



Diseño y construcción de un robot móvil tipo diferencial didáctico para aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE sede Latacunga

Macas Llamuca, Johan Enrique y Recalde Simancas, Luis Fernando

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de ingeniero en mecatrónica

Ing. Constante Prócel, Patricia Nataly

19 de marzo de 2021



DEPARTAMENTO DE ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "***DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL TIPO DIFERENCIAL DIDÁCTICO PARA APLICACIÓN DE TÉCNICAS DE CONTROL BASADO EN ALGORITMOS TRADICIONALES E INTELIGENTES EN EL LABORATORIO DE LA CARRERA INGENIERÍA MECATRÓNICA DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE SEDE LATACUNGA.***" fue realizado por los señores ***Macas Llamuca, Johan Enrique y Recalde Simancas, Luis Fernando.*** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 19 marzo del 2021

Firma:



Firmado electrónicamente por:
PATRICIA NATALY
CONSTANTE PROCEL

Ing. Constante Prócel, Patricia Nataly

C.C.:0503354029



Document Information

Analyzed document	Recalde_Macas_Robot_movil_algoritmos_control.pdf (D98712401)
Submitted	3/17/2021 11:52:00 PM
Submitted by	
Submitter email	lfrecalde1@espe.edu.ec
Similarity	9%
Analysis address	pnconstante.espe@analysis.arkund.com

Sources included in the report

SA	Universidad de las Fuerzas Armadas ESPE / Documento_tesis.pdf Document Documento_tesis.pdf (D54369965) Submitted by: djmendoza@espe.edu.ec Receiver: djmendoza.espe@analysis.arkund.com		30
SA	Universidad de las Fuerzas Armadas ESPE / TESIS BAEZ LARREA Capitulo12345.pdf Document TESIS BAEZ LARREA Capitulo12345.pdf (D78897346) Submitted by: phmejia@espe.edu.ec Receiver: phmejia.espe@analysis.arkund.com		1
W	URL: https://www.engineeringpassion.com/10-best-microcontroller-boards-for-engineers-an ... Fetched: 3/17/2021 11:53:00 PM		3
SA	L-T-0013.pdf Document L-T-0013.pdf (D20990677)		1
W	URL: https://doi.org/10.3390/app9194093 Fetched: 3/17/2021 11:53:00 PM		1
W	URL: http://www.mecamex.net/Libros/2017-Libro-InnovacionesEnMecatronica.pdf Fetched: 12/14/2020 7:23:46 PM		9
W	URL: https://doi.org/10.1155/2014/126317 Fetched: 3/17/2021 11:53:00 PM		2
W	URL: https://doi.org/10.3390/s19153371 Fetched: 3/17/2021 11:53:00 PM		2
SA	Universidad de las Fuerzas Armadas ESPE / BASANTES_BAUTISTA_PROYECTO_TITULACION.pdf Document BASANTES_BAUTISTA_PROYECTO_TITULACION.pdf (D36473223) Submitted by: vhandaluz1@espe.edu.ec Receiver: vhandaluz1.espe@analysis.arkund.com		3
SA	Universidad de las Fuerzas Armadas ESPE / TESIS MAESTRÍA_ANDALUZ GABRIELA_MODELACIÓN Y CONTROL PREDICTIVO DE UN ROBOT MÓVIL ... Document TESIS MAESTRÍA_ANDALUZ GABRIELA_MODELACIÓN Y CONTROL PREDICTIVO DE UN ROBOT MÓVIL ... (D31837897)		11

Firma:



Firma electrónica por:
**PATRICIA NATALY
CONSTANTE PROCCEL**

Ing. Constante Prócel, Patricia Nataly

DIRECTORA



DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Macas Llamuca, Johan Enrique** y **Recalde Simancas, Luis Fernando**, con cédulas de ciudadanía 0704421643 y 1717893166, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Diseño y construcción de un robot móvil tipo diferencial didáctico para aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE sede Latacunga.”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 19 de marzo del 2021

Macas Llamuca, Johan Enrique

C.C.:0704421643

Recalde Simancas, Luis Fernando

C.C.:1717893166



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

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Macas Llamuca, Johan Enrique y Recalde Simancas, Luis Fernando**, con cédulas de ciudadanía 0704421643 y 1717893166, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Diseño y construcción de un robot móvil tipo diferencial didáctico para aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE sede Latacunga.”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 19 de marzo del 2021

Macas Llamuca, Johan Enrique

C.C.:0704421643

Recalde Simancas, Luis Fernando

C.C.:1717893166

Dedicatoria

Este pequeño logro se lo dedico a mi padre "Carlos Recalde", de quien aprendí la disciplina, la importancia de prestarle atención a los detalles de la vida; mi madre "Sonia Simancas" gracias a ti logre entender que no importa lo complicada que la vida se torne, siempre tenemos que sonreír y seguir adelante.

Por supuesto a mis dos hermanos "Valeria Recalde" que me enseno que todo sacrificio al final tiene una recompensa y debemos luchar por nuestros sueños; finalmente pero no menos importante "Andrés Recalde", que no aprendí de ti, si fuiste tu quien me acerco al mundo de la música y entendí que la creatividad y la imaginación son partes muy importantes de la vida.

No quiero que se sientan orgullosos, quiero que se sientan felices, que sepan que soy muy Feliz haciendo lo que me gusta, y que gran parte de lo que soy ahora fue gracias a ustedes, gracias.

Luis F Recalde

Dedicatoria

Primeramente, le agradezco a mi padre, que, con su ayuda, su mano firme y sus consejos, me ha educado a lo largo de mi vida, dándome un ejemplo a seguir. Sin él esto no sería posible. A mi madre que con sus consejos y su forma de ver el mundo me ha ayudado a mantenerme firme y constante a lo largo de la carrera. A mi hijo por ser mi fuente de motivación, para superarme día a día y levantarme constantemente cada vez que caigo. Animándome con su personalidad alegre e inocente. A mi esposa por su sacrificio, por sus palabras de aliento llenas de amor, y por luchar cada día a mi lado sin flaquear.

Finalmente, a mi amigo de tesis con el que compartimos conocimientos para poder culminar este proyecto, el cual nos deja un recuerdo de cual sentirnos orgullosos.

Johan Macas

Agradecimiento

Agradecemos a nuestras familias sin su apoyo nada de esto se hubiese podido realizar, a cada compañero gracias por permitirnos aprender de ustedes y a los docentes que nos brindaron un poco de su vasto conocimiento.

A cada persona que conocimos y aprendimos de ustedes muchas gracias.

Luis F Recalde, Johan Macas

Tabla de contenidos

Carátula.....	1
Certificación.....	2
Reporte de similitud.....	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento.....	8
Tabla de contenidos.....	9
Índice de tablas.....	14
Índice de figuras.....	15
Resumen.....	19
Abstract.....	20
Aspectos generales.....	21
Planteamiento del problema.....	21
Antecedentes.....	23
Justificación e importancia.....	25
Objetivos del proyecto.....	27
<i>Objetivo general.....</i>	27
<i>Objetivos específicos.....</i>	27
Hipótesis.....	28
Variables de la investigación.....	28
Fundamentación teórica.....	29
Uso de tecnologías modernas para el aprendizaje.....	29

Robótica.....	30
Robótica móvil	31
Robot móvil tracción diferencial	32
Cinemática de robot móvil.....	33
Dinámica de robot móvil.....	34
Redes neuronales aplicadas control	36
Lógica difusa aplicada a control	37
Optimización aplicada a control.....	38
Diseño asistido por computador.....	39
Elementos de un robot móvil	40
<i>Motores dc</i>	40
<i>Microcontroladores</i>	41
<i>Microprocesadores</i>	43
<i>Sensor Lidar</i>	45
<i>Unidad de medición inercial</i>	46
Sistema comunicación	47
Introducción a ROS	47
<i>Conceptos ROS</i>	50
<i>El sistema de archivos de ROS</i>	50
<i>El gráfico de cálculo de ROS</i>	51
<i>El nivel de la comunidad de ROS</i>	53
Simuladores robóticos	54
<i>Webots</i>	54
Diseño y construcción del sistema mecánico, electrónico e interface	56
Diseño mecánico del robot móvil	56
Diseño y análisis de esfuerzos.....	59
<i>Diseño y análisis de esfuerzos en la base de carga</i>	60
<i>Diseño y análisis de esfuerzos en la base del robot</i>	63

Disposición final del robot Wall3	65
Ensamble del prototipo	65
Construcción de la estructura prototipo	66
Construcción de la base	66
Construcción de espacio de carga.....	66
Construcción de la caja de control.....	67
Construcción de la puerta delantera y posterior.....	67
Construcción de los brazos del robot.....	68
Ensamble del robot.....	69
Ensamble de la base del robot	69
Ensamble del espacio de carga y caja de control del robot..	70
Ensamble de las puertas y brazos del robot.....	70
Ensamble de la tapa superior del robot y el soporte Lidar...	71
Ensamble Final.....	72
Electrónica del robot móvil	72
Selección de componentes electrónicos.....	72
Selección del microprocesador.....	73
Selección de microcontrolador	74
Selección del sensor Lidar.....	75
Selección de la unidad de medida inercial	76
Selección de Servo Motores	77
Desarrollo de interfaz del robot móvil	78
Menú conexión	78
Menú control cinemático	80
Ejecución.....	80
Resultados	82
Información	82
Control manual.....	83
Ejecución.....	83
Simulador Webots.....	85

Estructura del sistema	87
Modelación matemática y algoritmos de control	89
Modelación cinemática	89
<i>Modelo generalizado plataformas móviles</i>	<i>89</i>
<i>Modelo cinemático plataforma móvil tipo diferencial</i>	<i>89</i>
Modelación dinámica	91
<i>Modelo dinámico plataforma móvil tipo diferencial</i>	<i>91</i>
Identificación de parámetros del modelo dinámico.....	98
<i>Identificación y validación experimental del dinámico</i>	<i>101</i>
Algoritmos de control para seguimiento de trayectoria.....	102
<i>Controlador proporcional basado en modelo cinemático. ..</i>	<i>102</i>
<i>Análisis de estabilidad</i>	<i>103</i>
Controlador fuzzy adaptativo	105
<i>Diseño de un controlador difuso</i>	<i>105</i>
<i>Controlador PD autoajustable con lógica difusa.....</i>	<i>107</i>
<i>Reglas difusas para autoajuste de Kp</i>	<i>111</i>
<i>Reglas difusas para autoajuste de Kd</i>	<i>112</i>
Controlador neuronal adaptativo	113
Aproximación de funciones redes neuronales artificiales.....	117
Controlador P autoajustable con en redes neuronales artificiales	120
Controlador MPC.....	123
Compensación dinámica	126
Pruebas y resultados	129
Pruebas experimentales	129
Control proporcional basado en modelo cinemático	130
Controlador PD autoajustable basado en lógica difusa	133
Controlador P autoajustable con redes neuronales artificiales.....	136

Controlador predictivo basado en el modelo MPC	140
Comparación de controladores.....	142
Descripción de las personas de prueba	143
Prueba de usabilidad	143
Cuestionario de usabilidad.....	144
Encuesta de satisfacción.....	146
Hipótesis.....	149
<i>Obtención de la muestra</i>	<i>149</i>
<i>Validación de la hipótesis</i>	<i>150</i>
Costos.....	155
Conclusiones y recomendaciones	157
Conclusiones.....	157
Recomendaciones.....	158
Bibliografía.....	160
Anexos.....	167

Índice de tablas

Tabla 1 <i>Esfuerzo (MPa) de acuerdo al porcentaje de relleno</i>	57
Tabla 2 <i>Propiedades Mecánicas del PLA</i>	58
Tbla 3 <i>Pesos de componentes del robot</i>	59
Tabla 4 <i>Selección de componentes</i>	73
Tabla 5 <i>Comparación SBC</i>	74
Tabla 6 <i>Análisis para selección de microcontrolador</i>	75
Tabla 7 <i>Análisis para selección Lidar</i>	76
Tabla 8 <i>Análisis para selección de unidad de medición inercial</i>	77
Tabla 9 <i>Selección de servomotores</i>	78
Tabla 10 <i>Reglas difusas para ajuste de K_p</i>	112
Tabla 11 <i>Reglas difusas para ajuste de K_d</i>	113
Tabla 12 <i>Promedio de los datos obtenidos del cuestionario de usabilidad</i>	144
Tabla 13 <i>Promedio de los datos obtenidos de la encuesta de satisfacción</i>	147
Tabla 14 <i>Notas de las pruebas realizadas</i>	150
Tabla 15 <i>Parámetros de Test Shapiro-Wilk</i>	152
Tabla 16 <i>Valores de análisis T-Student</i>	153
Tabla 17 <i>Parámetros para el análisis T-Student</i>	154
Tabla 18 <i>Costos del Proyecto</i>	155

Índice de figuras

Figura 1 Robot androide "iCub".....	30
Figura 2 Robot Móvil "e-puck".....	31
Figura 3 Esquema robot móvil diferencial.....	33
Figura 4 Sistema de coordenadas robot móvil diferencial.....	34
Figura 5 Dinámica directa e inversa.....	35
Figura 6 Esquema de una red neuronal de 4 capas.....	37
Figura 7 Diagrama del controlador difuso de 2 entradas y 1 salida.....	38
Figura 8 Interfaz de Blender.....	40
Figura 9 Motor dc.....	41
Figura 10 Arduino R3.....	42
Figura 11 Teensy 4.0.....	43
Figura 12 Raspberry pi 4.....	44
Figura 13 Jetson AGX Xavier.....	45
Figura 14 Mapa generado por Lidar.....	46
Figura 15 Unidad de Medición Inercial.....	47
Figura 16 Ecuación ROS.....	50
Figura 17 Comunicación entre nodos y master ROS.....	53
Figura 18 Entorno Webots.....	55
Figura 19 Diseño de la base de carga del robot.....	56
Figura 20 Diseño final.....	57
Figura 21 Pieza Base_2.....	60
Figura 22 Representación de la carga distribuida.....	61
Figura 23 Análisis por software de la pieza Base_2.....	62
Figura 24 Representación de la carga distribuida en la superficie.....	63

Figura 25 <i>Análisis por software de la pieza Base_1</i>	64
Figura 26 <i>Ensamble final del robot Wall3</i>	65
Figura 27 <i>Base del robot Wall3</i>	66
Figura 28 <i>Espacio de carga del robot Wall3</i>	67
Figura 29 <i>Caja de control del robot Wall3</i>	67
Figura 30 <i>Puertas del robot Wall3</i>	68
Figura 31 <i>Brazos del robot Wall3</i>	69
Figura 32 <i>Ensamble de la Base del robot Wall3</i>	69
Figura 33 <i>Ensamble del Espacio de carga del robot Wall3</i>	70
Figura 34 <i>Ensamble del robot</i>	70
Figura 35 <i>Ensamble del robot</i>	71
Figura 36 <i>Ensamble final del robot Wall3</i>	72
Figura 37 <i>Menú conexión</i>	79
Figura 38 <i>Menú conexión botones</i>	80
Figura 39 <i>Menú cinemático tiempos</i>	81
Figura 40 <i>Menú cinemático ganancias</i>	81
Figura 41 <i>Menú cinemático inicio</i>	82
Figura 42 <i>Menú cinemático resultados</i>	83
Figura 43 <i>Menú cinemático información</i>	83
Figura 44 <i>Menú manual tiempos</i>	84
Figura 45 <i>Menú manual potenciómetros</i>	85
Figura 46 <i>Entorno simulado en Webots</i>	86
Figura 47 <i>Robot Wall3 en el entorno simulado</i>	86
Figura 48 <i>Entorno 2 para simulación</i>	87
Figura 49 <i>Estructura funcionamiento del sistema</i>	88

Figura 50 <i>Parámetros modelo cinemático</i>	90
Figura 51 <i>Señales de identificación y modelo</i>	101
Figura 52 <i>Validación del modelo dinámico</i>	102
Figura 53 <i>Esquema control cinemático</i>	103
Figura 54 <i>Fuzzificación de variables de entrada</i>	106
Figura 55 <i>Esquema control PD autoajustable</i>	108
Figura 56 <i>Estructura del control difuso</i>	109
Figura 57 <i>Funciones membresía entrada</i>	110
Figura 58 <i>Funciones membresía salida</i>	110
Figura 59 <i>Superficies de control</i>	111
Figura 60 <i>Representación neurona</i>	114
Figura 61 <i>Red neuronal capa simple</i>	115
Figura 62 <i>Red neuronal multicapa</i>	116
Figura 63 <i>Estructura de una red neuronal para aproximación</i>	117
Figura 64 <i>Esquema algoritmo de retro propagación</i>	117
Figura 65 <i>Esquema control red neuronal</i>	121
Figura 66 <i>Esquema control MPC</i>	125
Figura 67 <i>Robot móvil Wall3</i>	129
Figura 68 <i>Trayectoria control proporcional basado en el modelo</i>	131
Figura 69 <i>Errores de posición controlador proporcional</i>	131
Figura 70 <i>Velocidades reales y de control en el proporcional</i>	132
Figura 71 <i>Trayectoria control PD autoajustable basado en lógica difusa</i>	134
Figura 72 <i>Errores de posición del PD autoajustable lógica difusa</i>	135
Figura 73 <i>Velocidades control PD basado en lógica difusa</i>	135
Figura 74 <i>Ganancias PD autoajustable basado en lógica difusa</i>	136

Figura 75 <i>Trayectoria control P autoajustable basado en redes neuronales...</i>	137
Figura 76 <i>Estimación de estados internos</i>	138
Figura 77 <i>Ganancias control P autoajustable basado en redes neuronales....</i>	139
Figura 78 <i>Errores de posición P autoajustable basado en redes neuronales..</i>	139
Figura 79 <i>Velocidades reales P basado en redes neuronales artificiales.....</i>	140
Figura 80 <i>Trayectoria control MPC</i>	141
Figura 81 <i>Velocidades de control y reales del controlador MPC</i>	142
Figura 82 <i>Errores de posición del controlador MPC.....</i>	142
Figura 83 <i>Comparativa controladores.....</i>	143
Figura 84 <i>Promedio de los resultados del cuestionario de usabilidad</i>	145
Figura 85 <i>Promedio de los resultados de la encuesta de satisfacción</i>	148
Figura 86 <i>Análisis T de Student.....</i>	155

Resumen

Este trabajo se basa en el diseño y construcción de un robot móvil tipo diferencial, el cual será usado como módulo de pruebas de algoritmos de control tradicionales e inteligentes. El robot móvil fue diseñado para que pueda transportar hasta 1 kg y se utilizó Ros como sistema de comunicación entre el robot y la el computador encargado de ejecutar los algoritmos. Los algoritmos de control tradicionales que fueron implementados son: (1) Control proporcional basado en el modelo, (2) Control predictivo basado en el modelo y los controladores inteligentes son: (1) Controlador Proporcional Derivativo adaptativo basado en lógica difusa y (2) Controlador P adaptativo basado en redes neuronales artificiales. El desempeño de los controladores fue validado en el robot diseñado en este trabajo Wall3, demostrando el correcto funcionamiento de los mismos, demostrando la estabilidad de los controladores y la robustez del sistema. Además, se generó un simulador usando Webots donde de igual forma de valido los controladores obteniendo resultados similares al robot real. Finalmente, para un mejor uso del sistema se desarrolló una interfaz de usuario donde, el operario puede seleccionar de forma sencilla entre los distintos tipos de controladores, además ir verificando su funcionamiento ante diferentes configuraciones en los controladores.

Palabras clave:

- **MÓVIL DIFERENCIAL**
- **CONTROLADORES TRADICIONALES**
- **CONTROLADORES INTELIGENTES**

Abstract

This work is based on the design and construction of a differential type mobile robot, which will be used as a test module for traditional and intelligent control algorithms. The mobile robot was designed so that it can carry up to 1 kg and Ros was used as a communication system between the robot and the computer in charge of executing the algorithms. The traditional control algorithms that were implemented are: (1) Proportional control based on the model, (2) Predictive control based on the model and the intelligent controllers are: (1) Adaptive Proportional Derivative Controller based on fuzzy logic and (2) Adaptive P controller based on artificial neural networks. The performance of the controllers was validated in the robot designed in this work, Wall3, demonstrating their correctness, demonstrating the stability of the controllers and the robustness of the system. In addition, a simulator was generated using Webots where the controllers were equally valid, obtaining results similar to the real robot. Finally, for a better use of the system, a user interface was developed where the operator can select between the different types of controllers, in addition to verifying their operation before different configurations in the controllers.

Key words:

- **DIFFERENTIAL MOBILE**
- **TRADITIONAL CONTROL ALGORITHMS**
- **INTELLIGENT CONTROL ALGORITHMS**

Capítulo I

1. Aspectos generales

1.1 Planteamiento del problema

En los últimos años ha surgido un gran interés por robots móviles terrestres, los robots móviles tienen un gran potencial en aplicaciones de manufactura, reactores nucleares, construcción y exploración planetaria (Li & Ge, 2013). Entre las posibles áreas de aplicación está el área médica donde se realiza operaciones de soporte, ayuda y análisis en laboratorios, aplicaciones en agricultura como puede ser la automatización en la recolección de frutos, la plantación de semillas y limpieza de los mismos cultivos, en el área de inspección y vigilancia de zonas peligrosas como pueden ser la detección y desactivación de minas en campos minados, revisión de reactores nucleares y limpieza de alcantarillados, en misiones espaciales como puede ser la inspección y mantenimiento de satélites, de igual modo en exploraciones planetarias, en el área militar cuenta con robots de reconocimiento y finalmente en el área de asistencia médica a adultos mayores como son sillas autónomas (Klančar, Zdešar, Blažič, & Škrjanc, 2017).

Las plantas industriales modernas y los productos tecnológicos a menudo se requieren para realizar tareas complejas con alta precisión, en condiciones no muy bien definidas. Las técnicas de control tradicionales pueden no ser bastante efectivas en estos sistemas, mientras que las técnicas de control inteligentes tienen un tremendo potencial. El énfasis del módulo didáctico es

práctico para aplicaciones de técnicas de control inteligente y tradicionales (J. Liu, 2017).

En la Universidad De Las Fuerzas Armadas ESPE Sede Latacunga, el laboratorio de ingeniería mecatrónica, debido a la alta demanda existente en la carrera requiere un mayor número de estaciones para realizar prácticas en las asignaturas, debido al alto costo que tienen estos dispositivos se opta por el desarrollo de un módulo didáctico robótico en el cual los estudiantes puedan implementar distintos algoritmos de control en un mismo sistema, afectando directamente a la comprensión de los algoritmos impartidos en clase. Los robots móviles son sistemas donde se pueden aplicar distintas técnicas de control, son un foco importante de investigaciones modernas y es muy importante que cada universidad tenga uno o más módulos que se centren en la investigación de robots móviles.

Este proyecto busca adentrarse en la problemática social permitiendo que el uso de robots móviles sirva para satisfacer las necesidades de la sociedad para aplicaciones que necesiten de manera simultánea tanto la capacidad de navegación y transporte de objetos. Gracias a la modelación cinemática del robot móvil se puede implementar distintas técnicas de control para llevar a cabo tareas de transporte de objetos.

Por todo lo mencionado en párrafos anteriores se puede concluir que solucionar la problemática planteada se desarrollará mediante la construcción de

un robot móvil capaz de realizar tareas de transporte en ambientes estructurados, dicho robot sirva también para la evaluación de algoritmos de control basados en la cinemática del robot móvil, de igual modo este módulo didáctico robótico beneficiará a las asignaturas de la formación profesional de la carrera tales como, Aplicaciones de técnicas de control, Control Digital, Robótica Industrial, Microprocesadores y Microcontroladores e Instrumentación aplicada a la mecatrónica.

1.2 Antecedentes

La robótica hace muchos años tiene mucha participación en el sector industrial moderno, debido a que en esta área se realizan tareas repetitivas para las cuales los robots presentan un buen desempeño ya sea en ambientes estructurados o no estructurados. En la actualidad existen robots con inteligencia artificial con capacidad de tomar las mejores decisiones posibles de acuerdo a los cambios que se presenten en él entorno (G. M. Andaluz et al., 2016a)

La robótica puede ser clasificada en robots industriales y de servicio, los robots pueden ser usados en tareas, en donde idea principal es realizar actividades que faciliten la vida del usuario, de este modo donde se presentaron mayores avances es en el área de la medicina donde se desarrollaron robots que van desde poder ayudar a personas en tareas del hogar hasta robots capaces de realizar operaciones quirúrgicas.

La plataforma móvil puede moverse y alcanzar lugares que puedan llegar a ser hostiles para los seres humanos, realizando operaciones en un amplio espacio de trabajo. Se puede encontrar este tipo de robot en un amplio rango de aplicaciones como son: robots de seguridad y servicio aplicados a fábricas, oficinas y hogares, búsqueda y rescate en ambientes peligrosos y operaciones y exploraciones en lugares remotos (Bayle, Fourquet, & Renaud, 2003).

Dichos sistemas incluyen una gran dificultad y complejidad en el área modelado Cinemático y Dinámico los cuales son usados al momento de diseñar sistemas de control. Una gran cantidad de avances en el análisis de sistemas no lineales y diseño de sistema de control ofrecen herramientas poderosas y conceptos fundamentales para el control donde el control inteligente, el control adaptativo y el control robusto toman lugar (Li & Ge, 2013).

Dentro de los controladores inteligentes se encuentra a las redes neuronales y los sistemas difusos los cuales pueden llegar a controlar cualquier función no lineal en un conjunto compacto donde el error de modelado tiende a cero. A diferencia del control tradicional, el control inteligente posee la capacidad de que el error del sistema tienda a cero incluso con información incompleta sobre el comportamiento del sistema. Existen muchas técnicas para poder implementar un controlador inteligente como son: controladores expertos, controladores difusos, controladores basados en redes neuronales y algoritmos inteligentes de búsqueda (J. Liu, 2017).

Un controlador experto es una técnica de control que usa el conocimiento y experiencia de un experto en el sistema, los controladores basados en lógica difusa utilizan la lógica difusa para representar el conocimiento de las personas en un área específica con las cuales realizan las inferencias o acciones respectivas sobre el sistema, las redes neuronales pueden ser entrenadas para representar un sistema no lineal altamente complejo, los cuales son un gran problema al momento de implementar estrategias de control, las configuraciones de redes neuronales BP y RBF pueden aproximar casi cualquier función no lineal, las cuales pueden ser usadas para el diseño de un controlador neuronal adaptativo (Angelov & Xydeas, 2006; Åström, Anton, & Årzén, 1986; Hornik, Stinchcombe, & White, 1989)

1.3 Justificación e importancia.

El conocimiento es adquirido de mejor manera cuando se experimenta directamente, no es suficiente solo ver o escuchar, por lo tanto, en el área de la educación tener un lugar donde probar los conocimientos adquiridos es muy importante, por esta razón este proyecto propone desarrollar un robot móvil tipo diferencial didáctico el cual sirva de puente entre la teoría y la práctica, en donde se puedan probar distintos algoritmos de control avanzado para la navegación autónoma y determinar si el comportamiento es el adecuado.

Al realizar el proyecto se pretende llevar a la práctica los conocimientos teóricos, en robótica se aborda el tema de manipuladores, pero en la actualidad también se consideran importante los robots móviles debido a que son un foco de investigaciones y es importante que cada universidad tenga uno o más

laboratorios que se centren en investigaciones de este tipo, por lo que disponer de un robot móvil para poder visualizar el comportamiento los algoritmos de control en la vida real es un gran aporte para la enseñanza de aplicación de técnicas de control, control digital y robótica industrial.

Las últimas investigaciones han presentado notorios avances en el desarrollo de robots móviles para tareas de servicio, actuación sobre objetos del entorno, como puede ser el transporte de objetos, entrenamiento de robots, tareas domésticas entre otras. El robot móvil de esta investigación está dirigido a tareas de servicio para que pueda ser instalado en un entorno estructurado como el Laboratorio de Ingeniería Mecatrónica.

Poco sirve modelar un robot si este no es capaz de hacer nada, la navegación autónoma desempeña un papel importante en el proyecto propuesto, ya que para llevar a cabo tareas que sean útiles para la sociedad el robot debe estar dotado de diferentes habilidades, se debe asegurar de que éste posee las habilidades de navegar por el área de trabajo sin ningún problema y que pueda transportar objetos si se lo requiere. Las fábricas actuales exigen movilidad en donde los robots móviles toman acción, dichos robots se mueven de forma autónoma sobre el espacio disponible, pueden llegar a actuar en grupos y ofrecen la flexibilidad que la industria necesita, lo cual es muy relevante en el ámbito de la logística interna.

1.4 Objetivos del proyecto

1.4.1 *Objetivo general*

Diseñar y construir un robot móvil tipo diferencial didáctico para aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el Laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.

1.4.2 *Objetivos específicos*

- Investigar el estado actual de la tecnología basado en la recolección de información a través de publicaciones e información técnica de los robots móviles para obtener parámetros de diseño.
- Diseñar la estructura, dimensionando y seleccionando materiales de forma adecuada el cual permita trasladar un kilogramo de peso.
- Modelar la cinemática robot móvil diferencial considerando el centro de masa de la plataforma como punto de interés para la obtención de la representación matemática del sistema.
- Desarrollar los esquemas de control basados en algoritmos tradicionales e inteligentes para tareas de navegación autónoma y transporte de un robot móvil en espacios de trabajos parcialmente estructurados.
- Validar la hipótesis a través de pruebas experimentales virtuales y reales con el objetivo de comprobar el desempeño de los algoritmos de control propuestos.

1.5 Hipótesis

¿El diseño y construcción de un robot móvil tipo diferencial didáctico permitirá la aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el Laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Sede Latacunga?

1.6 Variables de la investigación.

Variables independientes

Robot móvil tipo diferencial didáctico.

Variables dependientes

Aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes.

Capítulo II

2. Fundamentación teórica

2.1 Uso de tecnologías modernas para el aprendizaje

En la actualidad la educación está sufriendo un cambio abrupto obligando a actualizarse con nuevos métodos y tecnologías, es por este motivo que se están enfocando en el uso de las “tecnologías de la información y la comunicación” (TICs), teniendo como objetivo desarrollar competencias y aptitudes en los estudiantes para que sean competitivos en este mundo cambiante. El papel que desarrollan las TICs es de suma importancia porque ofrece servicios como el correo electrónico, plataformas virtuales, aulas virtuales, comercio electrónico, búsqueda de información, descarga de archivos, videoconferencias, entre otros. (“¿Qué son las TIC?,” n.d.)

El aprendizaje a través de estos servicios como la educación online permite tener una educación de calidad sin necesidad de ir a una institución física, trayendo una amplia gama de beneficios, entre ellos y el más importante la bioseguridad.

El uso de las TICs está permitiendo evolucionar e innovar en estrategias de estudio y métodos de enseñanza permitiendo actualizar un sistema de educación monótono que lleva la educación a través de ya muchos años.

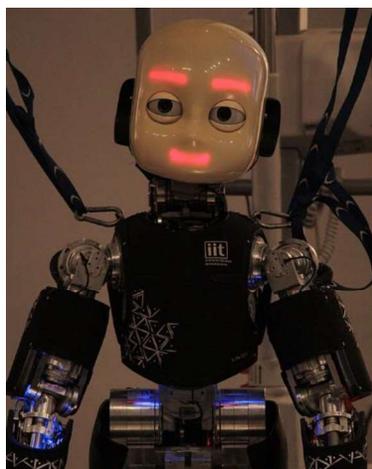
2.2 Robótica

La Robótica es un campo de la tecnología, que une conocimientos matemáticos y prácticos que permiten automatizar y realizar sistemas basados en estructuras mecánicas articuladas, combinando diversos conocimientos como la electrónica, física, inteligencia artificial, mecánica e ingeniería de control.

Un sistema robótico es el que puede recibir información, comprender su entorno y controlar o supervisar una operación (Beira et al., 2006). Se pueden clasificar según su cronología, partiendo de la primera generación, segunda generación hasta la tercera generación; o por su estructura, pudiendo ser poliarticulados, móviles, andróides, zoomórficos o híbridos (“Industry Spotlight: Robotics from Monster Career Advice,” n.d.).

Figura 1

Robot androide “iCub”



Nota. Tomado de “Design of the robot-cub (iCub) head” Beira, R., Lopes, M., Praça, M., Santos-Victor, J., Bernardino, A., Metta, G., ... Saltarén, R, 2006,

IEEE International Conference on Robotics and Automation.

2.3 Robótica móvil

Este tipo de robot se caracteriza por poder desplazarse fácilmente en su entorno, su diseño es basado frecuentemente en plataformas o automóviles, su nivel de inteligencia es relativamente alto y además disponen de un sistema de tracción rodante. Siguen su trayectoria mediante sus sensores o por telemando (“Moviles - ROBOTICA,” n.d.).

En la actualidad son sumamente importantes para las investigaciones, por esto las universidades priorizan el desarrollo de este tipo de robots, teniendo a su disposición uno o más laboratorios específicos (W. Liu & Winfield, 2011). También se encuentran en el ámbito industrial y de servicios como los domésticos o de entretenimiento (“Robot móvil - Wikipedia, la enciclopedia libre,” n.d.)

Figura 2

Robot Móvil “e-puck”



Nota. Tomado de “Open-hardware e-puck Linux extension board for

experimental swarm robotics research” por Liu, W., & Winfield, A. F. T, 2011, Microprocessors and Microsystems

2.4 Robot móvil tracción diferencial

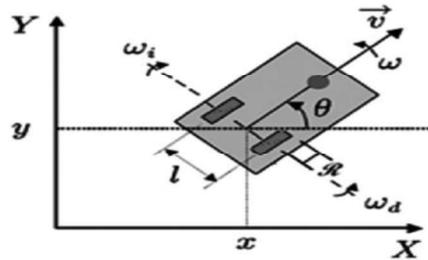
El movimiento de este robot móvil se basa en la tracción diferencial, la misma que es una configuración frecuentemente utilizada en lugares interiores porque permite girar en su propio eje al robot, pudiendo moverse con facilidad en su entorno.

Este sistema de tracción utiliza dos ruedas que son individualmente controladas, con una rueda loca siendo el tercer punto de apoyo del robot, generalmente los motores que proveen el movimiento a las ruedas son de corriente continua.

Los parámetros como la posición, la velocidad o el ángulo, pueden ser obtenidos mediante encoders, teniendo en cuenta la geometría del robot (Solaque Guzmán, Molina Villa, & Rodríguez Vásquez, 2014).

Figura 3

Esquema robot móvil diferencial



Nota. Tomado de “Seguimiento de trayectorias con un robot móvil de configuración diferencial” por Solaque Guzmán, L. E., Molina Villa, M. A., & Rodríguez Vásquez, E. L, 2014, Ingenierías USBMed

2.5 Cinemática de robot móvil

El modelo cinemático estudia el movimiento de sistemas mecánicos, en el cual se analiza la velocidad, la posición y la aceleración del robot sin tomar en consideración las fuerzas que interfieren en el movimiento. En algunos casos es conveniente emplear la cinemática directa ya que es una técnica que hace uso de gráficos en 3D por computadora, facilitando los cálculos (Craig, 2006).

En el caso de un robot de tracción diferencial consiste en representar la velocidad del robot móvil de los ejes de las ruedas, tomando en cuenta los parámetros geométricos del mismo.

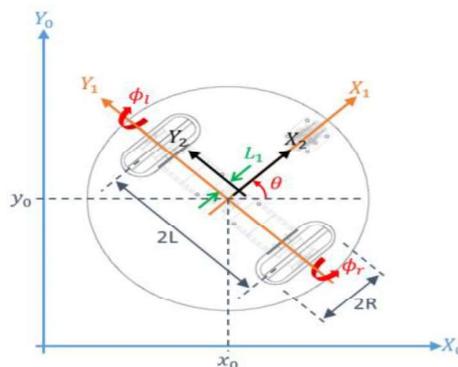
- Se debe tener en cuenta los parámetros del entorno.
- Si la superficie es plana, la energía potencial es constante.
- Debido a su geometría los ejes del robot móvil son perpendiculares a la superficie.

- El contacto entre la rueda y la superficie existe solamente en un solo punto de esta forma se facilita el análisis.

Teniendo en cuenta todas estas consideraciones se puede realizar el análisis bidimensional con menor complejidad, colocando un sistema de referencia convenientemente en el plano y otro sistema de referencia en el robot móvil, se obtiene las componentes de orientación y traslación (Solaque Guzmán et al., 2014).

Figura 4

Sistema de coordenadas robot móvil diferencial



Nota. Tomado de “Seguimiento de trayectorias con un robot móvil de configuración diferencial” por Solaque Guzmán, L. E., Molina Villa, M. A., & Rodríguez Vásquez, E. L., 2014, Ingenierías USBMed

2.6 Dinámica de robot móvil

Después de analizada la cinemática se inicia el desarrollo del modelo dinámico, el cual se basa en las leyes de la mecánica enfocándose en 3 componentes: inercia, elasticidad y fricción, los cuales están presentes en

cualquier sistema mecánico real como puede ser un robot. La dinámica de un robot móvil es un campo de estudio notoriamente complejo por lo que tiene gran interés por parte de los investigadores e ingenieros (Martins, Celeste, Carelli, Sarcinelli-Filho, & Bastos-Filho, 2008; Velagic, Lacevic, & Osmic, 2008).

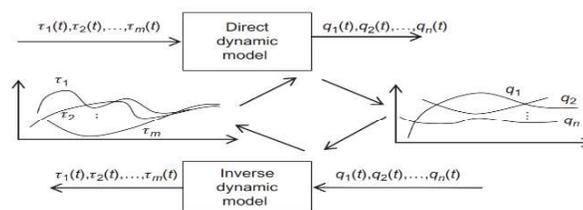
Con la modelación dinámica se obtienen las ecuaciones del movimiento considerando las fuerzas que lo producen, esto se puede realizar usando dos metodologías (Tzafestas, 2014).

- Método de Newton Euler
- Método de Lagrange

Al igual que en la cinemática, en la dinámica se puede obtener el modelo directo e inverso se puede visualizar de mejor forma en Figura 5.

Figura 5

Dinámica directa e inversa



Nota. Tomado de "Mobile Robot Dynamics" por Tzafestas, S. G, 2014, Introduction to Mobile Robot Control

La dinámica directa proporciona las ecuaciones que describen la respuesta del robot bajo ciertas fuerzas o torques los cuales suelen tener lugar en los motores. La dinámica inversa proporciona los torques o fuerzas

necesarios para cierta trayectoria o estados del robot, esta representación es muy usada al momento de diseñar controladores (Tzafestas, 2014).

2.7 Redes neuronales aplicadas control

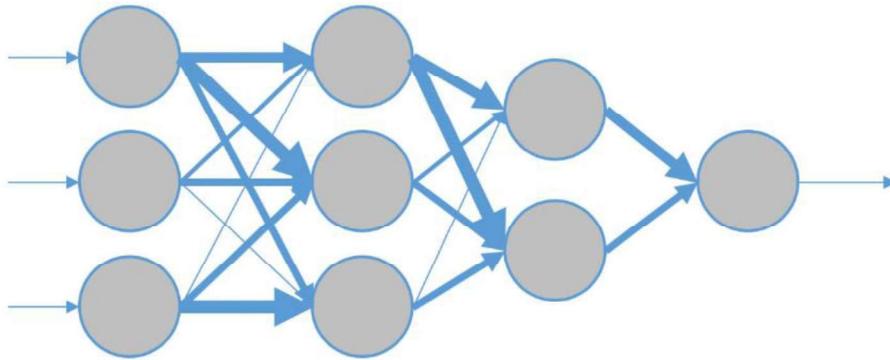
Las redes neuronales artificiales están inspiradas al comportamiento de las neuronas biológicas, siendo estas un modelo computacional constituido de un conjunto de neuronas artificiales, donde las neuronas están conectadas a otras mediante uniones permitiendo la transmisión de señales.

El proceso que se realiza consiste en que la información entra en la red neuronal artificial se observa en Figura 6 dando como resultado unos ciertos valores de salida. El valor de salida de cada enlace se multiplica por un valor denominado “peso”. Los pesos pueden aumentar o aislar el estado de activación de las neuronas vecinas. Este principio también es utilizado para los valores de salida de cada neurona, ya que puede existir una función de umbral o limitadora modificando el valor del resultado, este proceso es conocido como función de activación (van Gerven & Bohte, 2017).

Este tipo de sistemas se distinguen de otros porque no son programados, puesto que tienen un aprendizaje automático, sobresaliendo en la detección de soluciones en donde la programación convencional puede hacerlo por la complejidad que conlleva. El objetivo es poder resolver problemas de igual forma que lo resolvería el cerebro humano (van Gerven & Bohte, 2017).

Figura 6

Esquema de una red neuronal de 4 capas



Nota. Tomado de “Artificial neural networks as models of neural information processing” por van Gerven, M., & Bohte, S, 2017, *Frontiers in Computational Neuroscience*

2.8 Lógica difusa aplicada a control

El control difuso hace uso de la lógica difusa, la cual puede ser explicada como un control mediante el uso lingüístico que interpretan sentencias se usan en vez de ecuaciones, no obstante, las variables de entradas se miden en números su estructura se puede observar en Figura 7.

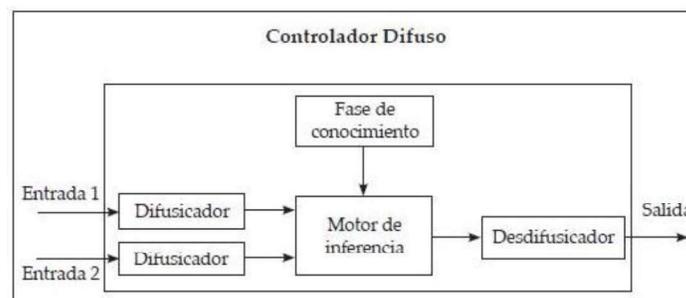
Entonces para que exista el control difuso se requiere una etapa de fusificación, en esta etapa se le asigna un grado de pertenencia a los datos de entrada dentro de los conjuntos borrosos definidos, luego se establecen las relaciones lógicas a través de reglas como “if-then”.

Luego se realizan las relaciones mediante la conjunción que permiten generar una acción deseada de salida. Finalmente llega a la etapa de

defusificación, la cual convierte la acción lingüística de salida en un número, para que posteriormente un conversor digital- análogo la convierta en una señal y pueda ser aplicada al sistema (Calderon, n.d.).

Figura 7

Diagrama del controlador difuso de 2 entradas y 1 salida



Nota. Tomado de “Control Mediante Lógica Difusa” por Calderon, J. E. J.

2.9 Optimización aplicada a control

El control óptimo de sistemas no lineales es una de las áreas más activas en la teoría de control, una de las principales dificultades del control optimo clásico es determinar controladores óptimos para sistemas no lineales en donde el Hamiltoniano del sistema debe ser resuelto, el cual muy rara vez se resuelve de forma analítica, las soluciones que se encuentran son a través de métodos numéricos (Polak, 1997a, 1997b; Tzafestas, 2014)

Una alternativa a este problema es el control predictivo basado en el modelo del sistema, el cual es uno de los métodos con mejores resultados en ámbitos industriales , aunque en la actualidad se dispone de algunas técnicas de control óptimo para sistemas lineales y no lineales por ejemplo en el área de la

inteligencia artificial como redes neuronales artificiales, sistemas de lógica difusa, algoritmos genéticos, y la optimización por enjambre de partículas, todas estas tienen como base la optimización sólo que aplicada desde distintos puntos (Grass, Caulkins, Feichtinger, Tragler, & Behrens, 2008; Prasad, Tyagi, & Gupta, 2014).

En la actualidad el uso de optimización en control ha aumentado debido que es posible la implementación de estas técnicas en algoritmos de tiempo real, lo que le permite al sistema un mejor desempeño en las no linealidades. Lo cual modifica el alcance de estimación y control permitiendo una mayor flexibilidad y precisión en los sistemas.

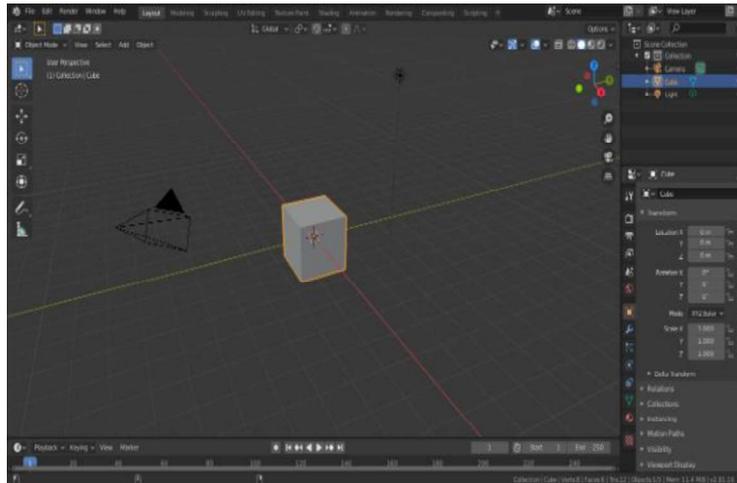
2.10 Diseño asistido por computador

El diseño asistido por computador (CAD) hace referencia a la utilización de computadores para asistir a la creación, análisis, optimización o modificación de un diseño. Son softwares basados en la geometría vectorial, pudiendo realizar superficies y sólidos, en Figura 8 se puede observar unos de los programas de diseño Open Source más populares (Kröger, Dekiff, & Dirksen, 2017).

El desarrollo de estos programas permite incrementar la productividad del diseñador, verificar el diseño antes de pasar a la etapa de la elaboración, lo cual se traduce en ventajas económicas como productivas, trayendo beneficios al diseñador y a la industria (“Diseño asistido por computadora - EcuRed,” n.d.).

Figura 8

Interfaz de Blender



Nota. Tomado de “3D printed simulation models based on real patient situations for hands-on practice” por Kröger, E., Dekiff, M., & Dirksen, D, 2017, European Journal of Dental Education

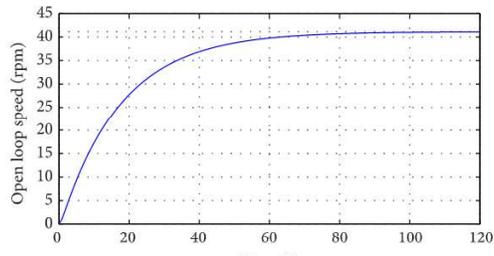
2.11 Elementos de un robot móvil

2.11.1 Motores dc

Un motor de corriente continua (DC) es un dispositivo que convierte la energía eléctrica se basa en el principio de la mano izquierda de Fleming que menciona que cuando un conductor se coloca en un campo magnético que lleva corriente en él se genera un par el cual es capaz de mover al motor (“Difference Between Servo Motor and DC Motor - Circuit Globe,” n.d.), en la Figura 9 se muestra el comportamiento de un motor en bucle abierto (Sabir & Khan, 2014).

Figura 9

Motor dc



Nota. Tomado de “Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques” por Sabir, M. M., & Khan, J. A, 2014 *Advances in Artificial Neural Systems*

En una aplicación industrial en la cual se necesite regular la posición angular una de la principal alternativa es un motor DC, lo que se debe a que es un dispositivo de bajo costo de mantener, tiene un precio bajo en el mercado, entre las principales industrias que usan estos dispositivos se tiene: la industria textil, la industria del papel entre otras más (Sabir & Khan, 2014).

2.11.2 Microcontroladores

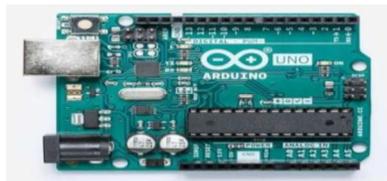
Un microcontrolador es un circuito integrado, el cual fue diseñado para gobernar una operación específica en un sistema. Un microcontrolador es un computador autónomo en un chip el cual incluye un procesador, memoria y periféricos de entrada / salida (E / S) en un solo chip. Por ejemplo, la bomba de gasolina necesita algo de potencia de procesamiento local para procesar las entradas del teclado del usuario para seleccionar la gasolina adecuada, active

las bombas, calcule el total costo de gasolina, es en estos casos donde son muy útiles los microcontroladores (Barrett & Pack, 2005).

A continuación, se presenta una lista de varias marcas de microcontroladores con algunas de sus especificaciones. Arduino uno R3 este microcontrolador está basado en un chip ATmega328p, tiene a disposición 14 pines, incluye conexión USB por lo que puede conectarse fácilmente a un computador personal, en la Figura 10 se puede observar este microcontrolador (“10 Best Microcontroller Boards for Engineers and Geeks - Engineering Passion,” n.d.).

Figura 10

Arduino R3



Nota. Tomado de “10 Best Microcontroller Boards for Engineers and Geeks”,2011 <https://www.engineeringpassion.com/10-best-microcontroller-boards-for-engineers-and-geeks/>

Teensy 4.0 como se observa en Figura 11 es una de las placas más recientes en el mercado, presenta una velocidad de procesamiento de 600 MHz y es una de las más rápidas en la actualidad, dispone de un cable USB con el que puede conectarse a un computador personal y ser programado a través de

Ide de Arduino (“10 Best Microcontroller Boards for Engineers and Geeks - Engineering Passion,” n.d.).

Figura 11

Teensy 4.0



Nota. Tomado de “10 Best Microcontroller Boards for Engineers and Geeks”,2011 <https://www.engineeringpassion.com/10-best-microcontroller-boards-for-engineers-and-geeks/>

2.11.3 Microprocesadores

Es la unidad central de un sistema informático que realiza operaciones aritméticas y lógicas, las cuales en su mayoría incluyen operaciones de sumar, restar, transferir números de un área de trabajo a otra ser comparados. A menudo se lo conoce como procesador o unidad central de procesamiento, es esencialmente el cerebro de la computadora

Una computadora de placa unida o como sus siglas en inglés “Single Board Computer”, es una computadora completa construida en una sola placa o circuito en la que se dispone de: microprocesador, memoria, entradas y salidas, y otras características propias de un computador. Estos sistemas suelen utilizarse para desarrollo de prototipos, sistemas educativos, controladores de computadora integrados ya que no dependen de ranuras de para funciones

periféricas (*COM-Based SBCs: The Superior Architecture for Small Form Factor Embedded Systems A White Paper from Diamond Systems Corp, n.d.*).

A continuación, se presentan los computadores de placa más populares en el mercado. Raspberry pi 4 Figura 12, es una tarjeta con todas las funciones computadora que es capaz de ejecutar aplicaciones como una PC de escritorio estándar. El dispositivo está basado en Broadcom BCM2835, que contiene un ARM1176JZFS de 700 MHz CPU con unidad de punto flotante de hardware. Memoria a bordo es de 512 MB (Sobota, PiŠl, Balda, & Schlegel, 2013).

Figura 12

Raspberry pi 4



Nota. Tomado de “Raspberry Pi and Arduino boards in control education” por Sobota, J., PiŠl, R., Balda, P., & Schlegel, M, 2013, IFAC Proceedings Volumes

Jetson AGX Xavier es un computador que permite la implementación de aplicaciones de robótica, observar la Figura 13 con inteligencia artificial, ya sea en el área industrial, agricultura o investigación. Compatible con los SDK de NVIDIA JetPack y DeepStream, así como con las bibliotecas de software CUDA®, cuDNN y TensorRT, en la se puede observar este dispositivo (Hossain & Lee, 2019; “Jetson AGX Xavier Developer Kit | NVIDIA Developer,” n.d.).

Figura 13

Jetson AGX Xavier



Nota. Tomado de “Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices” por Hossain, S., & Lee, D, 2019, Sensors.

2.11.4 Sensor Lidar

En los últimos años, Lidar (un acrónimo de detección de luz y rango) ha progresado desde una técnica de medición útil adecuada para estudios de cartografía aérea a automóviles autónomos. Lidar es una técnica de medición conocida desde el siglo pasado, se basa en un principio de funcionamiento simple basado en contar el tiempo entre eventos en magnitudes realizadas por la luz. A partir de estas mediciones de tiempo, la velocidad de la luz en el aire se utiliza para calcular distancias o realizar mapas observar en Figura 14. Lidar se ha vuelto tan relevante que se desarrolló una industria completa relacionada con la teledetección y la cartografía (Royo & Ballesta-Garcia, 2019).

Figura 14

Mapa generado por Lidar

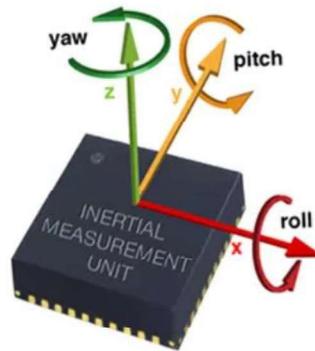


Nota. Tomado de “An Overview of Lidar Imaging Systems for Autonomous Vehicles” por Royo, S., & Ballesta-Garcia, M, 2019, Applied Sciences.

2.11.5 Unidad de medición inercial

Una de las principales ventajas de usar IMU en vehículos es que los datos de aceleración, rotación angular y actitud se proporcionan en altas tasas de actualización. Por lo tanto, la posición y velocidad del móvil puede ser evaluado con una alta frecuencia. A diferencia de los codificadores de rueda, una IMU no se ve afectada por el deslizamiento de las ruedas, que se encuentra en la mayoría de los vehículos terrestres.

La estructura general de una IMU se compone de tres acelerómetros, tres giroscopios y barómetro, observar la Figura 15. Estos conjuntos de sensores proporcionan la aceleración, la velocidad de rotación y la inclinación del vehículo, lo que se utiliza para calcular la odometría del robot (Sukkarieh, Nebot, & Durrant-Whyte, 1999).

Figura 15*Unidad de Medición Inercial*

Nota. Tomado de “A high integrity IMU/GPS navigation loop for autonomous land vehicle applications” por Sukkarieh, S., Nebot, E. M., & Durrant-Whyte, H. F, 1999, IEEE Transactions on Robotics and Automation

2.12 . Sistema comunicación

2.12.1 Introducción a ROS

El sistema operativo de robots o ROS es una colección de framework para crear aplicaciones de software en robótica. El objetivo principal de ROS es proporcionar las capacidades para crear robótica robusta, aplicaciones que pueden reutilizarse para otros robots. ROS tiene una colección de herramientas de software, bibliotecas y una colección de paquetes que facilita el desarrollo de software de robots (Joseph, 2015).

ROS es un proyecto de código abierto BSD, se puede usar para investigación y aplicaciones comerciales. Aunque ROS significa sistema operativo de robot, no es un sistema operativo real, más bien, es un sistema

meta-operativo, que proporciona las características de un sistema operativo real (“ROS/Introduction - ROS Wiki,” n.d.).

Estas son las principales características que proporciona ROS:

- **Transmisión de mensajes:** esta es la característica principal de ROS, y permite comunicación entre procesos, usando esta capacidad de pasar mensajes, permite comunicarse con sistemas vinculados e intercambiar datos.
- **Abstracción de hardware:** ROS tiene un grado de abstracción que permite a los desarrolladores crear software independiente, este tipo de aplicación se puede usar con cualquier tipo de robot; los desarrolladores solo deben preocuparse por el hardware del robot subyacente.
- **Gestión de paquetes:** los nodos ROS se organizan en paquetes, llamados paquetes ROS. Los paquetes ROS consisten en códigos fuente, archivos de configuración, archivos de compilación.
- **Integración de bibliotecas de terceros:** ROS está integrado con muchas bibliotecas de terceros, como Open-CV, PCL, Open NI, etc. Esto ayuda a los desarrolladores a crear todo tipo de aplicaciones en ROS.
- **Control de dispositivos de bajo nivel:** cuando se trabaja con robots, es posible que se necesite trabajar con dispositivos de bajo nivel, como los que controlan pines de E / S, enviando datos a través de puertos seriales, etc. Esto también se puede hacer usando ROS.

- **Computación distribuida:** la cantidad de computación requerida para procesar los datos de los sensores del robot es muy alta, usando ROS, se puede distribuir fácilmente el cálculo a un grupo de nodos informáticos. Esto distribuye la computación y le permite procesar los datos de manera más rápida a lo que sucedería solo usando un computador.
- **Reutilización de código:** el objetivo principal de ROS es la reutilización de código, la reutilización del código permite el crecimiento de una buena comunidad de investigación y desarrollo en todo el mundo.
- **Independencia del lenguaje:** en ROS se puede programar usando lenguajes populares (como Python, C ++ y Lisp). Los nodos se pueden escribir en cualquier lenguaje y pueden comunicarse a través de ROS sin ningún problema.
- **Pruebas sencillas:** ROS tiene un marco de pruebas de unidad incorporado llamado ros test para prueba los paquetes ROS.
- **Escalado:** ROS se puede escalar para realizar cálculos complejos en robots.
- **Fuente libre y abierta:** el código fuente de ROS es abierto y es absolutamente gratis para utilizar, la parte central de ROS está bajo una licencia BSD y puede reutilizarse en Productos comerciales y de código cerrado.

ROS es una combinación de transmisión de mensajes, herramientas, capacidades y ecosistema, dispone de herramientas poderosas para depurar y

visualizar los datos del robot, tiene incorporado capacidades, como navegación de robot, localización, mapeo, manipulación, entre otras, las cuales ayudan a crear potentes aplicaciones de robótica (Joseph, 2015).

En la Figura 16 se muestra la estructura base de ros.

Figura 16

Ecuación ROS



Nota. Tomado de “Learning robotics using Python : design, simulate, program, and prototype an interactive autonomous mobile robot fom scratch with the help of Python” por Joseph, L,2015, ROS, and Open-CV

2.12.2 Conceptos ROS

Hay tres niveles principales en ROS:

- El sistema de archivos ROS
- El gráfico de cálculo ROS
- La comunidad ROS

2.12.3 El sistema de archivos de ROS

El sistema de archivos ROS abarca la organización de los archivos ROS en el disco, a continuación, se presentan los términos principales que se debe entender al trabajar con el sistema de archivos ROS:

- **Paquetes:** los paquetes ROS son la unidad individual del marco de software ROS. Un paquete ROS puede contener código fuente, bibliotecas de terceros, configuración archivos, etc.
- **Manifiestos del paquete:** el archivo de manifiesto (`package.xml`) tendrá todos los detalles de los paquetes, incluidos el nombre, la descripción, la licencia y, lo que es más importante, las dependencias del paquete.
- **Tipos de mensajes (msg):** las descripciones de los mensajes se almacenan en la carpeta `msg` en un paquete, los mensajes ROS son estructuras de datos para enviar datos a través de ROS.
- **Tipos de servicio (srv):** las descripciones de servicio se almacenan en la carpeta `srv` con la extensión. `srv`. El archivo `srv` define la estructura de datos de solicitud y respuesta para el servicio en ROS.

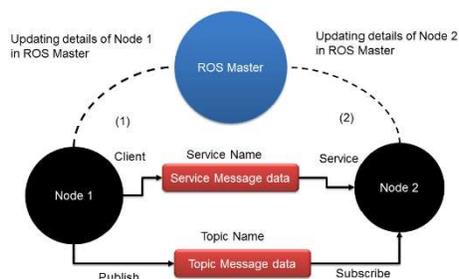
2.12.4 El gráfico de cálculo de ROS

Es la red punto a punto de sistemas ROS que procesa datos, las características básicas de ROS son nodos, ROS Master, el parámetro servidor, mensajes y servicios los cuales se detalla a continuación:

- **Nodos:** el nodo ROS es un proceso que utiliza funcionalidades ROS para procesar los datos, un nodo lo que realiza es computar por ejemplo puede procesar el láser escáner de datos para verificar si hay alguna colisión.
- **ROS master:** los nodos ROS pueden conectarse entre sí mediante un programa llamado Maestro ROS, esto proporciona el nombre, el registro y la búsqueda del resto del gráfico de cálculo.

- **Servidor de parámetros:** los datos o parámetros de ROS son valores estáticos que se almacenan en una ubicación global, llamada el servidor de parámetros. Es en el servidor de parámetros donde todos los nodos pueden acceder a estos valores y presentar una comunicación paralela.
- **Topics ROS:** los nodos ROS se comunican entre sí mediante un bus llamado ROS topic. Los datos fluyen a través del topic en forma de mensajes, el envío de mensajes sobre un topic se llama publicación y recibir los datos a través de un topic se llama suscribirse.
- **Mensajes:** un mensaje ROS es un tipo de dato que puede consistir en distintos tipos de datos primitivos, como enteros, puntos flotantes y booleanos.
- **Servicios:** Los servicios de ROS funcionan de manera similar a los ROS Topics en los que tienen una definición de tipo de mensaje. Usando esa definición de mensaje, se puede enviar la solicitud de servicio a otro nodo que proporcione el resultado del servicio.
- **Bolsas:** estos son formatos en los que puede guardar y reproducir los Topics de ROS. Las bolsas ROS son una herramienta importante para registrar los datos del sensor y los datos procesados.

El siguiente diagrama muestra cómo funcionan los Topics y servicios entre los nodos y el Maestro:

Figura 17*Comunicación entre nodos y master ROS*

Nota. Tomado de “Learning robotics using Python : design, simulate, program, and prototype an interactive autonomous mobile robot fom scratch with the help of Python” por Joseph, L,2015, ROS, and Open-CV

En la Figura 17 se aprecia a dos nodos ROS con el respectivo ROS maestro o Master el cual se encuentra entre los dos nodos, un punto importante de mencionar es que ROS master debe ser inicializado antes de cualquier procedimiento a utilizar. ROS master funciona como intermediario entre los dos nodos, con lo cual será el encargado de administrar la información de los nodos para establecer la comunicación entre ellos.

2.12.5 El nivel de la comunidad de ROS

La comunidad ROS está formada por desarrolladores e investigadores de ROS, que pueden crear y mantener paquetes e intercambiar nueva información relacionada con paquetes existentes o recientemente publicados, la comunidad ROS proporciona los siguientes servicios.

- **Distribuciones:** una distribución ROS tiene un conjunto de paquetes que vienen con una versión.

- **Repositorios:** Los repositorios en línea son los lugares donde se guarda los paquetes, los desarrolladores mantienen un conjunto de paquetes similares llamados meta paquetes en un repositorio.
- **La wiki de ROS:** La wiki de ROS es el lugar donde casi toda la documentación de ROS está disponible. Se puede aprender sobre ROS, desde sus conceptos más básicos hasta programación más avanzada.

2.13 Simuladores robóticos

2.13.1 *Webots*

Webots es un software de simulación robótica móvil que le proporciona un rápido entorno de creación de prototipos para modelado, programación y simulación de robots móviles. Las bibliotecas de robots proporcionadas le permiten transferir sus programas de control a varios robots móviles reales disponibles comercialmente. Webots te permite definir y modificar una configuración completa de robótica móvil, incluso varios robots diferentes que comparten el mismo entorno. Para cada objeto, puede definir una serie de propiedades, como forma, color, textura, masa, fricción, etc. Puede equipar cada robot con una gran cantidad de sensores y actuadores disponibles. Puede programar estos robots usando su favorito entorno de desarrollo, simularlos y, opcionalmente, transferir los programas resultantes a sus robots reales Colocar que es Webots y cosas así, (Michel, 2004; Zea, Guevara, Recalde, & Andaluz, 2021)

Capítulo III

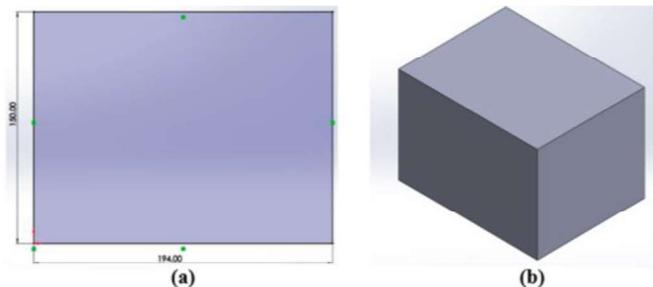
3. Diseño y construcción del sistema mecánico, electrónico e interface

3.1 Diseño mecánico del robot móvil

Para dimensionar los diferentes elementos que compondrán el robot móvil, es necesario delimitar el área y el volumen de carga que va a transportar, tomando en cuenta los diferentes objetos dentro del laboratorio, con lo cual se inicia que el área establecida para este robot es de 194 mm en **X**, 150 mm en **Y** como se presenta en la Figura 19 (a) y una altura de 40 mm en **Z** dando un espacio de trabajo como se representa en la Figura 19 (b)

Figura 19

Diseño de la base de carga del robot



Nota. En el gráfico a) Área de carga del robot b) Volumen de carga del robot.

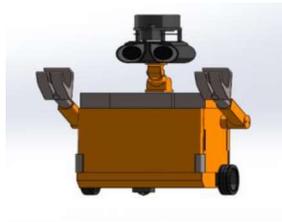
El robot se diseñó para soportar una carga máxima de 1000 g lo cual se considera suficiente para aplicaciones educativas dentro del laboratorio, en la Figura 20 se observa el robot en su diseño final.

Para el diseño del robot se priorizo que debe ser didáctico, de bajo costo y fácil mantenimiento, por consiguiente, se utilizaron materiales económicos y de

fácil acceso, por lo tanto, se utilizara el PLA en impresión FDM 3D para la totalidad de la estructura del robot.

Figura 20

Diseño final



Para el análisis estructural del robot se utilizará la Tabla 1 ya que ofrece un análisis de cómo va a comportarse la impresión de acuerdo a la norma ISO 527-1, ofreciendo una relación entre la altura por capa y porcentaje de relleno (Alvarez, Lagos, & Aizpun, 2016)

Tabla 1

Esfuerzo (MPa) de acuerdo al porcentaje de relleno

	Porcentaje de relleno							
	10	30	50	70	80	90	100	
Altura por capa (mm)	0.1	4	9	15	24	29	34	40
	0.15	8	13	19	28	33	38	44
	0.2	10	15	21	30	35	40	46
	0.25	11	16	22	31	37	42	48
	0.3	11	16	22	31	36	41	48

Nota. Tomado de “Influencia del porcentaje de relleno en la resistencia mecánica en impresión 3D, por medio del método de Modelado por Deposición Fundida

(FDM)” por Alvarez, K. L. C., Lagos, R. F. C., & Aizpun, M.,2016, Ingeniare.

De acuerdo a lo indicado en la Tabla 1, utilizando una técnica de impresión 3D en FDM, con una altura de capa de 0.3mm, un grosor de pared de 1.6, obteniendo 4 paredes lo cual otorga mayor rigidez y un 50% de relleno, obteniendo un esfuerzo máximo de 22MPa

Tabla 2

Propiedades Mecánicas del PLA

Propiedades	Unidades	Valores
Densidad	kg/m^3	1240
Resistencia a la tracción	N/mm^2	47
Coefficiente de Poisson	-	0.38
Límite elástico	N/mm^2	55
Módulo de elasticidad	N/mm^2	3300

Nota. Tomado de “Influencia del porcentaje de relleno en la resistencia mecánica en impresión 3D, por medio del método de Modelado por Deposición Fundida (FDM)” por Alvarez, K. L. C., Lagos, R. F. C., & Aizpun, M.,2016, Ingeniare

Para realizar el análisis de los componentes se emplearán los parámetros de la Tabla 1 y Tabla 2. A continuación, en la Tabla 3 se detallan los pesos individuales de los componentes del robot y su peso total, los cuales serán empleados para realizar el análisis de esfuerzos.

Tabla 3*Pesos de componentes del robot*

Elemento	Cantidad	Peso (g)
Base_1	1	230.37
Base_2	1	344.68
Base_3	1	160.05
Pinza_d	1	146.22
Pinza_i	1	146.22
Tapa_delantera	1	221.79
Tapa trasera	1	146.13
Tapa superior	1	72.25
Soporte ojos	1	48.84
Ojos	1	70.72
Tapa ojos	1	12.60
Sujetador bateria	1	3.80
Tapa bateria	1	9.28
Total		1612.5

3.2 Diseño y análisis de esfuerzos

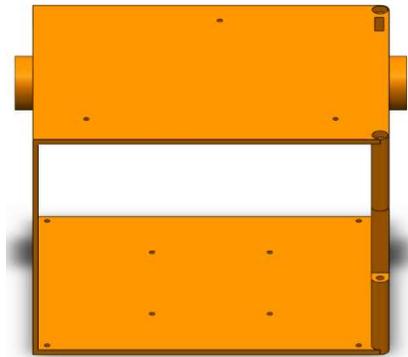
Debido a su ligero peso estructural no requiere un análisis minucioso de cada elemento que lo conforma, puesto que por las características del material si soporta su peso. Por lo tanto, únicamente se requiere analizar los componentes que van a soportar la carga.

3.2.1 *Diseño y análisis de esfuerzos en la base de carga*

El diseño de esta pieza tiene el propósito de soportar la carga que se le va aplicar en el laboratorio con un máximo de 1 kg, debido al amplio espacio en voladizo que tenía este elemento, se decidió propinarle 4 puntos de apoyo en el centro de la base para obtener un mejor comportamiento y distribución de la carga, quedando sujeta en 8 puntos mediante pernos M 3mm como se muestra en la Figura 21, y a su vez su perfil inferior reposa en el perfil de la “Base_1” otorgándole una mayor superficie de apoyo.

Figura 21

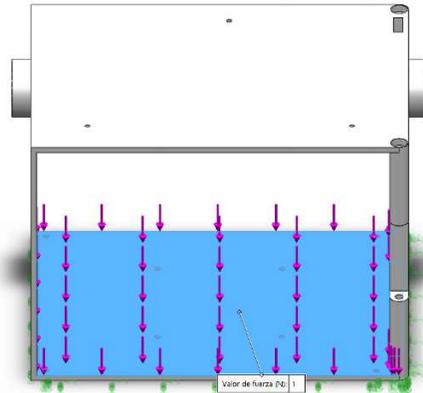
Pieza Base_2



Para el análisis mecánico del robot se realizarán 2 consideraciones, la primera en donde el peso que soporta la superficie de carga es una carga distribuida como se puede observar en la Figura 22, y la segunda en donde también se tomará en cuenta el peso de la base de carga con la fuerza gravitacional, para poder obtener los resultados más reales posibles.

Figura 22

Representación de la carga distribuida



Se determina el peso de la carga distribuida y se consideran los apoyos

$$W_1 = mg = 9.81N \quad (1)$$

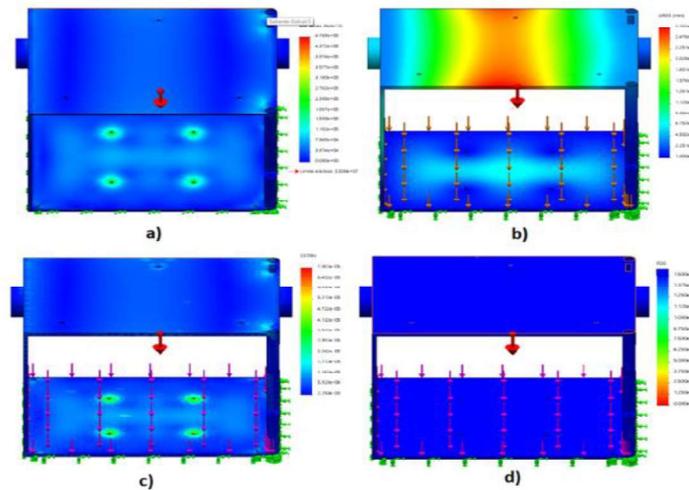
$$W_2 = mg = 1.098N \quad (2)$$

Tomando en cuenta que el PLA es un material dúctil, y el área de carga va a estar bajo carga estática, se considera un factor de diseño recomendable entre 1.25 y 2.0, para el robot utiliza un factor de diseño de $n= 3$.

Se usará un valor de 3mm de tamaño de la malla para el análisis mecánico de la pieza, debido a que, al reducir el tamaño de malla, el tiempo de procesamiento de datos es mayor y el resultado es muy similar. El análisis por elementos finitos se muestra en la figura 16, en donde se analizó el esfuerzo de Von- Mises, la deformación, deformación unitaria y el factor de seguridad.

Figura 23

Análisis por software de la pieza Base_2



Nota. El gráfico representa a) Esfuerzo Von-Mises b) Deformación c) Deformación unitaria d) Factor de diseño

De la Figura 23 se pueden observar los valores obtenidos en el análisis del software de SolidWorks:

- Esfuerzo Von- Mises: 0 a 0.4769 MPa
- Deformación: 1×10^{-30} a 2.701×10^{-2}
- Deformación unitaria: 2.29×10^{-8} a 7.082×10^{-5}
- Factor de diseño 12

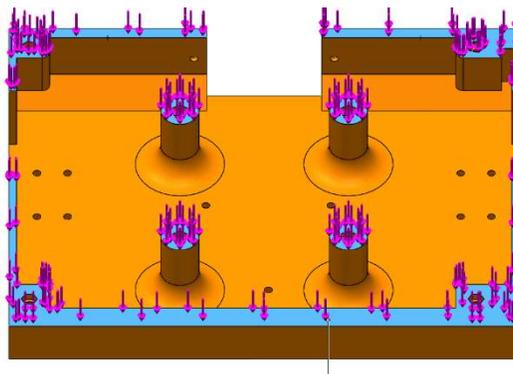
Debido a que el factor de diseño calculado por el software es de 12 muestra que la estructura está sobredimensionada para la carga que debe soportar, por lo tanto, asegura que la estructura no fallará.

3.2.2 Diseño y análisis de esfuerzos en la base del robot

El diseño de esta pieza debe ser capaz de soportar la carga de la estructura del robot que está en la parte superior a esta, y el peso de la carga que debe transportar el robot, por lo tanto, se le aplicará una carga distribuida en las áreas de apoyo como se muestra en la Figura 24.

Figura 24

Representación de la carga distribuida en la superficie



$$m_{estructura} = 344.68 + 160.05 + 146.22 + 146.22 + 221.79 + 146.13 + 72.25 + 48.84 + 70.72 + 12.6 = 1369.5g \quad (3)$$

$$m_{estructura} = 119g \quad (4)$$

$$m_{carga} = 1000g \quad (5)$$

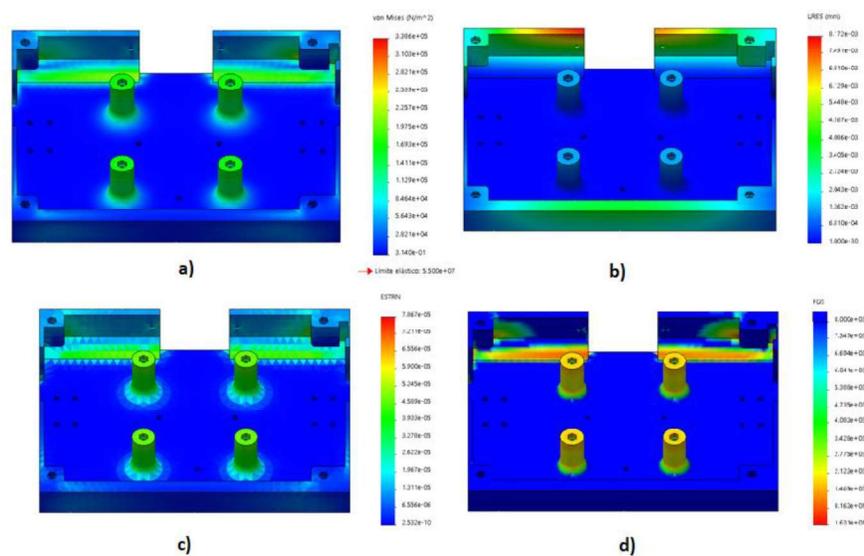
$$m_{total} = 2469.5g \quad (6)$$

$$W_{total} = 24.23N \quad (7)$$

Se utiliza un valor de malla de 3mm para el análisis mecánico de la pieza, debido a que, al reducir el tamaño de esta, aumenta el tiempo de procesamiento y el resultado es muy similar.

Figura 25

Análisis por software de la pieza Base_1



Nota. El gráfico representa a) Esfuerzo Von-Mises b) Deformación c) Deformación unitaria d) Factor de diseño.

Como se puede apreciar en la Figura 25 en la cual se ha realizado el análisis por elementos finitos, obteniendo los valores del esfuerzo de Von-Mises, la deformación, deformación unitaria y el factor de diseño, siendo estos los siguientes:

- Esfuerzo Von- Mises: 0.0000314 a 0.3386 MPa
- Deformación: 1×10^{-30} a 8.17×10^{-2}
- Deformación unitaria: 2.532×10^{-10} a 7.867×10^{-5}
- Factor de diseño 16.3

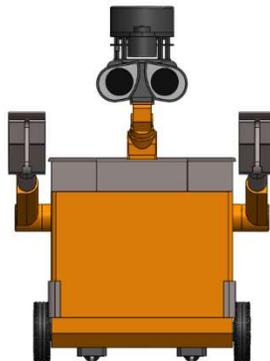
Debido a que el factor de diseño calculado por el software es de 16 en donde este es mayor a 3, se comprueba es capaz de soportar cargas estáticas, por lo tanto, asegura que la estructura no fallará.

3.3 Disposición final del robot Wall3

El ensamble final del robot móvil se muestra en la Figura 26, el cual muestra todos los componentes con anterioridad armados y dispuestos para su correcto funcionamiento.

Figura 26

Ensamble final del robot Wall3



3.4 Ensamble del prototipo

Finalizado el diseño del robot, se emplearán el proceso de impresión 3D por "FDM" (Deposición de material fundido) para cada una de las piezas que conforman al robot, a continuación, se describe el proceso que se empleó para cada pieza. Para la elaboración de estas piezas, hay que considerar varios parámetros, entre los más importantes están el espesor de las paredes, la altura de la capa y el porcentaje de relleno.

3.5 Construcción de la estructura prototipo

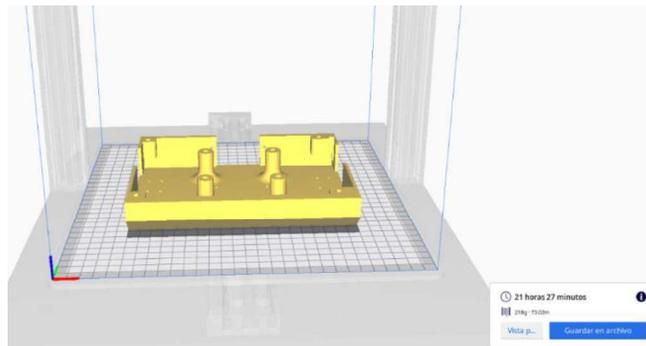
3.5.1 Construcción de la base

Para la elaboración de la base se determina un espesor de pared de 1.6 mm, tomando en cuenta la Tabla 1, la altura de capa es de 0.3 mm, y el porcentaje de relleno de un 50%, ya que con estos parámetros se obtiene una mejor resistencia a los esfuerzos.

Finalmente, para prevenir el “Warping” ya que el tiempo de impresión es superior a las 21 horas, se procede a colocar un borde en el perfil de la base de la pieza, este proceso se observa en la Figura 27.

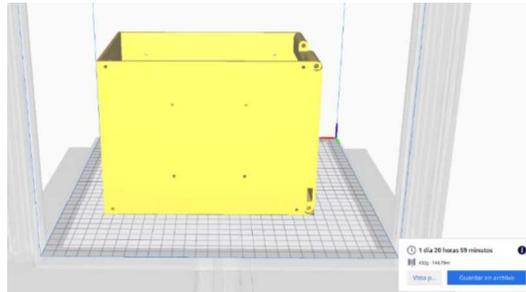
Figura 27

Base del robot Wall3

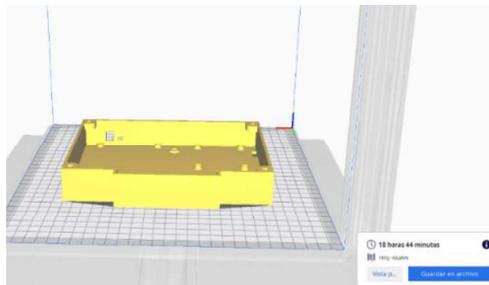


3.5.2 Construcción de espacio de carga

Para la construcción de esta pieza como se puede apreciar en la Figura 28, se consideró la carga que debe soportar, esta carga ya fue analizada previamente, por lo tanto, el porcentaje de relleno es del 50%, la altura de la capa de impresión es de 0.3 mm y el espesor de pared es de 2 mm, esta última consideración es tomada para prevenir pandeos.

Figura 28*Espacio de carga del robot Wall3***3.5.3 Construcción de la caja de control**

La caja de control almacenará los componentes electrónicos de control, previamente diseñado los agujeros para sujetarlos y los espacios para el pin de carga de la tarjeta Open Cr y el switch de on/ off para poder acceder a estos desde el exterior con facilidad, tal como se puede apreciar en la Figura 29.

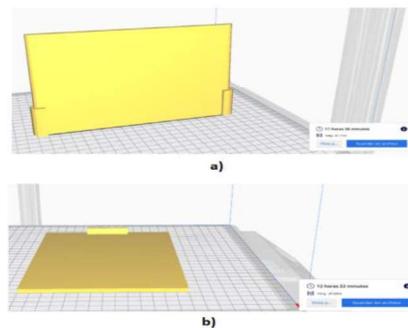
Figura 29*Caja de control del robot Wall3***3.5.4 Construcción de la puerta delantera y posterior**

Las puerta delantera y posterior tienen un diseño de una bisagra en su lateral derecho como se muestra en la Figura 30, que es sujeta por un tornillo M4 permitiendo su correcto funcionamiento, gracias a la fricción que se produce

entre la puerta y la base del robot, las puertas no requieren de un seguro para mantenerlas cerradas.

Figura 30

Puertas del robot Wall3



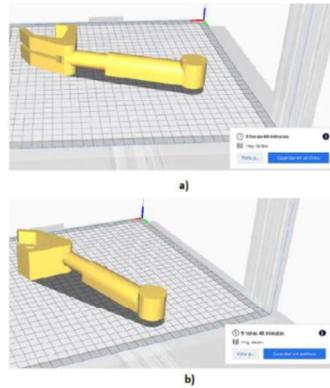
Nota. En el gráfico a) Puerta delantera b) Puerta posterior

3.5.5 Construcción de los brazos del robot

Los brazos del robot, están diseñados para sujetar objetos que no quepan en el espacio de carga, estos deben ser livianos de hasta 200 g, debido a que la carga no representa una fuerza considerable, no requiere de análisis. Estos están sujetos al cuerpo del robot mediante 2 tornillos M3 x 30 mm, como se puede apreciar en la Figura 31. El porcentaje de relleno en estas juntas es del 60% para mejorar el comportamiento mecánico de la pieza, y el espesor de pared es de 1.6mm.

Figura 31

Brazos del robot Wall3



Nota. En el gráfico a) Brazo izquierdo b) Brazo derecho

3.6 Ensamble del robot

3.6.1 Ensamble de la base del robot

En la base del robot se encuentran los motores, la batería y la rueda loca, los cuales están sujetos mediante tornillos M3 x 10mm como se muestra en la Figura 32, luego se procede a colocar tuercas M3 en su respectiva ubicación para poder ajustar la pieza que soporta la carga.

Figura 32

Ensamble de la Base del robot Wall3



3.6.2 *Ensamble del espacio de carga y caja de control del robot*

Para ensamblar esta pieza, simplemente se debe colocarla en la ubicación correcta de acuerdo al diseño CAD para que coincidan los agujeros y poderlos asegurar con un tornillo M3 X 10mm, posteriormente se coloca la pieza Caja de control como se puede observar en la Figura 33.

Figura 33

Ensamble del Espacio de carga del robot Wall3



3.6.3 *Ensamble de las puertas y brazos del robot*

A continuación, se procede a colocar las puertas delantera y posterior del robot, mediante dos tornillos M4 x 60 mm, el primero se lo inserta en la parte superior de las puertas y el segundo en la parte inferior de las mismas, estos están ajustados mediante dos tuercas M4 como se muestra en la Figura 34 (a). una vez colocadas las puertas del robot se procede a colocar los brazos izquierdo y derecho, estos están sujetos mediante tornillos M3 x 30 mm como se muestra en la Figura 34 (b).

Figura 34

Ensamble del robot



Nota. El gráfico representa a) Ensamble de las puertas b) Ensamble de los brazos

3.6.4 Ensamble de la tapa superior del robot y el soporte del sensor

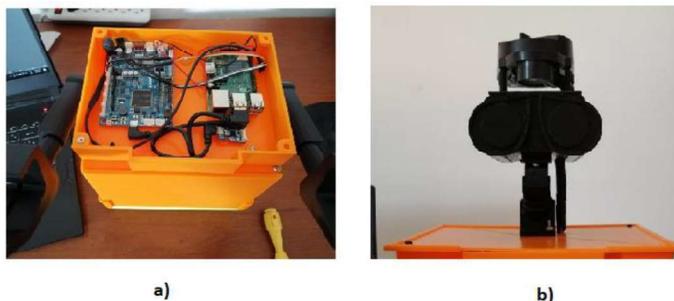
Lidar

Se procede a colocar y ajustar las tarjetas de control y el conector USB mediante tornillos M3 X 6 mm a la caja de control como se muestra en la Figura 35 (a), a continuación, se conecta los cables de los motores, batería y del sensor Lidar en los pines correspondientes de acuerdo a la conexión electrónica que se detalla a continuación.

Seguidamente se coloca la tapa superior que cubre la caja de control mediante 4 tornillos M3 en cada una de sus esquinas, después se ensambla el soporte donde va colocado el sensor Lidar y se procede a ajustarlo con la tapa superior como se muestra en la Figura 35 (b).

Figura 35

Ensamble del robot



Nota. El gráfico representa a) Colocación de las tarjetas de control b) Ensamble de la tapa superior del robot y el soporte del sensor Lidar

3.7 Ensamble Final

En la Figura 36 se muestra el ensamble final del robot Wall3, en donde se puede observar todos los componentes anteriormente detallados que lo conforman.

Figura 36

Ensamble final del robot Wall3



3.8 Electrónica del robot móvil

3.8.1 Selección de componentes electrónicos

Para la selección de componentes electrónicos se definió que los algoritmos de control serán procesados en una computadora principal debido al costo computacional de los algoritmos y el robot móvil dispondrá de una mini computadora encargada de gestionar y administrar los sensores y los actuadores, además el robot móvil debe ser controlado de manera inalámbrica por lo cual se toman las siguientes consideraciones:

- Conexión inalámbrica
- Procesamiento matemático para los algoritmos de control
- Sistema de comunicación estable y escalable.

- Autonomía del robot.

En la Tabla 4 se muestra la función de cada uno de los componentes a utilizar en el robot móvil.

Tabla 4

Selección de componentes

Componente	Descripción
Microprocesador	Comunicación con la computadora principal y gestión de sensores
Controlador embebido	Control de periféricos
Sensor Lidar	Permite determinar la ubicación .
IMU	Permite conocer las aceleraciones y velocidades.
Servo motor	Motores con PID internos
Batería lipo	Alimentación eléctrica al robot móvil

3.8.2 Selección del microprocesador

El microprocesador es muy importante para el robot móvil debido a que es el encargado de recibir, procesar y responder los datos enviados desde la computadora encargada del control, además se debe tomar en cuenta que, debido al canal de comunicación a usar es muy importante que este microprocesador sea capaz de correr un sistema operativo, en este caso Ubuntu Mate. Gracias a las características de un sistema operativo se encarga de gestionar los sensores y actuadores, de igual modo se puede aumentar más sensores u otros elementos robóticos por lo que el sistema se vuelve muy escalable.

Para la selección de esta placa se tomó varias consideraciones las cuales se mostrarán en la Tabla 5.

Tabla 5

Comparación SBC

Criterios de selección	Raspberry pi 3 B+	Raspberry pi 4	BeagleBone Black
Costo	+	-	+
Conexiones Wireless	+	+	+
Documentación	+	0	+
Procesamiento	-	+	-
Disponibilidad	+	0	0
Suma (+)	4	2	3
Suma (-)	1	1	1
Suma (0)	0	2	1
Evaluación	3	1	2

3.8.3 Selección de microcontrolador

Una vez seleccionado el microprocesador se selecciona el microcontrolador el cual debe ser capaz de sostener una comunicación estable, manejar los periféricos de manera rápida y adecuada.

En la Tabla 6 se presentan algunos modelos acordes, la placa selección es la Open CR ya que presenta partes físicas y de procesamiento muy útiles en este proyecto, además de compatibilidad con motores Dynamixel.

Tabla 6*Análisis para selección de microcontrolador*

Criterios de selección	Arudino Due	Open Cr	Arduino Uno
Costo	+	+	+
Procesador	+	+	-
Documentación	-	0	+
Protocolos	0	+	0
Comunicacion			
Adaptabilidad	+	+	0
Suma (+)	3	4	2
Suma (-)	1	0	1
Suma (0)	1	1	2
Evaluación	2	4	1

3.8.4 Selección del sensor Lidar

El sensor tipo Lidar es muy importante para la navegación autónoma del robot móvil, ya que permite una reconstrucción del entorno, lo cual le permite determinar al robot la evolución del entorno y en el caso de ser posible planificar rutas alternativas.

Este tipo de sensor es muy usado al momento de realizar SLAM también conocido como localización y mapeo simultáneo lo cual es muy útil para el robot ya que permite saber sus estados con mucha exactitud.

En el mercado se encuentra una variedad muy extensa de estos sensores, pero los puntos a tomar en cuenta son el precio y el tipo de

comunicación que disponen, ya que este tipo de sensores envían una gran cantidad de datos lo que ralentiza el proceso de control y los precios pueden llegar a ser muy elevados. Las posibles alternativas se muestran en Tabla 7, la ganadora es el sensor Lidar HLS-LFCD2.

Tabla 7

Análisis para selección Lidar

Criterios de selección	OS1	HLS-LFCD2
Costo	-	+
Disponibilidad	0	+
Ángulo de escaneo	+	0
Número de revoluciones	+	0
Suma (+)	2	2
Suma (-)	1	0
Suma (0)	1	2
Evaluación	1	2

3.8.5 Selección de la unidad de medida inercial

La selección de un correcto módulo de medida inercial es de vital importancia para el sistema robótico, ya que es la que permite calcular la odometría del sistema, lo cual se utiliza en cada una de las leyes de control.

En la Tabla 8 presentan dos alternativas, pero debido a disponibilidad y al algoritmo del cálculo de la odometría, se optó por utilizar un microcontrolador OPEN CR que ya dispone de este sensor listo para ser usado.

Tabla 8

Análisis para selección de unidad de medición inercial

Criterios de selección	CS-IM100	OPEN CR
Costo	-	+
Disponibilidad	0	0
Precisión	+	+
Documentación	+	+
Suma (+)	2	3
Suma (-)	1	0
Suma (0)	1	1
Evaluación	1	3

3.8.6 Selección de Servo Motores

Para la selección de los servomotores se debe tomar en cuenta que deben disponer control por velocidad y de posición lo cual ayudará en gran medida a no realizar algoritmos de control de bajo nivel.

En la Tabla 9 se presenta a dos tipos de motores con características muy distintas y mercables, la elección son los motores Dynamixel XL 430 debido a su precisión y fácil escalabilidad.

Tabla 9*Selección de servomotores*

Criterios de selección	Parallax Feedback 360°	Dinamixel XL 430
Costo	+	0
Disponibilidad	0	0
Precisión	-	+
Velocidad	+	+
Suma (+)	2	2
Suma (-)	1	0
Suma (0)	1	2
Evaluación	1	2

3.9 Desarrollo de interfaz del robot móvil

La interfaz de usuario fue desarrollada en “appdesginer”, el cual es un complemento de Matlab, que dispone de varias herramientas desarrolladas para facilitar el diseño de “GUI’s graphical user interface”. Appdesginer integra las dos tareas principales de la creación de aplicaciones: diseñar los componentes visuales de una interfaz gráfica de usuario (GUI) y programar el comportamiento de la aplicación. Es el entorno recomendado para crear aplicaciones en MATLAB (“MATLAB App Designer - MATLAB,” n.d.).

3.9.1 *Menú conexión*

El diseño del menú se presenta en la Figura 37, la cual consta de 5 parámetros que deben ser introducidos por el usuario.

- La dirección IP del robot móvil la cual ya se detalló en la sección anterior.

- La dirección IP del computador Remoto o donde se está ejecutando la interfaz.
- El nombre del Tópico de Ros para los comandos de velocidades del Robot.
- El nombre del Tópico de Ros de la odometría del Robot
- El nombre del Tópico de Ros del sensor Lidar del Robot

Figura 37

Menú conexión



Luego de ser colocados estos valores se procede a conectar el robot, para esto se tiene dos botones que cumplen dicho propósito, esto se puede observar en Figura 38.

- Conectar, establece la conexión entre la interfaz de usuario y el Robot
- Desconectar cierra la comunicación con el Robot, para evitar conflictos de comunicación.

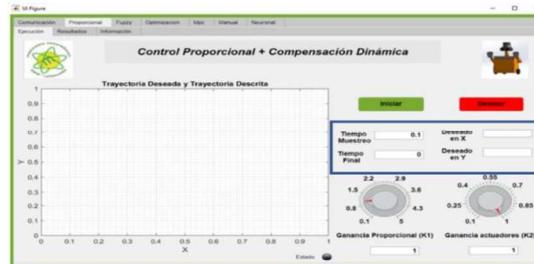
Figura 38*Menú conexión botones*

3.10 Menú control cinemático

3.10.1 Ejecución

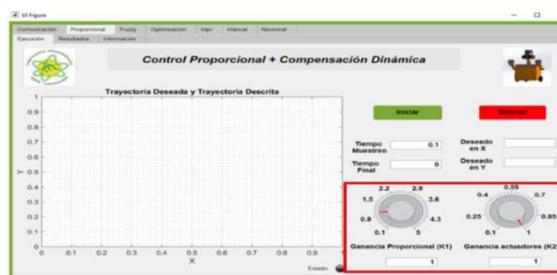
En esta sección los parámetros que debe proporcionar el usuario se muestran en Figura 39, a continuación, se los detalla:

- Tiempo muestreo es el intervalo de tiempo en que se toman datos en el sistema.
- Tiempo Final delimita hasta que valor se va a ejecutar el programa.
- Deseado en X define el punto deseado en el sistema de referencia con respecto a X.
- Deseado en Y define el punto deseado en el sistema de referencia con respecto a Y.

Figura 39*Menú cinematográfico tiempos*

El usuario podrá modificar las ganancias antes o durante la ejecución del programa se los puede observar en Figura 40 además a continuación se detallan:

- Ganancia Proporcional este valor está inicializado en 1 pero el usuario tiene la libertad de cambiarlo, su valor máximo y mínimo son 5 y 0.1 respectivamente.
- Ganancia de Actuadores este valor se inicializa en 1, el usuario lo puede modificar durante o al inicio del programa, el valor máximo y mínimo son 1, 0.1 respectivamente.

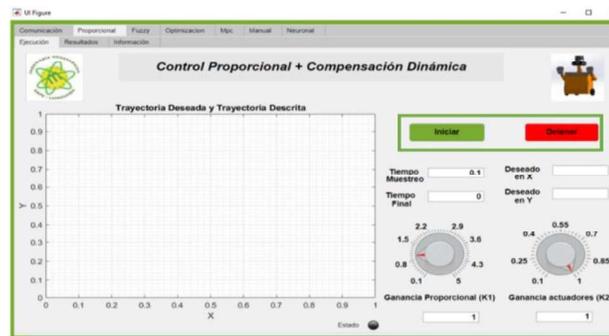
Figura 40*Menú cinematográfico ganancias*

Finalmente, el usuario dispondrá de dos botones para iniciar el programa o detenerlo esto se muestra en Figura 41, estos botones se diseñaron con el

objetivo de si ocurre algún accidente en el programa se pueda parar inmediatamente.

Figura 41

Menú cinemático inicio



3.10.2 Resultados

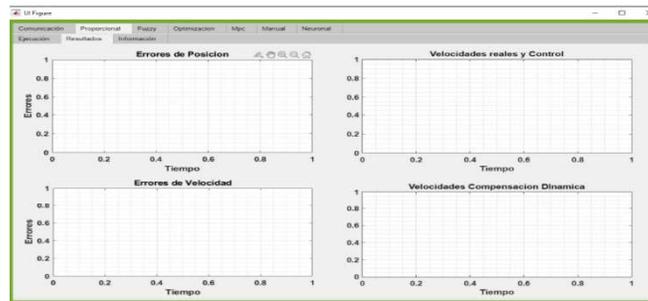
En esta sección se presentan los gráficos con los errores de control, los errores de velocidad, las velocidades reales que se producen en el robot, las velocidades del control cinemático, finalmente las velocidades de compensación dinámica, estos valores son tratados más a fondo en el siguiente capítulo. En la Figura 42 se muestran las ventanas de todas estas gráficas.

3.10.3 Información

En esta ventana se muestra el esquema de control utilizado, así como las ecuaciones de los respectivos controladores, esto se desarrolla con el objetivo que el usuario disponga de información de acceso rápido para entender cómo se encuentra el sistema y que ecuaciones están siendo implementadas, observar la Figura 43 .

Figura 42

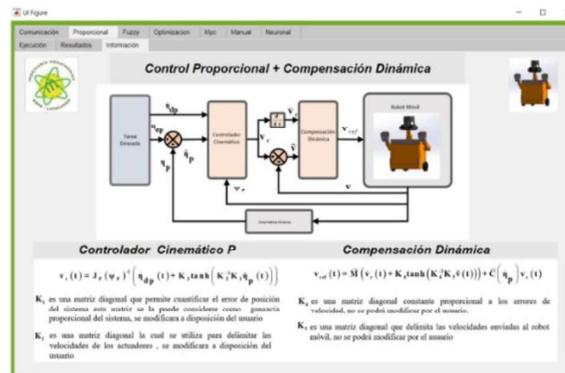
Menú cinemático resultados



Para los diferentes algoritmos se realizaron ventanas muy similares por lo tanto mencionarlas a todas no es algo muy práctico, pero en la que sí se tendrían cambios notorios serían en el control manual.

Figura 43

Menú cinemático información



3.11 Control manual

3.11.1 Ejecución

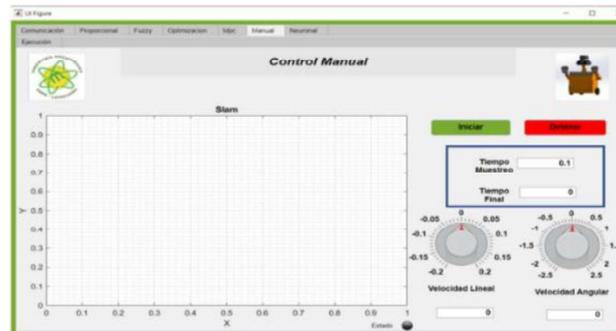
En esta sección se presenta el gráfico del movimiento del robot, incluyendo los datos del sensor, para así poder realizar un Pseudo “Slam Simultaneos localization and mapping”. En esta ventana se presentan 2 valores

que deben ser introducidos por el usuario observar en Figura 44, los mencionamos a continuación:

- Tiempo muestreo intervalo de tiempo en que se toman datos del sistema.
- Tiempo final indica hasta qué instante se va a realizar el programa.

Figura 44

Menú manual tiempos



La ventana dispone de dos potenciómetros los cuales ayudarán al usuario a definir las velocidades, observar en Figura 45, con esto se selecciona a qué velocidad desean en el robot, a continuación, se los detalla.

- Velocidad lineal: este potenciómetro ayudará al usuario a seleccionar a qué velocidad desea que se mueva el robot móvil, este parámetro puede cambiar durante la ejecución del programa.
- Velocidad angular, el usuario debe utilizarlo para definir qué tan rápido podrá girar el robot, de igual forma este valor se podrá cambiar durante la ejecución del programa.

Figura 45*Menú manual potenciómetros*

Finalmente, se dispone de dos botones de conexión y desconexión, los cuales tienen el mismo funcionamiento de los botones de las anteriores ventanas.

3.12 Simulador Webots

Para poder utilizar el robot desarrollado en este proyecto se utiliza la herramienta “SolidWorkd to URDF Exporter” el cual permite transformar la información del modelo CAD al formato URDF conocido como “Unified Robot Description Format”, el cual es un estándar para representar robots.

En el simulador Webots se permite desarrollar controladores para cada robot que se esté utilizando en la escena, por lo que se desarrolló el respectivo controlador para el robot móvil. Este controlador basa en que el robot pueda comunicar la odometría, el valor del sensor Lidar usando el framework ROS, con lo cual se logra que la interfaz de usuario funcione tanto para el robot real como para la simulación, todo esto se detalla en el manual de usuario.

Se desarrolló el entorno de simulación basado en las instalaciones del laboratorio de la carrera Ingeniería en mecatrónica, a continuación, en la Figura 46 se presenta el entorno desarrollado.

Figura 46

Entorno simulado en Webots



Una vez creado el ambiente de simulación se incluye al robot de este proyecto, de esta forma se puede utilizar la interfaz sin la necesidad de tener al robot real, lo cual es muy útil para los estudiantes y su proceso de aprendizaje, en la Figura 47 se observa en el entorno.

Figura 47

Robot Wall3 en el entorno simulado



Además de este entorno se desarrolló otro el cual está compuesto de menos elementos con el objetivo de minimizar los recursos que exige una simulación en un entorno complejo. En la Figura 48 se muestra este ambiente de

simulación el cual exige menos recursos, con el robot implementando en este proyecto

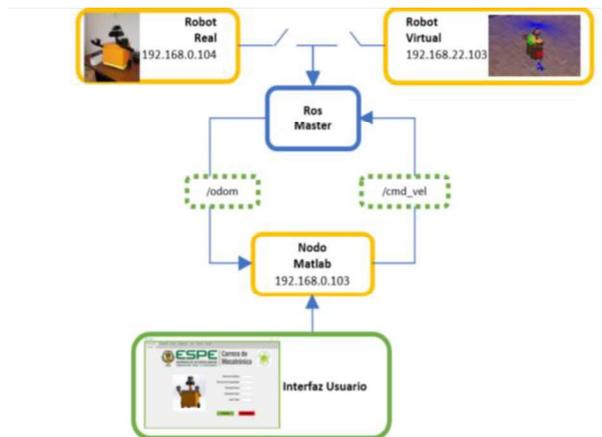
Figura 48

Entorno 2 para simulación



3.13 Estructura del sistema

Ya que se utiliza ROS como un medio de comunicación se puede utilizar la interfaz de usuario tanto en el robot real como el simulador, para comprender esta estructura se presenta la Figura 49 donde se aprecia este funcionamiento, toda la información detallada de como configurar el sistema se encuentra en el manual de usuario, el cual se incluye en la documentación del proyecto.

Figura 49*Estructura funcionamiento del sistema*

Capítulo IV

4. Modelación matemática y algoritmos de control

4.1 Modelación cinemática

4.1.1 Modelo generalizado plataformas móviles

La configuración de una plataforma móvil está definida por un vector \mathbf{q} de n coordenadas independientes, llamadas coordenadas generalizadas, donde $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T = [\mathbf{q}_p]^T$ y \mathbf{q}_p representa las coordenadas generalizadas de la plataforma móvil, se nota que $n = n_a$, donde n_a es la dimensión de espacio de configuración generalizado por lo que el conjunto de todas las configuraciones se denota por \mathcal{N} . La posición y orientación del punto de interés de una plataforma móvil está definida por un vector $\boldsymbol{\eta}$ dado por $\boldsymbol{\eta} = [\eta_1 \ \eta_2 \ \dots \ \eta_m]^T = [\boldsymbol{\eta}_p]^T$ en \mathbf{R} de $\boldsymbol{\eta}_p$ coordenadas operacionales de la plataforma móvil, del mismo modo el conjunto de estas se denota por $\mathcal{M}(G)$. (M. Andaluz et al., 2016a; V. H. Andaluz et al., 2015)

4.1.2 Modelo cinemático plataforma móvil tipo diferencial

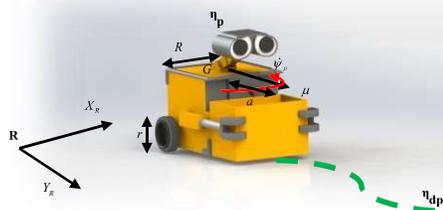
La Figura 50 muestra una plataforma móvil tipo diferencial, donde $\boldsymbol{\eta}_p = [x_p \ y_p \ \psi_p]^T$ es la posición y orientación del punto de interés con respecto a un sistema \mathbf{R} , el punto de interés está ubicado a una distancia a hacia delante del G el cual es el centro geométrico del eje de las ruedas, donde $(x_G \ y_G)$ es la posición de G con respecto a \mathbf{R} . Con las consideraciones mencionadas

anteriormente se elabora el modelo geométrico el cual está representado por (8)

$$\eta_{\mathbf{p}}(t) = \begin{bmatrix} x_p \\ y_p \\ \psi_p \end{bmatrix} = \begin{bmatrix} x_G + a \cos(\psi_p) \\ y_G + a \sin(\psi_p) \\ \psi_p \end{bmatrix} \quad (8)$$

Figura 50

Parámetros modelo cinemático



Derivando (8) se encuentra el modelo cinemático de la plataforma (9), donde $\dot{\eta}_{\mathbf{p}} = [\dot{x}_p \quad \dot{y}_p \quad \dot{\psi}_p]$ representa las velocidades del punto de interés en el sistema de coordenadas \mathbf{R} .

$$\dot{\eta}_{\mathbf{p}} = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\psi}_p \end{bmatrix} = \begin{bmatrix} \dot{x}_G - a \sin(\psi_p) \dot{\psi}_p \\ \dot{y}_G + a \cos(\psi_p) \dot{\psi}_p \\ \dot{\psi}_p \end{bmatrix} \quad (9)$$

Es útil poder representar el modelo cinemático de la plataforma en términos de su velocidad frontal m/s y angular rad/s (10) de esta manera se puede obtener una representación en su forma matricial (11) (Varela-Aldas, Andaluz, & Chicaiza, 2018).

$$\dot{\eta}_{\mathbf{p}} = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\psi}_p \end{bmatrix} = \begin{bmatrix} \mu \cos(\psi_p) - a \sin(\psi_p) \dot{\psi}_p \\ \mu \sin(\psi_p) + a \cos(\psi_p) \dot{\psi}_p \\ \dot{\psi}_p \end{bmatrix} \quad (10)$$

$$\dot{\eta}_{\mathbf{p}} = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\psi}_p \end{bmatrix} = \begin{bmatrix} \cos(\psi_p) & -a \sin(\psi_p) \\ \sin(\psi_p) & a \cos(\psi_p) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \dot{\psi}_p \end{bmatrix} \quad (11)$$

Este modelo cinemático puede ser representado en su forma compacta (12), donde $\mathbf{J}_p(\psi_p)$ es la matriz de transformación en función de la orientación y $\mathbf{v}(t)$ representa el vector de velocidades lineal y angular del robot móvil tipo diferencial.

$$\dot{\eta}_{\mathbf{p}}(t) = \mathbf{J}_p(\psi_p) \mathbf{v}(t) \quad (12)$$

4.2 Modelación dinámica

4.2.1 Modelo dinámico plataforma móvil tipo diferencial

Para encontrar el modelo dinámico se utiliza el método de Euler-Lagrange (13), donde K y P son la energía cinética y potencial, para este análisis se considera a una masa puntual ubicada en el punto de interés $\eta_{\mathbf{p}}$.

La energía cinética del sistema se representa mediante (14) donde m es la masa del robot diferencial, debido a que el robot no presenta desplazamiento a lo largo del eje z el sistema no posee energía potencial(15), desarrollando (14) y (15) se obtiene el Lagrangiano (16) (Cardona, Leal, & Ramirez, 2018; Tzafestas, 2013).

$$L = K - P \quad (13)$$

$$K = \frac{1}{2} \dot{\eta}_{\mathbf{p}}^T(t) m \dot{\eta}_{\mathbf{p}}(t) \quad (14)$$

$$P = 0 \quad (15)$$

$$L = \frac{1}{2} m (\dot{x}_p^2 + \dot{y}_p^2 - 2a \sin(\psi_p) \dot{\psi}_p \dot{x}_p + 2a \cos(\psi_p) \dot{\psi}_p \dot{y}_p + a^2 \dot{\psi}_p^2 + \dot{\psi}_p^2) \quad (16)$$

El método de Euler-Lagrange se basa en utilizar(17), para poder determinar los torques y fuerzas de la plataforma móvil tipo diferencial.

Derivando (16) con respecto a $\dot{\eta}_{\mathbf{p}}$ se obtiene(18), de igual manera derivando

(18) con respecto al tiempo se tiene(19).

$$\frac{d}{dt} \left(\frac{dL}{d\dot{\eta}_{\mathbf{p}}} \right) - \frac{dL}{d\eta_{\mathbf{p}}} \quad (17)$$

$$\frac{dL}{d\dot{\eta}_{\mathbf{p}}} = \begin{bmatrix} \frac{dL}{d\dot{x}_p} \\ \frac{dL}{d\dot{y}_p} \\ \frac{dL}{d\dot{\psi}_p} \end{bmatrix} = \begin{bmatrix} m(\dot{x}_p - a \sin(\psi_p) \dot{\psi}_p) \\ m(\dot{y}_p + a \cos(\psi_p) \dot{\psi}_p) \\ m(\dot{\psi}_p + a^2 \dot{\psi}_p + a \dot{y}_p \cos(\psi_p) - a \dot{x}_p \sin(\psi_p)) \end{bmatrix} \quad (18)$$

$$\frac{d}{dt} \left(\frac{dL}{d\dot{\eta}_{\mathbf{p}}} \right) = \begin{bmatrix} m\ddot{x}_p - am \cos(\psi_p) \dot{\psi}_p^2 - am \sin(\psi_p) \ddot{\psi}_p \\ m\ddot{y}_p - am \sin(\psi_p) \dot{\psi}_p^2 + am \cos(\psi_p) \ddot{\psi}_p \\ m\ddot{\psi}_p (a^2 + 1) - m\dot{\psi}_p (a\dot{y}_p \sin(\psi_p) + a\dot{x}_p \cos(\psi_p)) + ma\ddot{y}_p \cos(\psi_p) - ma\ddot{x}_p \sin(\psi_p) \end{bmatrix} \quad (19)$$

Derivando (16) con respecto a η_p se obtiene (20), aplicando el método de Euler-Lagrange en (19) y(20), se puede encontrar las fuerzas y torques con respecto a η_p (21).

$$\frac{dL}{d\eta_p} = \begin{bmatrix} \frac{dL}{dx_p} \\ \frac{dL}{dy_p} \\ \frac{dL}{d\psi_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -am \cos(\psi_p) \dot{\psi}_p \dot{x}_p - am \sin(\psi_p) \dot{\psi}_p \dot{y}_p \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} f_{\dot{x}_p} \\ f_{\dot{y}_p} \\ \tau_{\dot{\psi}_p} \end{bmatrix} = \begin{bmatrix} m\ddot{x}_p - am \cos(\psi_p) \dot{\psi}_p^2 - am \sin(\psi_p) \ddot{\psi}_p \\ m\ddot{y}_p - am \sin(\psi_p) \dot{\psi}_p^2 + am \cos(\psi_p) \ddot{\psi}_p \\ m\ddot{\psi}_p (a^2 + 1) + ma\ddot{y}_p \cos(\psi_p) - ma\ddot{x}_p \sin(\psi_p) \end{bmatrix} \quad (21)$$

Reordenando (21) se puede representar como(22), finalmente el sistema en su forma matricial compacta se representa por(23).

$$\begin{bmatrix} f_{\dot{x}_p} \\ f_{\dot{y}_p} \\ \tau_{\dot{\psi}_p} \end{bmatrix} = \begin{bmatrix} m & 0 & -am \sin(\psi_p) \\ 0 & m & am \cos(\psi_p) \\ -am \sin(\psi_p) & am \cos(\psi_p) & m(a^2 + 1) \end{bmatrix} \begin{bmatrix} \ddot{x}_p \\ \ddot{y}_p \\ \ddot{\psi}_p \end{bmatrix} + \begin{bmatrix} 0 & 0 & -am \cos(\psi_p) \dot{\psi}_p \\ 0 & 0 & -am \sin(\psi_p) \dot{\psi}_p \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\psi}_p \end{bmatrix} \quad (22)$$

$$\mathbf{f}(\mathbf{t}) = \mathbf{M}(\eta_p) \ddot{\eta}_p(\mathbf{t}) + \mathbf{C}(\eta_p, \dot{\eta}_p) \dot{\eta}_p(\mathbf{t}) \quad (23)$$

De la representación(23), se debe denotar a \mathbf{M} como la matriz de inercia, \mathbf{C} la matriz de fuerzas centrípetas y centrífugas, \mathbf{f} como el vector de fuerzas y torque generados en la plataforma móvil tipo diferencial.

Para poder realizar la compensación dinámica, la cual útil para desarrollar los algoritmos de control, es necesario representar al modelo dinámico en función de la velocidad lineal y angular del robot diferencial, por lo

tanto es necesario encontrar $\ddot{\eta}_p$ en función de dichas velocidades para lo cual se procede a derivar a (12) y se obtiene (24), la cual se reemplaza en (23) y se obtiene (25).

$$\ddot{\eta}_p(t) = \dot{J}_p(\psi_p)v(t) + J_p(\psi_p)\dot{v}(t) \quad (24)$$

$$f(t) = M(\eta_p)\ddot{\eta}_p(t) + M(\eta_p)J_p(\psi_p)\dot{v}(t) + C(\eta_p, \dot{\eta}_p)J_p(\psi_p)v(t) \quad (25)$$

Posteriormente se debe encontrar transformación que relacione las fuerzas del robot móvil con respecto a los torques de los motores derecho e izquierdo (26), donde τ_d es el torque de la rueda derecha, τ_i es el torque de la rueda izquierda, R es la distancia entre las dos ruedas, r es el radio de las ruedas, su representación matricial sería (27).

$$\begin{bmatrix} f_{x_p} \\ f_{y_p} \\ \tau_{\psi_p} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\psi_p)}{r} & \frac{\cos(\psi_p)}{r} \\ \frac{\sin(\psi_p)}{r} & \frac{\sin(\psi_p)}{r} \\ \frac{R}{r} & -\frac{R}{r} \end{bmatrix} \begin{bmatrix} \tau_d \\ \tau_i \end{bmatrix} \quad (26)$$

$$f(t) = B(\psi_p)\tau(t) \quad (27)$$

De la misma manera se tiene matrices de transformación que relacionan los torques de cada motor con respecto a sus velocidades angulares y voltajes inducidos en cada motor (28), donde v_d y v_i son los voltajes inducidos al motor derecho e izquierdo, k_a es una constante de reducción, k_b es constante contraelectromotriz, R_a es la resistencia eléctrica del motor, ω_d y ω_i son las velocidades angulares.

$$\begin{bmatrix} \tau_d \\ \tau_i \end{bmatrix} = \begin{bmatrix} \frac{k_a}{R_a} & 0 \\ 0 & \frac{k_a}{R_a} \end{bmatrix} \begin{bmatrix} v_d \\ v_i \end{bmatrix} - \begin{bmatrix} \frac{k_a k_b}{R_a} & 0 \\ 0 & \frac{k_a k_b}{R_a} \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} \quad (28)$$

Las velocidades angulares de cada rueda deben ser representadas en función de las velocidades generales del movimiento del robot (29) y del mismo modo se realiza una consideración de los voltajes de cada motor con respecto a los voltajes de velocidad lineal y angular donde se llega a la siguiente matriz de transformación(30).

$$\begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{R}{2r} \\ \frac{1}{r} & \frac{R}{2r} \end{bmatrix} \begin{bmatrix} \mu \\ \dot{\psi}_p \end{bmatrix} \quad (29)$$

$$\begin{bmatrix} v_d \\ v_i \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} v_u \\ v_{\dot{\psi}_p} \end{bmatrix} \quad (30)$$

Remplazando (29) y (30) en (28) se obtiene las matrices de transformación de los torques de cada motor con respecto a las velocidades generales del robot y sus voltajes respectivamente (31) ,su representación compacta se presenta de la siguiente forma (32), donde $\mathbf{v}(\mathbf{t})$ son los voltajes de alimentación de las velocidades generales y \mathbf{D} , \mathbf{E} son matrices de constantes.

$$\begin{bmatrix} \tau_d \\ \tau_i \end{bmatrix} = \begin{bmatrix} \frac{k_a}{R_a} & \frac{k_a}{R_a} \\ \frac{k_a}{R_a} & -\frac{k_a}{R_a} \end{bmatrix} \begin{bmatrix} v_\mu \\ v_{\dot{\psi}_p} \end{bmatrix} - \begin{bmatrix} \frac{k_a k_b}{R_a r} & \frac{R k_a k_b}{R_a r^2} \\ \frac{k_a k_b}{R_a r} & -\frac{R k_a k_b}{R_a r^2} \end{bmatrix} \begin{bmatrix} \mu \\ \dot{\psi}_p \end{bmatrix} \quad (31)$$

$$\boldsymbol{\tau}(\mathbf{t}) = \mathbf{D}\mathbf{v}(\mathbf{t}) - \mathbf{E}\dot{\mathbf{v}}(\mathbf{t}) \quad (32)$$

Los voltajes generales del robot pueden ser descritos en función de las velocidades generales del movimiento, debido a que el robot móvil utilizado en este trabajo cuenta con controladores PD de bajo nivel en cada rueda encargado de cumplir las velocidades generales de referencia generadas por los controladores de trayectoria, estas matrices de transformación son (33), donde u_{ref} y $\dot{\psi}_{p,ref}$ son los valores de referencia de la velocidad lineal y angular, k_{pt} y k_{dt} son las constantes proporcional y derivativa del controlador con respecto a la velocidad lineal, k_{pr} y k_{dr} son las constantes proporcional y derivativa del controlador con respecto a la velocidad angular, la representación compacta de (33) se representa como (34), donde $v_{ref}(t)$ son las velocidades de referencia y L, S son matrices de constantes de ganancia del controlador PD (V. H. Andaluz et al., 2015; Varela-Aldas et al., 2018).

$$\begin{bmatrix} v_{\mu} \\ v_{\dot{\psi}_p} \end{bmatrix} = \begin{bmatrix} k_{pt} & 0 \\ 0 & k_{pr} \end{bmatrix} \begin{bmatrix} \mu_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} - \begin{bmatrix} k_{pt} & 0 \\ 0 & k_{pr} \end{bmatrix} \begin{bmatrix} \mu \\ \dot{\psi}_p \end{bmatrix} - \begin{bmatrix} k_{dt} & 0 \\ 0 & k_{dr} \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \ddot{\psi}_p \end{bmatrix} \quad (33)$$

$$v(t) = Lv_{ref}(t) - Lv(t) - S\dot{v}(t) \quad (34)$$

Una vez obtenidas las transformaciones (27), (32) y (34) se procede a agruparlas en una sola transformación matricial tal como (35), la cual puede ser utilizada en (25) para obtener (36), la cual es útil para obtener el modelo dinámico compacto en función de las velocidades de referencia y aceleraciones reales del sistema finalmente se reordena (36) para así obtener (37), donde su representación matricial es (38).

$$\mathbf{f}(t) = \mathbf{B}(\psi_p) \left[\mathbf{D} \left[\mathbf{L} \mathbf{v}_{\text{ref}}(t) - \mathbf{L} \mathbf{v}(t) - \mathbf{S} \dot{\mathbf{v}}(t) \right] - \mathbf{E} \mathbf{v}(t) \right] \quad (35)$$

$$\begin{aligned} \mathbf{B}(\psi_p) \left[\mathbf{D} \left[\mathbf{L} \mathbf{v}_{\text{ref}}(t) - \mathbf{L} \mathbf{v}(t) - \mathbf{S} \dot{\mathbf{v}}(t) \right] - \mathbf{E} \mathbf{v}(t) \right] &= \mathbf{M} \left(\boldsymbol{\eta}_p \right) \mathbf{J}_p(\psi_p) \mathbf{v}(t) \dots \\ \dots + \mathbf{M} \left(\boldsymbol{\eta}_p \right) \mathbf{J}_p(\psi_p) \dot{\mathbf{v}}(t) + \mathbf{C} \left(\boldsymbol{\eta}_p, \dot{\boldsymbol{\eta}}_p \right) \mathbf{J}_p(\psi_p) \mathbf{v}(t) \end{aligned} \quad (36)$$

$$\begin{aligned} \mathbf{v}_{\text{ref}}(t) &= \mathbf{L}^{-1} \left[\mathbf{D}^{-1} \left(\mathbf{J}'_p(\psi_p) \mathbf{B}(\psi_p) \right) \mathbf{J}'_p(\psi_p) \mathbf{M} \left(\boldsymbol{\eta}_p \right) \mathbf{J}_p(\psi_p) + \mathbf{S} \right] \dot{\mathbf{v}}(t) \dots \\ \dots + \mathbf{L}^{-1} \left[\begin{array}{l} \mathbf{D}^{-1} \left(\mathbf{J}'_p(\psi_p) \mathbf{B}(\psi_p) \right) \mathbf{J}'_p(\psi_p) \mathbf{M} \left(\boldsymbol{\eta}_p \right) \mathbf{J}_p(\psi_p) \dots \\ + \mathbf{D}^{-1} \left(\mathbf{J}'_p(\psi_p) \mathbf{B}(\psi_p) \right) \mathbf{J}'_p(\psi_p) \mathbf{C} \left(\boldsymbol{\eta}_p, \dot{\boldsymbol{\eta}}_p \right) \mathbf{J}_p(\psi_p) \cdot \\ + \mathbf{D}^{-1} \mathbf{E} + \mathbf{L} \end{array} \right] \mathbf{v}(t) \end{aligned} \quad (37)$$

$$\mathbf{v}_{\text{ref}}(t) = \bar{\mathbf{M}} \dot{\mathbf{v}}(t) + \bar{\mathbf{C}} \left(\dot{\boldsymbol{\eta}}_p \right) \mathbf{v}(t) \quad (38)$$

Donde:

$$\bar{\mathbf{M}} = \begin{bmatrix} \frac{2k_{dt}k_a + R_a m r}{2k_a k_{pt}} & 0 \\ 0 & \frac{4R_a m r a^2 + 2k_{dr}k_a R + R_a m r}{2k_a k_{pr} R} \end{bmatrix}$$

$$\bar{\mathbf{C}} \left(\dot{\boldsymbol{\eta}}_p \right) = \begin{bmatrix} \frac{k_{pt} + \frac{k_b}{r}}{k_{pt}} & \frac{-(k_a k_b R^2 - k_a k_b R + 4R_a a m \dot{\psi}_p r^2)}{4k_a k_{pt} r} \\ \frac{R_a a m r \dot{\psi}_p}{k_a k_{pr} R} & \frac{k_b R^2 + k_b R + 4k_{pr} r}{4k_{pr} r} \end{bmatrix}$$

Con el objetivo de obtener una representación más compacta del sistema, que posteriormente será utilizada en el proceso de identificación y

validación del sistema se procede a asignar constantes a los elementos de las matrices $\bar{\mathbf{M}}$ y $\bar{\mathbf{C}}$, quedan el sistema representado de la siguiente manera (39).

$$\begin{aligned}\chi_1 &= \frac{2k_{dt}k_a + R_a mr}{2k_a k_{pt}} & \chi_2 &= \frac{4R_a mra^2 + 2k_{dr}k_a R + R_a mr}{2k_a k_{pr}R} \\ \chi_3 &= \frac{k_{pt} + \frac{k_b}{r}}{k_{pt}} & \chi_4 &= \frac{-(k_a k_b R^2 - k_a k_b R)}{4k_a k_{pt}r} \\ \chi_5 &= \frac{-(4R_a amr^2)}{4k_a k_{pt}r} & \chi_6 &= \frac{R_a amr}{k_a k_{pr}R} \\ \chi_7 &= \frac{k_b R^2 + k_b R + 4k_{pr}r}{4k_{pr}r}\end{aligned}$$

$$\begin{bmatrix} u_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} = \begin{bmatrix} \chi_1 & 0 \\ 0 & \chi_2 \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \ddot{\psi}_p \end{bmatrix} + \begin{bmatrix} \chi_3 & \chi_4 + \chi_5 \dot{\psi}_p \\ \chi_6 \dot{\psi}_p & \chi_7 \end{bmatrix} \begin{bmatrix} \mu \\ \dot{\psi}_p \end{bmatrix} \quad (39)$$

4.3 Identificación de parámetros del modelo dinámico

La identificación permite determinar un modelo matemático para un sistema desconocido a través de los datos de entrada, los cuales son utilizados para excitar al sistema y así obtener la salida del mismo.

Existen varias técnicas al momento de realizar la identificación de sistemas, en este trabajo se utilizará el método de mínimos cuadrados por lo que la ecuación(39) debe ser representada en su forma lineal (40).

$$\begin{bmatrix} \dot{u} & 0 & u & \dot{\psi}_p & \dot{\psi}_p^2 & 0 & 0 \\ 0 & \ddot{\psi}_p & 0 & 0 & 0 & u\dot{\psi}_p & \dot{\psi}_p \end{bmatrix} \begin{bmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \\ \chi_5 \\ \chi_6 \\ \chi_7 \end{bmatrix} = \begin{bmatrix} \mu_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} \quad (40)$$

Una vez representado el sistema en su forma lineal se procede a expresarlo de manera matricial (41), donde \mathbf{X}_k es la matriz de velocidades y aceleraciones generales del robot móvil en cada instante de tiempo, χ es el vector de parámetros desconocidos y \mathbf{Y}_k es el vector de señales de referencia del sistema en cada instante de tiempo. El método de mínimos cuadrados presenta una estructura a minimizar siguiente (42), donde $\mathbf{E} = [\mathbf{Y} - \mathbf{X}\chi]$ es el vector de errores de todas las muestras tomadas (G. M. Andaluz et al., 2016a).

$$\mathbf{X}_k \chi = \mathbf{Y}_k \quad (41)$$

$$\mathbf{J}(\chi) = \mathbf{E}^T \mathbf{E} \quad (42)$$

Para poder determinar los valores de χ que minimicen la función coste se procede a encontrar su derivada con respecto a χ , operando quedaría de la siguiente manera:

$$\mathbf{J}(\chi) = (\mathbf{Y} - \mathbf{X}\chi)^T (\mathbf{Y} - \mathbf{X}\chi) \quad (43)$$

$$\mathbf{J}(\chi) = (\mathbf{Y}^T - \chi^T \mathbf{X}^T) (\mathbf{Y} - \mathbf{X}\chi) \quad (44)$$

$$\mathbf{J}(\boldsymbol{\chi}) = (\mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X} \boldsymbol{\chi} - \boldsymbol{\chi}^T \mathbf{X}^T \mathbf{Y} + \boldsymbol{\chi}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\chi}) \quad (45)$$

$$\frac{d\mathbf{J}}{d(\boldsymbol{\chi})} = (-\mathbf{Y}^T \mathbf{X} + \boldsymbol{\chi}^T \mathbf{X}^T \mathbf{X}) \quad (46)$$

Ya que se dispone de la derivada de la función coste, se procese a igualarla a cero ya que con esto se garantiza encontrar los valores de $\boldsymbol{\chi}$ que garanticen un mínimo global de dicha función(46), reordenando se tiene los valores óptimos (49).

$$(\mathbf{Y}^T \mathbf{X})^T = (\boldsymbol{\chi}^T \mathbf{X}^T \mathbf{X})^T \quad (47)$$

$$\mathbf{X}^T \mathbf{Y} = \mathbf{X}^T \mathbf{X} \boldsymbol{\chi} \quad (48)$$

$$\boldsymbol{\chi} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (49)$$

La calidad de la estimación depende de las señales de excitación de referencia, la identificación se realizó evaluando diversos tipos de señales, las cuales fueron obtenidas a partir de las propuestas en(50), donde A_i y ω_j son constantes.

Las señales de referencia son iguales salvo por sus ganancias lo cual es muy importante al momento de la identificación ya que el robot móvil se moverá en una trayectoria circular de radio A_1 / A_2 . El otro tipo de señal a evaluar tiene la forma de (104) la cual de igual forma útil por la mayor cantidad de armónicos presente en la misma (G. M. Andaluz et al., 2016a).

$$\begin{bmatrix} u_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} = \begin{bmatrix} A_1 \sin(\omega_1 t) + A_1 \sin(\omega_2 t) + A_1 \sin(\omega_3 t) + \dots + A_1 \sin(\omega_n t) \\ A_2 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) + A_2 \sin(\omega_3 t) + \dots + A_2 \sin(\omega_n t) \end{bmatrix} \quad (50)$$

$$\begin{bmatrix} u_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} = \begin{bmatrix} \sqrt{0.1^2 + (0.1\pi \cos(0.2\pi t))^2} \\ \frac{-0.15\pi^2 \sin(0.2\pi t)}{1 + \pi^2 \cos^2(0.2\pi t)} \end{bmatrix} \quad (51)$$

4.3.1 Identificación y validación experimental del modelo dinámico

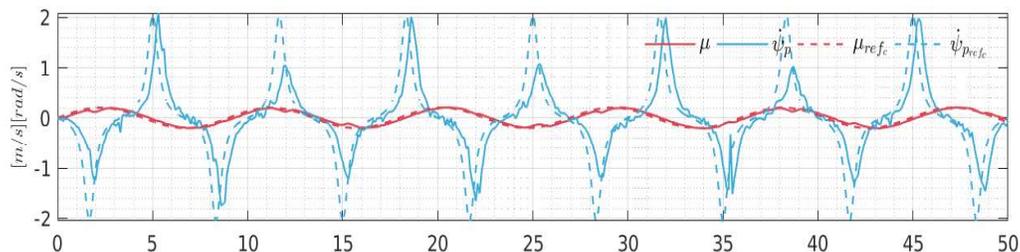
La identificación paramétrica se realiza para obtener el modelo dinámico, el cual debe ser validado para realizar la compensación dinámica. Las señales de entrada para la identificación se muestran en (50), los resultados de esta identificación se muestran en la Figura 51, los parámetros para el modelo dinámico del robot son:

$$\chi = [0.0903 \quad 0.0455 \quad 1.0078 \quad 0.0078 \quad 2.7e-06 \quad 3.4e-05 \quad 0.88880]^T$$

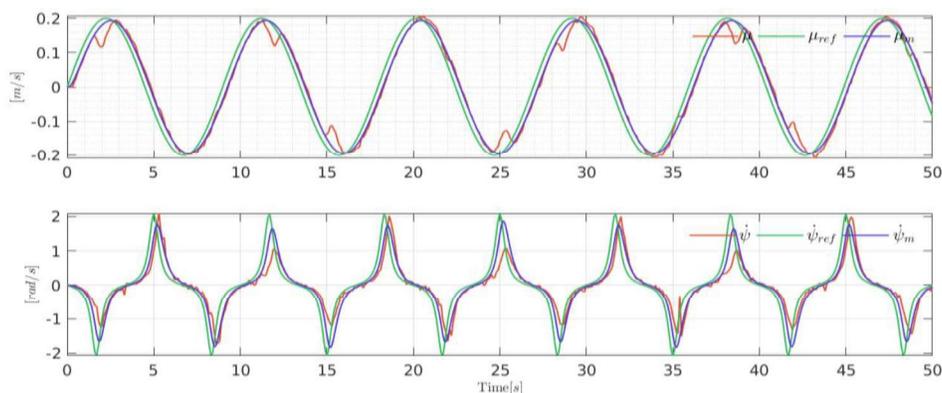
En la Figura 51 se muestra la señal de referencia de excitación y la señal obtenida mediante la identificación de parámetros.

Figura 51

Señales de identificación y modelo



En la Figura 52 se muestra los resultados de la validación, donde se nota que las señales de referencia del sistema no son exactamente iguales al modelo alcanzado mediante la identificación.

Figura 52**Validación del modelo dinámico**

Finalmente se puede concluir que el modelo dinámico del robot móvil representa de buena forma el comportamiento real del robot, lo cual se pudo observar en la Figura 52, donde μ , $\dot{\psi}_p$ son las señales del sistema real, μ_m , $\dot{\psi}_{p_m}$ son las señal del modelo dinámico, se puede notar que los datos reales y los simulados se aproximan entre sí.

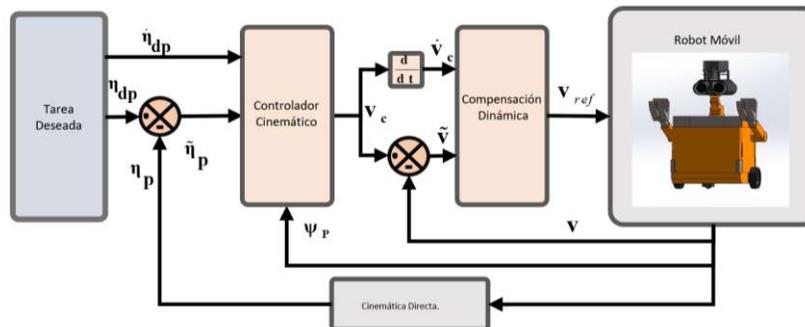
4.4 Algoritmos de control para seguimiento de trayectoria

4.4.1 Controlador proporcional basado en modelo cinemático.

Para desarrollar el controlador cinemático se parte del modelo del robot móvil (11), para este algoritmo se aplica el método del Jacobiano Inverso (52), donde $\mathbf{\eta}_{dp}(t) = [x_{dp} \quad y_{dp}]$ son las posiciones deseadas, $\dot{\mathbf{\eta}}_{dp}(t) = [\dot{x}_{dp} \quad \dot{y}_{dp}]$ son las velocidades de la trayectoria deseada y $\mathbf{v}_c(t) = [\mu_{ref,c} \quad \dot{\psi}_{p,ref,c}]^T$ son las velocidades de referencia generadas por el controlador, el esquema de este algoritmo se presenta en Figura 53.

Figura 53

Esquema control cinemático



$$v_c(t) = J_p(\psi_p)^{-1} \dot{\eta}_{dp}(t) \quad (52)$$

En el algoritmo de control se debe considerar los errores de posición para que pueda corregir errores de posición en las trayectorias previamente establecidas, por lo que se utiliza (53), donde $\tanh(\cdot)$ permite saturar los errores de posición, $\tilde{\eta}_p(t) = [\tilde{x}_p \quad \tilde{y}_p]$ son los errores de posición los cuales se obtienen a partir de $\tilde{\eta}_p(t) = [\eta_{dp}(t) - \eta_p(t)]$, K_1 es una matriz diagonal que permite cuantificar el error de posición del sistema esta matriz se la puede considerar como ganancia proporcional del sistema y K_2 es una matriz diagonal la cual se utiliza para delimitar las velocidades de referencia (Tzafestas, 2013).

$$v_c(t) = J_p(\psi_p)^{-1} \left(\dot{\eta}_{dp}(t) + K_2 \tanh \left(K_2^{-1} K_1 \tilde{\eta}_p(t) \right) \right) \quad (53)$$

4.4.2 Análisis de estabilidad

Para poder determinar la estabilidad de este controlador se utiliza herramientas de la teoría de control, una de estas herramientas se conoce como

Lyapunov, la cual es muy utilizada al momento de verificar la estabilidad de sistemas no lineales y controladores (Varela-Aldas et al., 2018).

Primeramente, se asume que tanto la velocidad de referencia del controlador tiende a ser igual a la velocidad del robot móvil $v(t) = v_c(t)$, por lo que se procede a reemplazaren (52), reordenando se tiene (54).

$$J_p(\psi_p)^{-1} \left(\dot{\mathbf{q}}_{dp}(t) + \mathbf{K}_2 \tanh \left(\mathbf{K}_2^{-1} \mathbf{K}_1 \tilde{\mathbf{q}}_p(t) \right) \right) = J_p(\psi_p)^{-1} \dot{\mathbf{q}}_p(t) \quad (54)$$

$$\dot{\mathbf{q}}_{dp}(t) - \dot{\mathbf{q}}_p(t) + \mathbf{K}_2 \tanh \left(\mathbf{K}_2^{-1} \mathbf{K}_1 \tilde{\mathbf{q}}_p(t) \right) = 0 \quad (55)$$

Se debe notar que en la ecuación (55) se presenta el error de velocidades de control la cual también se puede definir como

$\dot{\tilde{\mathbf{q}}}_p(t) = \left[\dot{\mathbf{q}}_{dp}(t) - \dot{\mathbf{q}}_p(t) \right]$ por lo tanto, la expresión final quedaría expresada de la siguiente manera (56).

$$\dot{\tilde{\mathbf{q}}}_p(t) = -\mathbf{K}_2 \tanh \left(\mathbf{K}_2^{-1} \mathbf{K}_1 \tilde{\mathbf{q}}_p(t) \right) \quad (56)$$

Para poder garantizar la estabilidad del algoritmo de control se procede a utilizar la herramienta de verificación de estabilidad de Lyapunov, por lo tanto, se propone una función candidata de la forma de errores cuadráticos la cual se representa como (57).

$$V(\tilde{\mathbf{q}}_p) = \frac{1}{2} \tilde{\mathbf{q}}_p(t)^T \tilde{\mathbf{q}}_p(t) \quad (57)$$

Se procede a derivar la función candidata de Lyapunov, por lo tanto, se tiene (58), la cual se debe reemplazar en (56) para encontrar la derivada en

función de los errores de la posición del sistema resultado de la siguiente manera (59).

$$\dot{V}\left(\tilde{\eta}_p\right)=\tilde{\eta}_p(t)^T \dot{\tilde{\eta}}_p(t) \quad (58)$$

$$\dot{V}\left(\tilde{\eta}_p\right)=-\tilde{\eta}_p(t)^T K_2 \tanh\left(K_2^{-1} K_1 \tilde{\eta}_p(t)\right) \quad (59)$$

El criterio de estabilidad de Lyapunov dice que $\dot{V}\left(\tilde{\eta}_p\right)$ debe ser definida negativa para garantizar la estabilidad del sistema, por lo tanto, en el algoritmo de control se debe considerar las siguientes restricciones: $K_1 > 0, K_2 > 0$ tal que $\tilde{\eta}_p(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

4.5 Controlador fuzzy adaptativo

La lógica difusa ha sido aplicada en varios campos desde teoría de control hasta la inteligencia artificial. En la lógica Booleana sólo pueden existir valores de cero o uno, es en este sentido donde la lógica difusa introduce el concepto de parcialmente verdadero, donde un valor verdadero se encuentra en un rango de completamente verdadero y completamente falso (J. Liu & Liu, 2018).

4.5.1 Diseño de un controlador difuso

El diseño de un controlador difuso se puede dividir en los siguientes pasos.

Fuzzificar una variable se pueden emplear conjuntos difusos, por ejemplo (60) y (61) donde NB es negativo grande, NS es negativo pequeño, ZO es cero, PS es positivo pequeño y PB es positivo grande.

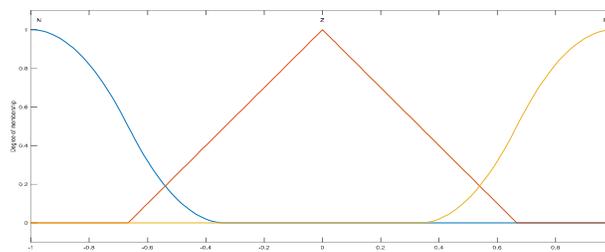
$$\mathbf{e}(\mathbf{t}) = \{NB, NS, ZO, PS, PB\} \quad (60)$$

$$\dot{\mathbf{e}}(\mathbf{t}) = \{NB, NM, NS, ZO, PS, PM, PB\} \quad (61)$$

Por ejemplo, como se puede observar en la Figura 54 se presenta un sistema difuso con 3 conjuntos en la entrada.

Figura 54

Fuzzificación de variables de entrada



Las reglas base consisten en un conjunto de reglas difusas, las cuales se formulan en relación al objetivo que se busque, los cuales pueden ser basados en objetivos de control o inteligencia artificial, las mismas que son realizadas por un experto en una determinada área.

Por ejemplo, en lógica difusa de control se puede diseñar reglas base como se muestra en (62), donde \mathbf{e} representa el error, $\dot{\mathbf{e}}$ es la razón de cambio

del error con respecto al tiempo, K_p es la ganancia del control proporcional y K_d es la ganancia del controlador derivativo.

$$\text{IF } e \text{ is N and } \dot{e} \text{ is N then } K_p \text{ is N and } K_d \text{ is Z} \quad (62)$$

Inferencia Difusa y defuzzificar donde partiendo de las reglas base, se puede obtener se puede obtener una matriz de relación difusa R , por lo tanto si se tiene nueva información del sistema A y B se puede calcular el nuevo resultado basándose en (63), la matriz de conclusión difusa C es un vector por lo tanto este necesita de algún método de defuzzificar para poder ser aplicado al mundo real, en este trabajo se utilizara el método del centroide.

$$C = (A \times B) \circ R \quad (63)$$

4.5.2 Controlador PD autoajustable con lógica difusa

El diseño de este controlador consta de dos partes la primera se basa en la cinemática inversa y la segunda en PD autoajustable con lógica difusa. Para el desarrollo de este algoritmo de control se debe considerar la estructura de un controlador proporcional derivativo aplicado al seguimiento de trayectorias de un robot móvil. La estructura de un controlador PD se presenta en (64), esta estructura del controlador puede se puede unir al modelo del robot móvil para así obtener directamente las velocidades generales de referencia (65), donde K_1 es la ganancia proporcional, K_3 es la ganancia derivativa, $\dot{\eta}_p(t)$ es la razón de cambio de los errores de control (Angelov & Xydeas, 2006; Tibaduiza, Amaya, Rodríguez, Mejía, & Flórez, 2011).

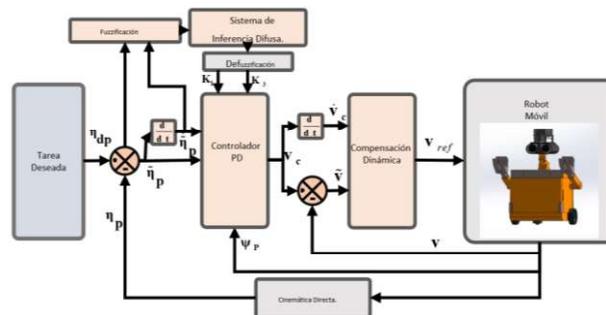
$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \quad (64)$$

$$\mathbf{v}_c(t) = \mathbf{J}_p(\boldsymbol{\psi}_p)^{-1} \left(\dot{\boldsymbol{\eta}}_{dp}(t) + \mathbf{K}_2 \tanh\left(\mathbf{K}_2^{-1} \mathbf{K}_1 \tilde{\boldsymbol{\eta}}_p(t)\right) + \mathbf{K}_2 \tanh\left(\mathbf{K}_2^{-1} \mathbf{K}_3 \dot{\tilde{\boldsymbol{\eta}}}_p(t)\right) \right) \quad (65)$$

Se puede notar que este algoritmo presenta una estructura muy similar al control basado en el Jacobiano Inverso, esto se debe a que algoritmo presenta un controlador proporcional basado en el modelo del sistema, el esquema de control se puede visualizar en la Figura 55.

Figura 55

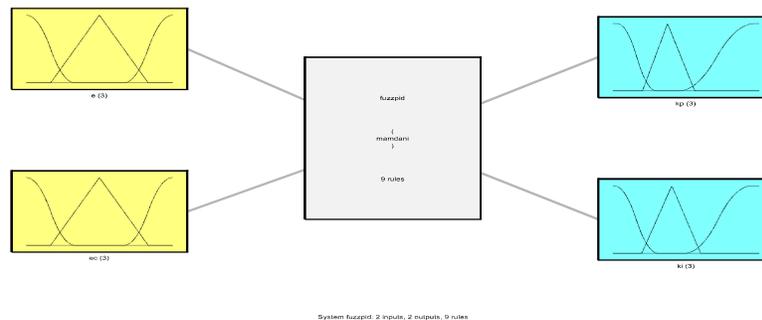
Esquema control PD autoajustable



En el controlador PD, el ajuste de las matrices \mathbf{K}_1 y \mathbf{K}_3 es un interrogante muy importante debido a que si se eligen de manera aleatoria es muy posible que el sistema no sea estable y presente oscilaciones indeseables. Para el diseño de este algoritmo de control adaptativo se considera inferencias difusas del tipo Mamdani como se muestra en la Figura 56, las entradas del controlador son $e(t)$ el cual es el error en el sistema de referencia que se elija, $ec(t)$ es la razón de cambio del error en un sistema de referencia.

Figura 56

Estructura del control difuso



El controlador difuso usa reglas para ajustar los parámetros del controlador PD, encontrando la relación difusa entre los las ganancias del controlador PD y los errores de control. Para el controlador PD adaptativo se elige 3 entradas difusas: (N) es negativo, (Z) es cero y (P) es positivo, las cuales son asignadas tanto al error y a la razón de cambio del error.

Se debe tomar en cuenta que e es un escalar y en el controlador a utilizar se tiene un vector de errores $\tilde{\mathbf{n}}_p(\mathbf{t}) = [\tilde{x}_p \quad \tilde{y}_p]$, por lo tanto, el controlador de adaptación difuso realizará la adaptación de manera independiente en cada eje respectivo, quedando las matrices \mathbf{K}_1 y \mathbf{K}_3 representadas de la siguiente manera:

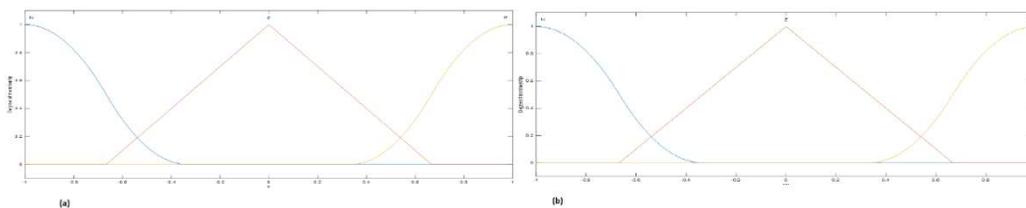
$$\mathbf{K}_1 = \begin{bmatrix} K_{p_x} & 0 \\ 0 & K_{p_y} \end{bmatrix}$$

$$\mathbf{K}_3 = \begin{bmatrix} K_{d_x} & 0 \\ 0 & K_{d_y} \end{bmatrix}$$

El control Pd adaptativo incluye dos entradas $e(t)$, $ec(t)$ y dos salidas ΔK_p , ΔK_d , las cuales son aplicadas para los valores de las ganancias del controlador, las funciones de membresía utilizados son triangulares y en sigmoides, los rangos de estas funciones $\in \{-1,1\}$ debido a $\tanh(\)$ del controlador. Las funciones de membresía de la entrada se describen en Figura 57 y las funciones de membresía de la salida se describen en la Figura 58.

Figura 57

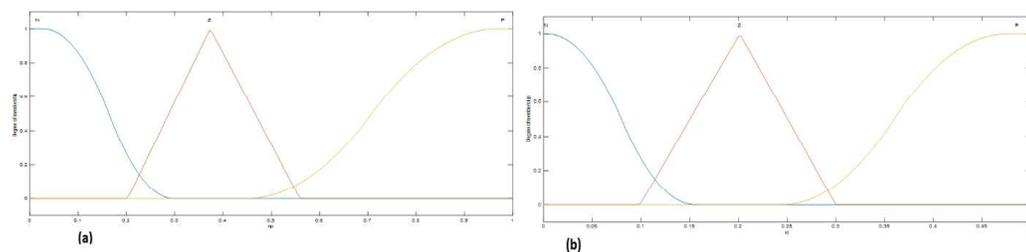
Funciones membresía entrada



Nota. El gráfico representa (a) membresía de entrada y (b) membresía de entrada

Figura 58

Funciones membresía salida

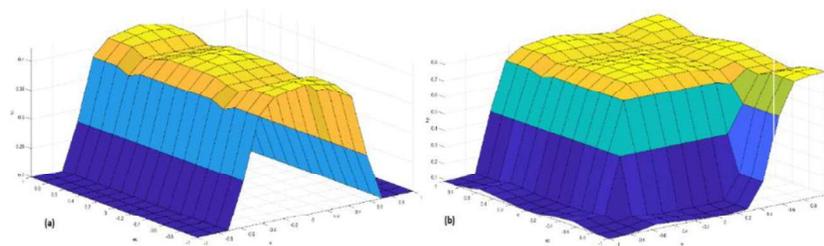


Nota. El gráfico representa (a) membresía de salida y membresía de salida

Una manera de verificar el funcionamiento del controlador autoajutable es visualizar la superficie de control, que en este caso no es tan suave como se desearía, pero para el control de trayectorias presentó un buen desempeño, estas superficies se presentan en la Figura 59.

Figura 59

Superficies de control



Nota. El gráfico representa (a) superficie generada para ΔK_p y (b) superficie generada para ΔK_i

4.5.3 Reglas difusas para autoajuste de K_p

Los principios para el auto ajuste de K_p son los siguientes: cuando $e(t)$ es positivo se debe incrementar K_p por lo tanto ΔK_p debe ser positiva, cuando $e(t)$ es negativa, se presenta un sobre impulso por lo que K_p debe disminuir por lo que ΔK_p debe ser negativo.

Cuando $e(t)$ se encuentra cerca de cero se toman en cuenta tres condiciones adicionales pero sobre la razón de cambio $ec(t)$, por lo que si $ec(t)$ es negativo el valor del sobre impulso tiende a ser mayor por lo que ΔK_p debería ser negativa, si $ec(t)$ se aproxima a cero para disminuir el error en

estado estable ΔK_p debería ser positiva, si $ec(t)$ es positiva el error tiende a incrementar por lo que ΔK_p se torna positiva, estas reglas se encuentran condensadas en Tabla 10.

Tabla 10

Reglas difusas para ajuste de K_p

$ec \setminus e$	N	Z	P
N	N	N	N
Z	N	P	P
P	P	P	P

Nota. Se considera (N) es negativo, (Z) es cero y (P) es positivo

4.5.4 Reglas difusas para autoajuste de K_d

Los principios para el auto ajuste de K_d se basan en cuando el error es muy pequeño es elije ΔK_d muy pequeña, por lo contrario, cuando el error sea muy grande ΔK_d será muy grande lo cual le permite al controlador el seguimiento de trayectorias que varíen con respecto al tiempo, estas reglas se muestran en Tabla 11. La forma en autoajustar K_p y K_d en línea es la siguiente (66).

$$K_p(k+1) = K_p(k) + \Delta K_p, K_d(k+1) = K_d(k) + \Delta K_d \quad (66)$$

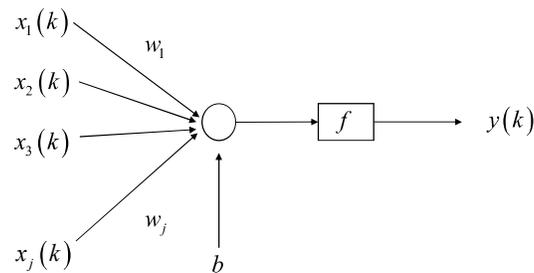
Tabla 11*Reglas difusas para ajuste de K_d*

$e \setminus e$	N	Z	P
N	Z	Z	Z
Z	P	P	P
P	Z	Z	Z

Nota. Se considera (*N*) es negativo, (*Z*) es cero y (*P*) es positivo

4.6 Controlador neuronal adaptativo

Las redes neuronales artificiales son modelos basados en procesos biológicos los cuales se encargan de procesar información, específicamente el sistema nervioso y su unidad básica conocida como neurona, las señales se propagan en forma de diferencias de potencial entre la entrada y salida de cada neurona, con las consideraciones previamente mencionadas se construye un modelo matemático de una neurona (67), donde w_j son los pesos que mapean la entrada de la neurona con su salida, b es conocido como umbral, $x_j(k)$ son las señales de entrada en el instante k , finalmente $y(k)$ es un escalar de la salida de la neurona la cual depende de la función de activación f que puede ser lineal o no lineal, su representación se encuentra en Figura 60. (J. Liu & Liu, 2018)

Figura 60*Representación neurona*

$$y(k) = f\left(\sum_{j=1}^n w_j x_j(k) + b\right) \quad (67)$$

Las funciones de activación son seleccionadas a partir de los distintos criterios de aplicación entre las más comunes se tiene (68),(69),(70)y (71), estas funciones de activación presentan una cualidad que su derivada puede ser fácilmente computada lo cual será muy útil al momento de realizar lo algoritmos de entrenamiento.

$$\text{Activación Lineal } y = x \quad (68)$$

$$\text{Sigmoide } y = \frac{1}{1 + e^{-x}} \quad (69)$$

$$\text{Sigmoide Simetrica } y = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (70)$$

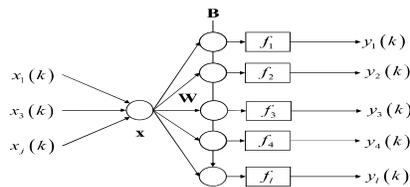
$$\text{Tangente Hiperbolica } y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (71)$$

La ecuación presentada en (67) solo es la representación de una sola neurona por lo que es necesario construir un modelo que represente un conjunto

de neuronas, lo cual es conocido como redes neuronales de una capa. Su representación se muestra en Figura 61 y su modelo se muestra en (72)

Figura 61

Red neuronal capa simple



$$y_l(k) = f \left(\sum_{j=1}^n w_{lj} x_j(k) + b_l \right) \quad (72)$$

La ecuación (72) puede ser escrita de forma más compacta en su forma matricial, la cual se define a continuación en (73).

$$\mathbf{y} = \mathbf{f}(\mathbf{W}\mathbf{X} + \mathbf{B})$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{L1} & w_{L2} & \dots & w_{Ln} \end{bmatrix} \quad (73)$$

$$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_L \end{bmatrix}$$

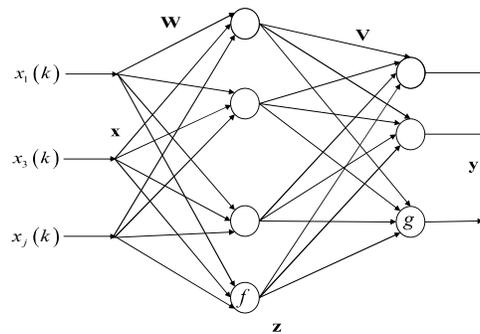
El vector de entrada se puede representar como $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ y el vector de salidas como $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_L]^T$.

Las configuraciones previamente mencionadas si bien presentan un buen desempeño, para aplicaciones más complejas es necesario introducir un mayor

número de capas, esto se representa en la Figura 62 en donde la primera capa se conoce con el nombre de capa de entrada, la última capa como de salida y el conjunto de neuronas que se encuentran en el centro se conoce con el nombre de capas ocultas.

Figura 62

Red neuronal multicapa



El modelo matemático que representa este comportamiento se describe en (74), (75) el cual de igual modo puede ser escrito en su forma matricial con la siguiente representación (76), (77).

$$z_l = f\left(\sum_{j=1}^n w_{lj}x_j\right) \quad l = 1, 2, \dots, L \quad (74)$$

$$y_i = g\left(\sum_{l=1}^L v_{il}z_l\right) \quad i = 1, 2, \dots, m \quad (75)$$

$$\mathbf{z} = \mathbf{f}(\mathbf{W}\mathbf{x}) \quad (76)$$

$$\mathbf{y} = \mathbf{g}(\mathbf{V}\mathbf{z}) \quad (77)$$

4.7 Aproximación de funciones redes neuronales artificiales.

El algoritmo de retro propagación utilizar se presenta en las Figura 63 y Figura 64 , la aproximación basada en este algoritmo se describe a continuación, para una mejor comprensión de este algoritmo se realiza de manera escalar para posteriormente tener su representación matricial y pueda ser aplicado a casos más generales.

Figura 63

Estructura de una red neuronal para aproximación

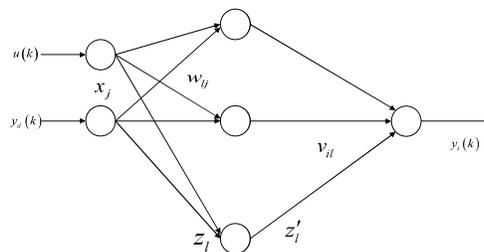
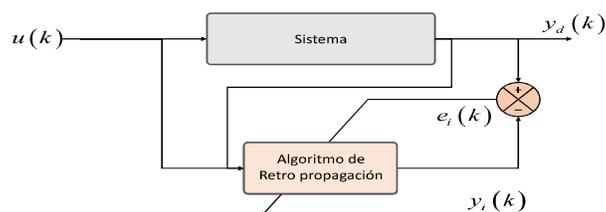


Figura 64

Esquema algoritmo de retro propagación



Propagación hacia delante, en esta sección del algoritmo se toma en cuenta las entradas del sistema (78).

$$z_l = \sum_{j=1}^n w_{lj} x_j \quad (78)$$

La salida final de esta sección se representa como (79) y de igual modo su derivada en función de z_l es (80)

$$z'_l = f(z_l) = \frac{1}{1 + e^{-z_l}} \quad (79)$$

$$\frac{\partial z'_l}{\partial z_l} = z'_l (1 - z'_l) \quad (80)$$

La salida de la capa oculta resulta en la siguiente ecuación (81) donde el valor de $i = 1$.

$$y_i = \sum_{l=1}^L v_{il} z'_l \quad (81)$$

Entonces el error de aproximación puede expresarse como (82) y la función a minimizar se diseña tal que (83).

$$e_i(k) = y_{di}(k) - y_i(k) \quad (82)$$

$$E = \frac{1}{2} e_i(k)^2 \quad (83)$$

Algoritmo de aprendizaje se toma en cuenta las consideraciones del algoritmo de descenso de gradiente, la adaptación de los valores de v_{il} son (84), (85) y (86).

$$\Delta v_{il} = -\eta \frac{\partial E}{\partial v_{il}} = \eta e_i(k) \frac{\partial y_i}{\partial v_{il}} = \eta e_i(k) z'_l \quad (84)$$

$$\Delta v_{il} = v_{il}(k+1) - v_{il}(k) \quad (85)$$

$$v_{il}(k+1) = v_{il}(k) + \eta e_i(k) z'_l \quad (86)$$

Aplicando los mismos conceptos se puede obtener el algoritmo de adaptación de w_{lj} el cual se muestra a continuación en (87) y (88)

$$\Delta w_{lj} = -\eta \frac{\partial E}{\partial w_{lj}} = \eta e_i(k) \frac{\partial y_i}{\partial w_{lj}} \quad (87)$$

$$\frac{\partial y_i}{\partial w_{lj}} = \frac{\partial y_i}{\partial z'_l} \frac{\partial z'_l}{\partial z_l} \frac{\partial z_l}{\partial w_{lj}} = v_{il} \frac{\partial z'_l}{\partial z_l} x_j = v_{il} z'_l (1 - z'_l) x_j \quad (88)$$

La adaptación de estos valores se muestra a continuación en (89) y (90).

$$\Delta w_{lj} = w_{lj}(k+1) - w_{lj}(k) \quad (89)$$

$$w_{lj}(k+1) = w_{lj}(k) + \eta e_i(k) v_{il} z'_l (1 - z'_l) x_j \quad (90)$$

Considerando los efectos del cambio de los pesos se puede escribir las ecuaciones (86) y (90) con los momentos, donde η es la tasa de aprendizaje y α es una tasa de momento, las cuales están representadas en (91) y (92)

$$v_{il}(k+1) = v_{il}(k) + \eta e_i(k) z'_l + \alpha (v_{il}(k) - v_{il}(k-1)) \quad (91)$$

$$w_{lj}(k+1) = w_{lj}(k) + \eta e_i(k) v_{il} z'_l (1 - z'_l) x_j + \alpha (w_{lj}(k) - w_{lj}(k-1)) \quad (92)$$

Usando este algoritmo de retro propagación es posible calcular el valor de Jacobiano de un sistema, el cual es una matriz de mapeo entre velocidades de espacio de configuración a velocidades del espacio de la tarea, a continuación, se presenta esta aproximación tomando en cuenta que el Jacobiano es un escalar (93).

$$\frac{\partial y_d(k)}{\partial u(k)} \approx \frac{\partial y_i(k)}{\partial u(k)} = \frac{\partial y_i}{\partial z'_l} \frac{\partial z'_l}{\partial z_l} \frac{\partial z_l}{\partial x_1} = \sum_l v_{il} z'_l (1 - z'_l) w_{1j} \quad (93)$$

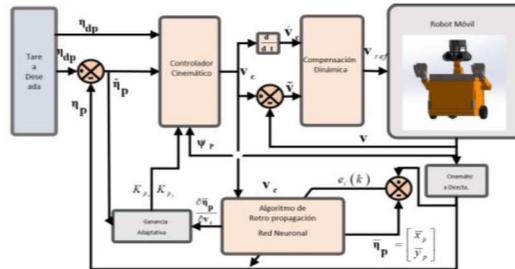
4.8 Controlador P autoajustable con en redes neuronales artificiales

En controlador basado en el modelo del robot que se dio en la ecuación (53), se puede notar que este algoritmo presenta una matriz de ganancias constantes \mathbf{K}_1 la cuales como se mencionó anteriormente es una matriz definida positiva que presenta la siguiente estructura (94), donde K_{p_x} es una constante proporcional a los errores en el eje x , K_{p_y} es una constante proporcional a los errores en el eje y , la desventaja que presenta este algoritmo es que las constantes deben ser elegidas con cuidado, si bien es cierto se puede dar un valor de uno de manera general presenta pequeños errores para distintas trayectorias. Por lo tanto, se propone un algoritmo basado en Jacobiano inverso con ganancias adaptativas en el cual dichas ganancias pueden variar de acuerdo a la trayectoria previamente definida, el esquema de este controlador se muestra en Figura 65 (Hernández-Alvarado, García-Valdovinos, Salgado-Jiménez, Gómez-Espinosa, & Fonseca-Navarro, 2016; Yang, Han, & Lin, 2017).

$$\mathbf{K}_1 = \begin{bmatrix} K_{p_x} & 0 \\ 0 & K_{p_y} \end{bmatrix} \quad (94)$$

Figura 65

Esquema control red neuronal



La ecuación (95) muestra la estructura matemática que se utilizara para adaptar las ganancias del controlador, la cual está sujeta a problemas de optimización, ya que se hace uso de una función costo a minimizar.

$$\mathbf{J} = \frac{1}{2} \tilde{\eta}_p^T \gamma \tilde{\eta}_p \quad (95)$$

$$\gamma = \begin{bmatrix} \gamma_x & 0 \\ 0 & \gamma_y \end{bmatrix}$$

Las ganancias del controlador se consideran parte de la función costo, este es optimizada de acuerdo al método de descenso del gradiente. Las ganancias del controlador se las puede representar como $\kappa = \begin{bmatrix} K_{p_x} & K_{p_y} \end{bmatrix}$ por lo tanto la derivada parcial (95) con respecto a κ se obtiene (96).

$$\frac{\partial \mathbf{J}}{\partial \kappa} = \tilde{\eta}_p^T \gamma \frac{\partial \tilde{\eta}_p}{\partial \kappa} \quad (96)$$

$$\frac{\partial \mathbf{J}}{\partial \kappa} = \tilde{\eta}_p^T \gamma \frac{\partial}{\partial \kappa} (\eta_{dp} - \eta_p) \quad (97)$$

$$\frac{\partial \mathbf{J}}{\partial \kappa} = \tilde{\eta}_p^T \gamma \frac{\partial \eta_p}{\partial \kappa} \quad (98)$$

$$\frac{\partial \eta_{\mathbf{p}}}{\partial \kappa} = \frac{\partial \eta_{\mathbf{p}}}{\partial \mathbf{v}_c} \frac{\partial \mathbf{v}_c}{\partial \kappa} \quad (99)$$

Se utiliza la regla de la cadena para poder obtener la derivada parcial del error con respecto a las ganancias del controlador(98), por lo tanto, la primera derivada de la ecuación (99) se puede definir como la matriz Jacobiana con en la cual se tiene como entradas las velocidades de control, Para el cálculo de esta matriz se utiliza la expresión encontrada en (100).

$$\frac{\partial \eta_{\mathbf{p}}}{\partial \mathbf{v}_c} = \begin{bmatrix} \frac{\partial x_p}{\partial \mu_{ref_c}} & \frac{\partial x_p}{\partial \dot{\psi}_{p,ref_c}} \\ \frac{\partial y_p}{\partial \mu_{ref_c}} & \frac{\partial y_p}{\partial \dot{\psi}_{p,ref_c}} \end{bmatrix} \quad (100)$$

La derivada del segundo término en la ecuación (99) se calcula a partir de la derivada parcial de la ley de control y las ganancias de este algoritmo resultado en (101)

$$\frac{\partial \mathbf{v}_c}{\partial \kappa} = \begin{bmatrix} \cos(\psi_p) \tilde{x}_p & \sin(\psi_p) \tilde{y}_p \\ -\frac{\sin(\psi_p)}{a} \tilde{x}_p & \frac{\cos(\psi_p)}{a} \tilde{y}_p \end{bmatrix} \quad (101)$$

Por lo tanto, combinando las derivadas parciales (100),(101) y reemplazándolas en (98) se obtiene (102).

$$\frac{\partial \mathbf{J}}{\partial \kappa} = \begin{bmatrix} \tilde{x}_p & \tilde{y}_p \end{bmatrix} \begin{bmatrix} \gamma_x & 0 \\ 0 & \gamma_y \end{bmatrix} \begin{bmatrix} \frac{\partial x_p}{\partial \mu_{ref_c}} & \frac{\partial x_p}{\partial \dot{\psi}_{p,ref_c}} \\ \frac{\partial y_p}{\partial \mu_{ref_c}} & \frac{\partial y_p}{\partial \dot{\psi}_{p,ref_c}} \end{bmatrix} \begin{bmatrix} \cos(\psi_p) \tilde{x}_p & \sin(\psi_p) \tilde{y}_p \\ -\frac{\sin(\psi_p)}{a} \tilde{x}_p & \frac{\cos(\psi_p)}{a} \tilde{y}_p \end{bmatrix} \quad (102)$$

Con lo anteriormente deducido lo que se busca es que las ganancias del controlador cambien y se adapten logrando que la función costo tienda a cero por lo tanto de acuerdo al algoritmo del descenso del gradiente el cual se

muestra a continuación representado en (103), finalmente la forma a implementar se muestra en (104).

$$\Delta \kappa = \begin{bmatrix} -\eta_{\kappa_x} & 0 \\ 0 & -\eta_{\kappa_y} \end{bmatrix} \frac{\partial \mathbf{J}^T}{\partial \kappa} \quad (103)$$

$$\kappa(k+1)^T = \kappa(k)^T - \begin{bmatrix} \eta_{\kappa_x} & 0 \\ 0 & \eta_{\kappa_y} \end{bmatrix} \frac{\partial \mathbf{J}^T}{\partial \kappa} \quad (104)$$

4.9 Controlador MPC

Es un tipo de controlador basado en el modelo del sistema con la particularidad que se realizan proyecciones hacia adelante en el tiempo, con el objetivo de minimizar los errores de control tomando en cuenta las predicciones de dicho modelo (Klančar et al., 2017; Kühne, Gomes, & Fetter, 2005). El modelo predictivo de control o conocido también como (MPC por sus siglas en inglés), es un algoritmo de control que usa el modelo de un sistema para optimizar su comportamiento en una ventana de predicciones, también conocido como horizonte deslizante (G. M. Andaluz et al., 2016b)

En esta sección se plantea un modelo matemático uniendo la dinámica y cinemática del robot móvil, esto se presenta a continuación en (105).

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\psi}_p \\ \dot{\mu} \\ \ddot{\psi}_p \end{bmatrix} = \begin{pmatrix} \mu \cos(\psi_p) - a \sin(\psi_p) \dot{\psi}_p \\ \mu \sin(\psi_p) + a \cos(\psi_p) \dot{\psi}_p \\ \dot{\psi}_p \\ -\frac{\mu \chi_3}{\chi_1} - \frac{\chi_4 \dot{\psi}_p}{\chi_1} - \frac{\chi_5 \dot{\psi}_p \dot{\psi}_p}{\chi_1} \\ -\frac{\chi_6 \dot{\psi}_p \mu}{\chi_2} - \frac{\chi_7 \dot{\psi}_p}{\chi_2} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\chi_1} & 0 \\ 0 & \frac{1}{\chi_2} \end{bmatrix} \begin{bmatrix} u_{ref} \\ \dot{\psi}_{p,ref} \end{bmatrix} \quad (105)$$

El modelo puede ser representado de forma compacta como se muestra en (106), donde $\mathbf{x}(t) = [x_p \quad y_p \quad \psi_p \quad \mu \quad \dot{\psi}_p]^T$ se le conoce como el vector de estados del sistema, finalmente se tiene que $\mathbf{v}(t) = [u_{ref} \quad \dot{\psi}_{p,ref}]^T$ es el vector de control del sistema.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{v}(t)) \quad (106)$$

Es necesario utilizar un método numérico que permita conocer la evolución del sistema, para esto se utilizara el método de la aproximación de Euler, por lo tanto, se tiene esta solución en la ecuación (107).

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T_o[\mathbf{f}(\mathbf{x}(k)) + \mathbf{g}(\mathbf{v}(k))] \quad (107)$$

Finalmente, para obtener la salida del sistema se utiliza una matriz \mathbf{C} , la cual permite realizar un mapeo entre todos los estados para así solo trabajar con las posiciones y orientación del robot móvil. Esto se puede apreciar de mejor forma en (108), con lo que se obtiene $\mathbf{y}(k) = [x_p(k) \quad y_p(k) \quad \psi_p(k)]^T$, el cual es el vector de salidas del sistema.

Con todas estas consideraciones se procede a implementar el control predictivo basado en el modelo.

$$\mathbf{y}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}(k) \quad (108)$$

Gran parte de los controladores MPC se basan en minimizar una función coste o costo que la mayor parte del tiempo se denota por \mathbf{J} , siendo esta una función basada en normas al cuadrado de los errores de control futuro más la

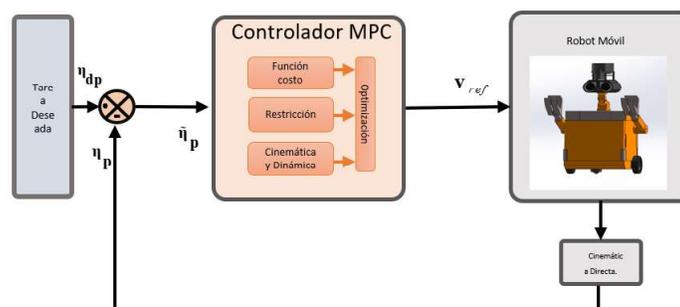
norma al cuadrado de los incrementos de la acción de control. Los errores de control se muestran en (109), donde i son las predicciones en el horizonte deslizando.

$$\mathbf{e}(k+i|k) = \mathbf{y}(k+i|k) - \mathbf{y}_d(k+i) \quad (109)$$

Tomando estas consideraciones se define una función costo (110), donde δ y λ son valores de penalización con el cual se selecciona el grado de importancia entre los criterios a minimizar, $\mathbf{y}_d(k+i)$ son los estados deseados proyectados en el horizonte deslizando, $\mathbf{y}(k+i|k)$ son los estados proyectados en el horizonte, el cual utiliza la información en el instante k para sí utilizar (105) y así obtener las proyecciones del sistema, N es el número de proyecciones que se tomarán en cuenta, \mathbf{Q}, \mathbf{R} son matrices definidas positivas, finalmente se tiene $\Delta \mathbf{v}(k+i|k) = \mathbf{v}(k+i|k) - \mathbf{v}(k+i-1|k)$ el cual es el incremento de las acciones de control, el esquema de control se puede apreciar en Figura 66.

Figura 66

Esquema control MPC



$$\mathbf{J} = \sum_{i=1}^N \delta \mathbf{e}(k+i|k)^T \mathbf{Q} \mathbf{e}(k+i|k) + \lambda \Delta \mathbf{v}(k+i|k)^T \mathbf{R} \Delta \mathbf{v}(k+i|k) \quad (110)$$

De esta manera se puede definir \mathbf{J} como una función que depende de las acciones de control del sistema, el objetivo principal del controlador es obtener las acciones de control futuras, de tal forma que las salidas predichas del sistema se encuentren lo más cerca posible de los estados deseados, esto se logra mediante la minimización de \mathbf{J} con respecto a las acciones de control.

Cuando en la función coste a minimizar se incluye en su modelo no linealidades, el MPC debe solucionar un problema de optimización no lineal en cada paso de muestreo, lo que puede ser implementado de dos formas distintas: en forma secuencial o paralela, en este trabajo se utilizó la forma secuencial y se usó librerías de Matlab de optimización (Recalde, Guevara, Cuzco, & Andaluz, 2020).

4.10 Compensación dinámica

Debido que el robot móvil debe ser capaz de transportar objetos de un peso máximo de 1 Kg se presentan efectos en la dinámica por lo que se generan errores de velocidad $\tilde{\mathbf{v}} = [\mathbf{v}_c - \mathbf{v}]$, el cual es la diferencia entre las velocidades del control de trayectorias y las velocidades reales del robot, para reducir este efecto se propone una compensación la cual se realiza usando el modelo dinámico del robot.

La ley de control a utilizar se indica en (111), donde \mathbf{K}_4 es una matriz diagonal constante proporcional a los errores de velocidad, \mathbf{K}_5 es una matriz diagonal que delimita las velocidades enviadas al robot móvil.

$$\mathbf{v}_{\text{ref}}(\mathbf{t}) = \bar{\mathbf{M}}\left(\dot{\mathbf{v}}_c(\mathbf{t}) + \mathbf{K}_4 \tanh(\mathbf{K}_4^{-1} \mathbf{K}_5 \tilde{\mathbf{v}}(\mathbf{t}))\right) + \bar{\mathbf{C}}\left(\dot{\mathbf{q}}_p\right) \mathbf{v}(\mathbf{t}) \quad (111)$$

Para poder determinar la estabilidad de la compensación dinámica se utiliza herramientas de la teoría de control, una de estas herramientas se conoce como Lyapunov.

Primeramente, se asume que tanto la velocidad de referencia del controlador tiende a ser igual a la velocidad del robot móvil $\mathbf{v}(\mathbf{t}) = \mathbf{v}_{\text{ref}}(\mathbf{t})$, por lo que se procede a reemplazaren (40) y se obtiene (112).

$$\bar{\mathbf{M}}\dot{\mathbf{v}}(\mathbf{t}) + \bar{\mathbf{C}}\left(\dot{\mathbf{q}}_p\right) \mathbf{v}(\mathbf{t}) = \bar{\mathbf{M}}\left(\dot{\mathbf{v}}_c(\mathbf{t}) + \mathbf{K}_4 \tanh(\mathbf{K}_4^{-1} \mathbf{K}_5 \tilde{\mathbf{v}}(\mathbf{t}))\right) + \bar{\mathbf{C}}\left(\dot{\mathbf{q}}_p\right) \mathbf{v}(\mathbf{t}) \quad (112)$$

Simplificando términos semejantes se llega a la siguiente expresión (113), esta operación se puede realizar si y solo si el modelo dinámico coincide con la dinámica real del sistema, por lo tanto, si se desea agregar más peso al sistema se debe realizar nuevamente la validación del modelo para que la compensación tenga un correcto funcionamiento.

$$\dot{\mathbf{v}}(\mathbf{t}) = \dot{\mathbf{v}}_c(\mathbf{t}) + \mathbf{K}_4 \tanh(\mathbf{K}_4^{-1} \mathbf{K}_5 \tilde{\mathbf{v}}(\mathbf{t})) \quad (113)$$

Se debe notar que en la ecuación (55) se presenta el error de aceleraciones de la cual también se puede definir como $\dot{\tilde{\mathbf{v}}}(\mathbf{t}) = [\dot{\mathbf{v}}_c(\mathbf{t}) - \dot{\mathbf{v}}(\mathbf{t})]$ por lo tanto, la expresión final quedaría expresada de la siguiente manera (114).

$$\dot{\tilde{\mathbf{v}}}(\mathbf{t}) = -\mathbf{K}_4 \tanh(\mathbf{K}_4^{-1} \mathbf{K}_5 \tilde{\mathbf{v}}(\mathbf{t})) \quad (114)$$

Para poder garantizar la estabilidad del algoritmo de control se procede a utilizar la herramienta de verificación de estabilidad de Lyapunov, por lo tanto, se

propone una función candidata de la forma de errores cuadráticos la cual se representa como (115) a la cual se aplica la derivada quedando finalmente (116).

$$V(\tilde{\mathbf{v}}) = \frac{1}{2} \tilde{\mathbf{v}}(t)^T \tilde{\mathbf{v}}(t) \quad (115)$$

$$\dot{V}(\tilde{\mathbf{v}}) = \tilde{\mathbf{v}}(t)^T \dot{\tilde{\mathbf{v}}}(t) \quad (116)$$

$$\dot{V}(\tilde{\mathbf{v}}) = -\tilde{\mathbf{v}}(t)^T \mathbf{K}_4 \tanh(\mathbf{K}_4^{-1} \mathbf{K}_5 \tilde{\mathbf{v}}(t)) \quad (117)$$

Finalmente reemplazando (114) en (116) se obtiene la función candidata en función a los errores de velocidades (117), por lo tanto el criterio de estabilidad de Lyapunov dice que $\dot{V}(\tilde{\mathbf{v}})$ debe ser definida negativa para garantizar la estabilidad del sistema, por lo tanto en nuestra algoritmo de control se debe considerar las siguientes restricciones: $\mathbf{K}_4 > 0, \mathbf{K}_5 > 0$ tal que $\tilde{\mathbf{v}}(t) \rightarrow 0$ cuando $t \rightarrow \infty$

Capítulo V

5. Pruebas y resultados

Este capítulo se centró en mostrar los resultados experimentales, obtenidos de la implementación de seguimiento de trayectoria en un robot móvil a través de los diferentes algoritmos desarrollados en este trabajo: Control Proporcional Basado en Modelo Cinemático, Controlador PD autoajustable basado en Lógica Difusa, Controlador P autoajustable basado en Redes Neuronales Artificiales, Controlador Predictivo basado en el Modelo y finalmente el Controlador para seguimiento de trayectorias con evasión de obstáculos.

5.1 Pruebas experimentales

Para las pruebas experimentales se utiliza el robot móvil desarrollado en este proyecto, el cual está conformado por: un microprocesador Raspberry pi 3 +, microcontrolador Open Cr, sensor Lidar HLS-LFCD2 y finalmente dos motores Dynamixel XL 430, el robot se puede apreciar en Figura 67.

Figura 67

Robot móvil Wall3



5.2 Control proporcional basado en modelo cinemático

Para la implementación de este controlador se consideran las ecuaciones (53) y (111) cuales generan las acciones del control proporcional y la compensación dinámica respectivamente, además se tomó en cuenta las siguientes consideraciones.

Condiciones iniciales

- $x_p(0) = -0.35[m]$
- $y_p(0) = -0.15[m]$

Trayectoria deseada

- $y_{dp} = 0.3 \cos(0.3t)[m]$
- $x_{dp} = 0.3 \cos(0.3t)[m]$

Constantes de experimentación

- Periodo de muestreo $T_o = 0.1[s]$
- Tiempo de experimentación $T_f = 20.0[s]$
- Carga de 1 Kg

Ganancias del controlador

- Ganancia proporcional $\mathbf{K}_1 = \text{diag}[1 \ 1]$
- Ganancia suavizado actuadores $\mathbf{K}_2 = \text{diag}[0.2 \ 0.2]$
- Ganancia interna compensación dinámica $\mathbf{K}_4 = \text{diag}[1 \ 1]$
- Ganancia interna compensación dinámica $\mathbf{K}_5 = \text{diag}[1 \ 1]$

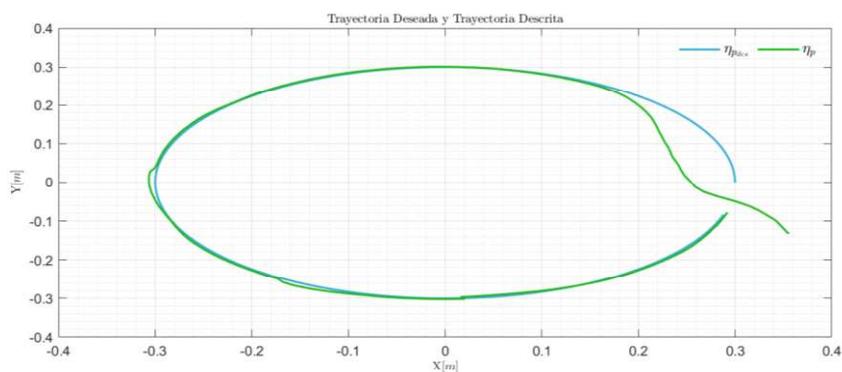
En Figura 68 se aprecia los resultados de este controlador, donde \mathbf{n}_{pdes}

es la trayectoria deseada del sistema, \mathbf{n}_p es la trayectoria generada por el robot

móvil, se aprecia que el robot móvil sigue correctamente la trayectoria de referencia.

Figura 68

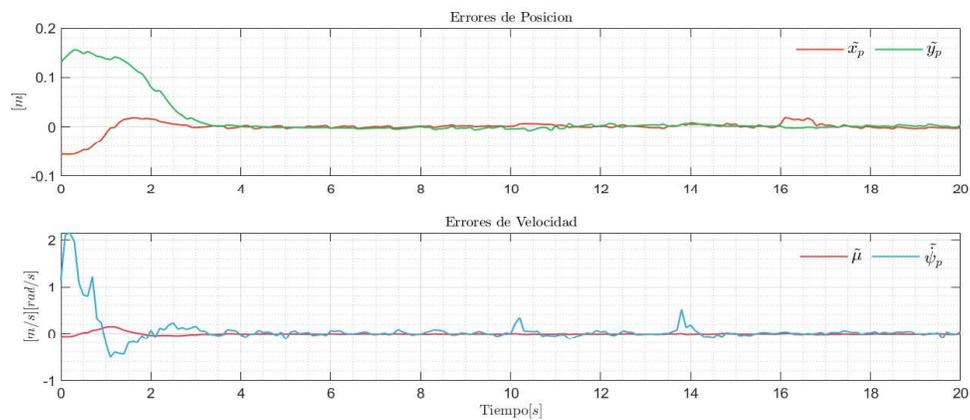
Trayectoria control proporcional basado en el modelo



En la Figura 69 se presentan los errores de seguimiento tanto en el eje x como en el eje y , donde se puede observar que tienden a cero y se posee estabilidad en el punto de equilibrio o trayectoria.

Figura 69

Errores de posición controlador proporcional



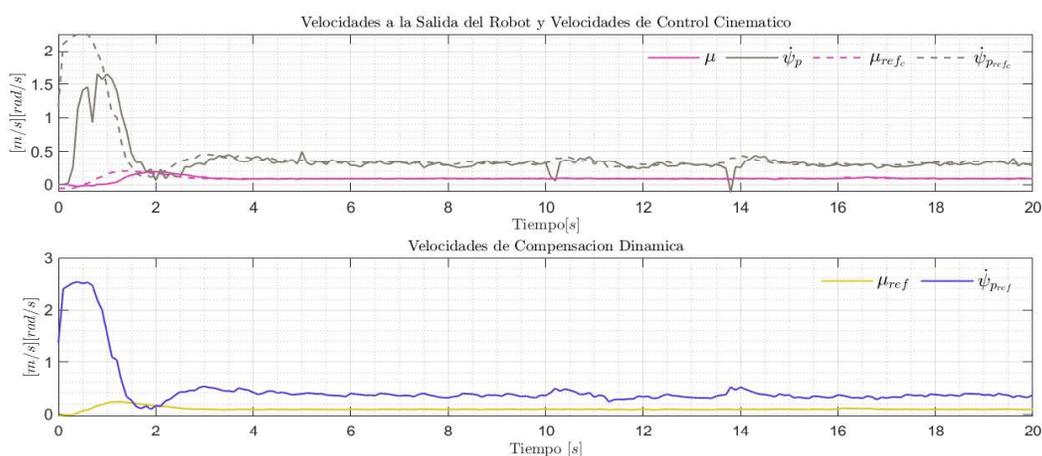
Además, en la Figura 69 se presenta los errores de velocidades

$\tilde{\mathbf{v}} = [\tilde{\mu} \quad \tilde{\psi}_p]$, los cuales se producen por la carga que transporta el robot

provocando cambios notorios en la dinámica del robot, se puede observar que tienden a cero.

Figura 70

Velocidades reales y de control en el proporcional



En Figura 70 se presentas las acciones del control proporcional

$\mathbf{v}_c = [\mu_{refc} \quad \dot{\psi}_{refc}]$ las cuales generan las velocidades que necesita el robot para

poder seguir una determinada trayectoria, $\mathbf{v} = [\mu \quad \dot{\psi}_p]$ son las velocidades

reales que se están produciendo en el robot, y finalmente para garantizar que

las velocidades reales tienden a las velocidades de control $\mathbf{v} \rightarrow \mathbf{v}_c$, se genera la

acción de compensación dinámica $\mathbf{v}_{ref} = [\mu_{ref} \quad \dot{\psi}_{ref}]$.

La compensación dinámica ayuda a los sistemas en donde se transporta cargas de peso considerable, mejorando el desempeño de los controladores de seguimiento de trayectorias, evasión de obstáculos entre otros.

5.3 Controlador PD autoajustable basado en lógica difusa

Para implementar este controlador se considera las ecuaciones (65), (66) y (111), las cuales generan las acciones de control proporcional derivativo, la adaptación de la respectivas, lo cual es una de las características principales de este controlador ya que actualiza sus constantes durante la experimentación, y la compensación dinámica respectivamente, además para este experimento se tomó en cuenta las siguientes consideraciones.

Condiciones iniciales

- $x_p(0) = 0.1[m]$
- $y_p(0) = 0.0[m]$

Trayectoria deseada

- $y_{dp} = 0.3 \cos(0.3t)[m]$
- $x_{dp} = 0.3 \cos(0.3t)[m]$

Constantes de experimentación

- Periodo de muestreo $T_o = 0.1[s]$
- Tiempo de experimentación $T_f = 20.0[s]$
- Carga de 1 Kg

Ganancias del controlador

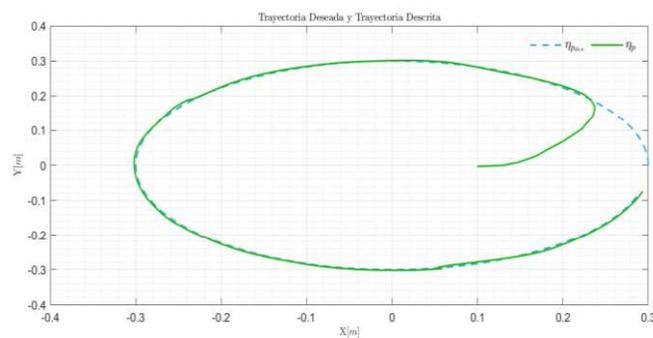
- $\mathbf{K}_4 = \text{diag}[1 \ 1]$
- $\mathbf{K}_5 = \text{diag}[1 \ 1]$

Es importante recalcar que las ganancias proporcional y derivativa se encontrara en cada instante de tiempo, es por eso que no se lo inicializa con un

valor. En Figura 71 se muestra los resultados de este controlador, donde η_{pdes} es la trayectoria deseada, η_p es el movimiento generado por el robot móvil, se aprecia que el robot móvil sigue correctamente la trayectoria de referencia.

Figura 71

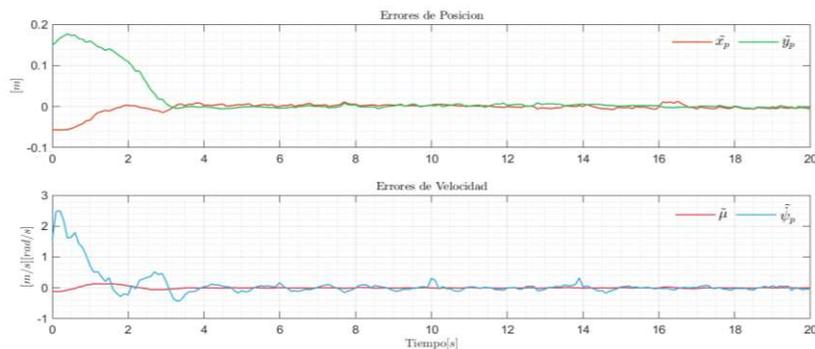
Trayectoria control PD autoajustable basado en lógica difusa



La Figura 72 muestra los errores de seguimiento $\tilde{\eta}_p(t) = \begin{bmatrix} \tilde{x}_p & \tilde{y}_p \end{bmatrix}$, donde se puede observar que tienden asintóticamente a cero tanto en el eje x como en el eje y, además presenta los errores de velocidad $\tilde{v} = \begin{bmatrix} \tilde{u} & \tilde{\psi}_p \end{bmatrix}$ generados durante el experimento, los cuales tienden a cero a pesar de que el robot móvil este transportando una carga de 1 Kg.

Figura 72

Errores de posición del PD autoajustable lógica difusa



La muestra Figura 73 las acciones de control tanto del controlador PD

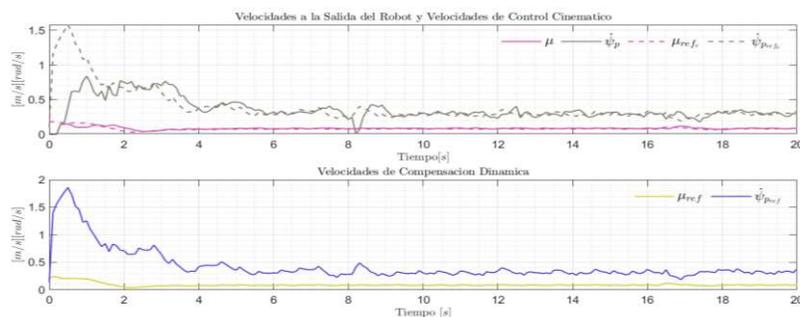
$\mathbf{v}_c = [\mu_{ref} \ \dot{\psi}_{ref}]$, como las de la compensación dinámica $\mathbf{v}_{ref} = [\mu_{ref} \ \dot{\psi}_{ref}]$, estas

últimas necesarias para cumplir que las velocidades reales del sistema

$\mathbf{v} = [\mu \ \dot{\psi}_p]$ tiendan a ser las del controlador PD.

Figura 73

Velocidades control PD basado en lógica difusa

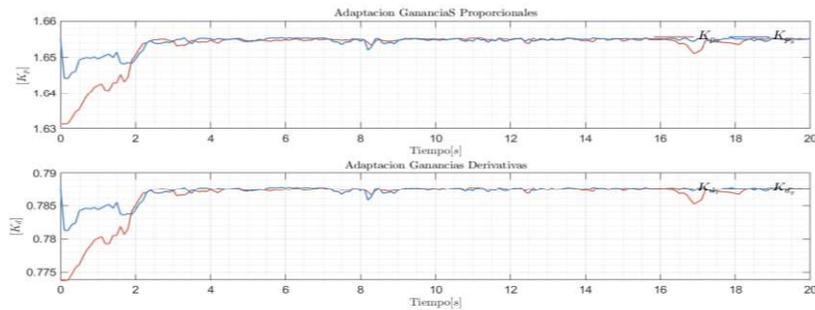


Finalmente, la Figura 74 en se muestra la evolución de las matrices

diagonales que son utilizadas como ganancias tanto para la parte proporcional como la derivativa.

Figura 74

Ganancias PD autoajustable basado en lógica difusa



5.4 Controlador P autoajustable basado en redes neuronales artificiales

Para la implementación de este controlador se tomó en cuenta las siguientes ecuaciones (53), (111), (102) y (104), con las cuales se tiene un controlador proporcional capaz de ajustar sus ganancias durante el experimento, esto es útil al momento de encontrar las ganancias adecuadas del sistema, además presenta su respectiva compensación dinámica, y la ecuación para el ajuste de las ganancias.

Para este controlador se tomó las siguientes consideraciones.

Condiciones iniciales

- $x_p(0) = 0.1[m]$
- $y_p(0) = 0.0[m]$

Trayectoria deseada

- $y_{dp} = 0.3 \cos(0.3t)[m]$
- $x_{dp} = 0.3 \cos(0.3t)[m]$

Constantes de experimentación

- Periodo de muestreo $T_o = 0.1[s]$
- Tiempo de experimentación $T_f = 20.0[s]$
- Carga de 1 Kg

Ganancias del controlador

- $\mathbf{K}_1(1) = \text{diag}[0.5 \ 0.5]$
- $\mathbf{K}_4 = \text{diag}[1 \ 1]$
- $\mathbf{K}_5 = \text{diag}[1 \ 1]$

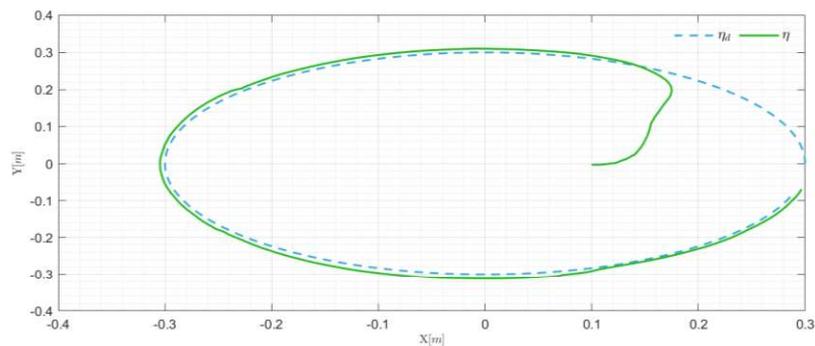
Numero neuronas internas

- 6 para estimación \bar{x}
- 6 para estimación \bar{y}
- 6 para estimación $\bar{\psi}_p$

La Figura 75 presenta los resultados de este controlador donde se puede notar que el robot móvil \mathbf{n}_p no sigue perfectamente la trayectoria deseada \mathbf{n}_{pdes} , pero si va mejorando con el tiempo esto se debe a que este tipo de controlador va modificando sus ganancias logrando que se ajusten en los valores óptimos.

Figura 75

Trayectoria control P autoajustable basado en redes neuronales

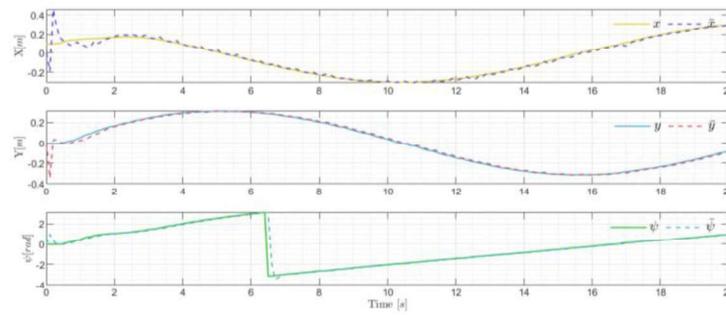


La Figura 76 muestra los resultados de las estimaciones de los estados internos del robot móvil, lo cual es útil al momento de emplear la adaptación de las ganancias, donde $[\bar{x} \ \bar{y} \ \bar{\psi}]$ son las estimaciones generadas por la red neuronal usando el algoritmo de retro propagación y $[x \ y \ \psi]$ son los estados reales del robot.

Para el algoritmo de retro propagación se utilizó el descenso de gradiente con lo cual esta red neuronal se entrena durante el experimento esto se detalla en las ecuaciones (91) y (92) .

Figura 76

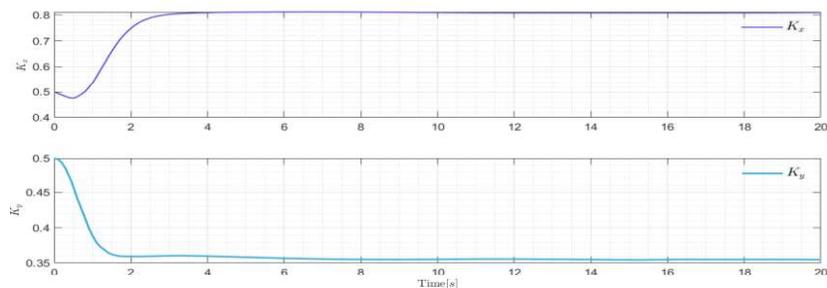
Estimación de estados internos



La Figura 77 muestra la evolución de las ganancias durante el experimento, $\mathbf{K}_1 = \text{diag}[K_x \ K_y]$ son las componentes de la matriz diagonal que actúa como ganancia proporcional del controlador, se puede observar que después de un cierto tiempo se mantienen constantes ya que en ese instante encuentra los valores óptimos.

Figura 77

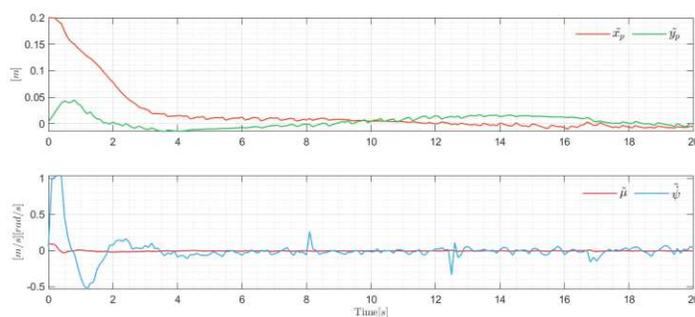
Ganancias control P autoajustable basado en redes neuronales



La Figura 78 muestra los errores de seguimiento $\tilde{\mathbf{q}}_p(t) = [\tilde{x}_p \quad \tilde{y}_p]$, donde se puede observar que tienden asintóticamente a cero, además presenta los errores de velocidad $\tilde{\mathbf{v}} = [\tilde{u} \quad \tilde{\psi}_p]$ generados durante el experimento, los cuales tienden a cero.

Figura 78

Errores de posición P autoajustable basado en redes neuronales



La muestra Figura 79 las acciones de control tanto del controlador

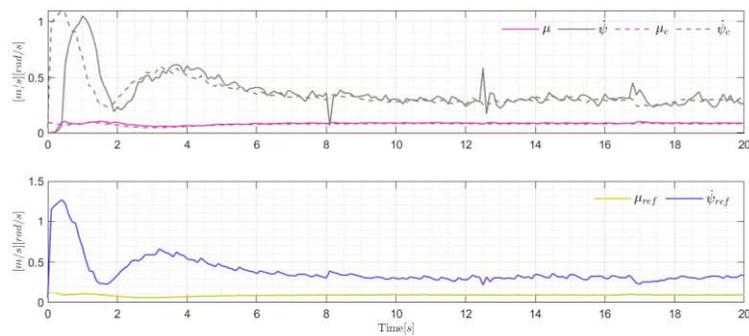
$\mathbf{v}_c = [\mu_{refc} \quad \dot{\psi}_{refc}]$, como las de la compensación dinámica $\mathbf{v}_{ref} = [\mu_{ref} \quad \dot{\psi}_{ref}]$, estas

últimas necesarias para cumplir que las velocidades reales del sistema

$\mathbf{v} = [\mu \quad \dot{\psi}_p]$ tiendan a ser las del controlador.

Figura 79

Velocidades reales P basado en redes neuronales artificiales



5.5 Controlador predictivo basado en el modelo MPC

Para la implementación de este controlador se utilizó la ecuación (110), para lo que se debe resolver el modelo del sistema aplicando N veces la ecuación (107), la cual necesita los estados actuales para poder realizar las predicciones. El algoritmo que se utilizó es de enfoque secuencial y se realizó a través de los paquetes de Matlab para optimización. A continuación, se presenta aspectos a tomar en cuenta.

Condiciones iniciales

- $x_p(0) = 0.1[m]$
- $y_p(0) = 0.0[m]$

Trayectoria deseada

- $y_{dp} = 0.3 \cos(0.3t)[m]$

- $x_{dp} = 0.3 \cos(0.3t)[m]$

Constantes de experimentación

- Periodo de muestreo $T_o = 0.1[s]$
- Tiempo de experimentación $T_f = 20.0[s]$
- Carga de 1 Kg
- $N = 10$

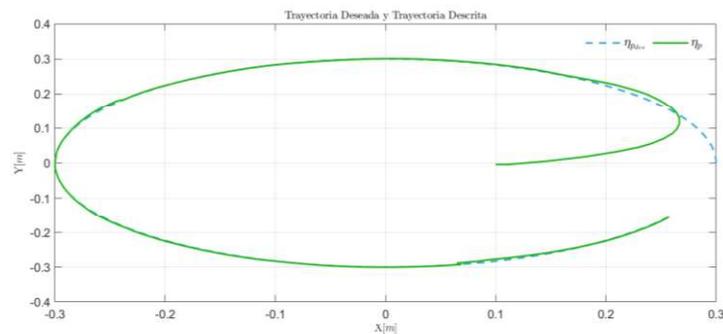
Ganancias del controlador

- $\mathbf{Q} = \text{diag}[1 \ 1]$
- $\mathbf{R} = \text{diag}[0.001 \ 0.001]$

La Figura 80 presenta los resultados de este controlador donde se puede notar que el robot móvil \mathbf{n}_p sigue perfectamente la trayectoria deseada \mathbf{n}_{pdes} , pero además presenta un movimiento muy suavizado lo cual se debe a (110) se considera minimizar error de posición y variaciones de acciones de control.

Figura 80

Trayectoria control MPC

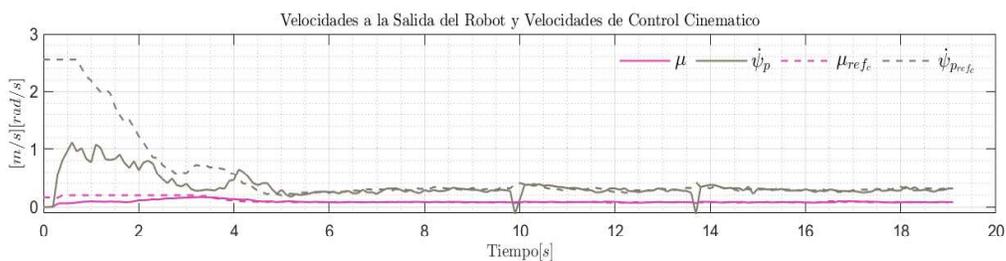


A continuación, se muestran las velocidades generadas por este controlador $\mathbf{v}_c = [\mu_{refc} \ \dot{\psi}_{refc}]$, además de presentar las velocidades reales del

sistema $\mathbf{v} = [\dot{\mu} \quad \dot{\psi}_p]^T$, se visualiza el suavizado en las acciones que genera dicho controlador.

Figura 81

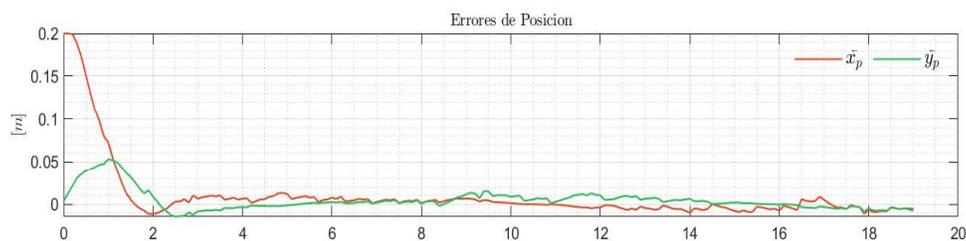
Velocidades de control y reales del controlador MPC



La Figura 82 presenta los errores de seguimiento con lo que se puede visualizar que tienden a cero con el paso del tiempo, verificando el funcionamiento del controlador.

Figura 82

Errores de posición del controlador MPC



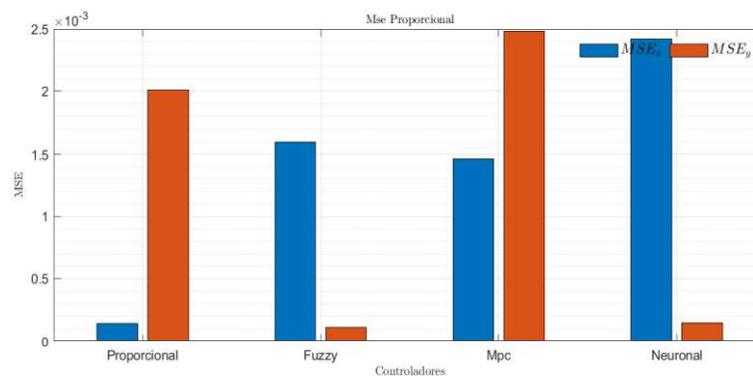
5.6 Comparación de controladores

En esta sección se presenta el error cuadrático medio analizado en cada uno de los controladores, dando como resultado la Figura 83, donde se visualiza que todos los controladores tienen un buen desempeño con errores de posición

menores a $0.0025[m]$, con lo que se puede visualizar que en el robot es proyecto es eficaz porque se cumplen con las trayectorias y es eficiente debido a que los errores de control son menores a $0.0025[m]$.

Figura 83

Comparativa controladores



5.7 Descripción de las personas de prueba

Estas pruebas de funcionamiento fueron realizadas con estudiantes de Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE-L, a través de las guías que ayudaran al estudiante a implementar y conocer diferentes algoritmos de control. Las pruebas se realizaron de forma remota a través de un simulador robótico.

5.8 Prueba de usabilidad

Para poder comprobar la usabilidad se dispuso a que los estudiantes de control digital interactúen con la interfaz y el robot Wall3. Luego de culminar la práctica con el robot se pudieron observar algunos detalles que permitió mejorar la interfaz, para que de esta manera exista una mejor experiencia del usuario.

5.9 Cuestionario de usabilidad

Mediante un cuestionario de usabilidad a los usuarios, permitirá denotar de una forma sencilla el grado de impacto de la interfaz y del robot, lo cual ayudará a tener mejoras en ciertos aspectos donde el usuario se haya sentido insatisfecho en su totalidad.

Este cuestionario de usabilidad fue revisado y validado por la docente tutora del proyecto, la calificación final de la interfaz y del robot están en función de los datos obtenidos de este cuestionario.

En la Tabla 12 muestra el promedio de los resultados del cuestionario de usabilidad aplicado a cada usuario. El sistema de puntuación de este cuestionario de usabilidad es de 5 puntos, desde: 1 (Completamente desacuerdo), hasta 5 (Completamente de acuerdo).

Tabla 12

Promedio de los datos obtenidos del cuestionario de usabilidad

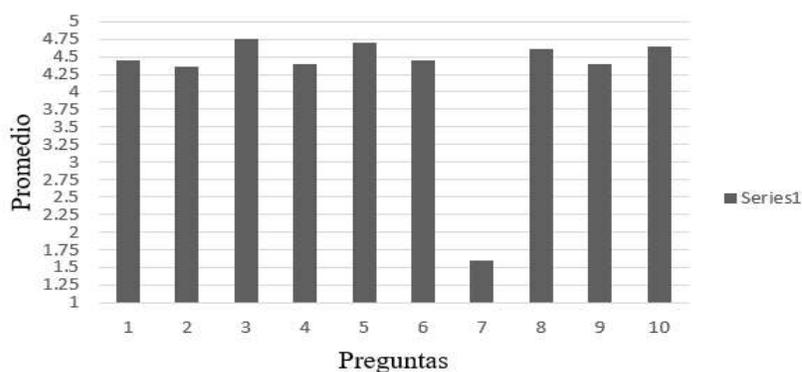
Preguntas	Promedio
Me gustaría utilizar este sistema con mayor frecuencia.	4.45
El sistema es muy fácil de utilizar.	4.35
Las gráficas de los resultados me ayudaron a comprender mejor el comportamiento del robot.	4.75
Las funciones de la interfaz se encuentran bien integradas.	4.4
El sistema permite aplicar diferentes técnicas de control.	4.7

Preguntas	Promedio
Es necesaria la ayuda del tutor para poder utilizar el sistema.	1.6
Los movimientos del robot se encuentran sincronizados correctamente con la interfaz.	4.6
Se puede aprender a utilizar el sistema rápidamente.	4.4
Reforcé mis conocimientos con el sistema.	4.65

Teniendo en cuenta los promedios obtenidos en la Tabla 12 en función de las preguntas del cuestionario de usabilidad aplicado a los usuarios, se puede realizar un análisis mediante un gráfico estadístico como se muestra en la Figura 84.

Figura 84

Promedio de los resultados del cuestionario de usabilidad



De acuerdo a la Figura 84 se puede analizar que 9 de las 10 preguntas tuvieron un promedio mayor a 4.25 por lo que se concluye que los usuarios tuvieron una buena experiencia de usabilidad del sistema, ayudándoles a

reforzar conceptos de los diferentes tipos de control que están presentes en el robot. La pregunta 7 (Es necesaria la ayuda del tutor para poder utilizar el sistema) obtuvo un promedio inferior a 1.75, lo que indica que no se necesita la ayuda de un tutor para poder utilizar el sistema.

Finalmente se puede analizar que las preguntas 3 (Las gráficas de los resultados me ayudaron a comprender mejor el comportamiento del robot) y 5 (El sistema permite aplicar diferentes técnicas de control) obtuvieron el mayor promedio, siendo superior a 4.5 con lo que se comprueba que la interfaz es óptima, permitiendo aplicar diferentes técnicas de control, y las gráficas de resultados son de gran utilidad para un análisis minucioso de cada usuario, facilitándoles la comprensión del comportamiento del robot.

5.10 Encuesta de satisfacción

Al aplicar una encuesta de satisfacción a los usuarios, se puede obtener una retroalimentación lo cual permite analizar sus experiencias con el sistema para posteriormente mejorar o modificar el sistema en caso de ser necesario.

Como se puede observar en la Tabla 13 se visualiza los promedios de las preguntas aplicadas a los usuarios, hay que tener en cuenta que la ponderación es de 5 puntos, desde: 1 (Muy insatisfecho o más baja), hasta 5 (Muy satisfecho o más alta).

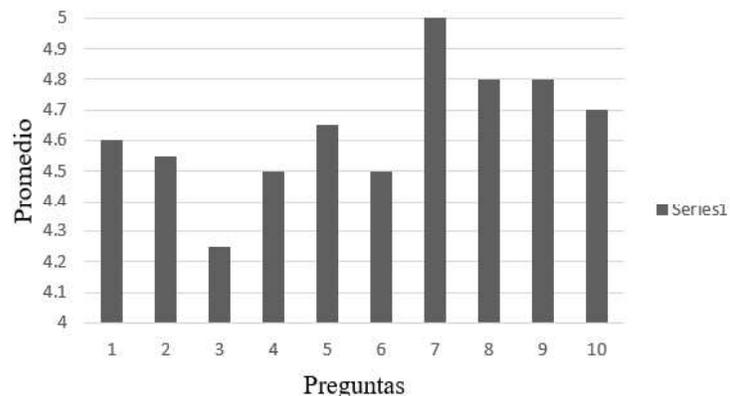
Tabla 13*Promedio de los datos obtenidos de la encuesta de satisfacción*

Preguntas	Promedio
Calidad del robot Wall3.	4.6
Experiencia de uso.	4.55
Servicio de ayuda a través de la interfaz.	4.25
Gráficas de resultados de la interfaz.	4.5
Facilidad de mantenimiento.	4.65
Interacción con la interfaz.	4.5
Recomendarías el sistema a otro usuario.	5
Estaría interesado en utilizar nuevamente el sistema.	4.8
Correcto funcionamiento en el movimiento del robot.	4.8
Información del manual de usuario.	4.7

Teniendo en cuenta los datos proporcionados en la Tabla 13 se puede realizar un análisis mediante un gráfico estadístico como se muestra en la Figura 2.

Figura 85

Promedio de los resultados de la encuesta de satisfacción



De la Figura 85 se puede analizar que 9 de las 10 preguntas tuvieron un promedio mayor a 4.4 por lo tanto, permite determinar que la experiencia de los usuarios en general ha sido satisfactoria, teniendo en cuenta el correcto funcionamiento del robot, la interacción de la interfaz y la información proporcionada por la misma, la facilidad de mantenimiento explicado y la información proporcionada en el manual de usuario.

La pregunta 7 (Recomendarías el sistema a otro usuario) es la que tiene el mayor promedio siendo este de 5 puntos, con lo que se puede concluir que los usuarios se sienten completamente satisfechos llegando al grado de recomendárselo a otros usuarios, puesto que es un sistema innovador e inexistente en el laboratorio de Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE-L. También se puede apreciar que las preguntas 8 (Estaría interesado en utilizar nuevamente el sistema) y 9 (Correcto funcionamiento en el movimiento del robot) tienen un promedio mayor a 4.7,

pudiendo analizar que el sistema se comporta correctamente lo cual implica el deseo de los usuarios por volver a utilizar el sistema.

Finalmente se puede notar que la pregunta 3 (Servicio de ayuda a través de la interfaz) es la de menor promedio siendo este menor a 4.3 por lo que se concluye que los usuarios sugieren una mayor ayuda en los cuadros de dialogo de la interfaz.

5.11 Hipótesis

5.11.1 Obtención de la muestra

Para poder validar el prototipo, se tendrá como población general a los estudiantes de ingeniería mecatrónica del paralelo de control digital de 28 estudiantes, de donde se obtiene un tamaño de población de 28 personas para calcular el tamaño de la muestra se debe emplear la ecuación (118).

$$n = \frac{NZ_a^2 pq}{d^2(N-1) + Z_a^2 pq} \quad (118)$$

Donde:

- N = Tamaño de la población.
- Z_a = Nivel de confianza.
- p = Probabilidad de éxito
- q = Probabilidad de fracaso
- d = Error máximo admisible.

Para determinar el tamaño de la muestra se utiliza un nivel de confianza del 95% ($Z_a=1.96$), y una probabilidad de éxito y fracaso de $p=0.5$ y $q= 0.5$.

$$n = \frac{(21)(1.96)^2(0.5)(0.5)}{(0.05)^2(19) + (1.96)^2(0.5)(0.5)} = 20.01 \quad (119)$$

5.11.2 Validación de la hipótesis

Hipótesis propuestas:

¿El diseño y construcción de un robot móvil tipo diferencial didáctico permitirá la aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el Laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Sede Latacunga?

Para empezar la validación de la hipótesis se debe analizar la variable dependiente e independiente, estas variables se presentan en Variables de la investigación.

Para poder obtener la validación de la hipótesis, se realizará una evaluación sobre las técnicas de control que conocen antes de utilizar al robot y otra después de su utilización como se puede observar en la Tabla 14

Tabla 14

Notas de las pruebas realizadas

N	Nota antes del uso del robot	Nota despues del uso del robot
1	3	6
2	5	10
3	3	8
4	5	10

N	Nota antes del uso del robot	Nota despues del uso del robot
6	4	8
7	3	7
8	5	9
9	4	8
10	5	9
11	3	7
12	5	9
13	4	8
14	4	8
15	3	7
16	6	9
17	4	7
18	4	9
19	3	7
20	5	8

Para poder validar la hipótesis se utilizará la prueba de T-Student la cual se usa cuando se trata de buscar que la hipótesis nula sea cierta para lo cual se planteara dos hipótesis, una nula y otra alternativa, estas son H_0 y H_1 respectivamente.

- H_0 : La utilización del robot móvil permitió aplicar varias técnicas de control tanto tradicionales e inteligentes, afianzando los conocimientos de los usuarios de forma didáctica y práctica.

- H_1 : La utilización del robot móvil no permitió aplicar varias técnicas de control tanto tradicionales e inteligentes, el robot no tuvo impacto en los conocimientos de los usuarios.

Continuando con el análisis se debe comprobar que la muestra este normalizada, para esto se empleara el test de Shapiro Wilk, donde todos los cálculos están desarrollados en Excel y los resultados se pueden observar en la Tabla 15.

Tabla 15

Parámetros de Test Shapiro-Wilk

	Antes de utilizar el robot	Después de utilizar el robot
<i>W</i>	0.8654	0.9257
<i>p-value</i>	0.01	0.1
α	0.05	0.05

Donde:

- *W*: Es el valor de Shapiro Wilk.
- *p-value*: Representa al valor de Shapiro Wilk linealizado.
- α : Es la significancia

Para poder comprobar si la muestra esta normalizada, el valor *p-value* debe ser mayor a 0.05, en este caso se tiene que $0.1 > 0.05$, validando que la muestra tiende a ser normal.

A continuación, se realiza el cálculo de las variables para poder comprobarla por la T-Student como se puede observar en la Tabla 16.

Tabla 16*Valores de análisis T-Student*

n_1	n_2	s_1^2	s_2^2	\bar{X}_1	\bar{X}_2	$\mu_1 - \mu_2$
20	20	0.8315	1.1473	4.1	8.1	0

Donde:

- n_1 : Es el número de pruebas realizadas antes de utilizar el robot.
- n_2 : Es el número de pruebas realizadas después de utilizar el robot.
- X_1 : Es el promedio de las calificaciones obtenidas por los estudiantes antes de utilizar el robot.
- X_2 : Es el promedio de las calificaciones obtenidas por los estudiantes después de utilizar el robot.
- S_1^2 : Es la varianza de las calificaciones obtenidas por los estudiantes antes de utilizar el robot.
- S_2^2 : Es la varianza de las calificaciones obtenidas por los estudiantes después de utilizar el robot.

A continuación, se procede a calcular la varianza combinada para las dos muestras se aplica la ecuación (120)

$$S_c^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{(n_1 - 1) + (n_2 - 1)} = 0.9894 \quad (120)$$

Para calcular el estadístico de prueba se aplica la ecuación (121).

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{s\sqrt{n_1^{-1} + n_2^{-1}}} = -1.2716 \quad (121)$$

Para calcular lo grados de libertad se utiliza la ecuación (122)

$$G_l = n_1 + n_2 - 1 = 38$$

(122)

Una vez obtenidos estos datos se procede a obtener los valores de T-Student de acuerdo con los grados de libertad como se puede observar en la Tabla 17.

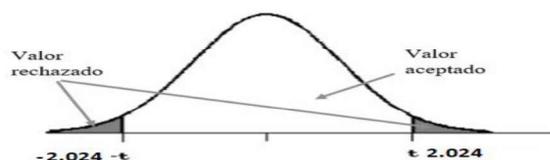
Tabla 17

Parámetros para el análisis T-Student

	Antes de utilizar el robot	Después de utilizar el robot
Media	4.1	8.1
Varianza	0.8315	1.1473
Estadístico t	-1.2716	
p-valor	0.2112	
Valor critico	2.02439	
Alpha	0.05	

Acorde a los valores que se encuentran en la Tabla 17 se observa que el valor critico es de 2.02439 a ambas colas de la distribución, el valor de estadístico de prueba (t) se encuentra dentro del área de aceptación de H_0 , comprobando que la hipótesis planteada es verdadera, teniendo una fiabilidad del 95%, es decir, se está 95% seguro de que la utilización del robot móvil permitió aplicar varias técnicas de control afianzando los conocimientos de los usuarios.

Como se observa en la Figura 86 el valor de $t = -1.2716$ se encuentra dentro de la zona de aceptación.

Figura 86*Análisis T de Student***5.12 Costos**

Para la elaboración de este proyecto se adquirieron varios componentes existentes en el mercado internacional y nacional, también se requirió fabricar las piezas diseñadas por los autores, en la siguiente Tabla 18 se detallan los gastos realizados.

Tabla 18*Costos del Proyecto*

Componente	Cantidad	Costo por unidad (\$)	Total (\$)
Raspberry Pi 3	1	75	75
Modulo USB	1	15	15
Tarjeta Open CR	1	80	80
Batería Lipo 11.1V 1800mA	1	40	40
Motor Dynamixel XL430	2	115	230
Sensor Lidar	1	50	50
Cables Conexión	1	5	5
Rueda loca	1	2	2
Impresión 3D	1	199.41	199.41
Tornillos M3x45mm	4	0.05	0.20
Tornillos M3x8mm	10	0.05	0.50

Componente	Cantidad	Costo por unidad (\$)	Total (\$)
Tornillos M4x60mm	4	0.10	0.40
Tornillos M2.5x15mm	4	0.05	0.20
Tornillos M2.5x8mm	12	0.05	0.60
Ruedas del robot	2	4	8
Tuercas M3	46	0.05	2.30
Tuercas M4	4	0.10	0.40
			710.61

En este costo total de 710.61 USD (dólares estadounidenses) no se toma en cuenta el costo del diseño, del ensamble y de la programación puesto que fueron realizadas por los autores.

Capítulo VI

6. Conclusiones y recomendaciones

6.1 Conclusiones

Se diseño y construyo un robot móvil tipo diferencial didáctico, el cual permite la aplicación de técnicas de control basado en algoritmos tradicionales e inteligentes en el Laboratorio de la carrera Ingeniería Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Sede Latacunga.

Se recopiló la información correspondiente sobre robots móviles, tanto para el desarrollo del robot como para la modelación matemática usada en la implementación de los algoritmos de control comúnmente usados en este tipo de sistema.

Se diseño la estructura del robot móvil enfocándose en que pueda trasportar una carga de 1 Kg, con lo cual se pudo conocer que material usar en la estructura y el torque necesario para los motores, tomando en cuenta que un módulo didáctico.

Se desarrollo el sistema de comunicación de forma que el robot móvil pueda enviar y recibir sus estados internos y los comandos de velocidad por WLAN, mientras en un computador se realizan los cálculos necesarios de cada algoritmo de control. Lo que es útil ya que el robot se puede utilizar de forma modular.

Se desarrollo una interfaz de usuario intuitiva que permita el aprendizaje de control usando el robot móvil diferencial como sistema de pruebas.

Se desarrollo un simulador robótico, donde se incluyó al robot realizado en este proyecto, donde se pueden realizar pruebas antes de utilizar el robot real, tanto el simulador como el robot real disponen el mismo sistema de comunicación.

Se desarrollo los algoritmos de control tradicionales usando el modelo matemático cinemático y dinámico del robot y para los algoritmos de control inteligentes se utilizó lógica difusa y redes neuronales artificiales, todos los algoritmos presentan un error cuadrático medio menor al 0.0025 m lo cual verifica que se puede aplicar una amplia gama de controladores.

Como resultados de las pruebas realizadas en estudiantes del sexto nivel de la carrera de Ingeniera mecatrónica mediante un módulo didáctico y una interfaz permitió mejorar notablemente el aprendizaje de ciertos conceptos y la aplicación de los mismos en la práctica, debido a que pueden visualizar de forma gráfica e interactiva conceptos de control y poder aplicarlos en un sistema real.

La interfaz o simplemente el robot puede ser usado por estudiantes de la carrera ingeniería mecatrónica que se encuentren del 6to nivel en adelante y quieran aprender conceptos básicos de robótica, control y funcionamiento de un robot móvil diferencial.

6.2 Recomendaciones

Para un mejor desempeño del robot se debe utilizar un computador de alto rendimiento, con esto se asegura que el computo de los controladores sea rápido y eficiente.

Para el correcto funcionamiento de los controladores no se recomienda usar trayectoria con velocidades muy altas ya que los motores son de bajas revoluciones y alto torque.

Para instalar los programas necesarios del simulador se recomienda usar las versiones descritas y no cambiar a otras ya que el simulador fue diseñado en específico para estas.

Durante el ensamblaje del robot Wall3, tomar en cuenta el patrón de colores de los cables de conexión de cada motor para un correcto funcionamiento del mismo.

Verificar la dirección IP del terminal de la computadora de cada usuario antes de utilizar el robot.

Al momento que ejecutar la interfaz del robot Wall3, digitar correctamente los comandos de comunicación para que no se presente errores en la ejecución de la interfaz.

Al momento de desarrollar un controlador basado en redes neuronales utilizar una ley de adaptación que garantice la convergencia de los errores de control a cero.

Bibliografía

- ¿Qué son las TIC? (n.d.). Recuperado 4 diciembre del 2020, from <http://tutorial.cch.unam.mx/bloque4/lasTIC>
- 10 Best Microcontroller Boards for Engineers and Geeks - Engineering Passion. (n.d.). Recuperado 4 diciembre del 2020, from <https://www.engineeringpassion.com/10-best-microcontroller-boards-for-engineers-and-geeks/>
- Alvarez, K. L. C., Lagos, R. F. C., & Aizpun, M. (2016). Recuperado 4 diciembre del 2020, Influencia del porcentaje de relleno en la resistencia mecánica en impresión 3D, por medio del método de Modelado por Deposición Fundida (FDM). *Ingeniare*, 24(ESPECIAL), 17–24. <https://doi.org/10.4067/S0718-33052016000500003>
- Andaluz, G. M., Andaluz, V. H., Terán, H. C., Arteaga, O., Chicaiza, F. A., Varela, J., ... Canseco, P. (2016a). Recuperado 4 diciembre del 2020, Modeling dynamic of the human-wheelchair system applied to NMPC. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9835 LNCS, 179–190. https://doi.org/10.1007/978-3-319-43518-3_18
- Andaluz, G. M., Andaluz, V. H., Terán, H. C., Arteaga, O., Chicaiza, F. A., Varela, J., ... Canseco, P. (2016b). Recuperado 4 diciembre del 2020, Modeling dynamic of the human-wheelchair system applied to NMPC. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9835 LNCS, 179–190. https://doi.org/10.1007/978-3-319-43518-3_18
- Andaluz, V. H., Canseco, P., Varela, J., Ortiz, J. S., Pérez, M. G., Morales, V., ... Carelli, R. (2015). Recuperado 4 diciembre del 2020, Modeling and control of a wheelchair considering center of mass lateral displacements. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9246, 254–270.

https://doi.org/10.1007/978-3-319-22873-0_23

Angelov, P., & Xydeas, C. (2006). Recuperado 23 diciembre del 2020, Fuzzy systems design: Direct and indirect approaches. *Soft Computing*, 10(9), 836–849. <https://doi.org/10.1007/s00500-005-0006-x>

Åström, K. J., Anton, J. J., & Årzén, K. E. (1986). Recuperado 23 diciembre del 2020 Expert control. *Automatica*, 22(3), 277–286. [https://doi.org/10.1016/0005-1098\(86\)90026-9](https://doi.org/10.1016/0005-1098(86)90026-9)

Barrett, S. F., & Pack, D. J. (2005). Recuperado 23 diciembre del 2020 Microcontrollers. In *Handbook of Networked and Embedded Control Systems* (pp. 295–322). https://doi.org/10.1007/0-8176-4404-0_13

Bayle, B., Fourquet, J.-Y., & Renaud, M. (2003). Recuperado 01 enero del 2021 Manipulability of Wheeled Mobile Manipulators: Application to Motion Generation. *The International Journal of Robotics Research*, 22(7–8), 565–581. <https://doi.org/10.1177/02783649030227007>

Beira, R., Lopes, M., Praça, M., Santos-Victor, J., Bernardino, A., Metta, G., ... Saltarén, R. (2006). Recuperado 01 enero del 2021 Design of the robot-cub (iCub) head. *Proceedings - IEEE International Conference on Robotics and Automation, 2006*, 94–100. <https://doi.org/10.1109/ROBOT.2006.1641167>

Calderon, J. E. J. (n.d.). *Control Mediante Lógica Difusa*.

Cardona, J. P., Leal, J. J., & Ramirez, J. L. (2018). Recuperado 01 enero del 2021 Mathematical modeling of mass center position of differential traction robot. An approach from Lagrangian mechanics. *Informacion Tecnologica*, 29(6), 307–320. <https://doi.org/10.4067/S0718-07642018000600307>

COM-Based SBCs: The Superior Architecture for Small Form Factor Embedded Systems A White Paper from Diamond Systems Corp. (n.d.). Retrieved from www.diamondsystems.com

Craig, J. (2006). La mecanica y el control de los manipuladores mecánicos . In *Robotica* (pp. 1–19). Recuperado 01 enero del 2021

[https://books.google.com.mx/books?id=hRzOp_qdxG8C&pg=PA1&dq=robot+manipulador+industrial&hl=es-419&sa=X&ved=0ahUKEwix0YiBj_boAhXKTN8KHR0wDLUQ6AEIbjAI#v=onepage&q=robot manipulador industrial&f=false](https://books.google.com.mx/books?id=hRzOp_qdxG8C&pg=PA1&dq=robot+manipulador+industrial&hl=es-419&sa=X&ved=0ahUKEwix0YiBj_boAhXKTN8KHR0wDLUQ6AEIbjAI#v=onepage&q=robot%20manipulador%20industrial&f=false)

Difference Between Servo Motor and DC Motor - Circuit Globe. (n.d.). Recuperado 01 enero del 2021, from <https://circuitglobe.com/difference-between-servo-motor-and-dc-motor.html>

Diseño asistido por computadora - EcuRed. (n.d.). Recuperado 15 enero del 2021, from https://www.ecured.cu/Diseño_asistido_por_computadora

Grass, D., Caulkins, J. P., Feichtinger, G., Tragler, G., & Behrens, D. A. (2008). Recuperado 15 enero del 2021 Optimal control of nonlinear processes: With applications in drugs, corruption, and terror. In *Optimal Control of Nonlinear Processes: With Applications in Drugs, Corruption, and Terror*. <https://doi.org/10.1007/978-3-540-77647-5>

Hernández-Alvarado, R., García-Valdovinos, L., Salgado-Jiménez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Recuperado 15 enero del 2021 Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. *Sensors*, *16*(9), 1429. <https://doi.org/10.3390/s16091429>

Hornik, K., Stinchcombe, M., & White, H. (1989). Recuperado 15 enero del 2021 Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

Hossain, S., & Lee, D. (2019). Recuperado 15 enero del 2021 Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices. *Sensors*, *19*(15). <https://doi.org/10.3390/s19153371>

Industry Spotlight: Robotics from Monster Career Advice. (n.d.). Retrieved June 4, 2020, from <https://web.archive.org/web/20070830231418/http://content.monster.com/articles/3472/18567/1/industry/12/home.aspx>

- Jetson AGX Xavier Developer Kit | NVIDIA Developer. (n.d.). Recuperado 15 enero del 2021 Retrieved February 11, 2021, from <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>
- Joseph, L. (2015). Recuperado 25 enero del 2021 *Learning robotics using Python : design, simulate, program, and prototype an interactive sutonomous mobile robot fom scratch with the help of Python, ROS, and Open-CV!* Packt Publishing.
- Klančar, G., Zdešar, A., Blažič, S., & Škrjanc, I. (2017). Recuperado 15 enero del 2021 *Wheeled Mobile Robotics*.
- Kröger, E., Dekiff, M., & Dirksen, D. (2017). Recuperado 15 enero del 2021 3D printed simulation models based on real patient situations for hands-on practice. *European Journal of Dental Education*, 21(4), e119–e125. <https://doi.org/10.1111/eje.12229>
- Künhe, F., Gomes, J., & Fetter, W. (2005). Recuperado 15 enero del 2021. Mobile Robot Trajectory Tracking Using Model Predictive Control. *II IEEE Latin-American Robotics ...*, 1–7. Retrieved from http://www.feng.pucrs.br/~fkuhne/files/sistrob/notasdeaula/robosmoveis/sbai05_10022.pdf
- Li, Z., & Ge, S. S. (2013). Recuperado 25 enero del 2021 *Fundamentals in modeling and control of mobile manipulators*. Taylor & Francis.
- Liu, J. (2017). Recuperado 25 enero del 2021 Intelligent control design and MATLAB simulation. In *Intelligent Control Design and MATLAB Simulation*. <https://doi.org/10.1007/978-981-10-5263-7>
- Liu, J., & Liu, J. (2018). Recuperado 25 enero del 2021 Introduction to Intelligent Control. In *Intelligent Control Design and MATLAB Simulation* (pp. 1–5). https://doi.org/10.1007/978-981-10-5263-7_1
- Liu, W., & Winfield, A. F. T. (2011). Recuperado 25 enero del 2021 Open-hardware e-puck Linux extension board for experimental swarm robotics research.

Microprocessors and Microsystems, 35(1), 60–67.
<https://doi.org/10.1016/j.micpro.2010.08.002>

Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., & Bastos-Filho, T. F. (2008). Recuperado 25 enero del 2021 An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, 16(11), 1354–1363. <https://doi.org/10.1016/j.conengprac.2008.03.004>

MATLAB App Designer - MATLAB. (n.d.). Recuperado 25 enero del 2021, from <https://www.mathworks.com/products/matlab/app-designer.html>

Michel, O. (2004). Recuperado 25 enero del 2021 Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1), 5. <https://doi.org/10.5772/5618>

Moviles - ROBOTICA. (n.d.). Recuperado 25 enero del 2021 Retrieved June 4, 2020, from <https://sites.google.com/a/unitecnica.net/dmhenao/articulos/movile>

Polak, E. (1997a). Recuperado 25 enero del 2021 *Optimal Control*. https://doi.org/10.1007/978-1-4612-0663-7_4

Polak, E. (1997b). Recuperado 25 enero del 2021 *Optimization*. <https://doi.org/10.1007/978-1-4612-0663-7>

Prasad, L. B., Tyagi, B., & Gupta, H. O. (2014). Recuperado 25 enero del 2021 Optimal control of nonlinear inverted pendulum system using PID controller and LQR: Performance analysis without and with disturbance input. *International Journal of Automation and Computing*, 11(6), 661–670. <https://doi.org/10.1007/s11633-014-0818-1>

Propiedades del PLA. (n.d.). Recuperado 25 enero del 2021 Retrieved February 26, 2021, from <https://sites.google.com/view/poliacidolactico-coma/poliácido-láctico/propiedades-del-pla>

Recalde, L. F., Guevara, B. S., Cuzco, G., & Andaluz, V. H. (2020). Recuperado 25 enero del 2021 Optimal control problem of a differential drive robot. In

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12144 LNAI (pp. 75–82). https://doi.org/10.1007/978-3-030-55789-8_7

Robot móvil - Wikipedia, la enciclopedia libre. (n.d.). Recuperado 25 enero del 2021, from https://es.wikipedia.org/wiki/Robot_móvil

ROS/Introduction - ROS Wiki. (n.d.). Recuperado 25 enero del 2021, from <http://wiki.ros.org/ROS/Introduction>

Royo, S., & Ballesta-Garcia, M. (2019). Recuperado 25 enero del 2021, An Overview of Lidar Imaging Systems for Autonomous Vehicles. *Applied Sciences*, 9(19). <https://doi.org/10.3390/app9194093>

Sabir, M. M., & Khan, J. A. (2014). Recuperado 04 febrero del 2021, Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques. *Advances in Artificial Neural Systems*, 2014, 1–8. <https://doi.org/10.1155/2014/126317>

Sobota, J., PiŠI, R., Balda, P., & Schlegel, M. (2013). Recuperado 04 febrero del 2021 Raspberry Pi and Arduino boards in control education. *IFAC Proceedings Volumes*, 46(17), 7–12. <https://doi.org/https://doi.org/10.3182/20130828-3-UK-2039.00003>

Solaque Guzmán, L. E., Molina Villa, M. A., & Rodríguez Vásquez, E. L. (2014). Recuperado 04 febrero del 2021 Seguimiento de trayectorias con un robot móvil de configuración diferencial. *Ingenierías USBMed*, 5(1), 26–34. <https://doi.org/10.21500/20275846.298>

Sukkarieh, S., Nebot, E. M., & Durrant-Whyte, H. F. (1999). Recuperado 04 febrero del 2021 A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3), 572–578. <https://doi.org/10.1109/70.768189>

Tibaduiza, D. A., Amaya, I., Rodríguez, S., Mejía, N., & Flórez, M. (2011). Recuperado 04 febrero del 2021 Implementación de un control fuzzy para el

- control cinemático directo en un robot manipulador. *Ingeniare*, 19(3), 312–322. <https://doi.org/10.4067/S0718-33052011000300002>
- Tzafestas, S. G. (2013). Recuperado 04 febrero del 2021 Introduction to Mobile Robot Control. In *Introduction to Mobile Robot Control*. <https://doi.org/10.1016/C2013-0-01365-5>
- Tzafestas, S. G. (2014). Recuperado 04 febrero del 2021, Mobile Robot Dynamics. In *Introduction to Mobile Robot Control* (pp. 69–99). <https://doi.org/10.1016/b978-0-12-417049-0.00003-1>
- van Gerven, M., & Bohte, S. (2017, December 19). Recuperado 04 febrero del 2021, Editorial: Artificial neural networks as models of neural information processing. *Frontiers in Computational Neuroscience*, Vol. 11. <https://doi.org/10.3389/fncom.2017.00114>
- Varela-Aldas, J., Andaluz, V. H., & Chicaiza, F. A. (2018). Recuperado 04 febrero del 2021, Modelling and control of a mobile manipulator for trajectory tracking. *Proceedings - 3rd International Conference on Information Systems and Computer Science, INCISCOS 2018, 2018-December*, 69–74. <https://doi.org/10.1109/INCISCOS.2018.00018>
- Velagic, J., Lacevic, B., & Osmic, N. (2008). Recuperado 04 febrero del 2021, Nonlinear Motion Control of Mobile Robot Dynamic Model. In *Motion Planning*. <https://doi.org/10.5772/5997>
- Yang, P., Han, S., & Lin, Y. (2017). Recuperado 04 febrero del 2021, Neural network control method for mobile robot trajectory tracking. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10607 LNAI, 199–212. https://doi.org/10.1007/978-3-319-69456-6_17
- Zea, D. J., Guevara, B. S., Recalde, L. F., & Andaluz, V. H. (2021). Recuperado 04 febrero del 2021, *Dynamic Simulation and Kinematic Control for Autonomous Driving in Automobile Robots*. https://doi.org/10.1007/978-3-030-63665-4_16

Anexos