 1 metro, además se tabuló el error y el porcentaje de error de cada prueba; graficando los mismos parámetros que las anteriores trayectorias.

Figura 117

Trayectoria en caracol con lazo de radio 1 metro.



El robot inicialmente partirá de la posición (0,0) al igual que las demás trayectorias, por lo cual el robot iniciará su recorrido desde el centro de la rosa hasta llegar al extremo derecho del mismo en sentido antihorario y regresar al mismo punto del cual partió, como se indica en la Figura 117.

Tabla 10

Pruebas del controlador PID y PID2DoF en trayectoria de 4 pétalos.

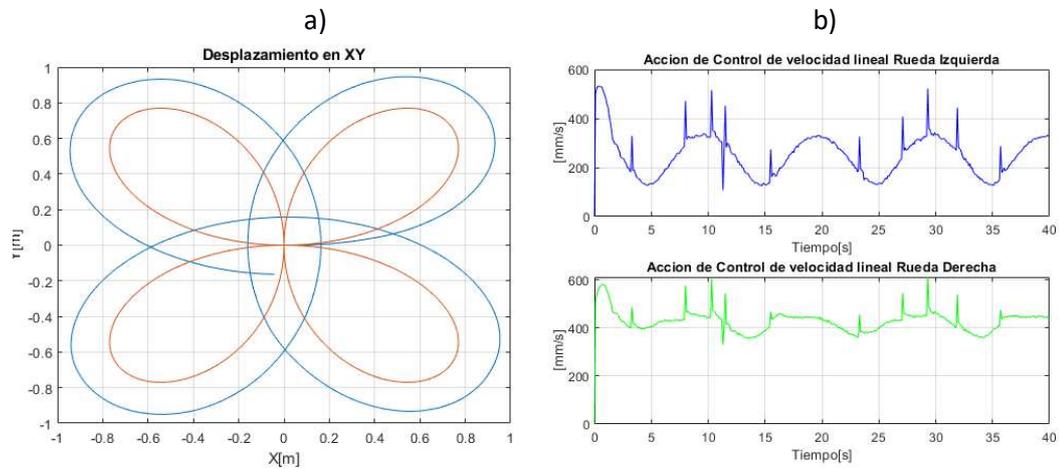
Rosa de cuatro pétalos										
Tiempo de trayectoria (segundos)	Control PID					Control PID2DoF				
	N° Prueba	Radio= 1 metro				N° Prueba	Radio= 1 metro			
		Error (m)		Porcentaje (Giro en cada pétalo)			Error (m)		Porcentaje (Giro en cada pétalo)	
		Min.	Max.	Min(%)	Max(%)		Min.	Max.	Min(%)	Max(%)
40	15	0.168	0.223	16.8	22.3	17	0.152	0.205	15.2	20.5
60	16	0.076	0.092	7.6	9.2	18	0.057	0.078	5.7	7.8

Análisis del Controlador PID en la Trayectoria de rosa de cuatro pétalos.

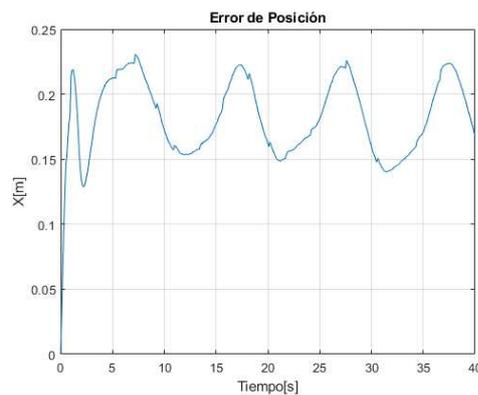
En la Figura 118 a) de la Prueba N°15 con control PID, se observa que el robot móvil realizó el desplazamiento en forma rosa de cuatro pétalos, debido a la dificultad de la trayectoria se produce un error de 0.223 m cuando éste sube para realizar el giro del pétalo, y otro error de posición de 0.168 m cuando se desplaza hacia adentro del pétalo, mientras realizó el recorrido en los 40 segundos como se indica en la tabla 10, con cambios de velocidad que oscilan entre $362.1 \frac{mm}{s}$ a $469.1 \frac{mm}{s}$, en la rueda derecha y $129.7 \frac{mm}{s}$ a $332.1 \frac{mm}{s}$ en la izquierda como se muestra en la Figura 118 b), los cuales se producen por los giros que realiza el móvil para formar los pétalos de la rosa. Al igual que las trayectorias anteriores la rueda derecha necesitó más velocidad debido a que la trayectoria se realizó en sentido antihorario, y es necesario compensar la velocidad para que las dos ruedas del robot se muevan simultáneamente. En esta ocasión los errores de posición varían cada 5 segundos, ya que es el tiempo en que tarda en girar el móvil para realizar el pétalo como se observa en la Figura 119.

Figura 118

Prueba N°15 PID: Trayectoria en el tiempo $t=40s$.

**Figura 119**

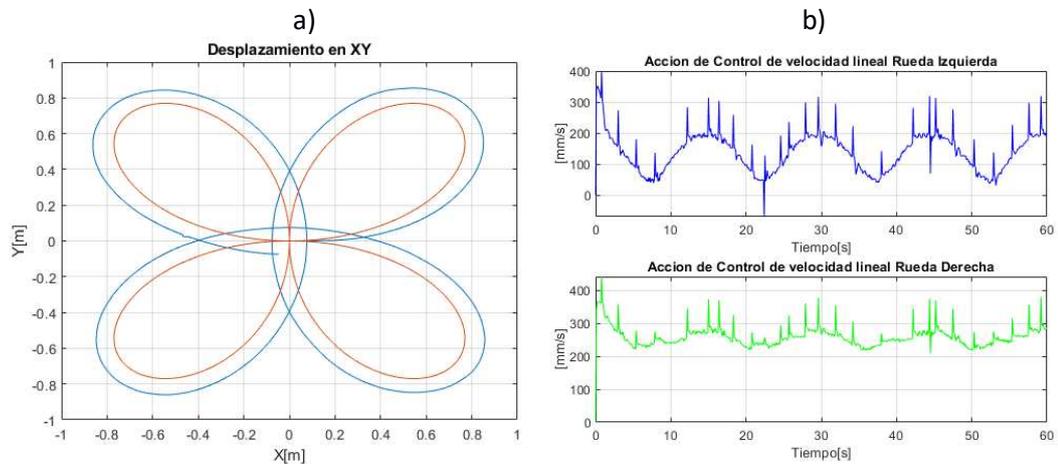
Prueba N°15 PID: Error de posición vs Tiempo.



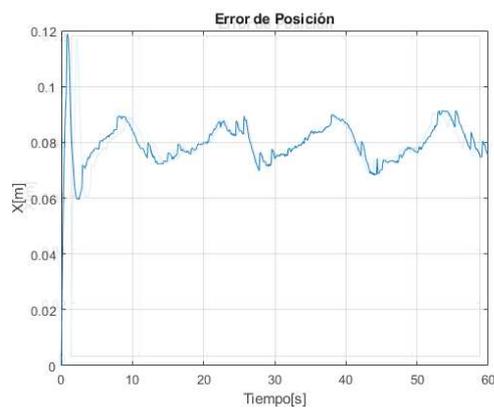
En la Figura 120 a) de la Prueba N°16 con control PID, se observa que el robot móvil realizó el desplazamiento en forma rosa de cuatro pétalos, el error que se generó es de 0.092 m cuando éste sube para realizar el giro del pétalo, y 0.076 m cuando se desplaza hacia adentro del pétalo, mientras realizó el recorrido en los 60 segundos como se indica en la tabla 10, con cambios de velocidad que oscilan entre $221 \frac{mm}{s}$ a $280 \frac{mm}{s}$ en la rueda derecha y $60.37 \frac{mm}{s}$ a $195.5 \frac{mm}{s}$ en la izquierda, como se muestra en la Figura 120 b), los cuales se produjeron por los cambios de giro. Los errores de posición en cada giro varían cada 10 segundos, como se observa en la Figura 121.

Figura 120

Prueba N°16 PID: Trayectoria en el tiempo t=60s.

**Figura 121**

Prueba N°16 PID: Error de posición vs Tiempo.



Al aumentar notablemente el tiempo de trayectoria se evidencia la reducción del error de posición en casi un 50%, al igual que las demás trayectorias cuando se aplicó el control PID.

Análisis del Controlador PID2DoF en la Trayectoria de rosa de cuatro pétalos.

En la Figura 122 a) de la Prueba N°17 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento en forma rosa de cuatro pétalos, produciendo un error de 0.205 metros (20.5 cm) cuando éste sube para realizar el giro del pétalo, y 0.152 metros (15.2 cm) cuando se desplaza hacia adentro del pétalo, mientras realizó el recorrido en los 40 segundos como se indica en la tabla 10 con cambios de velocidad que oscilan entre

362.1 $\frac{mm}{s}$ a, 469.1 $\frac{mm}{s}$ en la rueda derecha y 129.7 $\frac{mm}{s}$ a 332.1 $\frac{mm}{s}$ en la izquierda, como se muestra en la Figura 122 b). Los errores de posición varían cada 5 segundos, ya que es el tiempo en que tarda en girar el móvil para realizar el pétalo como se observa en la Figura 123.

Figura 122

Prueba N°17 PID2DoF: Trayectoria en el tiempo $t=40s$.

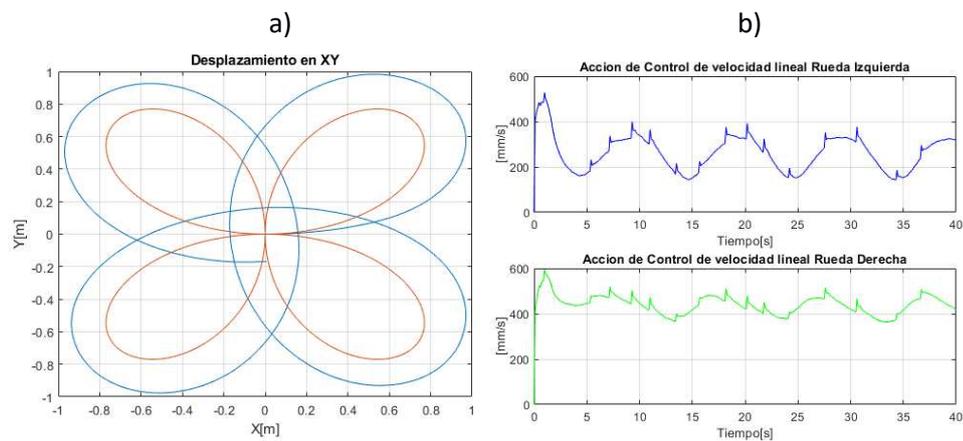
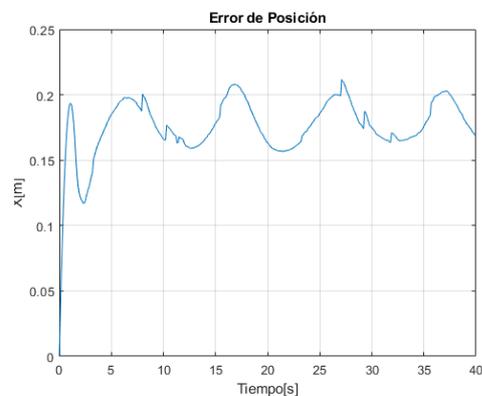


Figura 123

Prueba N°17 PID2DoF: Error de posición vs Tiempo.



En la Figura 124 a) de la Prueba N°18 con control PID2DoF, se observa que el robot móvil realizó el desplazamiento en forma rosa de cuatro pétalos, el error que se generó es de 0.078 m cuando éste sube para realizar el giro del pétalo, y 0.076 m cuando se desplaza hacia adentro del pétalo, mientras realizó el recorrido en los 60 segundos como se indica en la tabla 10, con cambios de velocidad que oscilan entre 218.7 $\frac{mm}{s}$ a 272.5 $\frac{mm}{s}$ en la rueda derecha y 45.98 $\frac{mm}{s}$ a 185 $\frac{mm}{s}$ en la izquierda, como se muestra en la Figura 124 b), los

cuales se produjeron por los cambios de giro. Los errores de posición en cada giro varían cada 10 segundos, y disminuyeron en casi 0.10 m con respecto al trayecto de 40 segundos, como se observa en la Figura 125.

Figura 124

Prueba N°18 PID2DoF: Trayectoria en el tiempo $t=60s$.

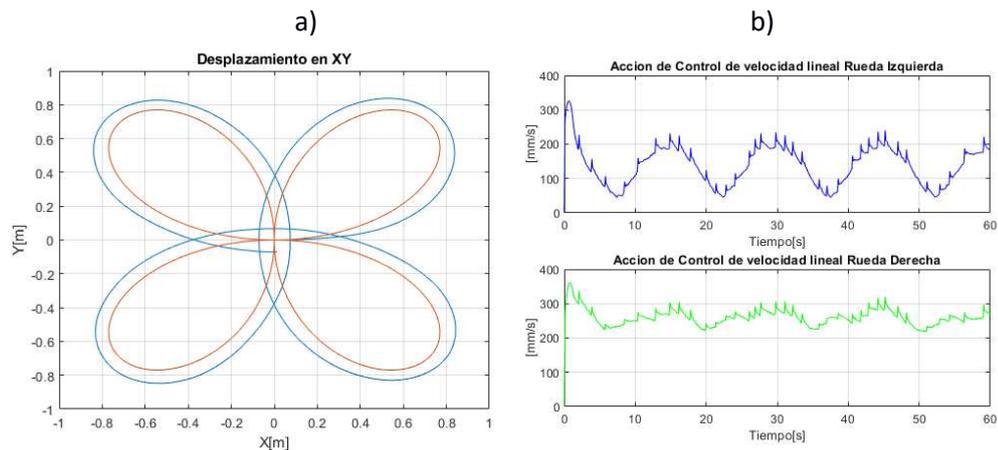
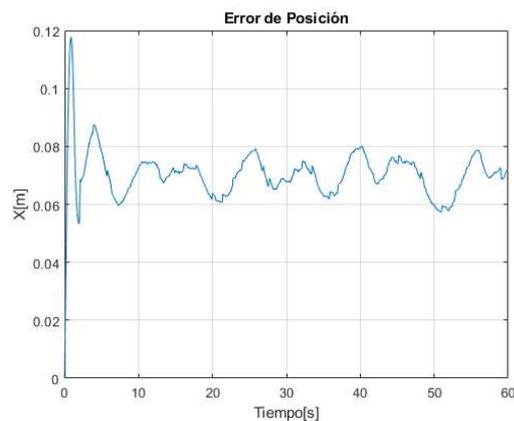


Figura 125

Prueba N°18 PID2DoF: Error de posición vs Tiempo.



Planificación de trayectorias

En esta etapa de prueba, se incorporó todos los componentes del sistema de navegación: modelo diferencial, planificación, sistema de control y evasión, con el objetivo de demostrar el funcionamiento en base a mapas del entorno de trabajo del robot. Por lo tanto, se utilizó un mapa donde se trazó una ruta colocando un punto inicial y un punto objetivo, obtenida a través de la ejecución de las planificaciones de trayectorias mencionadas en el capítulo 4. El mapa del entorno de trabajo del robot se basó en el patio

ubicado en los laboratorios de electrónica de la Universidad de las Fuerzas Armadas ESPE, incorporando dos obstáculos para el procesamiento de la ruta, además se añadió tres obstáculos al entorno de trabajo del robot, sin proyectarles en el mapa, como se indica Figura 126 y comprobar el sistema de evasión y a su vez el sistema de seguimiento para el robot.

Figura 126

Entorno de trabajo.



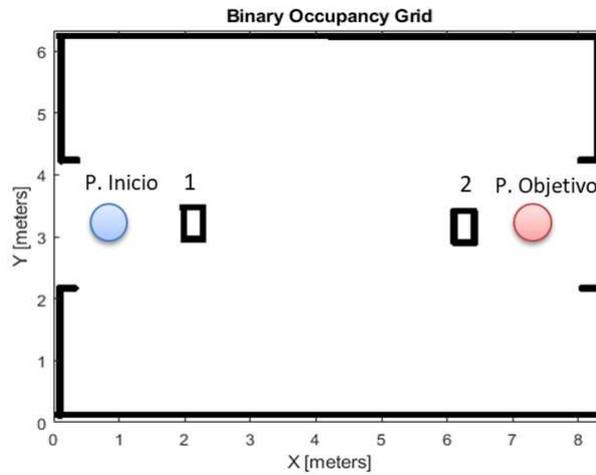
Para las trayectorias RRT, RRT* y Híbrido A*, el mapa se graficó en el software Mapper, permitiendo proyectar el entorno de trabajo con medidas en metros; posteriormente se guardó el archivo en extensión .bmp, en la carpeta de Matlab para crear un mapa de ocupación 2D, y poder representar el espacio de trabajo del robot móvil.

Las gráficas presentadas para las planificaciones mencionadas a continuación, se obtuvieron del simulador MobileSim, debido a que se logró los mismos resultados que el robot físico.

En la Figura 127, se presenta dos obstáculos (llamaremos 1 y 2) dentro del espacio de trabajo, cuyo meta fue alcanzar el punto objetivo del extremo derecho a una distancia de 4 m desde el punto inicio en el lado izquierdo.

Figura 127

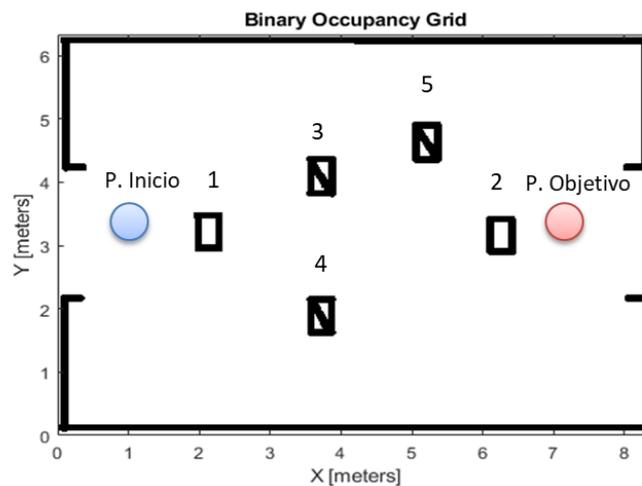
Mapa de ocupación del espacio de trabajo en Matlab.



Se colocarán tres obstáculos (llamaremos 3,4 y 5) que no se encontraban dentro del mapa como se indica en la Figura 128, para verificar la funcionalidad del sistema de evasión y a su vez el sistema de control para el seguimiento.

Figura 128

Obstáculos fuera del mapa de ocupación.

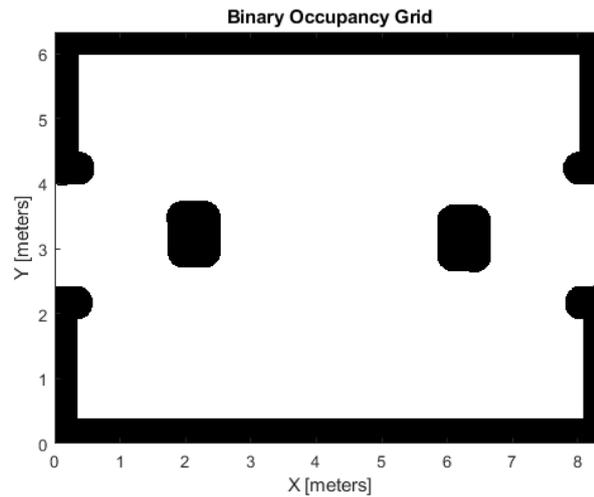


Luego, para evitar colisiones con las paredes o cualquier objeto y asegurar que el robot no tenga rosamientos se infla las paredes como se indica en la Figura 129. Los valores

se modifican usando la inflación en escala de grises, para inflar valores de mayor probabilidad en toda el mapa de ocupación binaria.

Figura 129

Aumento de grosor al mapa de ocupación.

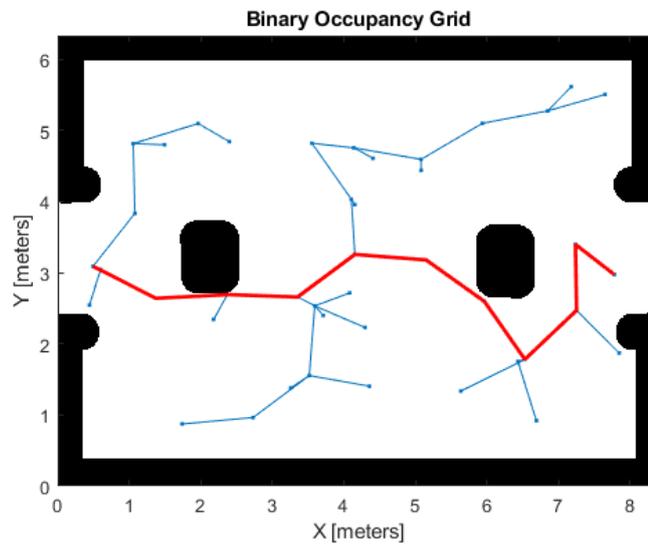


Planificación RRT

Como se había mencionado en el capítulo 4, para la planificación RRT, se definió tres parámetros: la distancia de conexión entre los nodos, el radio de giro y el número de iteraciones, cuyos valores son: 0.5 m, 0.9 m, y 3000, respectivamente. Debido a las características del mapa se obtuvo un árbol enraizado con pocos nodos alrededor, como se observa en la Figura 130. El punto inicial se tomó desde la posición (0.35, 3.05) m, hasta el punto objetivo (7.87, 2.98) m.

Figura 130

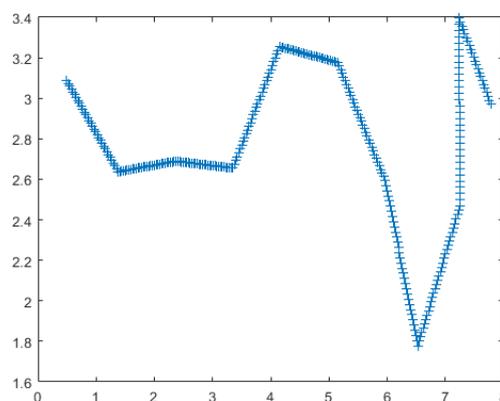
Ruta RRT en el patio del laboratorio.



Luego, se obtuvo los puntos de la ruta generada por la planificación como se indica en la Figura 131, los cuales se usaron para que el robot móvil siga esa ruta, obteniendo un total de 336 puntos que conectan el punto inicial hasta el objetivo sobre el mapa en el plano XY ; cuyo valor también se utilizó como tiempo de trayectoria necesario para trazar la ruta deseada con un tiempo de muestro de 0.1 segundos.

Figura 131

Obtención de los puntos de la ruta determinado por el algoritmo RRT.

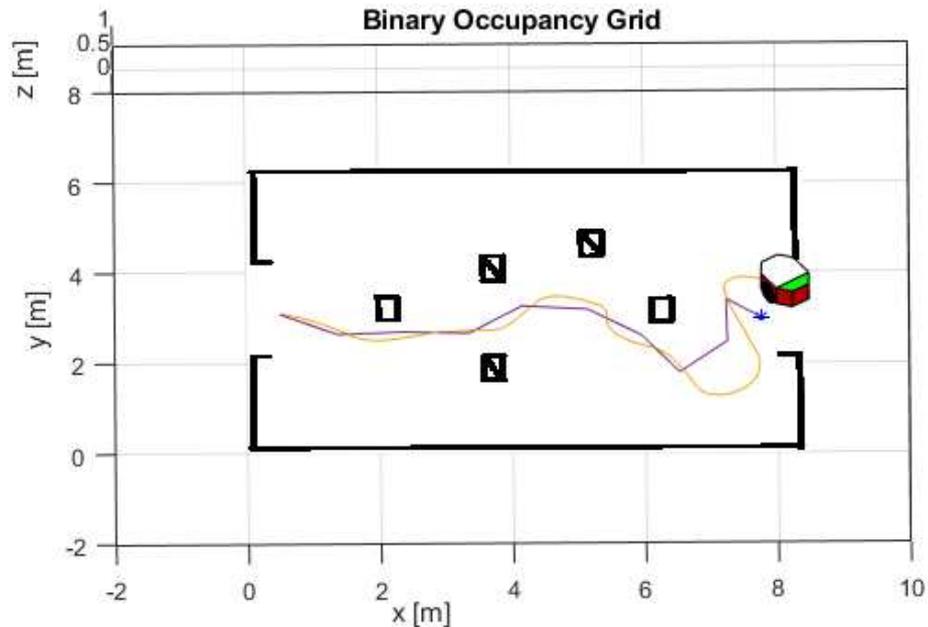


En la Figura 132, se observa que la trayectoria generada por el RRT, se produce debajo de los obstáculos número 1 y 2, donde se evidencia que existe un incremento de

radio en la trayectoria alrededor de los mismos, debido a que el robot móvil comienza a evadir cuando el sensor se encuentra a una distancia de 0.65 m.

Figura 132

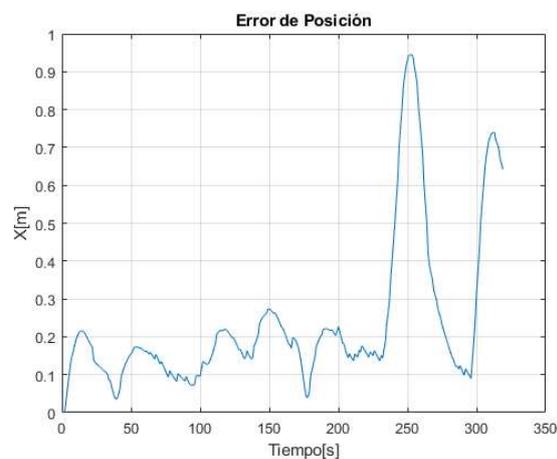
Trayectoria realizada por el móvil por la ruta extraída del algoritmo RRT.



En la Figura 133, se observa el error de posición del robot móvil con respecto la trayectoria deseada, con un error máximo de 0.9m en el tiempo de 250 segundos, debido a que la trayectoria tiene ángulos cerrados y el robot únicamente realiza movimientos circulares o en su propio eje.

Figura 133

Error de posición vs Tiempo de la ruta RRT.

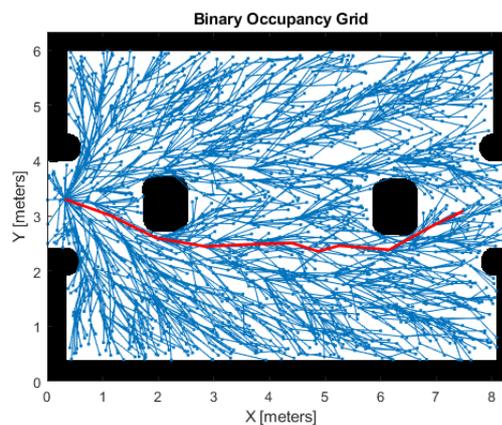


Planificación RRT*

En esta planificación al igual que la RRT se definen los mismos parámetros: distancia de 0.9 m entre cada nodo del árbol, radio de giro de 0.5 m y número de iteraciones de 3000, pero con la diferencia de que el árbol se propaga por todos los espacios vacíos dentro del mapa como se observa en la Figura 134. El punto inicial se tomó el punto (0.28, 3.45) m, hasta el punto (7.54, 3.08) m.

Figura 134

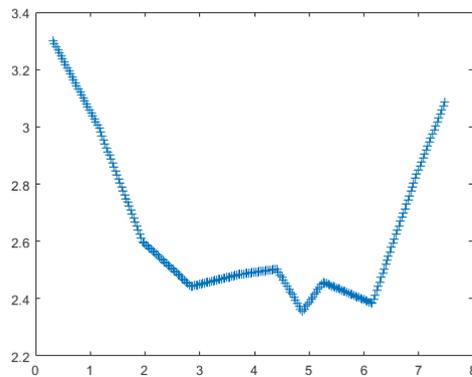
Ruta RRT en el patio del laboratorio.*



En la Figura 135, se observa los puntos extraídos de la ruta generado por el algoritmo, el cual tiene menos ángulos cerrados en comparación al algoritmo RRT debido a que existe más nodos creados en el mapa y construye una ruta con mayor posibilidad de control. La ruta se formó con un total 296 puntos que conectan el recorrido desde el punto inicial hasta el deseado sobre el mapa en el plano XY ; cuyo valor se utilizó al igual que la trayectoria RRT, como tiempo de trayectoria que emplea el robot para trazar la ruta deseada con un tiempo de muestro de 0.1 segundos.

Figura 135

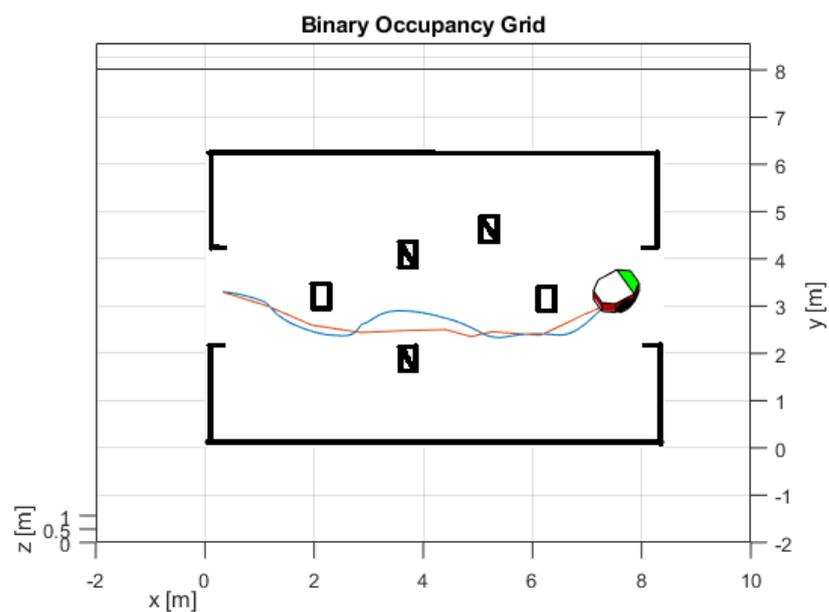
Representación de los puntos de la ruta determinado por el algoritmo RRT.*



En la Figura 136, se observa la trayectoria generada por el RRT* se produce debajo de los obstáculos número 1 y 2. Debido a que hay más nodos para suavizar la ruta se produce más conexiones rectilíneas por lo que existe mayor control en el seguimiento de la trayectoria. Sin embargo, el camino se sobrepone sobre el obstáculo número 4, por lo que existe un aumento en el radio de la trayectoria que pasa alrededor del mismo y el obstáculo 1, y a la evasión.

Figura 136

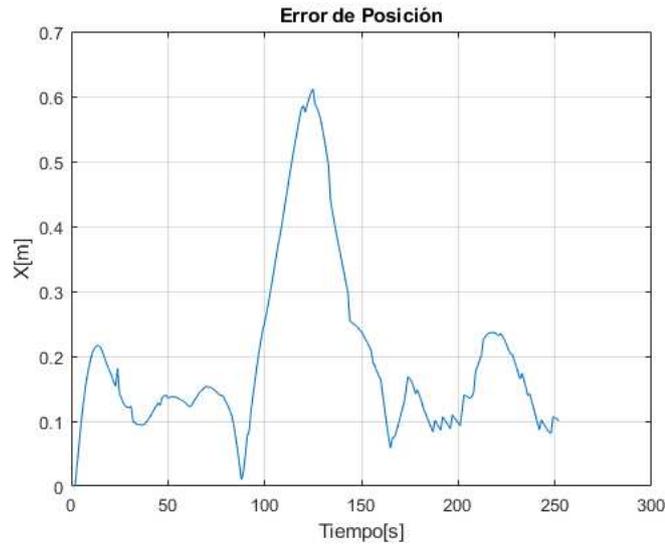
Trayectoria realizada por el móvil por la ruta extraída del algoritmo RRT.*



En la Figura 137, se observa que existe un error constante de 0.2m pero hay un incremento de 0.6m en el tiempo de 125 segundos, debido a la evasión que tuvo que realizar el robot por el obstáculo número 4.

Figura 137

*Error de posición vs Tiempo de la ruta RRT**

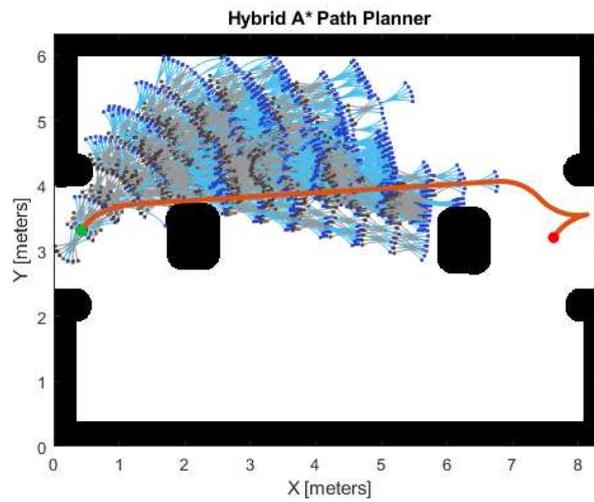


Planificación Híbrido A*

Para la planificación A* como se había mencionado en el capítulo 4, se definen dos parámetros: distancia de conexión la celdas vecinas y el radio de giro del móvil, cuyos valores son: 0.8 m y 0.5 m, respectivamente, donde se visualiza la propagación de las conexiones entre las celdas de menor coste, dentro del mapa como se observa en la Figura 138. El punto inicial se tomó el punto (0.46, 3.35) m, hasta el punto (7.62, 3.52) m.

Figura 138

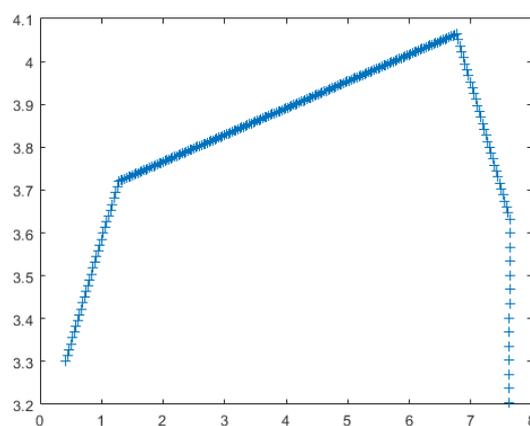
Ruta RRT en el patio del laboratorio.



En la Figura 139, se observa los puntos extraídos de la ruta generado por el algoritmo, el cual tiene menos conexiones con ángulos cerrados debido a que existe más nodos, crea un camino con mayor posibilidad de control. La ruta se formó con un total 295 puntos para la conexión entre el punto inicial y el deseado, tomando ese valor al igual que en las dos anteriores planificaciones, como tiempo de trayectoria.

Figura 139

Obtención de los puntos de la ruta del algoritmo A Híbrido.*

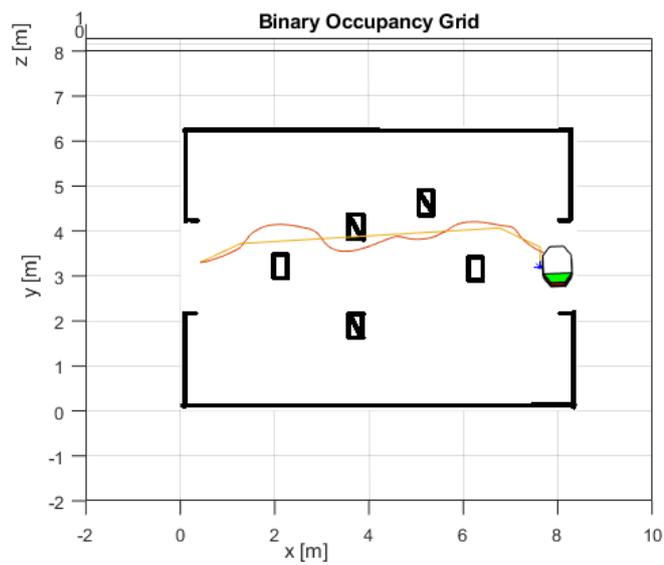


En la Figura 140, se observa la trayectoria generada por el híbrido A* se produce encima de los obstáculos número 1 y 2. La ruta extraída tiene más conexiones rectilíneas

por lo que existe mayor control en el seguimiento de la trayectoria al igual que el RRT*. Sin embargo, el camino se sobrepone sobre los obstáculos número 3 y 5, por lo que existe un incremento en el radio de la trayectoria que pasa alrededor de los mismos y del obstáculo 1, debido a la evasión.

Figura 140

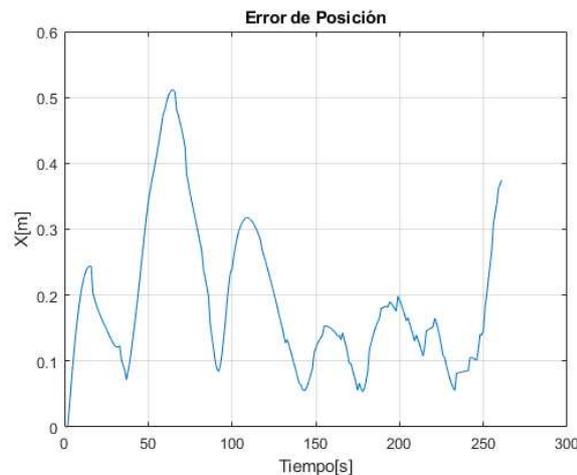
Trayectoria realizada por el móvil por la ruta del algoritmo Híbrido A.*



En la Figura 141, se observa el error de posición del robot móvil con respecto a la trayectoria deseada, con un error de 0.5m al evadir el obstáculo número 1, 0.3m de error en el obstáculo 3, y 0.2 m en el obstáculo 5.

Figura 141.

Error de posición vs Tiempo de la ruta Híbrido A.*



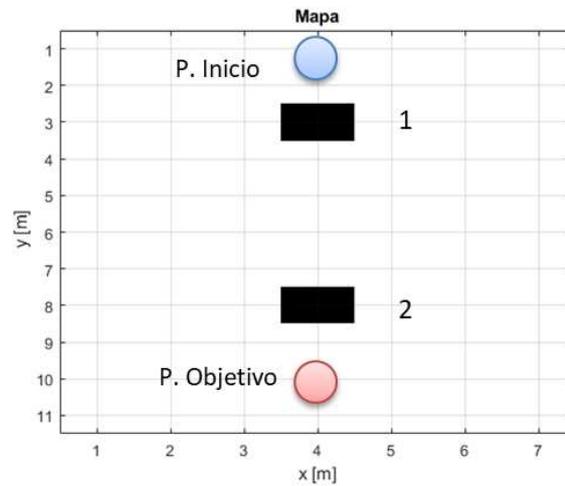
Al analizar los errores de posición, se determina que existe mayor control en el algoritmo A* Híbrido y RRT * debido a que creó una ruta con menos conexiones con ángulos cerrados entre los nodos, por lo cual el control reduce el cambio abrupto de velocidad. Se verifica que existe menor control en el algoritmo RRT, debido a que no muestrea el mapa como los dos anteriores algoritmos, generando menor cantidad de nodos, produciendo de esta manera conexiones con ángulos demasiados pequeños que en ocasiones el móvil no puede manejar y le es imposible continuar con la ruta.

Buscador A*

Para el algoritmo A*, como se había mencionado anteriormente no se utilizan las librerías de Matlab para generar la trayectoria, por lo que el mapa tiene que ser generado a través de matrices de 0 y 1 como se indica en la Figura 142, tomando en consideración la distancia de 4 metros entre los puntos de inicio y objetivo como las anteriores trayectorias y la dimensión del patio del laboratorio.

Figura 142

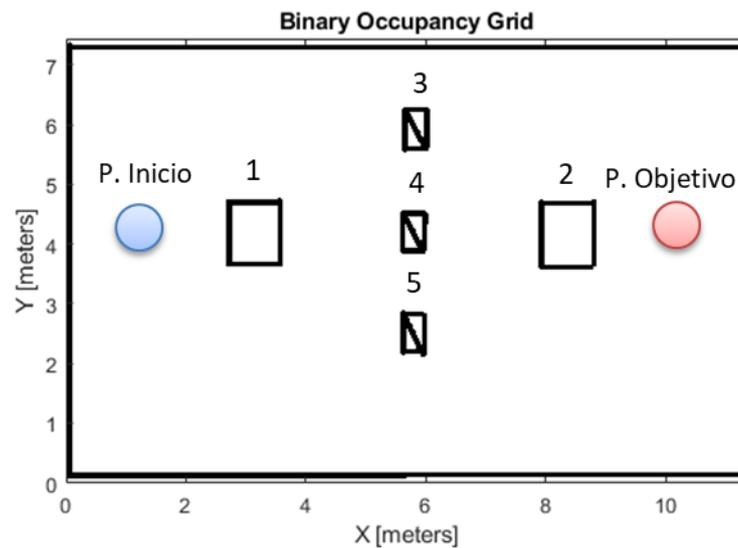
Mapa de ocupación para el algoritmo A*.



Para la visualización de la evasión a través de la interfaz gráfica de Matlab, se genera el mapa con los obstáculos al igual que las demás trayectorias como se indica en la Figura 143, con las mismas dimensiones que los obstáculos generados por el mapa de matrices de "0" y "1" y los que no están dentro del mismo.

Figura 143

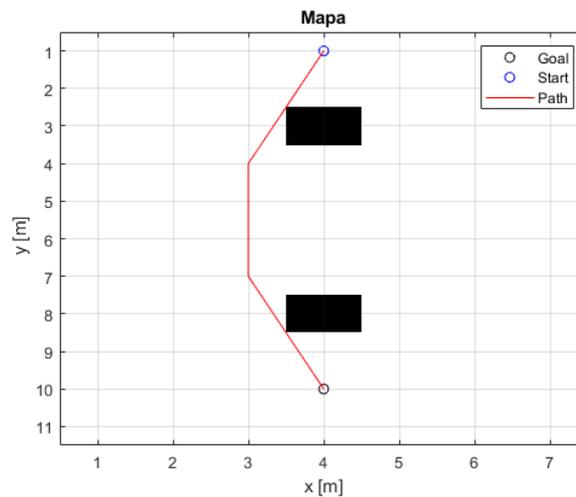
Mapa de ocupación del espacio de trabajo en Matlab.



En la planificación A* se definió tres parámetros como: punto inicial (4, 1) m, y punto (4, 10) m con una distancia de conexión de 3, es decir 32 direcciones en cada nodo para trazar la ruta como se indica en la Figura 144.

Figura 144

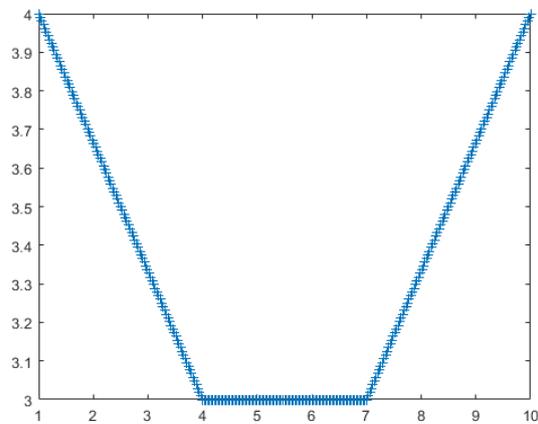
Ruta A en el patio del laboratorio.*



En la Figura 145, se observa los puntos extraídos de la ruta generado por el algoritmo A*, el cual tiene menos ángulos cerrados en comparación a las demás planificaciones, debido a que existe menos nodos por evaluar gracias a la matriz del mapa de ocupación, por lo tanto, la ruta se genera con el menor número de conexiones entre el punto inicial y final. El camino extraído tiene un total 303 puntos sobre el mapa en el plano XY; cuyo valor se utilizó al igual que las demás trayectorias, como tiempo de trayectoria para trazar la ruta deseada con un tiempo de muestro de 0.1 segundos.

Figura 145

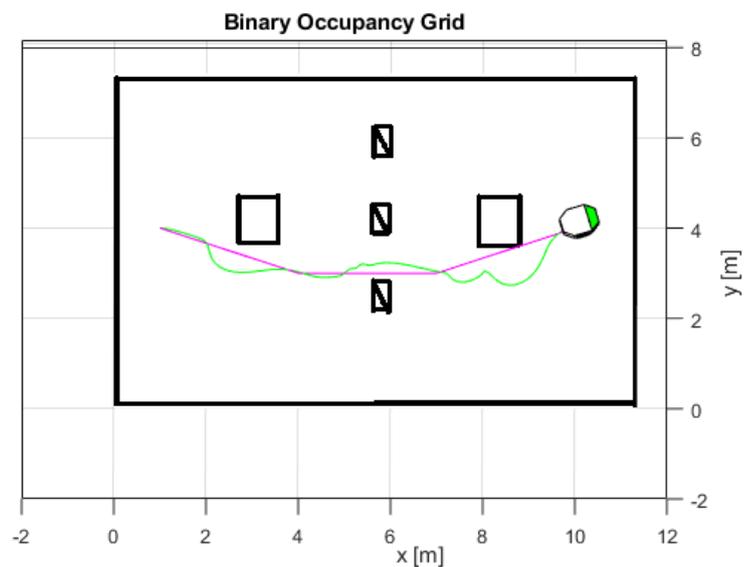
Representación de los puntos de la ruta determinado por el algoritmo A.*



En la Figura 146, se observa la trayectoria generada por el A* se produce debajo de los obstáculos número 1 y 2. La ruta extraída tiene tres conexiones rectilíneas por lo que existe mayor control en el seguimiento de la trayectoria al igual que la generación RRT* e híbrido A*. Sin embargo, el camino pasa muy cerca de los obstáculo número 1, 2 y 5, por lo que existe un incremento en el radio de la trayectoria alrededor de los mismos debido a la evasión.

Figura 146

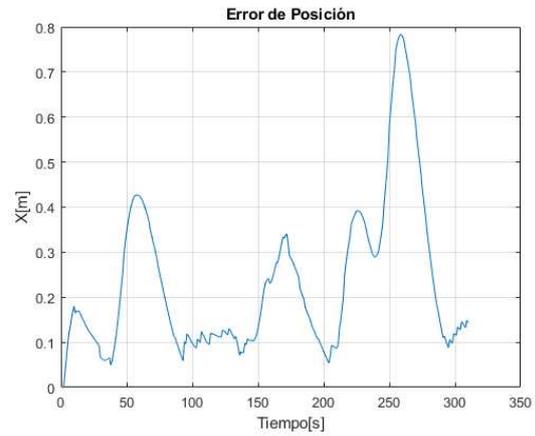
Trayectoria realizada por el móvil por la ruta extraída del algoritmo A.*



En la Figura 147, se observa que existe un error constante de 0.1 m pero hay un incremento de 0.4m al evadir el obstáculo 1, 0.5m en el obstáculo 5 y 0.8 m en el número 2.

Figura 147

Error de posición vs Tiempo de la ruta Híbrido A.*



Capítulo VI

Conclusiones y Recomendaciones

Conclusiones

- Se realizó un modelado general basado en un robot móvil de configuración diferencial, el cual utiliza principalmente las velocidades de las ruedas. Logrando, incorporar dos alternativas de sistemas de control, PID y PIF2DoF, y una opción de control evasión basado en el método de impedancia, cuyos sistemas pueden aplicarse en otros robots con la misma disposición, pero tomando en cuenta los sensores que posee para realizar el control ya sea de posición o velocidad.
- Se implementó dos formas de generación de trayectorias, la primera basada en fórmulas matemáticas y la segunda en la planificación de trayectorias, cuya programación se basa en dos formas: importación de librerías Matlab y desarrollo de pseudocódigo expresado en un diagrama de flujo.
- Se implementó dos tipos de control PID y PID2DoF, los cuales se utilizaron en el seguimiento de trayectorias, obtenidas de los dos tipos de generaciones mencionadas anteriormente.
- Se incorporó el sistema de control de evasión basado en el método por impedancia, el cual se incorporó a la cinemática inversa y al sistema de control PID, por facilidades en el diseño, ya que maneja menos señales de control a diferencia del PID2DoF, que maneja seis.
- Se concluye que el modelo cinemático directo tiene un error aproximado de 0.04 m, en movimientos traslacionales y 0.12 m en rotacionales, cuyo error aumenta cuando se aumenta la velocidad en las ruedas y disminuye cuando se reduce; esto ocurre debido al tiempo de muestro y al retardo que existe en el accionamiento de los motores ordenado por el comando utilizado `arobot_setwhelevls`.

- Se concluye que el modelo cinemático inverso tiene un error del 0.04 m en trayectorias rectilíneas, mientras que en trayectorias circulares realiza el recorrido deseado, pero con un error de posición de 0.1 m.
- Se concluye que el bloque derivativo en la obtención de las ganancias para el sistema de control del robot físico, el cual se suma al bloque del controlador PID, y PID2DoF, implica una reducción en el error de posición de 0.05 m, en comparación a usar únicamente el bloque del PID y PID2DoF, por lo tanto, se considera el uso de este bloque para tener mejores resultados en el seguimiento de trayectorias con errores mínimos de posición.
- Se concluye que al utilizar el controlador PID2DoF, se obtiene un error de posición aproximado de 0.01 m a 0.03 m en todas las trayectorias realizadas con tiempos mayores a 40 segundos, superando al controlador PID con 0.02 m con respecto al error de posición, obteniendo mejores resultados en el seguimiento de trayectorias.
- Se concluye que la planificación de trayectoria más eficiente es el híbrido A*, ya que, al generar primitivas de movimientos curvos, se obtiene una ruta más suave y con menos ángulos cerrados, al interpolar cada nodo. Por lo tanto, el robot móvil puede realizar menos cambios de velocidad abruptos sin perder el control de la trayectoria hasta llegar al objetivo final.
- Se concluye que la planificación RRT, tiene trayectorias más complejas a seguir debido a la conexión entre los nodos generados por el algoritmo dentro el mapa, el cual produce numerosos ángulos cerrados entre cada nodo durante la conexión, por lo cual el robot móvil no puede seguir ese tipo de trayectoria, perdiendo el control al intentar seguir el camino generado, causado por el cambio constante de velocidad en las ruedas.
- Se concluye que al utilizar el algoritmo RRT*, se obtiene una ruta con menos ángulos cerrados, ya que los nodos y el árbol generado se expande por todo el espacio vacío,

escogiendo los nodos con menos coste (menor distancia), generando un camino más suave para el robot móvil en comparación al RRT, consiguiendo mayor control en la velocidad en las ruedas.

- Se concluye que al utilizar el algoritmo A*, se obtiene menos ángulos cerrados debido a que se evalúan menos nodos, en un mapa de ocupación con menos celdas, por lo tanto, se obtiene una ruta con menos conexiones que las demás planificaciones, y el robot puede seguir la trayectoria con menor error de posición.

Recomendaciones

- Al realizar pruebas de la implementación de los algoritmos de los controladores en el modelo diferencial propuesto al robot móvil, se utilizó como realimentación las posiciones estimadas, obteniendo errores menores al 10% en las trayectorias trazadas, sin embargo, es recomendable utilizar como retroalimentación las velocidades, ya que son las que se encarga de mover el robot móvil.
- Al utilizar los sensores ultrasónicos como un indicativo para iniciar la evasión, se demuestra que el robot móvil puede evadir obstáculos y regresar a la trayectoria predefinida con el algoritmo propuesto, pero se recomienda utilizar mejores sensores para obtener mejor procesamiento del entorno del robot, en el que se está moviendo como: cámaras, GPS, láser. etc.
- Se recomienda utilizar un tiempo de muestro de 0.1 segundos, ya que al utilizar el comando `arobot_setwheellvels` se obtiene un error de posición mínimo de 0.465 m para una distancia de 1 m con una velocidad de $100 \frac{mm}{s}$, mientras que usando un tiempo de muestro de 0.001 se generan un error de 0.959 m con la misma distancia y velocidad; y por ende mayor error en distancias de mayor longitud.
- Se recomienda utilizar tiempos de trayectorias mayores a 40 segundos en trayectorias circulares con radios mayores a 0.5 m, siendo esta longitud mayor al

tamaño del móvil, ya que el robot no consigue moverse a velocidades menores, para realizar éstas trayectorias.

- Se recomienda realizar la evasión de obstáculo en lugares, que no alteren la sensibilidad de los sensores, además de lugares ideales como paredes y sin espacios con trasfondos, ya que los sensores comienzan a detectar objetos con una altura de 25 cm y no menores a éstos.

Referencias bibliográficas

- Agarwal, K., Mahtab, S., Bandyopadhyay, S., & Gupta, S. D. (2013). *A Proportional-Integral-Derivative Control Scheme of Mobile Robotic platforms using MATLAB*. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE).
<https://pdfs.semanticscholar.org/d9dc/9342f9b656d2f048afe2a258ef6aeff28035.pdf>
- Alfaro, V., & Vilanova, R. (2007). *Robust tuning and performance analysis of 2DoF PI controllers for integrating controlled processes*. ResearchGate. 23
https://www.researchgate.net/publication/291994451_Robust_tuning_and_performance_analysis_of_2DoF_PI_controllers_for_integrating_controlled_processes
- Avanti, P. (2008). *Una Introducción a los Robots Móviles*.
https://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf
- Bermúdez, G. (2002). *Robots Móviles. Aplicaciones y experiencias*. ResearchGate, 12.
https://www.researchgate.net/profile/Giovanni_Bermudez/publication/267798579_ROBOTS_MOVILES_TEORIA_APLICACIONES_Y_EXPERIENCIAS/links/54b57af10cf2318f0f99c81c.pdf
- Bermudez, G. (2003). *Modelamiento cinemático y odométrico de robots móviles: aspectos matemáticos*. Researchgate, 2.
https://www.researchgate.net/publication/233603982_Modelamiento_cinematico_y_odometrico_de_robots_moviles_aspectos_matematicos
- Capito, L., & Proaño, P. (2011). *Seguimiento de Trayectorias mediante cuatro Técnicas de Control utilizando una Plataforma Robótica Pioneer 3DX y el Sistema Operativo Robótico ROS*. [Tesis de Grado, Escuela Politécnica Nacional de Ecuador]. <https://bibdigital.epn.edu.ec/bitstream/15000/4321/1/CD-3931.pdf>

- Chin, T. (2019). *Medium: Robotic Path Planning: RRT and RRT**.
<https://medium.com/@theclassytim/robotic-path-planning-rrt-and-rrt-212319121378>
- Cova, W. J. (2005). *Control pid un enfoque descriptivo*. Revista de la Universidad Tecnológica Nacional,35.
http://www.frlr.utn.edu.ar/archivos/alumnos/electronica/catedras/38-sistemas-de-control-aplicado/Publicaciones/Control_PID_Enfoque_Descriptivo.pdf
- Cruz, H. (2008). *Introducción a los robots móviles*. Academia.edu, 86.
https://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf
- Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2008). *Practical Search Techniques in Path Planning for Autonomous Driving*. American Association for Artificial Intelligence. https://ai.stanford.edu/~ddolgov/papers/dolgov_gpp_stair08.pdf
- Espitia Cuchango, H. E., & Sofrony Esmeral, J. I. (2012). *Algoritmo para planear trayectorias de robots móviles*. Revista de tecnología Redalyc, 3-4.
<https://www.redalyc.org/pdf/911/91126903005.pdf>
- Ferreira, A., Pereira, F. G., Vassallo, R. F., Filho, T. F., & Filho, M. S. (2008). *Un enfoque para evitar obstáculos en la navegación de robots móviles: el escape tangencial*. Controle & Automação Sociedade Brasileira de Automatica, 8.
<http://www.sbai2015.dca.ufrn.br/download/artigo/163>
- Flores, M., & Proaño, A. (2013). *Evolución artificial y robótica autónoma desarrollada en el robot Pioneer 3-dx con aproximaciónn basada en comportamientos*. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].
<http://repositorio.espe.edu.ec/bitstream/21000/16071/1/T-ESPE-038629.pdf>
- Guzmán, L. E., Villa, M. A., & Vásquez, E. L. (2014). *Seguimiento de trayectorias con un robot móvil de configuración diferencial*. [Tesis de Grado, Universidad Militar Nueva Granada]. <http://revistas.usbbog.edu.co/index.php/IngUSBmed/article/view/298>

- Hedges, R. (2013). *GitHub: Aria_Matlab*. <https://github.com/reed-adept/aria-matlab/blob/master/sdefs.m>
- Holman, M., Jacinto, E., & Martínez, F. H. (2014). *Generación de Ruta Óptima para Robots Móviles a Partir de segmentación de movimientos*. <https://scielo.conicyt.cl/pdf/infotec/v26n2/art17.pdf>
- Jácome, K. (2016). *Diseño e Implementación de un robot móvil utilizando un arduino nano y controlado mediante bluetooth*. Guayaquil. [Tesis de Grado, Universidad Católica de Santiago de Guayaquil de Ecuador]. <http://repositorio.ucsg.edu.ec/bitstream/3317/5440/1/T-UCSG-PRE-TEC-ITEL-132.pdf>
- JuYi, B., & Kim, W. K. (2001). *Kinematic Modeling of Omni-Directional Mobile Robots as Parallel Manipulator*. IEEE Journal of Robotics and Automation, 11. <https://www.sciencedirect.com/science/article/pii/S1474667017342908>
- Kumar, U. D., Nisha, M., & Mathivanan, N. (2018). *Tracking of a PID Driven Differential Drive Mobile Robot*. International Journal of mechatronics, electrical and computer technology, 15. http://www.aeuso.org/includes/files/articles/Vol8_Iss27_3690-3704_Tracking_of_a_PID_Driven_Differenti.pdf
- Liang, Y. (2014). *Bidirectional RRT Algorithm for Collision Avoidance Motion Planning of FFSR*. 8. https://www.sensorsportal.com/HTML/DIGEST/august_2014/Vol_177/P_2309.pdf
- López, D., Brav, F. G., Cuesta, F., & Ollero, A. (2006). *Planificación de trayectorias con el algoritmo rrt. aplicación a robots no holonómicos*. Revista Iberoamérica de automática e información industrial, 13. <https://polipapers.upv.es/index.php/RIAI/article/view/8144/8287>
- Malu, S. K., & Majumdar, J. (2014). *Kinematics, Localization and Control of Differential Drive Mobile Robot*. Global Journal of Researches in Engineering, 9.

https://globaljournals.org/GJRE_Volume14/1-Kinematics-Localization-and-Control.pdf

Manual, P. 3. (2006). *MobileRobots Exclusive Advanced Robot Control & Operations Software*. MobileRobots Inc.

https://www.inf.ufrgs.br/~prestes/Courses/Robotics/manual_pioneer.pdf

Martha, A., Bautista, M., & Iruegas, J. (2007). *Diseño y control de robots móviles*. Instituto Tecnológico de Nuevo Laredo, 1-2.

<http://www.mecamex.net/anterior/cong02/papers/art24.pdf>

Martínez, D. A. (2008). *Control remoto por voz del robot móvil pioneer*. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].

<https://repositorio.espe.edu.ec/bitstream/21000/6873/1/AC-EAC-ESPE-047250.pdf>

Martínez, D. J. (2014). *Diseño de un sistema de conducción automático para un robot Pioneer 3-DX en entornos controlados e integración dentro del entorno Robocomp*. [Tesis de Grado, Universidad de Castilla-La Mancha de España] .

http://neithan.weebly.com/uploads/5/2/8/0/52807/memoria_tfg.pdf

mobilerobots, A. (2011). *Especificaciones de la plataforma Pioneer P3-DX*.

<https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf>

Morales, J., & Jaramillo, D. (2010). *Desarrollo de aplicaciones y documentación de la plataforma Pioneer P3-DX y P3-AT*. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].

<http://repositorio.espe.edu.ec/bitstream/21000/246/6/T-ESPE-027504.pdf>

Musuña, J., Zapata, B., Oñate, L., & Campusano, G. (2013). *Diseño y construcción de un robot móvil que permita la obtención de una nube de puntos del escaneo de habitaciones utilizando láser y webcams*.

https://www.researchgate.net/publication/318355901_Disenyo_y_construccion_de_

un_robot_movil_que_permite_la_obtencion_de_una_nube_de_puntos_del_escaneo_o_de_habitaciones_utilizando_laser_y_webcams

Parra Andrade, G. E. (2015). *Control del robot Pioneer 3D utilizando una FPGA Rio*. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].

<http://repositorio.espe.edu.ec/handle/21000/10792>

Petereit, J., Emter, T., & Frey, C. W. (s.f.). *Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments*. ResearchGate, 6.

https://www.researchgate.net/publication/260964861_Application_of_Hybrid_A_to_an_Autonomous_Mobile_Robot_for_Path_Planning_in_Unstructured_Outdoor_Environments

Petrović, E., Nikolić, V., Ćirić, I., Simonović, M., Pavlović, S., Mančić, M., & Rajković, B. (2016).

Kinematic model and control of robot mobile for trajectory tracking. International Journal of Engineering , 4. <http://annals.fih.upt.ro/pdf-full/2016/ANNALS-2016-2-25.pdf>

Guffanti, D., Proaño V.,(2015).*Control remoto por voz del robot móvil pioneer*. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].

P3-DX. <https://repositorio.espe.edu.ec/bitstream/21000/6873/1/AC-EAC-ESPE-047250.pdf>

Ramos, E. (2010). *Modelado, simulación y construcción de un robot móvil tipo diferencial*.

ResearchGate, 11.

https://www.researchgate.net/publication/47297317_Modelado_simulacion_y_construccion_de_un_robot_movil_de_ruedas_tipo_diferencial

Ramos, O. (2016). *Cinematika de Robots Móviles*. IEEE International Conference on Robotics and Automation.

https://scielo.conicyt.cl/scielo.php?script=sci_abstract&pid=S0718-33052016000100016&lng=es&nrm=iso

- Rosales, A. (2009). *Navegación de robots móviles, en entornos no estructurados utilizando álgebra lineal*. Revista Iberoamérica de Automática e Informática industrial, 9.
<https://www.sciencedirect.com/science/article/pii/S1697791209700962>
- Santillán, P. (2015). *Control neuronal del robot móvil Pioneer P3-DX*. Sangolquí. [Tesis de Grado, Universidad de las Fuerzas Armadas ESPE de Ecuador].
<https://repositorio.espe.edu.ec/bitstream/21000/10621/3/T-ESPE-049168-D.pdf>
- Siegwart, R., & Nourbakhsh, I. (2004). *Introduction to Autonomous Mobile Robot*. London, England: Massachusetts Institute of Technology.
<http://dl.icdst.org/pdfs/files/87a07ba36230cb43e6b48791e929c3b0.pdf>
- Swift, N. (27 de 02 de 2017). *Medium: Búsqueda de caminos fácil A **.
<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>
- Ueland, E. (5 de 08 de 2019). *Github: Algoritmo de búsqueda A * (Astar / A Star)*.<https://github.com/EinarUeland/Astar-Algorithm>
- Valencia, J., Montoya, A., & Ríos, L. (2009). *Modelo cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odométrica*. Revista Scientia et Technica , 6. <https://dialnet.unirioja.es/descarga/articulo/4729008.pdf>
- Vinssa, S. (2009). *Vinssa: Industrial Solutions*. <https://blog.vinssa.com/tecnologia-en-la-industria-robots-moviles>
- Wai, R., & Liu, C. (2009). *Design of dynamic Petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot*. IEEE transactions on Industrial Electronics, 16.
<https://www.sciencedirect.com/science/article/pii/S1877050911004479/pdf?md5=4e8eb95f265263082dbd6e859302f56d&pid=1-s2.0-S1877050911004479-main.pdf>

