



Diseño y construcción de un sistema multi-robot tipo enjambre, basado en la asignación de tareas.

Ayala Bravo, Edison Leonardo y Mejía González, Leonardo David

Departamento de Ciencias de la Energía y Mecánica

Carrera de Mecatrónica

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Mecatrónica


Ing. Loza Matovelle, David César

14 de febrero del 2023



CERTIFICADO DE ANÁLISIS
magister

TESIS AYALA - MEJIA

< 1% Similitudes  < 1% Texto entre comillas
0% similitudes entre comillas
0% Idioma no reconocido

Nombre del documento: TESIS AYALA - MEJIA.pdf
ID del documento: 36255f3817b4cc53b69a16d1e646755fee717cc0
Tamaño del documento original: 2,35 Mo


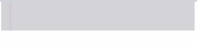


Depositante: DAVID CESAR LOZA MATOVELLE
Fecha de depósito: 15/2/2023
Tipo de carga: interface
fecha de fin de análisis: 15/2/2023

Número de palabras: 18.722
Número de caracteres: 117.064

Ubicación de las similitudes en el documento:



Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 dspace.uazuay.edu.ec https://dspace.uazuay.edu.ec/bitstream/datos/9411/1/15048.pdf#:~:text=Resumen—En este trabajo s...	< 1%		Palabras idénticas : < 1% (12 palabras)
2	 repositorio.espe.edu.ec Diseño e implementación de un sistema SCADA en el proc... http://repositorio.espe.edu.ec:8080/bitstream/21000/29357/4/T-ESPEL-MEC-0241.pdf.txt	< 1%		Palabras idénticas : < 1% (13 palabras)

Firma:

Ing. Loza Matovelle, David César
C. C 1708661549



Departamento de Ciencias de la Energía y Mecánica

Carrera de Mecatrónica

Certificación

Certifico que el trabajo de integración curricular: **“Diseño y construcción de un sistema multi-robot tipo enjambre, basado en la asignación de tareas”** fue realizado por los señores **Ayala Bravo, Edison Leonardo y Mejía González, Leonardo David**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 14 de febrero del 2023

Firma:

Ing. Loza Matovelle, David César

C. C 1708661549



Departamento Ciencias de la Energía y Mecánica

Carrera de Mecatrónica

Responsabilidad de Autoría

Nosotros, **Ayala Bravo, Edison Leonardo y Mejía González Leonardo David**, con cédulas de ciudadanía N° 1003976220 y N° 1718170994, declaramos que el contenido, ideas y criterios del trabajo de titulación: **"Diseño y construcción de un sistema multi-robot tipo enjambre, basado en la asignación de tareas"** 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.

Sangolquí, 14 de febrero del 2023

Firmas

Edison Ayala

Ayala Bravo, Edison Leonardo
CC: 1003976220

Leonardo Mejía

Mejía González Leonardo David
CC: 1718170994



Departamento Ciencias de la Energía y Mecánica

Carrera de Mecatrónica

Autorización de Publicación

Nosotros, **Ayala Bravo, Edison Leonardo y Mejía González Leonardo David**, con cédulas de ciudadanía N° 1003976220 y N° 1718170994, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "**Diseño y construcción de un sistema multi-robot tipo enjambre, basado en la asignación de tareas**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 14 de febrero del 2023

Firmas

Ayala Bravo, Edison Leonardo
CC: 1003976220

Mejía González Leonardo David
CC: 1718170994

Dedicatorias

Dedicado a mi padre Edison Ayala que siempre estuvo apoyándome en el transcurso de la carrera y a mi hermana. Quienes han sabido motivarme hasta poder alcanzar mis metas en mis años de estudio.

Ayala Bravo, Edison Leonardo

Quiero dedicar este trabajo a mi familia, mi padre Javier Mejía, mi madre María de los Ángeles González y a mi hermana Camila Mejía quienes han sido testigos del esfuerzo y dedicación a lo largo de todos estos años.

Mejía González, Leonardo David

Agradecimientos

A mi querido padre, mi mayor inspiración; quien ha sido mi fuente de inspiración, siendo un gran ejemplo para mí. Nunca decidió rendirse con los desafíos de la vida, enseñándome el valor del esfuerzo y trabajo que al final siempre traen sus recompensas.

A mi familia y mis amigos/as dentro y fuera de la universidad, que siempre me mostraron su apoyo y supieron animarme positivamente a seguir, con quienes he compartido gratos momentos, llenos de alegrías y tristezas, que me han ayudado a sobrellevar mi día a día

Ayala Bravo, Edison Leonardo

Mi dedicatoria y agradecimientos van especialmente dirigidos a mi familia quienes han sido apoyo fundamental en este largo trayecto en la universidad, simplemente les quiero decir gracias por la paciencia que me han tenido.

Agradezco a mis amigos más cercanos *The Boyz* conformado por Pedro, Gua, JP, Valdez, Villa, Guillo, Cuvi, Alex, Joe, Pol, Stivi y Joel que gracias a ellos estos años de universidad han sido llenos de risas, llantos y buenos momentos. No tengo palabras para describir a estos individuos, simplemente quiero decirles gracias por todo.

También quiero hacer mención especial para mis amigos Erick y Mauricio que han sabido estar en los momentos más duros de la carrera.

Finalmente, quiero agradecer todo el apoyo que he tenido por parte de mi amigo y compañero de tesis Edison que ha sufrido tanto como yo durante todo el proceso de realizar este trabajo.

Mejía González, Leonard David

Índice de contenidos

Dedicatorias.....	6
Agradecimientos	7
Resumen	17
Abstract.....	18
Generalidades.....	19
Antecedentes.....	19
Trabajos Nacionales	20
Trabajos Internacionales.....	23
Justificación	25
Objetivos.....	27
<i>Objetivo General</i>	<i>27</i>
<i>Objetivos Específicos</i>	<i>27</i>
Alcance	27
Estructura del documento	28
Estado Del Arte.....	29
Robots móviles	29
<i>Clasificación de robots móviles</i>	<i>30</i>
Locomoción por piernas.....	30
Locomoción por ruedas.	31
Componentes de los robots móviles	31
Sistemas multi-robots	32

<i>Tipos de sistemas multi-robots basado en su organización</i>	33
<i>Sistemas centralizados</i>	34
<i>Sistemas tipo enjambre</i>	35
<i>Sistemas distribuidos</i>	36
Sistemas de asignación de tareas	37
<i>Aproximación basada en mercado</i>	37
<i>Acercamiento basado en optimización</i>	40
Optimización determinista.....	41
Optimización estocástica.	41
Resumen	41
Metodología	43
Requerimientos.....	44
Diseño de subsistemas.....	44
<i>Diseño del subsistema mecánico</i>	45
Integración del sistema mecánico.	50
<i>Subconjunto de tracción</i>	50
<i>Subconjunto estructural</i>	50
<i>Conjunto (Subconjunto estructural + tracción)</i>	52
<i>Diseño del subsistema electrónico</i>	53
Driver o controlador para motor DC.	54
Sensores.	55
Tarjeta de adquisición de datos.	58
Unidad de control.....	61
Alimentación.	62

<i>Reguladores de voltaje lineales y conmutados</i>	62
Integración de componentes.....	63
<i>Diseño de la placa de potencia</i>	64
<i>Diseño de la placa de control</i>	66
<i>Diseño del subsistema de control</i>	69
Diseño del control PID para la velocidad angular de ambos motores.....	73
Diseño de controlador PID de orientación (seguimiento en línea recta).....	75
Diseño de controlador PID de orientación (orientación hacia punto final).....	76
<i>Diseño del subsistema de software</i>	78
Adquisición y procesamiento de información.....	79
Comunicación entre los dispositivos.....	81
Selección de método de subasta.....	83
Algoritmo de ordenamiento.....	85
Integración del subsistema de ordenamiento.....	86
Algoritmo de funcionamiento global.....	89
Algoritmo de odometría.....	92
Algoritmo de prevención de colisiones.....	92
Resumen.....	93
Validación y análisis de resultados	94
Validación del subsistema mecánico.....	94
<i>Soporte de motor</i>	95
<i>Placa inferior</i>	97
Validación del sistema electrónico.....	99
<i>Placa de potencia</i>	99
<i>Placa de control</i>	100

Validación de odometría	100
<i>Prueba de giro, seguimiento en línea recta y distancia (GSD)</i>	100
<i>Prueba de trayectoria de un cuadrado</i>	106
Validación de algoritmos de asignación de tareas.....	109
<i>Comparación de algoritmo de asignación de tarea paralelo y secuencial</i>	109
<i>Ejemplo de cálculo de una subasta paralela para n agentes</i>	111
<i>Validación del ordenamiento cuando existen distancias iguales</i>	113
Validación de sistema descentralizado	115
<i>Desconexión de un agente simple</i>	116
<i>Desconexión del agente reina</i>	117
Resumen	118
Conclusiones	119
Recomendaciones	120
Trabajos futuros	120
Bibliografía	122
Apéndices	125

Índice de Tablas

Tabla 1 <i>Trabajos nacionales relacionados</i>	20
Tabla 2 <i>Trabajos relacionados internacionales</i>	23
Tabla 3 <i>Tipos de subastas</i>	38
Tabla 4 <i>Matriz de evaluación para la selección de materiales</i>	46
Tabla 5 <i>Matriz de evaluación para selección de motores</i>	49
Tabla 6 <i>Matriz de selección de drivers para motores DC</i>	55
Tabla 7 <i>Matriz de selección para sensores de orientación</i>	56
Tabla 8 <i>Matriz de selección para sensores de detección externa</i>	58
Tabla 9 <i>Tipos de señales de los sensores</i>	59
Tabla 10 <i>Matriz de selección para tarjeta de adquisición de datos</i>	60
Tabla 11 <i>Características de microordenadores</i>	61
Tabla 12 <i>Voltajes y corrientes de sensores y actuadores en el robot móvil</i>	62
Tabla 13 <i>Consumo de corriente de la placa de potencia</i>	65
Tabla 14 <i>Consumo de corriente de la placa de control</i>	67
Tabla 15 <i>Coeficientes de los motores izquierdo y derecho</i>	72
Tabla 16 <i>Cálculo de constantes para PID</i>	73
Tabla 17 <i>Constantes para PID de ambos motores</i>	74
Tabla 18 <i>Variables para selección del ordenamiento</i>	83
Tabla 19 <i>Selección del método para organización</i>	85
Tabla 20 <i>Criterios de selección para el ordenamiento</i>	86
Tabla 21 <i>Posibles métricas para valores de puja</i>	87
Tabla 22 <i>Matriz de selección de métrica</i>	88
Tabla 23 <i>Propiedades mecánicas del acrílico</i>	94
Tabla 24 <i>Análisis de tensiones en soporte de motor</i>	96

Tabla 25 <i>Análisis de tensiones en placa inferior</i>	98
Tabla 26 <i>Datos de prueba GSD</i>	101
Tabla 27 <i>Datos de prueba GSD para J2</i>	102
Tabla 28 <i>Estadísticos descriptivos para J1</i>	103
Tabla 29 <i>Estadísticos descriptivos para J1</i>	105
Tabla 30 <i>Prueba de trayectoria cerrada para J1</i>	107
Tabla 31 <i>Prueba de trayectoria cerrada para J2</i>	107
Tabla 32 <i>Resumen de estadísticos descriptivos para ambos robots</i>	108

Índice de figuras

Figura 1 <i>Robot cuadrúpedo Kotetsu</i> _____	30
Figura 2 <i>Sistemas básicos de un robot móvil</i> _____	32
Figura 3 <i>Norma VDI 2206</i> _____	43
Figura 4 <i>División del sistema mecánico</i> _____	45
Figura 5 <i>Diagrama de cuerpo libre</i> _____	47
Figura 6 <i>Sub ensamble del sistema de tracción</i> _____	51
Figura 7 <i>Subconjunto estructural</i> _____	52
Figura 8 <i>Estabilidad proporcionada por 3 puntos de apoyo</i> _____	53
Figura 9 <i>Resumen del subsistema electrónico</i> _____	54
Figura 10 <i>Distribución de voltaje</i> _____	63
Figura 11 <i>Distribución de voltajes</i> _____	64
Figura 12 <i>Esquema de la placa de potencia</i> _____	65
Figura 13 <i>Pistas para placa de potencia</i> _____	66
Figura 14 <i>Esquema de la placa de control</i> _____	67
Figura 15 <i>Pistas para placa de control</i> _____	68
Figura 16 <i>Circuito equivalente de un motor DC</i> _____	70
Figura 17 <i>Muestreo de la planta en lazo abierto</i> _____	71
Figura 18 <i>Respuesta al escalón en lazo abierto para los motores DC</i> _____	72
Figura 19 <i>Respuesta al escalón en lazo cerrado para los motores DC</i> _____	74
Figura 20 <i>Diagrama de flujo para corrección de trayectoria</i> _____	76
Figura 21 <i>Orientación del robot para llegar a su punto final</i> _____	77
Figura 22 <i>PID de orientación sobre el centro de giro</i> _____	78
Figura 23 <i>Resumen del subsistema de software</i> _____	79
Figura 24 <i>Comunicación entre los agentes participantes</i> _____	82

Figura 25 Comunicación entre tarjeta de control y DAQ	83
Figura 26 Algoritmo de ordenamiento	88
Figura 27 Comunicación y control entre los agentes participantes	90
Figura 28 Estructura de los agentes simples	91
Figura 29 Estructura del agente reina	91
Figura 30 Diagrama de flujo de odometría	92
Figura 31 Estructura de colisiones	93
Figura 32 Puntos de apoyo y fuerza equivalente del soporte de motor	95
Figura 33 Análisis de tensiones en soporte de motor	96
Figura 34 Análisis de tensiones en soporte de motor	96
Figura 35 Puntos de apoyo y fuerza aplicada en la placa inferior	97
Figura 36 Análisis de tensiones en placa inferior	98
Figura 37 Análisis de tensiones en placa inferior	98
Figura 38 Placa de potencia ensamblada	99
Figura 39 Placa de control ensamblada	100
Figura 40 Prueba GSD	101
Figura 41 Prueba Z para $\Delta\phi$	103
Figura 42 Prueba Z para Δd	104
Figura 43 Prueba Z para $\Delta\phi$	105
Figura 44 Prueba Z para Δd	106
Figura 45 Asignación de tareas por método secuencial	110
Figura 46 Asignación de tareas por método paralelo	110
Figura 47 Simulación de ordenamiento	111
Figura 48 Ejemplo de cálculo de matriz para asignación de tareas	112
Figura 49 Matriz de cálculo - eliminación de agente ganador	113
Figura 50 Vectores de compra a cada actividad	114

Figura 51: <i>Ordenamiento a distancias iguales</i> _____	115
Figura 52: <i>Desconexión del segundo agente simple</i> _____	116
Figura 53: <i>Desconexión del agente reina</i> _____	117

Resumen

En el presente trabajo se muestra el proceso de diseño y construcción de robots diferenciales para realizar seguimiento a un líder, sus acciones están guiadas por un sistema multi-robot tipo enjambre, para comunicarse se emplea el protocolo MQTT en una red local, además trabaja en conjunto con ROS, es decir entre la computadora de cada robot, se comunica con su tarjeta de adquisición de datos para controlar los subsistemas de percepción y movimiento, además de ser usada para envío y recepción de información entre los agentes, por otro lado entre los robots y la computadora central, se emplearon algoritmos de asignación de tareas basado en el mercado por la simplicidad matemática que conlleva a comparación de los algoritmos de optimización que se han encontrado en la literatura.

Para validar el proceso de comunicación y control se desarrollaron algoritmos para establecer un líder y los seguidores, el líder recibe el objetivo al cual debe llegar para cumplir su tarea y cada agente su posición inicial estimada mediante una cámara cenital, mediante estimación de posición y control de orientación se establece la distancia recorrida por los agentes líder-seguidores. Finalmente se presenta un robot móvil modular para que pueda ser escalable y reproducible fácilmente en base a la disponibilidad del mercado local.

Palabras clave: robot diferencial, enjambre, algoritmo de asignación de tareas, escalable

Abstract

This paper shows the designing and building of differential robots to track a leader, its actions are guided by a multi robot swarm-type system, the MQTT protocol is used in a local network for the communication, and it's essential that works together with ROS, because, between the computer of each robot and its data acquisition card it's control subsystems of perception or movements, but it's core function is to send and receive information between the agents and the queen to organize the swarm, on the other hand, between the robots and the queen, budget algorithms were used of market-based tasks due to the mathematical simplicity compared to some of the optimization algorithms that have been found in the literature entails.

For the validation of the communication and control systems, algorithms were developed to establish different leaders and followers, the master receives the goal to which he must reach to fulfill his task and each agent has to estimated his initial position through an overhead camera, using position estimation and orientation control, the distance traveled by the leader-follower agents is established. Finally, a modular mobile robot is presented so it can be easily scalable and reproducible based on local market availability.

Key Words: differential robot, swarm, budget algorithms, scalable

Capítulo I

Generalidades

Antecedentes

La facultad de adaptación de los sistemas determina la capacidad de colaboración y aprendizaje colectivo que es requerido para la solución de problemas, por lo cual es necesario buscar una mayor flexibilidad mediante la aplicación de robótica colaborativa para la resolución de problemas complejos.

En los años 90 empieza el desarrollo del concepto “multi-robot” para ejecutar tareas conjuntas como: búsqueda, agrupamiento, dispersión, entre otros. Además, también surge el concepto de robótica de enjambres, encargado de estudiar el diseño de robots a nivel constructivo como de comportamiento, respecto a los patrones de comportamiento colectivos al interactuar con el entorno y sus semejantes se toma como objeto de estudio a los insectos sociales. (Sempere-Tortosa, 2014)

En 1990 surge el MARTHA (Mosquera, 2017) uno de los primeros trabajos sobre sistemas multi-robots donde los agentes individuales son robots móviles, en el proyecto se implementa por primera vez la tolerancia de fallos, refiriéndose a eventos inesperado tales como: Obstáculos nuevos fuera del mapeo de las rutas, nivel insuficiente de batería para impulsar a sus actuadores, reportes de fallas por los diferentes módulos que componen al sistema, como también fallas en envío de mensajes para la comunicación del enjambre. Por lo cual, se crea una capa de ejecución, en la cual el robot supervisor se encarga de interpretar y cambiar la planificación del enjambre, siendo la responsable de las interacciones entre los robots.

Para conocer el avance investigativo en el Ecuador sobre sistemas multi-robots es necesario comparar las publicaciones realizadas localmente con las internacionales, debido a

que en literaria realizada por Knudsen (Knudsen & KaiVo-Oja, 2020) se muestra el crecimiento de publicaciones internacionales, en los gráficos presentados por su autor se observa que tomo 20 años desde 1996 sobre pasar los 20 trabajos publicados anualmente, aunque la tendencia cambiaria ya que solo en el 2017 el volumen creció a más de 100 artículos por año, hasta finales del 2019 el crecimiento no ha parado pues en este se registró 193 publicaciones.

Trabajos Nacionales

En la sección revisaremos los trabajos nacionales publicados sobre sistemas multi-robot para conocer el avance en investigación. En la Tabla 1 se presenta el título del artículo, sus autores y un resumen del contenido expuesto en cada trabajo.

Tabla 1

Trabajos nacionales relacionados

Títulos	Autores	Descripción de la investigación	Citas
Implementation of an IoT Architecture based on MQTT for a Multi-Robot System	González David, Verdugo Cabrera Alexandra del Carmen, Escobar Luis, Loza David	El trabajo consiste en la implementación de una arquitectura IoT basado en MQTT, entre las funciones del sistema se encuentra la navegación, se enfocan en el tiempo de respuesta, costo computacional, la funcionalidad de la arquitectura y las capacidades multi-robot	(Efrain et al., 2018)
Desarrollo de un controlador centralizado de seguimiento de	Christian Beltran Tuquerrez	El trabajo se enfoca en el desarrollo e implementación de seguimiento de trayectorias centralizado, prevención y evasión de colisiones en sistemas	(Beltran Tuquerrez, 2019)

Títulos	Autores	Descripción de la investigación	Citas
<p>Trayectoria para un Sistema Multi-Robot</p> <p>Multi-Robot platform with features of Cyber-physical systems for education applications</p>	<p>Luis Escobar, Cristian Moyano, Gabriel Aguirre, Gabriel Guerra, Luis Allauca, David Loza</p>	<p>multi-robot, además de la comunicación maestro-esclavo entre ellos mediante red inalámbrica estrella.</p> <p>Se desarrolla un robot omnidireccional el cual debe ser flexible y de características escalabres al igual que debe ser fácil de replicar, se usa el protocolo MQTT para la gestión conjunta del sistema, el control se lo realiza usando ROS en una Rasberry pi, con el uso de los Arduino se controla sensores y actuadores.</p>	<p>(Escobar et al., 2020)</p>
<p>Architecture for Multi-robot Systems with Emergent Behavior</p>	<p>Angel Gil, Jose Aguilar, Rafael Rivas, Eladio Dapena, Kevin Hernandez</p>	<p>La investigación se divide en 3 niveles, en el primero se encargan del manejo de acciones, percepción, comunicación de los sistemas multi-robots, en el segundo nivel se provee soporte los procesos colectivos del sistema, así como los mecanismos de cooperación, planeamiento, negociación, finalmente el tercer nivel se encarga del aprendizaje individual y colectivo.</p>	<p>(Gil et al., 2018)</p>

De la Tabla 1, el primer trabajo revisado se publica en octubre del 2018, en el que se explora la implementación de arquitecturas IoT basado en MQTT debido a su conectividad entre múltiples dispositivos, para probar su navegación y funciones complementarias se utiliza criterios de evaluación de IoT: respuesta en el tiempo, costo computacional, funcionalidad de la arquitectura y capacidad de los multi-robots. El resultado muestra una gran oportunidad en el área de comunicación y escalabilidad para sistemas multi-robot.

En diciembre del mismo año Ángel Gil y sus compañeros analizan el comportamiento de los sistemas multi-robot al ser manejados por una arquitectura de 3 capas: soporte local, manejo de acciones, percepción y comunicación, en el estudio se considera los aspectos reactivos, cognitivos, al igual que sociales del robot, donde se introduce un componente afectivo el cual influencia el comportamiento, la manera de interactuar con el entorno y sus semejantes.

En 2019 Christian Beltrán publica el desarrollo de un controlador centralizado de seguimiento de trayectoria para un sistema multi-robot, en el trabajo se expone la creación de una red inalámbrica tipo estrella para realizar la comunicación del maestro con los esclavos, implementación de un sistema de seguimiento de trayectorias basados en un controlador de trayectoria cinemático, la implementación de métodos de prevención y evasión de colisiones.

Finalmente, en el año 2020 se publica “Plataforma multi-robótica para aplicaciones educativas”, en la investigación se establece el uso de protocolo MQTT para la gestión conjunta de tareas y ROS en una Raspberry pi en conjunto con Arduino para el control mediante sensores y actuadores, obteniendo en conjunto un robot omnidireccional el cual es flexible y de características escalables, el cual es simple de replicar.

Trabajos Internacionales

En la Tabla 2 encontramos trabajos internacionales que se han considerado como guías en la investigación para conocer mejor el problema a solucionar, en la tabla encontramos títulos, autores y un breve resumen del contenido existente en las investigaciones.

Tabla 2

Trabajos relacionados internacionales

Títulos	Autores	Descripción de la investigación	Citas
Coordinated multi-robot exploration	W. Burgard; M. Moors; C. Stachniss; F.E. Schneide	Se enfocan en la distribución de los distintos robots para explorar se deben escoger puntos estratégicos de tal forma que cada robot explore diferentes regiones simultáneamente y los algoritmos a seguir en situaciones donde se vean limitados por el rango de comunicación.	(Burgard et al., 2005)
Multi-robot target detection and tracking: taxonomy and survey	Cyril Robin & Simon Lacroix	En el trabajo se realiza una recopilación sobre varios artículos relacionados de sistemas multi-robots donde se revisará la taxonomía y se señalaran las diferentes aproximaciones al problema, modelos y resultados. Finalmente proponen una síntesis transversal en la cual se analizan las aproximaciones, modelos	(Robin & Lacroix, 2016)

Títulos	Autores	Descripción de la investigación	Citas
Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods	Kelin Jose Dilip Kumar Pratihar	y carencias de los problemas analizados. En este estudio se implementa el sistema de planeación de trayectorias y prevención de colisión en una planta, también se realiza la asignación óptima del número de tareas para cada robot a partir de un algoritmo genérico.	(Jose & Pratihar, 2016)
Complete coverage path planning for multi-robots based on	Adiyabaatar Janchiv, Dugarjav Batsaikhan, Gook hwan Kim, Soon-Geul Lee	La investigación se centra en la cobertura completa de planeación de trayectorias para 2 pisos internos donde sus tareas son limpieza e inspección de pisos industrial o áreas públicas, el algoritmo propuesto divide las áreas de limpieza en celdas mediante el método de descomposición celular, luego provee una cobertura eficiente basado en la distancia de los robos con el objetivo de reducir el consumo de energía y tiempo de trabajo.	(Janchiv et al., 2011)

De la Tabla 2, uno de los primeros trabajos internacionales revisados fue realizado por Burgard (Burgard et al., 2005), en su investigación realiza la distribución de distintos robots para exploración, se enfoca en la obtención de puntos estratégicos los cuales son usados y

enviados a cada individuo para explorar diferentes regiones simultáneamente, adicionalmente el algoritmo debe considerar situaciones donde la comunicación se vea limitada.

En 2011 en el trabajo "*Complete coverage path planning for multi-robots based on*" (Janchiv et al., 2011), se centra en planear trayectorias con cobertura completa en 2 pisos internos para labores de limpieza e inspección, se implementa el método de descomposición celular para dividir las áreas de limpieza en celdas las cuales considerando la distancia entre robots asigna las tareas para reducir consumo de energía y tiempo de trabajo.

Jose Kelin (Jose & Pratihari, 2016) en su trabajo implementa un sistema de asignación óptima de número de tareas para cada robot a partir de un algoritmo genérico, además posee una capa para prevención de colisiones y planeación de trayectorias en una planta.

La revisión literaria realizada por Robin Cyril y Simon Lacroix (Robin & Lacroix, 2016) reúne los trabajos relacionados sobre sistemas multi-robots donde se señalan las diferentes aproximaciones al problema, modelo y conclusiones de carencias sobre los problemas analizados, esto se presenta en una síntesis transversal.

Justificación

El enfoque del proyecto radica en diseñar un sistema multi-robot tipo enjambre que conste de un robot líder y seguidores, mediante dicha configuración se buscará realizar tareas de forma eficiente, teniendo en consideración que el líder informa al miembro sobre los movimientos a seguir para mantener la estructura.

Mediante la implementación de multi-robots y algoritmos de asignación de tareas se pretende realizar una primera aproximación dentro de este campo en la Universidad de las Fuerzas Armadas ESPE.

Con el avance tecnológico a través de los años los robots se han integrado a nuestra vida cotidiana robots para facilitar tareas, reducir costos, al igual que tiempos de operación

entre otros. En muchos casos estos robots no se comunican, desaprovechando las 3 ventajas potenciales que representa tener un sistema multi-robot: La primera de ellas es la eficiencia debido a su distribución espacial reduciendo el mapa de incertidumbre, en segundo lugar la fiabilidad y robustez en caso de falla o error de una unidad para evitar que la misión no se comprometa debido a la redundancia del sistema y en tercer lugar la especialización de los robots al poseer diferentes sensores (Rocha et al., 2005).

Por lo cual, surge el concepto de multi-robots, los cuales pueden desempeñar tareas más eficientemente en términos de tiempo o inclusive eficiencia energética como ya se revisó anteriormente.

Aunque la coordinación de sistemas multi-robots sigue en constante desarrollo, existen ejemplos de su uso y los podemos encontrar en diversos sectores como el industrial donde han implementado robots que, a través de programación, son capaces de realizar diversas actividades como: clasificar, ordenar o soldar.

Por otro lado, en los hogares se puede encontrar su utilidad en unidades robóticas aspiradoras que se encargan de la limpieza del hogar, etc.

En la actualidad los sistemas robóticos no tienen la suficiente capacidad para adaptarse a las diferentes tareas que se pueden presentar, en la Universidad de las Fuerzas Armadas ESPE se ha desarrollado varios proyectos de robótica, pero todos presentan “acoples débiles” es decir pueden reconocer su presencia en el entorno, mas no identificarse como individuos los cuales se pueden comunicar, organizar e interactuar entre sí , por lo cual el propósito de este trabajo es empezar una nueva área de investigación dentro de la Universidad de las Fuerzas Armadas ESPE, el acoplamiento fuerte entre robots.

Objetivos

Objetivo General

Diseñar y construir un sistema multi-robot tipo enjambre, basado en la asignación de tareas.

Objetivos Específicos

- Diseñar e implementar robots diferenciales modulares y embebidos
- Implementar el sistema de ubicación en los robots para evitar colisiones
- Desarrollar una plataforma para gestionar las actividades a cumplir por cada robot
- Analizar los datos obtenidos para validar el sistema multi-robot

Alcance

Se plantea el diseño y construcción de dos robots móviles para la implementación de un sistema multi-robots tipo enjambre. Los robots operarán en un espacio de pruebas controlado, donde se establecerá un objetivo al cual llegar ayudados por un sistema asignación de tareas, el cual debe cumplir eficientemente, todo este proceso se realizará mediante algoritmos de puja y ordenamiento.

Los agentes son robots diferenciales con diseño modular en su chasis para una fácil construcción y adaptación para las diferentes tareas que se puedan requerir, poseen una tarjeta de control y una de adquisición de datos, las cuales son: Jetson nano 2GB encargada de las funciones esenciales del robot como la comunicación con los distintos agentes (central y sus homólogos), choques, puja, entre otras y una ESP32 encargada de la odometría del robot diferencial

Estructura del documento

En el capítulo I se plantea un problema a resolver mediante el uso de sistemas multi-robots, se analizarán trabajos nacionales e internacionales, se proponen los objetivos generales y específicos y el alcance del proyecto donde se habla de las limitaciones y hasta qué punto debe avanzar el proyecto de titulación.

El capítulo II consiste en la revisión de diversas fuentes bibliográficas en donde se explorarán conceptos sobre sistemas multi-robots y robótica móvil, con el fin de introducir y explicar el tipo de sistema abordado en el proyecto. Adicionalmente, se revisa el estado del arte sobre tipos de asignación de tareas con el objetivo de tener una variedad de opciones de la cual elegiremos la más viable para la realización del proyecto.

En el capítulo III se detalla todo el proceso de selección de conceptos para posteriormente construir un agente simple, la elección de conceptos se basa en matrices de selección donde se plantean varias opciones y mediante el uso de ponderaciones se escoge la mejor opción.

En el capítulo IV se validarán las selecciones realizadas en el capítulo III mediante varias pruebas ya sean simulaciones o pruebas prácticas para validar el diseño propuesto.

Finalmente, en el capítulo V se abordan conclusiones, recomendaciones y trabajos futuros que nacen a raíz de todo el proceso de diseño, construcción y validación.

Capítulo II

Estado Del Arte

A lo largo del capítulo II se expondrán todos los conceptos necesarios para comprender el trabajo de titulación, tales como conceptos de robots móviles y de sus diferentes formas de locomoción, sistemas multi-robots para conocer las ventajas que se puede llegar a tener sobre robótica de un solo individuo, algoritmos de asignación de tarea, clasificación de sistemas multi-robots, entre otros.

Robots móviles

Actualmente en la industria existe gran aceptación de la robótica puesto que permite crear sistemas flexibles de manufactura, permite disminuir tiempos de construcción y además optimiza mejor los tiempos de ciertos procesos. Los robots como los KUKA son conocidos por estar fijados a una superficie por lo que su campo de trabajo es limitado, a raíz de su anclaje se tiene una gran desventaja de los brazos robóticos empleados en la industrial, por lo tanto, se plantea el concepto de robótica móvil. (Siegwart et al., 2011)

Los sistemas móviles se pueden definir como sistemas que no están fijos al ambiente y que pueden moverse libremente dentro de un área de trabajo y dependiendo del medio en que se muevan se pueden clasificar en tres grupos:

- Sistemas móviles terrestres
- Sistemas móviles aéreos
- Sistemas móviles acuáticos o subacuáticos

Para que un robot sea considerado móvil requiere tener cierta autonomía, por lo que se establecen dos puntos muy importantes que son:

- Desde el punto de vista de energías, el robot debe tener un sistema de alimentación propio.

- Desde el punto de vista de toma de decisiones, el robot debe ser capaz de tomar ciertas decisiones.

Clasificación de robots móviles

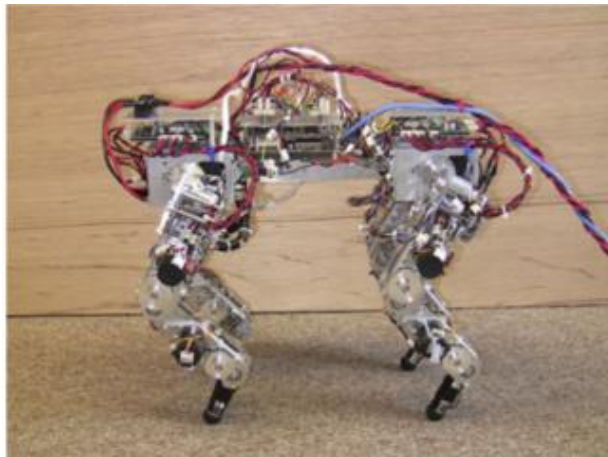
Dentro del área de la robótica móvil se pueden encontrar dos subgrupos que son:

- Locomoción por piernas
- Locomoción con ruedas

Locomoción por piernas. La locomoción de algunos de los animales más desarrollados es por el uso de piernas, este mecanismo de locomoción permite que algunos animales e insectos sea capaz de moverse a través de campos con múltiples obstáculos e irregularidades, dentro de la robótica un robot dotado de piernas para que sea funcional debe ser estáticamente estable, la condición se cumple si el centro de gravedad está dentro del polígono generado debido al área de contacto entre las piernas y el piso. (Tzafestas, 2014)

Figura 1

Robot cuadrúpedo Kotetsu



Nota. Tomado de *Introduction to mobile robot control*, por Tzafestas, S. G, 2013.

Como se puede observar en la Figura 1 se tiene un robot cuadrúpedo es decir que su movilidad y estabilidad es proporcionada por sus cuatro piernas o patas.

Locomoción por ruedas. Los robots con este tipo de locomoción tienen tres grados de libertad y se caracterizan por tener mucha maniobrabilidad la cual es requerida al momento de operar en superficies planas, dentro de este tipo de locomoción se pueden dividir en dos grupos que son robots móviles no holonómicos y holonómicos.

La holonomía se refiere a si un robot móvil tiene algún tipo de restricción en su movilidad, por ejemplo, los robots holonómicos son aquellos que pueden moverse en todas las direcciones y son muy usados en áreas muy pequeñas, a este tipo de robots se los conoce como omnidireccionales, mientras que los robots no holonómicos si poseen restricción en una dirección, por ejemplo, un robot diferencial no puede moverse hacia los lados debido a la configuración de sus ruedas.

Componentes de los robots móviles

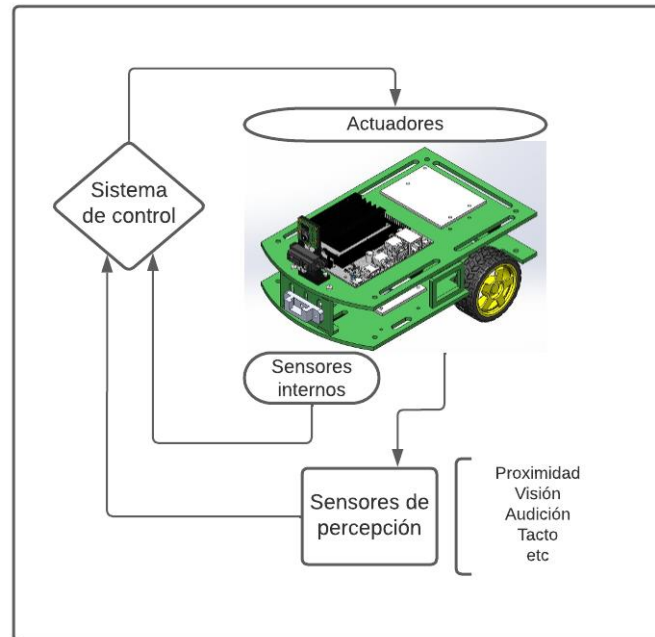
En la Figura 2 podemos observar los componentes básicos propuestos en libro Robótica: Manipuladores y robots móviles (Baturone, 2005), los sistemas que componen a un robot son:

- **Sistema mecánico:** Un sistema robótico puede ser distinguido por dos variantes debido a su efector final: brazo mecánico o un vehículo.
- **Sistema de percepción:** se recopila la información a través de sensores, es necesario tratar la señal proveniente, poder interpretarla y ser usada por el sistema de control para dictar los movimientos de los actuadores, los sensores podemos dividirlos en propioceptivos y exteroceptivos. El primero se encarga de medir el estado interno del robot, el segundo censa el entorno del robot dotándolo de “sentidos”.

- **Sistema de control:** control es el encargado de proporcionar una señal de retorno con la cual se cierra la cadena cerrada señal de salida-señal correctora-señal de salida.
- **Actuadores:** Son los encargados de generar la fuerza necesaria para llevar a cabo el desplazamiento del robot, la fuerza proviene principalmente de 3 energías: Eléctrica, neumática e hidráulica.

Figura 2

Sistemas básicos de un robot móvil



Sistemas multi-robots

El desarrollo de robots se inició con el propósito de realizar múltiples tareas, es decir que sean multipropósito y que a su vez sean reprogramables con el fin de que pueden cumplir varias tareas únicamente cambiando el actuador final, reprogramando sus rutas, etc. El avance de la tecnología permite el desarrollo de robots complejos y completos con el objetivo de realizar más tareas disminuyendo su tiempo de ejecución, pero esto implica mayores costos de

fabricación, incrementa la dificultad al momento de seleccionar las técnicas de control y su programación.

En raíz a dicho problema se desarrollan los sistemas multi-robot, concepto el cual es referido al conjunto de agentes robóticos que se integran y realizan un esfuerzo en conjunto cuando la tarea lo requiera, una ventaja notada diversos fabricantes es que la producción de múltiples robots representa una disminución en su costo en comparación a construir uno único robot es más sencillo. A pesar de su mayor facilidad de construcción, existen algunos retos durante su programación, puesto que el objetivo de un sistema multi-robot, es el desarrollo de múltiples robots de iguales características, de los pilares respecto a la resolución de problemas es evitar colisiones entre ellos.

Un acercamiento para prevenir las colisiones realizada por (Reynolds, 1987), propone tres reglas para tener éxito al momento de simular computacionalmente bandadas de pájaros:

- Centrado de la bandada: Mantenerse cerca a los compañeros de la bandada.
- Esquivar obstáculos: Evitar colisiones con compañeros de la bandada.
- Igualar velocidades: Intentar igualar las velocidades con los compañeros de la bandada.

Tipos de sistemas multi-robots basado en su organización

Para la clasificación de sistemas multi-robot, José Guerrero (Sastre Guerrero, 2011) propone una clasificación en 3 grupos centrados en diferenciar la creación de coaliciones y métodos para la creación de tareas con restricciones temporales, la clasificación en función de la distribución del proceso en un sistema multi-robot: Sistemas centralizados, sistemas tipo enjambre (*Swarm* u auto-organizados) y sistemas distribuidos (basados en comunicación)

Sistemas centralizados

Un sistema centralizado es compuesto por varios individuos, los cuales están siempre conectados con único agente central el cual es el encargado de la tomar todas las decisiones, por lo tanto, toda la información recogida por los robots sobre el entorno es enviado a la central. La centralización en el procesamiento ocasiona que los tiempos de ejecución sean elevados, además al existir un punto único de envío ocasiona fallos y saturación en el sistema de comunicación. Las características expuestas dan paso a las principales ventajas y desventajas de los sistemas centralizados:

- Sencillez: al ser usada para mecanismos basados en programación lineal facilita su implementación en sistemas multi-robots.
- Robustez: una de sus grandes desventajas ocasionada por su restricción a un agente central (único punto de fallo), donde si falla la central deriva en el paro de todo el sistema.
- Escalabilidad: Al estar conectado a un único agente central, por lo tanto, la comunicación está limitada a las características de la central, derivando en un cuello de botella.
- Flexibilidad: No permite una reacción ante cambios pues todo el procesamiento de la información se lo lleva en un solo punto.

Un ejemplo de la aplicación en sistemas centralizados la encontramos en el trabajo *“Heterogeneous Multirobot Coordination with Spatial and Temporal Constraints”* (Koes et al., 2005) se aplica un sistema multi robots para el sistema de búsqueda y rescate de víctimas en desastres naturales, donde se recibe el mapeo de la zona del desastre, la locación de las víctimas, mediante el uso de técnicas de programación lineal se asignan las tareas y planifica el camino a recorrer.

Otro enfoque de solución es proporcionado por Stephen Smith en su trabajo "*The Dynamic Team Forming Problem: Throughput and Delay for Unbiased Policies*" (S. L. Smith & Bullo, 2009) donde para disminuir el tiempo de retraso entre la llegada de la tarea y su finalización se implementa una reglas o "políticas" en base a la frecuencia de llegada de las tareas.

Sistemas tipo enjambre

Los enjambres consisten de muchos individuos, simples, homogéneos o heterogéneos, es muy común que los enjambres interactúen entre sus miembros, pero sin un sistema central de control, es decir descentralizado, además deben actuar con un comportamiento simple y local. Solamente a través de interacciones un comportamiento colectivo nace y que es capaz de resolver tareas complejas. Las características previamente nombradas dan paso a la principal ventaja de los enjambres, que son las siguientes:

- Sencillez y bajo coste: dados, los escasos requerimientos en robots tipo enjambre, se permiten robots pequeños y de bajo costo.
- Robustez: Al tener varios robots enlazados, si existiese un fallo de alguno de ellos, este no afecta de manera global a la colonia o enjambre
- Flexibilidad: Los robots pueden coordinar sus comportamientos de múltiples maneras, de tal forma que pueden otorgar múltiples soluciones a un mismo problema. (Sastre Guerrero, 2011)

Los agentes poseen capacidad de procesamiento, comunicación y detección en su estructura, con el propósito de permitir la interacción con otros miembros del sistema, además de permitirles de forma autónoma reaccionar a la variabilidad del entorno, los agentes pueden ser homogéneos o heterogéneos, pero siempre deberán estar interconectados

Al ser un campo en auge para la investigación, se basa en la creación de robots que cooperan e interactúan entre ellos para alcanzar un objetivo, uno de los principales aspectos es

que son sistemas descentralizados. En un sistema tipo enjambre, aunque se tengan robots totalmente autónomos, todos los robots del enjambre deben actuar como un todo para resolver problemas que uno solo no puede hacerlo debido a restricciones físicas o capacidades limitadas.

Sistemas distribuidos

Los sistemas distribuidos están basados en mecanismos donde se llevan cabo negociaciones explícitas sobre las tareas a realizar por cada robot, Los protocolos están basados en el trabajo de *Smith Contract Net Protocol (CNP)* (R. G. Smith, 1980), en este cada máquina posee un nodo para la comunicación entre sí, el cual debía determinar la solución a los problemas de forma cooperativa y distribuida, el nodo manager envía un mensaje a los nodos para anunciar una nueva tarea, estos responden con una puja o valor que representa lo adecuados que son para llevar a cabo la ejecución del nuevo problema, el nodo manager selecciona entre todos los valores para llevar a cabo la tarea.

Campbell (Campbell et al., 2008) idea un nuevo método parcialmente basado en subastas, donde cada robot puede decidir no seleccionar una tarea, siendo la opción seleccionada para ejecutarla al incluir tareas con *deadlines*. En el modelo propuesto por Smith solo se tiene en cuenta la puja por una única tarea, en el método el subastas combinatorias los robots pujan por un conjunto de tareas, en el trabajo de "*Task allocation in rescue operations using combinatorial auctions*" (López & Suárez, 2003) se usa en la competición de *RoboCup Rescue*, el método permite la distribución de robots en un entorno dinámico donde al asignar la tarea, debe ser cumplida para poder pujar otra.

El sistema distribuido es sencillo, y permiten restricciones temporales, además comparado a los sistemas centralizados estos algoritmos son menos complejos evitando problemas como la saturación o el punto único de fallo. Por otro lado, el tiempo total de

ejecución por los robots y por lo tanto la energía que consumen se ve incrementada a cambio de mejorar el rendimiento del sistema y la ejecución de tareas más complejas.

Sistemas de asignación de tareas

En la revisión literaria realizada por A. Khamis se establece 2 aproximaciones comúnmente utilizadas para MRTA (*Multi-Robots Task Allocation*): Aproximación basada en el mercado y aproximación basada en optimización.

Aproximación basada en mercado

La aproximación basada en mercado surge en el concepto económico de subastas, el cual se define como un mecanismo de reglas para intercambio, en el proceso de subasta se asignan bienes o servicios a los licitadores de acuerdo a sus ofertas. En la revisión literaria Multi-robot Task Allocation (Khamis et al., 2015) se menciona que la atención ganada en el campo de la robótica es debido a sus características deseables como eficiencia en satisfacer la función objetivo, robustez y escalabilidad, además de ser una forma simple de asignar tareas en sistemas multi-agente. Se plantea un algoritmo de subasta el cual consta de 4 fases:

- **Fase de anuncio:** en la etapa el agente con el rol de subastador anuncia la tarea o conjunto de tareas las cuales están disponibles para ser pujadas por los compradores.
- **Fase de sumisión:** Cada agente habilitado para realizar una compra establece su valor de puja basado en una función objetiva, una vez calculado el valor se envía la puja al anunciante.
- **Fase de selección:** una vez recibida la información de compra, el subastador evaluara las pujas en base a una estrategia de optimización para determinar el agente ganador.
- **Fase de contrato:** finalmente al comprador ganador se le asigna un contrato para ejecutar la tarea, dependiendo el tipo de subasta el proceso puede ser reiterativo.

Como se mencionó en la fase del contrato existen diferentes tipos de subastas, en el trabajo realizado por Michael Otte y sus colegas establecen 6 algoritmos para realizar aplicables a MRTA(Otte et al., 2020), cada opción se diferencian en el número de rondas para realizar la subasta, número de ítems o tareas asignadas al finalizar cada ronda, tamaño de los mensajes enviados/recibidos, entre otros, en la Tabla 3 se detalla cada tipo.

Tabla 3

Tipos de subastas

Tipo	Características	Descripción
Paralelo	Rondas: 1 Tareas anunciadas en la ronda r: m Tareas asignadas por ronda: m	En una única ronda se realiza la subasta de todos los ítems simultáneamente, los agentes compradores envían una lista con las pujas para cada tarea. Finalmente, el subastador envía a los ganadores las tareas a realizar
Secuencial	Rondas: m Tareas anunciadas en la ronda r: 1 Tareas asignadas por ronda: 1	Para este caso se vende únicamente un ítem por cada ronda, la selección de una tarea no subastada, la cual puede ser elegida de forma al azar, adicionalmente se realiza el anuncio del ganador y reconocimiento.
G-Prim	Rondas: m Tareas anunciadas en la ronda r: m - r	Similar a la subasta secuencial se subasta un ítem por cada una de las m rondas, aunque en la subasta se ofertan múltiples

Tipo	Características	Descripción
Paralelo repetido	Tareas asignadas por ronda: 1	ítems, durante cada ronda se puja por la tarea no vendida de mayor valor. Durante la ronda r el subastador anuncia las primeras n actividades que no han sido vendidas, en caso de ser n se anunciarán
	Rondas: m / n	todas, cada agente puja cada ítem, la primera tarea se asignará al pujador con la mejor oferta para ese ítem, la c -ésima tarea será asignada a la mejor oferta del comprador que no cuente con una actividad asignada.
	Tareas anunciadas en la ronda r: n	
G-Prim repetido	Tareas asignadas por ronda: n	Similar al algoritmo G-prim cuando el anunciador avisa sobre todas las actividades no asignadas en cada ronda, cada pujador envía su oferta para la tarea que considera de mayor valor, por cada ronda los agentes oferta por el n -ésimo ítem que consideren de mayor valor
	Rondas: m / n	
	Tareas anunciadas en la ronda r: $m - rn$	
Combinatorio	Tareas asignadas por ronda: n	
	Rondas: 1	
	Tareas anunciadas en la ronda r: m	Cada comprador oferta para todas las actividades posibles, el subastador calcula la forma óptima de división en equipos.
	Tareas asignadas por ronda: m	

De la Tabla 3 Podemos encontrar seis principales formas de subasta, la estructura paralela se caracteriza por realizar la subasta de todas las tareas en una ronda, una vez

receptada la puja de los agentes compradores se asignan los premios en base a la mejor oferta para cada tarea.

Respecto a la forma secuencial se realiza únicamente la venta de una actividad por cada ronda, la elección de la tarea a subastar puede ser aleatorio u otra implementación, cada pujador envía su oferta y al final de la ronda el subastador elige y notifica al ganador el cual ya no será tomado en cuenta para las siguientes pujas.

En el tipo G-prim se realiza la subasta de múltiples ítems, aunque únicamente se asigna la unidad no vendida de mayor valor.

En cuanto al Paralelo repetido se realiza la subasta de múltiples ítems (la cantidad puede ser definida por el programador), en caso de la cantidad sobrante sea menor se subastarán todos, una vez recibidas las ofertas, se asigna cada tarea para la mejor oferta por cada tarea.

El tipo G-prim repetida el anunciante avisa de las tareas no asignadas en cada ronda, los agentes compradores envían su oferta por el enésimo ítem que consideren de mayor valor en cada ronda.

Finalmente, la combinatoria los agentes pujan por todas las actividades posibles, aunque el subastador es el encargado de realizar la división óptima en equipos para el cumplimiento de las actividades.

Acercamiento basado en optimización

Para encontrar las soluciones posibles se aplican la matemática, dichas soluciones están restringidas por criterios y constantes las cuales ayudaran a seleccionar la respuesta óptima, los criterios son impuestos en base a una descripción cuantitativa de la meta, que a su vez dependen de la aproximación que se realiza a la naturaleza y el grado de complejidad del problema a resolver, por lo cual podremos encontrar varias formas de resolución por

optimización. El trabajo de *Multi-Robots Task Allocation: A review of the state of the art* (Khamis et al., 2015) nos propone la siguiente clasificación general de las técnicas de optimización:

Optimización determinista. La característica principal de la optimización determinista es que, mientras se mantenga el mismo punto de partida se seguirá el mismo camino sin importar cuando se corra el algoritmo. La optimización determinística sigue un riguroso procedimiento en el cual tanto las variables de diseño y las funciones se pueden repetir, en la categoría podemos encontrar técnicas basadas en: métricas, métodos clásicos, gradientes, métodos hessianos entre otros.

Optimización estocástica. Las técnicas estocásticas siempre tendrán un componente aleatorio, dentro de la optimización estocástica podemos clasificar en métodos basados en la trayectoria y algoritmos basados en población. El algoritmo basado en trayectorias usa un agente o solución única la cual se mueve por el espacio diseñado o buscado, se aceptan mejores soluciones, aunque movimiento no tan aceptable tienen la probabilidad de ser seleccionados, así estos pasos o movimientos crean una trayectoria en el campo de trabajo que visto globalmente son óptimos. Por otro lado, los algoritmos basados en poblaciones se buscan por una solución óptima mediante múltiples agentes.

Resumen

Los robots móviles pueden propiciar varias ventajas dentro de la industria debido a su carencia de un punto de anclaje en el suelo, para poder obtener un robot móvil es necesario determinar qué tipo de locomoción va a tener dado que en función de este parámetro puede incrementar su nivel de dificultad al momento de implementarlo y realizar un control sobre el mismo.

Para maximizar las ventajas que puede ofrecer un robot móvil se plantean conceptos de sistemas multi-robots, los cuales cuentan con varios individuos y dependiendo de la categoría

en la que se basa su organización pueden resolver tareas de mayor complejidad, se destaca el sistema centralizado la facilidad de programación, asignación por métodos lineales de tareas y la falta de robustez en su sistema, por otro lado los sistemas tipo enjambre son conformados por individuos sencillos y de bajo costo pero debido a su estructura de comunicación ofrecen robustez en su funcionamiento, finalmente los sistemas distribuidos basan su funcionamiento en CNP o *Contract Net Protocols*, comparado con un sistema centralizado poseen mayor capacidad de resolución de tareas aunque no ofrecen la robustez necesaria.

Los sistemas de asignación de tareas los podemos dividir en basados en el mercado y optimización matemática, donde el primer sistema de asignación ofrece un buen desempeño en la optimización de tareas acompañada con una facilidad relativa en su implementación, por otro lado, los algoritmos matemáticos presentan mejor optimización para el desempeño individual pero la obtención de ecuaciones que gobiernan el algoritmo es su principal desventaja.

Capítulo III

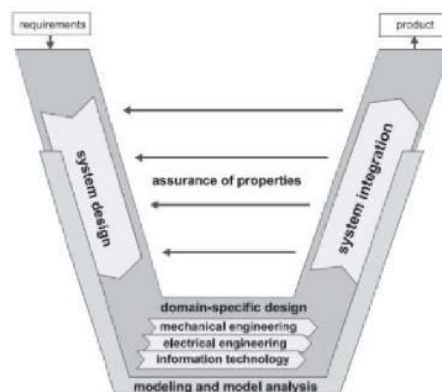
Metodología

Dentro del capítulo III se encontrará todo el desarrollo del robot, como la selección de materiales del chasis, dimensionamiento de los motores DC para el sistema de locomoción, selección de sensores, diseño de los diferentes controladores PID tanto de velocidad angular de ambos motores como controladores para orientación y seguimiento de línea recta. Finalmente se explica la organización de los algoritmos desarrollados para el correcto funcionamiento. (odometría, asignación de tareas y prevención de colisiones).

Para determinar y seleccionar los diferentes elementos que van a componer al robot móvil se planteó la selección de conceptos mediante el uso de matrices de evaluación, dichas matrices constan de una columna denominada “Peso”, “Ca” que es la columna de ponderación, y el producto de ambas da como resultado una columna de evaluación ponderada denominada “Ep”.

Figura 3

Norma VDI 2206



Nota. Tomado *The new V-Model of VDI 2206 and its validation*, por Graessler, I., & Hentze, J., 2020.

De acuerdo a como propone (Graessler & Hentze, 2020) la norma VDI 2206 (Figura 3) es empleada para el desarrollo de productos mecatrónicos, la norma consta de 3 fases que son diseño del sistema, diseño específico e integración del sistema. El desarrollo del proyecto será basado con respecto al diseño del robot y el entorno.

La norma VDI se compone de 7 etapas que son:

- Requerimientos
- Diseño de sistemas
- Dominio específico del diseño
- Integración de sistemas
- Verificación/Validación
- Modelado y análisis del modelo
- Producto

Requerimientos

Durante la etapa se determinará todos los aspectos que debe tener cada robot con el fin de cumplir con los objetivos del proyecto, a continuación, se detallaran dichos requerimientos:

- Aplicación de robótica móvil
- Implementación de un sistema de comunicación entre robots
- Asignación de un líder
- Modularidad
- Bajo costo de elaboración y producción

Diseño de subsistemas

Se propone dividir al proyecto en los siguientes subsistemas con el fin de obtener mejores resultados y tener más control sobre el proceso. El proyecto se divide en:

- Subsistema mecánico

- Subsistema electrónico
- Subsistema de control
- Subsistema de software y algoritmos

Diseño del subsistema mecánico

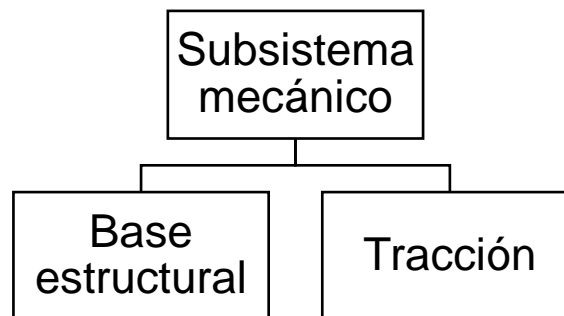
La base fundamental para cualquier diseño mecatrónico es el subsistema mecánico puesto que se planifica las dimensiones en bruto del vehículo, las cargas que va a soportar, y la ubicación de componentes.

Como se mencionó en el capítulo II los robots se pueden clasificar de dos formas de acuerdo a su tipo de locomoción (piernas o ruedas) en base a estos conceptos se plantea el uso de ruedas clásicas para la locomoción, esto gracias a su facilidad de instalación y determinación de la cinemática propia de un robot móvil con ruedas

El diseño del subsistema mecánico cuenta de dos partes:

Figura 4

División del sistema mecánico



El chasis del vehículo debe contar con ciertos parámetros para el cual se han seleccionado tres posibles materiales, entre ellos MDF que es un material hecho por fibras de madera unidas por una fibra sintética y normalmente son comprimidas con presión y calor, acrílico que es un material termoplástico rígido transparente, en su estado natural es incoloro pero puede ser pigmentado para obtener varios colores, es inerte a materiales corrosivos, y

finalmente acero, es un material que posee elasticidad y es compuesto a base de hierro y carbono.

La selección de materiales se basó en 3 principios: resistencia a la flexión, durabilidad, facilidad de maquinado, costo.

Tabla 4

Matriz de evaluación para la selección de materiales

Criterios	Peso [%]	MDF		Acrílico		Acero	
		Ca	Ep	Ca	Ep	Ca	Ep
Resistencia a la flexión	10	2	0.2	4	0.4	5	0.5
Durabilidad	10	2	0.2	4	0.4	5	0.5
Facilidad de maquinado	30	5	1.5	5	1.5	3	0.9
Costo	50	5	2.5	5	2.5	3	1.5
Total			4.4		4.8		3.4

En la Tabla 4 podemos observar que el mejor material para el propósito del proyecto es el acrílico, con este material se propone el diseño de todo el sistema estructural del vehículo, además de emplear tornillos y tuercas M3 en el caso de ser necesario. El chasis del vehículo deberá ser capaz de cargar el peso de los siguientes componentes:

- Puente H L298n (30g)
- Baterías LiPo (300g)
- Ruedas (50g)
- Jetson Nano (200g)
- Placas de potencia y control (30g)
- Sistema de tracción (100g)

Todos estos elementos serán detallados a profundidad más adelante en este documento.

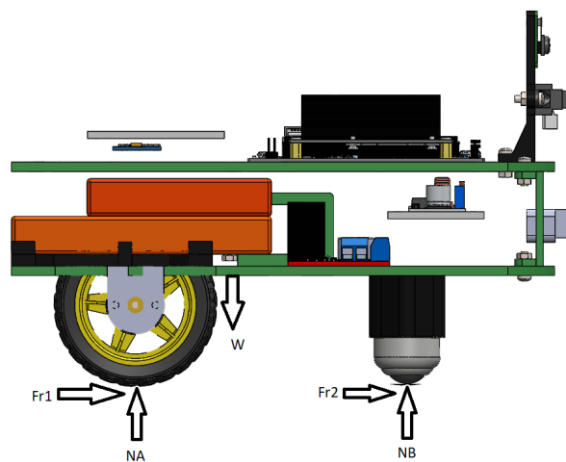
Una vez que se ha determinado el material del vehículo, se debe dimensionar todo el sistema de locomoción puesto que debe ser capaz de impulsar el vehículo a una velocidad determinada y constante. Como se mencionó anteriormente existen diferentes formas de locomoción, debido a que el objetivo es implementar un sistema de locomoción sencillo y fácil de controlar, simplemente se optó por la siguiente configuración:

- **Sistema de locomoción:** Rodante
- **Número de ruedas:** 3 (2 para tracción y 1 para estabilidad)
- **Configuración de las ruedas:** Diferencial
- **Tipo de ruedas:** Clásicas y rueda loca

Finalmente, para impulsar el vehículo se deben considerar el peso del conjunto, por lo que se estima que con el chasis y el subsistema electrónico tendrá un peso de alrededor de 1.2kg. Por lo que se plantea el siguiente diagrama:

Figura 5

Diagrama de cuerpo libre



Donde:

- F_{r_1} y F_{r_2} : Fuerza de fricción
- N_A y N_B : Fuerza normal entre el suelo y el robot
- W : Peso del vehículo

Para el dimensionamiento de los motores es importante determinar la dinámica del robot, dentro de la literatura existen dos métodos para determinar la dinámica de un robot, una de ellas es el método de Newton-Euler y la segunda forma de cálculo es por el método de Euler-LaGrange.

Para determinar la dinámica del robot por el método de Euler-LaGrange se emplearán las siguientes ecuaciones,

$$\dot{v} = \frac{1}{mr} (\tau_r + \tau_l) \quad (1)$$

$$\dot{\omega} = \frac{2a}{Ir} (\tau_r - \tau_l) \quad (2)$$

Donde:

- \dot{v} : Aceleración lineal
- $\dot{\omega}$: Aceleración angular
- τ_r y τ_l : Torque del motor derecho e izquierdo
- m : Masa del robot
- r : El radio de las ruedas del robot
- I : Inercia del robot
- a : la distancia entre ruedas

El vehículo debe moverse a velocidad constante, pero inicialmente el vehículo parte del reposo y debe llegar a una velocidad de 0.4 m/s en 0.2 segundos, por lo que la aceleración del sistema es de

$$a = \frac{0.4}{0.2} = 2 \left[\frac{m}{s^2} \right] \quad (3)$$

Como se requiere que el robot se mueva en línea recta los torques de ambas ruedas deben ser iguales, por lo que reescribiendo la ecuación 1 se obtiene que,

$$\tau = \frac{mra}{2} \quad (4)$$

$$\tau = 0.0315[N \cdot m] = 0.321[kgf \cdot cm] \quad (5)$$

De la ecuación 5 se obtiene que se requiere un motor DC capaz de entregar a las ruedas un torque de 0.3212[kgf.cm].

Dentro del mercado existen diversos tipos de motores DC, entre ellos encontramos motores a pasos (A), motores con escobillas con caja reductora (B) y servomotores (C), por lo que se plantea seleccionar un motor que sea capaz de proporcionar el torque calculado, que sea pequeño y fácil de controlar.

Tabla 5

Matriz de evaluación para selección de motores

Criterios	Peso [%]	A		B		C	
		Ca	Ep	Ca	Ep	Ca	Ep
Liviano	10	4	0.4	4	0.4	2	0.2
Alto torque	40	3	1.2	5	2.0	4	1.6
Facilidad de controlar	40	3	1.2	5	2.0	2	0.8
Costo	10	5	0.5	2	0.2	3	0.3
Total			3.3		4.8		2.9

En base a la Tabla 5 se seleccionan motores DC con caja reductora, para el proyecto se optó por comprar motores DC de la marca ADAFRUIT. A continuación, se detallan algunas características relevantes para tomar a consideración.

- Voltaje de alimentación: 7VDC
- Relación de la caja reductora: 1:19.52
- Dimensiones sin el eje: 47.3mm x 24.8mm

Integración del sistema mecánico. Anteriormente se hablo acerca del sistema de tracción o locomoción, la base estructural, los materiales, y los diversos elementos que lo conforman como son ruedas, motores y elementos de sujeción como tornillos y tuercas, ahora se hablará acerca de la integración de esos sistemas.

Subconjunto de tracción. Se compone de:

- 2 ruedas clásicas
- 1 rueda loca
- 2 motores DC con caja reductora Adafruit

Para establecer un ensamble de estos componentes se diseñó un acople entre las ruedas clásicas y los motores, dicho acople se muestra en la Figura 6.

Como se puede observar a la izquierda se encuentra el acople entre la rueda y el motor. Se puede observar que en la figura 6a, el acople se lo fabrico mediante el uso de manufactura aditiva, empleándose impresión en resina. Mientras que en la figura 6b podemos observar una vista explotada del subconjunto de tracción.

Subconjunto estructural. Este subconjunto se encuentra conformado por las siguientes piezas:

- 2 plataformas
- Elementos para sujetar el subconjunto de tracción
- Elemento de nivelación de rueda loca

Todos los elementos mencionados anteriormente se los fabrico en acrílico a excepción del último elemento que se empleó técnicas de manufactura aditiva, empleándose impresión en 3D.

En la Figura 7 se puede observar todos los elementos que conforman la base estructural (se han suprimido los elementos de sujeción, en el apéndice se detallaran las características de los mismos).

Figura 6

Sub ensamble del sistema de tracción

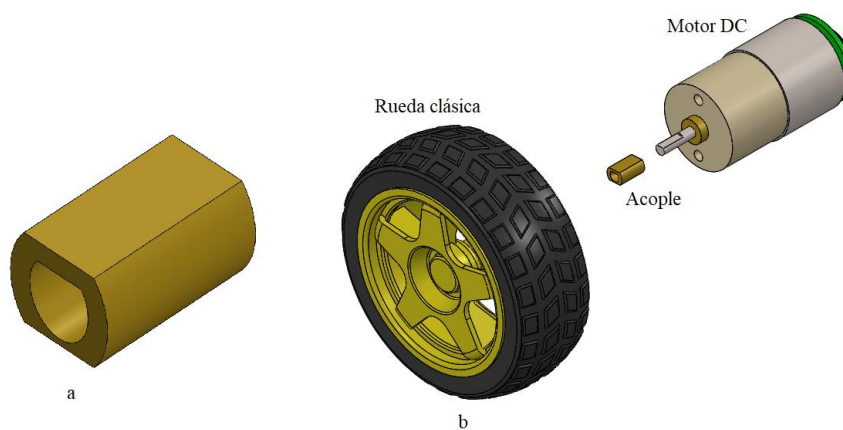
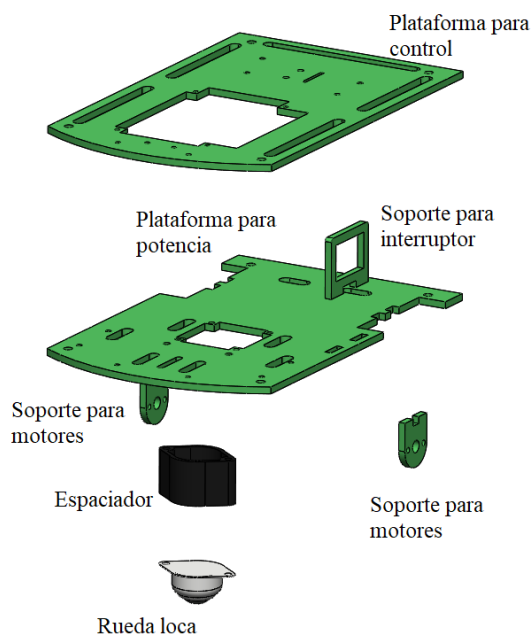


Figura 7

Subconjunto estructural

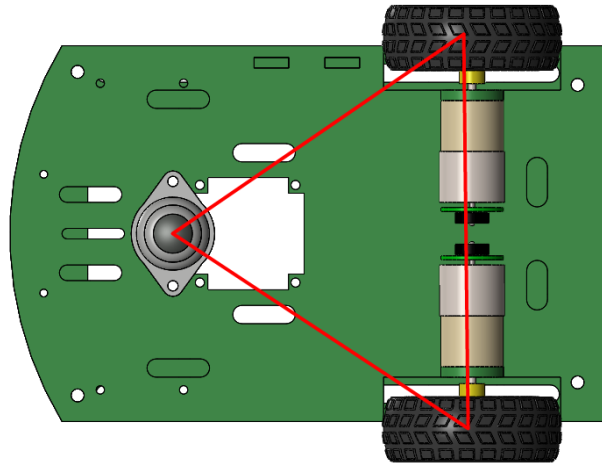


Conjunto (Subconjunto estructural + tracción). El vehículo cuenta con dos plataformas como se puede observar en la Figura 7, la plataforma inferior se encarga de albergar todos los elementos de potencia como son el módulo L298n, baterías LiPo y la placa de potencia y distribución (circuito impreso), además de brindar soporte al subconjunto de tracción, mientras que la plataforma superior se encarga de albergar los elementos de control placa de control (circuito impreso), Jetson Nano, cámara y sensores.

Por lo que el vehículo se encuentra dotado de dos pisos en la cual se buscó que el centro de masa del vehículo sea lo suficientemente bajo para que sea estable debidos a empujes laterales y que además dada la configuración de las 3 ruedas (ruedas clásicas y rueda loca) se proporcione estabilidad en los 3 puntos de apoyo como se puede observar en la Figura 8.

Figura 8

Estabilidad proporcionada por 3 puntos de apoyo



Diseño del subsistema electrónico

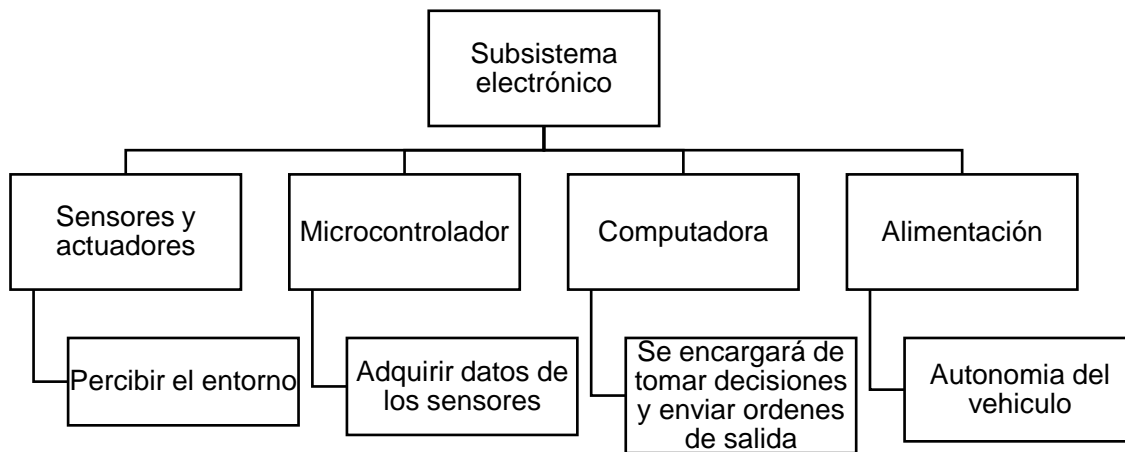
El diseño del subsistema electrónico se basa en el análisis y selección de ciertos componentes electrónicos con el fin de garantizar el óptimo funcionamiento del proyecto, para lo cual se plantean las siguientes tareas dentro del diseño del subsistema electrónico:

- Selección de sensores
- Selección de tarjeta de adquisición de datos
- Selección de la computadora
- Diseño de circuitos impresos (PCB)
- Dimensionamiento de baterías y componentes de distribución de voltaje

Se debe tomar en cuenta que en la sección anterior ya se seleccionó los motores, por lo que en la sección se seleccionará un driver para poder controlar los mismos. En la Figura 9 se detallarán mejor estos puntos.

Figura 9

Resumen del subsistema electrónico



Driver o controlador para motor DC. Parte fundamental de un robot móvil es poder controlar los motores que proporcionan el torque necesario para impulsar al robot. En base a la selección de motores que se detalló anteriormente el driver deberá tener las siguientes características:

- Corriente de consumo por motor: 145mA (en vacío)
- Alimentación del driver: 5-7 V
- Variación de velocidad por PWM
- Número de motores a controlar: 2

Con las características vistas anteriormente se presentan tres posibles drivers:

DRV8830 (A), L298n (B), TB9051ftg (C).

Tabla 6

Matriz de selección de drivers para motores DC

Criterio	A	B	C
Corriente	Si	Si	Si
Alimentación	Si	Si	Si
PWM	Si	Si	Si
Control de velocidad	No	Si	No
Inversión de giro	Si	Si	Si
Total	3	5	3

Como podemos observar en la Tabla 6 se va a emplear el Driver L298n, debido a su versatilidad, facilidad de uso, y disponibilidad en el mercado.

Sensores.

- **Medición de velocidad y posición**

Como se mencionó anteriormente el robot debe moverse con velocidad constante por lo que es necesario tener sensores que permitan desarrollar de dicha tarea, este tipo de sensores cae en la categoría de propioceptivos, es decir que miden el estado interno y la dinámica del robot móvil.

Los motores adquiridos vienen incorporados con *encoders* de efecto hall en cuadratura por lo cual procederá a trabajar con los mismos.

Los *encoders* de efecto hall se basan en detectar una alteración de un campo magnético y además como se mencionó son en cuadratura, es decir que el sensor emite dos señales digitales, una de las señales se enviará primero dependiendo del sentido de giro del motor.

Estos sensores se encuentran anclados al rotor del motor DC por lo que permiten medir la variación de posición del rotor dependiendo de cuantos pulsos envíe el *encoder* dentro de un determinado intervalo de tiempo y de forma similar se puede medir la posición relativa del robot dentro de un marco de referencia conocido.

- **Sensores de orientación**

Un parámetro muy importante dentro de robótica móvil es la orientación del robot, con este sensor y en conjunto de un sistema de posición, es posible enviar al robot a un punto dentro del plano cartesiano desde un punto inicial conocido.

Se proponen algunos sensores que permiten medir la orientación:

- Magnetómetro (A)
- Unidad de medición inercial (B)
- GPS (C)

Para la selección de un sensor de orientación es imprescindible que sea preciso, sea compatible con el microcontrolador, indique la orientación absoluta del robot.

Tabla 7

Matriz de selección para sensores de orientación

Criterios	Peso [%]	A		B		C	
		Ca	Ep	Ca	Ep	Ca	Ep
Precisión	30	4	1.2	4	1.2	3	0.9
Orientación absoluta	30	1	0.3	5	1.5	4	1.2
Compatibilidad	40	5	2.0	5	2.0	5	2.0
Total			3.5		4.7		4.1

Como observamos en la Tabla 7 el sensor de orientación que mejor se acomoda para el proyecto es una Unidad de Medición Inercial, para nuestro caso usaremos el sensor MPU 6050

que es un giroscopio y acelerómetro de 6 grados de libertad que va a medir la aceleración lineal y las velocidades angulares en los 3 ejes principales, debido a que se requiere medir la variación de posición angular se requiere integrar la velocidad angular en el eje Z.

$$\omega = \frac{d\theta}{dt} \quad (6)$$

$$\theta = \theta_o + \omega \cdot \Delta t \quad (7)$$

Como observamos en la ecuación 7 se puede determinar la orientación del robot conociendo la velocidad angular del robot en un instante dado.

Sensor exteroceptivo

Los sensores exteroceptivos proporcionan información del entorno como puede ser temperatura, humedad, obstáculos, etc. dentro de robótica móvil es importante adquirir información del entorno puesto que el robot deberá ser capaz de evitar colisiones, por lo que se requiere de algún sensor que sea capaz de proporcionar información acerca de algún cuerpo externo al robot.

Se pueden emplear sensores ópticos que se valen de diodos infrarrojos para la detección de objetos y sensores ultrasónicos que se valen del envío y recepción de señales de ultrasonido y estimar cuando tiempo se demora en regresar dicha onda para así determinar la presencia de un objeto.

Ambos sensores cumplen bien con su función, pero el objetivo del robot es tener una respuesta rápida ante un obstáculo, que pueda trabajar en rangos de detección pequeños (10-15cm) y que además no requiera demasiados pines para su funcionamiento.

Tabla 8

Matriz de selección para sensores de detección externa

Criterios	Peso [%]	Óptico		Ultrasónico	
		Ca	Ep	Ca	Ep
Respuesta rápida	20	5	1.0	5	1.0
Rango de detección	30	5	1.5	5	1.5
# de pines	50	5	2.5	3	1.5
Total			5.0		4.0

En la Tabla 8 se selecciona el sensor óptico debido a que solamente requiere de un pin para la detección de obstáculos que a diferencia del sensor ultrasónico requiere de dos pines para su funcionamiento (se obvia los pines de alimentación para ambos casos).

Para el proyecto se empleó el sensor óptico SHARP que es un sensor infrarrojo que envía una señal analógica hacia la tarjeta de adquisición de datos, el valor analógico que envía es inversamente proporcional con la distancia del objeto, es decir que entre más alto sea el valor leído por el ADC de la tarjeta de adquisición de datos (que será discutido más adelante en este documento) menor será la distancia entre el objeto y el sensor.

Una vez seleccionado todos los sensores necesarios para el funcionamiento del robot se procede a determinar que tarjeta se usara para poder enlazar todos estos sensores propioceptivos y exteroceptivos.

Tarjeta de adquisición de datos. Según (Enríquez et al., 2016) las tarjetas de adquisición de datos son dispositivos diseñados para la adquisición y medición de señales que consiste en la toma de muestras de variables físicas del mundo real para generar datos que puedan ser manipulados por un programa dentro de una tarjeta de adquisición de datos.

En secciones anteriores ya se discutió que clase de sensores se va a emplear en el robot móvil, ahora se deberá identificar qué tipo de señales se va a recibir la tarjeta de adquisición de datos.

Tabla 9

Tipos de señales de los sensores

Sensor	Tipo de señal	Variable a medir
<i>Encoder</i> efecto hall	Digital	Velocidad angular y posición de las ruedas del robot
Unidad de medición inercial	I2C	Orientación del robot
Sensor de detección	Analógico	Detección de obstáculos

En la Tabla 9 se detallan los tipos de señales que cada sensor envía, notamos que la Unidad de Medición Inercial tiene un tipo de comunicación I2C, por lo que la tarjeta que se seleccione deberá contar con esa característica para poder emplear dicho sensor.

Otra característica importante que debe tener la tarjeta de adquisición de datos es que tenga salidas PWM (Modulación por ancho de pulsos), puesto que debe ser capaz de controlar la velocidad de los motores y finalmente la tarjeta también deberá contar con varios pines disponibles.

Dentro del mercado se encuentran varias tarjetas de código abierto para su uso, las más populares son ARDUINO MEGA (A), ESP32 (B), ESP8266 (C).

Tabla 10

Matriz de selección para tarjeta de adquisición de datos

Criterios	Peso [%]	A		B		C	
		Ca	Ep	Ca	Ep	Ca	Ep
Pines de interrupción	10	3	0.3	5	0.5	4	0.4
Salidas PWM	20	5	1.0	5	1.0	4	0.8
Comunicación I2C	20	5	1.0	5	1.0	5	1.0
<i>Open Source</i>	10	5	0.5	5	0.5	5	0.5
Disponibilidad de pines	20	5	1.0	5	1.0	3	0.6
ADC	10	5	0.5	5	0.5	4	0.4
Tamaño	10	1	0.1	5	0.5	4	0.4
Total			4.4		5.0		4.1

De todas las tarjetas previamente nombradas se escoge a la tarjeta ESP32, no solamente por dichas características, sino que también es una tarjeta de 32 bits y posee gran capacidad de procesamiento de datos, su reducido tamaño y que integra tecnología Wifi en su arquitectura, por lo que dichas características la hacen una tarjeta rápida y confiable.

En base al *datasheet* del chip ESP32 se establece que tiene una frecuencia de trabajo entre 80 a 240 MHz, este valor es importante puesto que más adelante se hablara acerca de estrategias de control y saber la frecuencia del CPU es importante puesto que los *encoders* trabajan con interrupciones por lo que se pierde información al momento de realizar el muestreo de dicha señal.

Unidad de control. El proyecto consiste en que los robots puedan ser capaces de ordenarse de acuerdo a una métrica previamente definida que se hablará más adelante en el proyecto, enviar y recibir mensajes para la comunicación entre los robots, y que sean capaces de conectarse a internet para establecer el protocolo MQTT, como se dijo en la sección anterior, la tarjeta ESP32 viene incorporada con wifi, pero el coste de procesamiento de las tareas mencionadas anteriormente resulta muy alto, por lo que es imprescindible el uso de una unidad de control.

Dentro del mercado existen diversas placas de micordenadores que también son *open source*, ya que se desarrollan con software de uso libre. A continuación, se detallan las características más importantes de dos computadoras muy comúnmente empleadas

Tabla 11

Características de micordenadores

	Jetson Nano	Raspberry PI 4B
	Procesador ARM	Procesador ARM
CPU	Cortex-A57 MPCore de cuatro núcleos	Cortex-A57 MPCore de cuatro núcleos
GPU	NVIDIA Maxwell	Broadcom Videocore
Memoria RAM	4GB	4GB
USB	2x USB 3.0	4x USB 3.0
GPIO	40	40
Almacenamiento	Micro SD	Micro SD
Conexión a internet	Gigabit ethernet	Gigabit ethernet

Como podemos notar en la Tabla 11 entre las dos opciones no hay mucha diferencia en tanto a sus características, salvo en una que es la GPU, la unidad de control Jetson Nano

cuenta con gran procesamiento de imágenes, mientras que la Raspberry PI 4B no, la característica es deseada puesto que se implementará sistemas de reconocimiento para evitar colisiones.

Alimentación. Como todo dispositivo electrónico y eléctrico requiere de una fuente de alimentación, es importante dimensionarla y ajustarla acorde a la corriente necesaria y a cuánto tiempo deberá operar el robot (autonomía).

Tabla 12

Voltajes y corrientes de sensores y actuadores en el robot móvil

Componente	Cant.	Voltaje	Corriente
		[V]	[mA]
Motor DC	2	7	~145
<i>Encoder</i>	2	5	20
MPU6050	1	5	4
L298n	1	7	40
ESP32	1	5	~600
Jetson Nano	1	5	~1000

En base a la Tabla 12 se sacan las siguientes conclusiones:

- Tanto los motores como el driver requieren de alrededor de 7V.
- El resto de los componentes requiere 5V.

Reguladores de voltaje lineales y conmutados. Los reguladores de voltaje son un conjunto de elementos electrónicos que permiten tener un voltaje en la entrada que puede variar debido a la descarga natural de las baterías y a su salida mantener el voltaje constante. Los reguladores se pueden dividir en reguladores de voltaje lineales y reguladores de voltaje conmutados.

De acuerdo a (Erickson & Maksimović, 2001) los reguladores lineales ya sean en serie o paralelos trabajan de una misma manera, el elemento de control es un transistor que opera como una resistencia variable que es accionada por la señal de error que surge de comparar la tensión de salida con una referencia. Cuando un regulador lineal debe suministrar corrientes muy altas su eficiencia ronda el 40%.

En los reguladores de voltaje conmutados se emplea un transistor que opera en corte y saturación y que periódicamente aplica a la carga toda la tensión no regulada a través de un inductor por cortos intervalos de tiempo, de forma que se garantiza al menos una eficiencia del 85% a la salida del regulador de voltaje. En base a estos conceptos de regulación de voltaje se opta por emplear un regulador de voltaje conmutado de tipo *Step Down*.

Integración de componentes. Una vez seleccionado todos los componentes electrónicos del robot móvil se procede a dimensionar baterías y diseñar los circuitos impresos.

El diseño de la placa de potencia contará con dos baterías de tipo LiPo de dos celdas que proporcionaran un voltaje de 7.4V con el fin de cumplir con los requerimientos de voltaje expuestos en la Tabla 12.

Figura 10

Distribución de voltaje

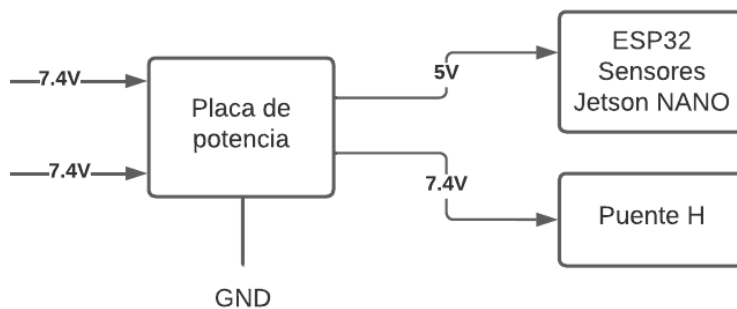
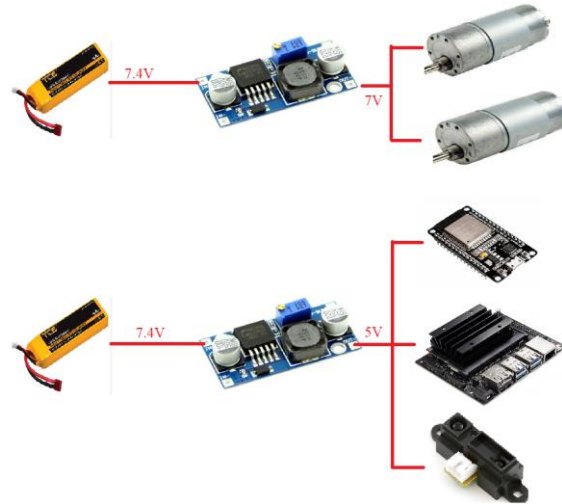


Figura 11

Distribución de voltajes



Como observamos en la Figura 10 y en la Figura 11 se van a emplear dos reguladores conmutados tipo *Step Down*, dichos reguladores conmutados están controlados por el circuito integrado XL4015 que en conjunto con su placa puede otorgar hasta 5A. Las baterías para alimentar el circuito serán de 7.4V, es decir serán baterías LiPo de dos celdas (2s).

Diseño de la placa de potencia. La placa será la encargada de distribuir los voltajes a los diferentes elementos, así como de regular el voltaje que cada elemento requiere en base a sus especificaciones técnicas.

Análisis de corriente para batería 1.

En la batería 1 se conectará el driver para motor DC, por lo tanto, en base a la Tabla 12 el consumo de corriente es el siguiente.

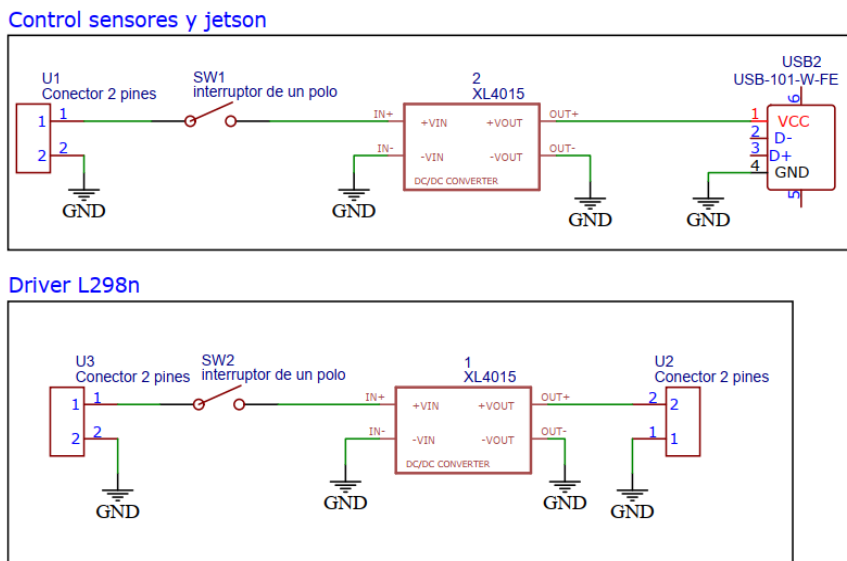
Tabla 13

Consumo de corriente de la placa de potencia

Componente	Cant.	Voltaje	Corriente	Potencia
		[V]	[mA]	[W]
Motor DC	2	7	~145	2.02
L298n	1	7	40	0.28
Total			185	2.3

Figura 12

Esquema de la placa de potencia



La implementación de la placa de potencia consta de los siguientes componentes:

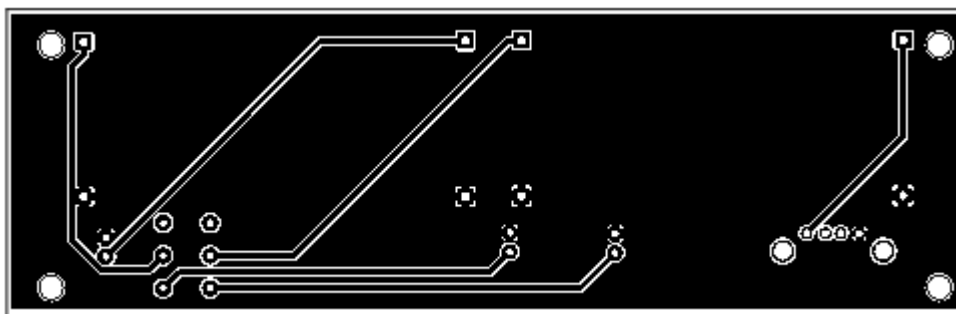
- 3 conectores MOLEX de 2 pines
- 1 interruptor DPST
- 2 reguladores conmutados XL4051

- 1 puerto USB hembra

Con la configuración de componentes descrita, se obtiene una placa totalmente modular y de fácil implementación.

Figura 13

Pistas para placa de potencia



Consumo de batería 1

Como se mencionó anteriormente se plantea usar baterías de 2s a 7.4V y una capacidad de 1500mAh

$$T_{b1} = \frac{1500[mAh]}{185[mA]} \quad (8)$$

$$T_{b1} = 8.1[h] \quad (9)$$

En base a la ecuación 9 se espera que los motores funcionen por al menos 8 horas consecutivas.

Diseño de la placa de control. La placa será la encargada de distribuir los voltajes a los diferentes elementos y manejar señales de control ya sean digitales o analógicas.

Análisis de corriente para batería 2.

En la batería 2 se conectarán los sensores, tarjeta de adquisición de datos y la unidad de control, en base a la Tabla 12, el consumo de corriente es el siguiente.

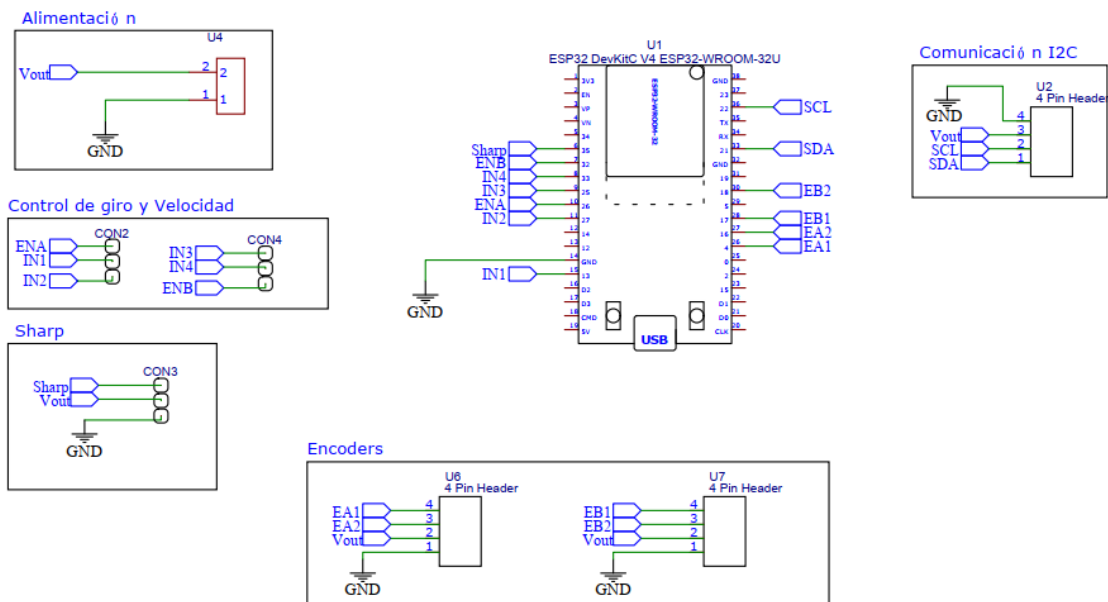
Tabla 14

Consumo de corriente de la placa de control

Componente	Cant.	Voltaje	Corriente	Potencia
		[V]	[mA]	[W]
Encoder	2	5	20	0.2
MPU6050	1	5	4	0.02
ESP32	1	5	~600	3
Jetson Nano	1	5	~1000	5
			1644	8.22

Figura 14

Esquema de la placa de control



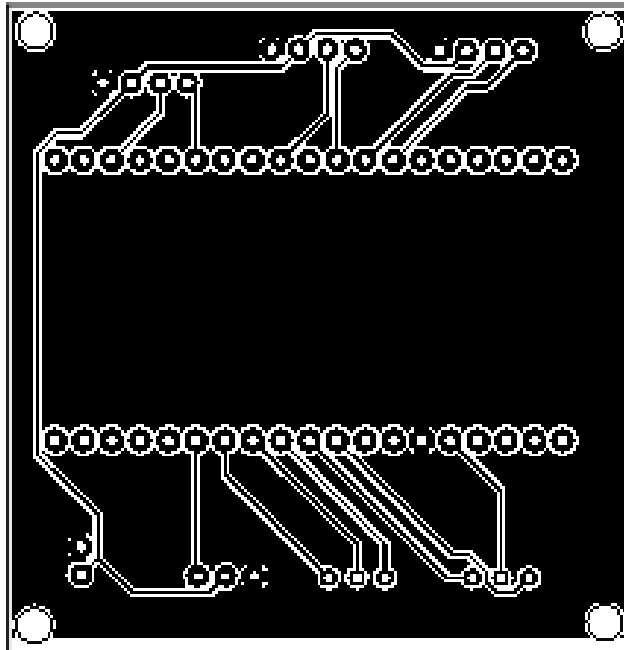
La implementación de la placa consta de los siguientes componentes:

- 1 conectores MOLEX de 2 pines
- 3 conectores MOLEX de 3 pines
- 3 conectores MOLEX de 4 pines
- 1 ESP32
- Regleta hembra

Con la configuración de componentes se obtiene una placa totalmente modular y de fácil implementación.

Figura 15

Pistas para placa de control



Consumo de batería 2

Como se mencionó anteriormente se plantea usar baterías de 2s a 7.4V y una capacidad de 2200mAh

$$T_{b2} = \frac{2200[mAh]}{1644[mA]} \quad (10)$$

$$T_{b2} = 1.33[h] \quad (11)$$

En base a la ecuación 11 se espera que todos los elementos de la Tabla 14 funcionen por al menos 1.33 horas o 1 hora con 20 minutos aproximadamente.

Comparando los tiempos entre la batería 1 y la batería 2 se espera que el robot tenga una autonomía de al menos 1.33 horas o una hora y veinte minutos (1:20)

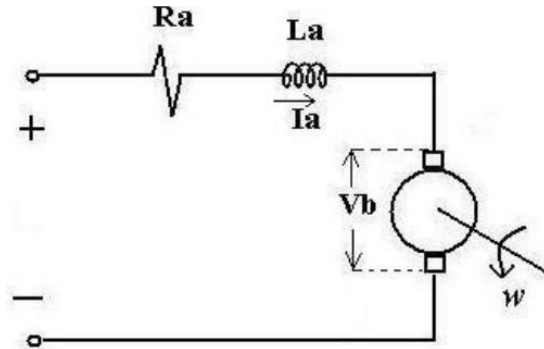
Diseño del subsistema de control

El control automático se ha convertido en una parte importante e integral en los sistemas de vehículos espaciales, en los sistemas robóticos, en los procesos modernos de fabricación y en cualquier operación industrial que requiera el control de temperatura, presión, humedad, flujo, etc. (Ogata, 2003)

Para el desarrollo del proyecto es imprescindible implementar un controlador para algunas variables presentes en el robot diferencial, una de ellas y que es la más importante, es realizar el control de velocidad de ambos motores ya que la variable permite controlar la orientación y posición del robot.

Figura 16

Circuito equivalente de un motor DC



Nota. Tomado de *Control PID para el control de velocidad de un motor DC*, por Gil Garcés, L. A., & Rincón Gaviria, J. L., 2014.

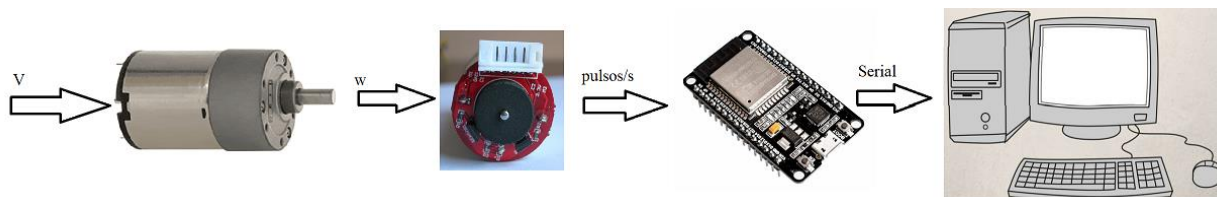
Como observamos en la Figura 16 existen dos partes fundamentales dentro del circuito que son el circuito eléctrico y la parte mecánica del motor, existen varios métodos para determinar la función de transferencia del motor, dado que no es pertinente la demostración de la función de transferencia en este trabajo de titulación se procede a mostrar solamente la función de transferencia. (Garces & Gaviria, 2014)

$$\frac{\omega(s)}{V(s)} = \frac{k_m}{LJs^2 + (RJ + LB)s + (RB + k_m k_a)} \quad (12)$$

En la ecuación 12 notamos que se requiere saber de algunos parámetros electromecánicos del motor que no es tan sencillo determinar, por lo que se planteó un reconocimiento de la planta empleando los sensores de efecto hall y realizando un algoritmo para determinar las revoluciones por minuto del motor. Para realizar el reconocimiento de planta se plantea el siguiente diagrama de bloques en lazo abierto

Figura 17

Muestreo de la planta en lazo abierto



En donde se tiene una entrada V que corresponde a un voltaje que posteriormente es transformado a movimiento rotatorio; el cual es leído por un *encoder* de efecto hall, que posteriormente cuenta cuantos pulsos ha dado en un intervalo de tiempo dentro de la tarjeta ESP32 y que finalmente es leído por el puerto serial de la computadora.

Parte importante para el cálculo de revoluciones por minuto del motor es transformar el conteo de pulsos proporcionado por la tarjeta de adquisición de datos, para ello se debe conocer la relación de transmisión de la caja reductora del motor y cuantos pulsos cuenta el *encoder* por cada revolución

$$ppr = 12 \left[\frac{pulsos}{rev} \right] \quad (13)$$

$$i = 19.52 \quad (14)$$

En donde ppr e i son los pulsos por revolución dados por el *encoder* y la relación de transmisión entre el eje del rotor del motor y el eje de salida de la caja reductora respectivamente. El valor de i se lo debe entender como, por cada revolución que da el eje de salida de la caja reductora, el rotor del motor da 19.52 revoluciones, con los valores establecidos en (13) y (14) se procede a calcular el factor de conversión k de pulsos por segundo a revoluciones por minuto

$$k = 0.256 \quad (15)$$

Con los valores calculados previamente se procede a realizar el muestreo de ambos motores con un algoritmo que emplea el método por mínimos cuadrados para identificación de plantas que se basa en aproximación por mínimos cuadrados, a continuación, se muestran los coeficientes de la función de transferencia para cada motor.

$$G(z) = \frac{Az^{-1}}{1 + Bz^{-1} + Cz^{-2}} \quad (16)$$

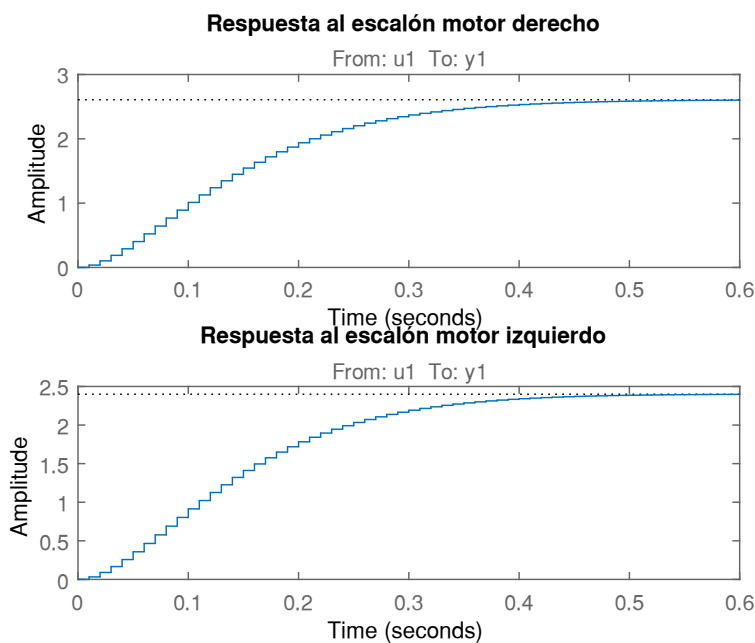
Tabla 15

Coeficientes de los motores izquierdo y derecho

	A	B	C
Motor derecho	0.0368	-1.769	0.783
Motor izquierdo	0.0322	-1.781	0.794

Figura 18

Respuesta al escalón en lazo abierto para los motores DC



Como se dijo en la sección de “Diseño del subsistema mecánico” el robot deberá moverse a una velocidad constante de $0.4 \left[\frac{m}{s} \right]$, por lo tanto,

$$\omega_d = \frac{v}{R} \quad (17)$$

$$\omega_d = 11.9 \left[\frac{rad}{s} \right] = 113.6 [rpm] \quad (18)$$

Donde:

- ω_d : Velocidad angular de la rueda derecha
- v : Velocidad lineal del robot diferencial
- R : Radio de la rueda

Así se establece el *setpoint* de velocidad angular para ambos motores.

Diseño del control PID para la velocidad angular de ambos motores. Las características de diseño del controlador PID para los motores es que no tengan sobre pico en el estado transitorio y que el tiempo de estabilización de ambos motores sea de aproximadamente 0.2 segundos, puesto que con ese tiempo de estabilización se puede alcanzar el torque requerido para cambiar el estado estático del robot a uno dinámico.

El controlador PID se diseñó en base a los parámetros de Ziegler – Nichols para el cual se requiere conocer las ganancias ultimas de ambas plantas.

Tabla 16

Cálculo de constantes para PID

	Kp	Ki	Kd
PID	$0.6K_u$	$\frac{1.2K_u}{T_u}$	$0.075K_uT_u$

De donde se obtienen las siguientes constantes para ambos motores

Tabla 17

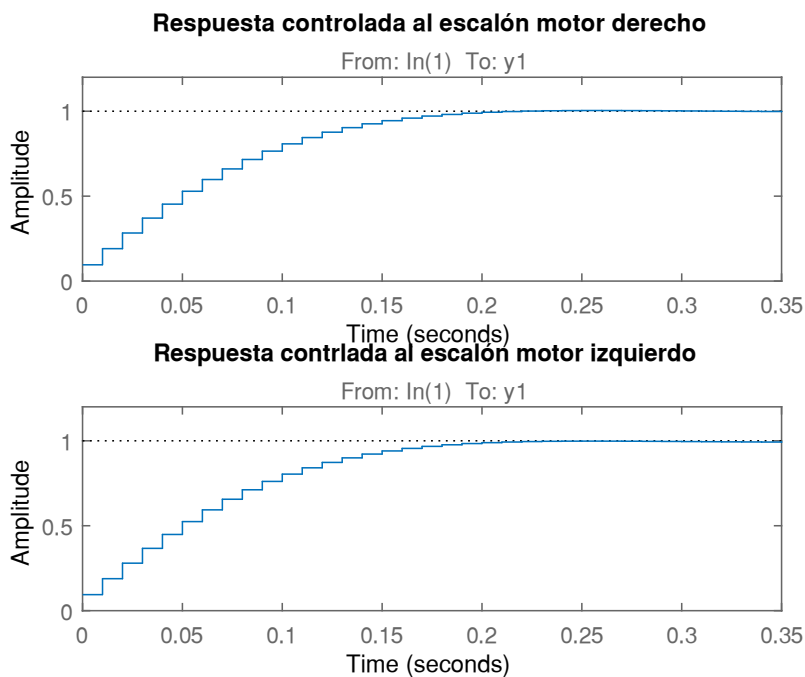
Constantes para PID de ambos motores

	Kp	Ki	Kd
Motor derecho	0.945	6.172	0.0288
Motor izquierdo	1.028	6.444	0.0323

Se nota una ligera diferencia en las constantes, pero esto se debe a que los motores electromecánicamente no son iguales, por lo que el motor derecho requiere de constantes más altas para estar en equilibrio con el motor izquierdo.

Figura 19

Respuesta al escalón en lazo cerrado para los motores DC



De la Figura 19 notamos que ambos motores responden de igual manera por lo que el arranque será controlado y en el tiempo planificado en la ecuación 1, debido a que alcanza la velocidad deseada en 0.2 segundos.

Diseño de controlador PID de orientación (seguimiento en línea recta). Dado que el robot debe llegar a un punto final siguiendo una trayectoria rectilínea se plantearon algunos modelos de controlador, uno de ellos es el uso constante de la unidad inercial IMU para detectar cualquier variación de posición angular mientras el robot se dirige hacia su ubicación final, pero debido a la falta de amortiguación del robot y las irregularidades del piso, estas producen vibraciones en el sensor lo que provoca errores de lectura considerables.

Finalmente se planteó el uso de los propios encoders para determinar si el robot ha variado su orientación, entonces se plantea que la variable de control para este PID sea la siguiente,

$$\Delta p = p_d - p_i \quad (19)$$

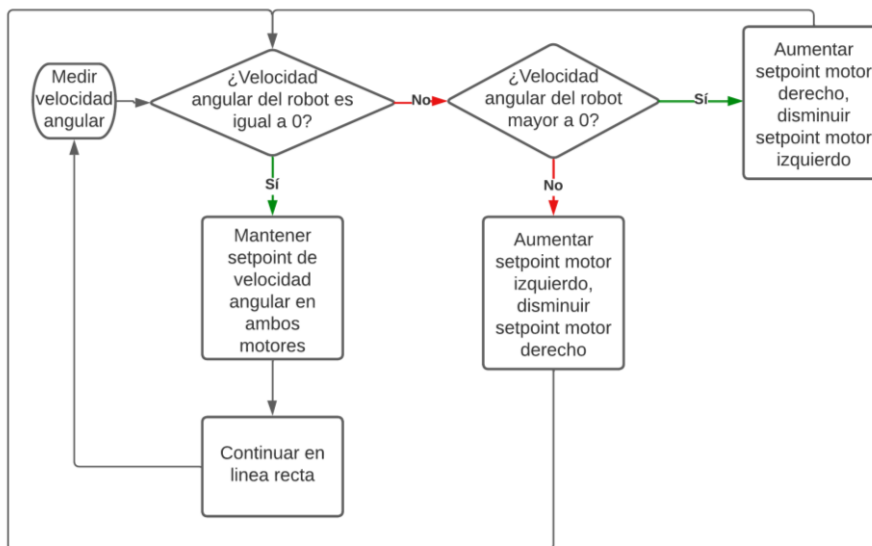
Donde:

- Δp : Variación de pulsos
- p_d : Pulsos de la rueda derecha
- p_i : Pulsos de la rueda izquierda

De esta forma si $\Delta p > 0$ el robot ha tenido un cambio de orientación positivo, mientras que si $\Delta p < 0$ el robot ha tenido un cambio de orientación negativo, lo que se busca es el conteo de pulsos de ambas sea igual, por lo que se espera que $\Delta p = 0$.

Figura 20

Diagrama de flujo para corrección de trayectoria



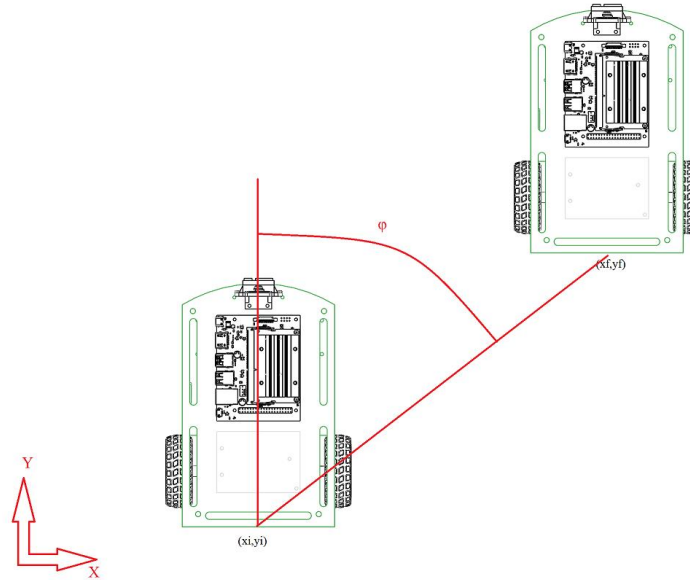
El controlador para este sistema se diseñó en base a un controlador en cascada donde se establece que la señal de control del PID de orientación no es un valor para PWM, sino que es una variación al punto de control del PID de velocidad angular de ambos motores.

El diseño para esta compensación del *setpoint* de velocidad angular de ambos motores se lo realizó en base a prácticas de prueba y error, de esta forma primero se sintonizó un controlador tipo proporcional (P) hasta que el sistema responda rápidamente y se acerque al *setpoint* de orientación (0°), después un control proporcional-derivativo (PD) de esta forma se estabiliza el sistema y se eliminan sobre picos y finalmente se establece un controlador proporcional-integral-derivativo (PID) con el fin de eliminar errores en estado estable.

Diseño de controlador PID de orientación (orientación hacia punto final). Como el robot aparte de avanzar en línea recta, también debe orientarse para ir a su punto final, es decir desde su posición actual debe llegar a un punto final debe girar sobre su centro de giro para orientarse, como se muestra a continuación.

Figura 21

Orientación del robot para llegar a su punto final



Para realizar la maniobra de giro se hace uso de una unidad inercial para determinar el ángulo de giro, esto se realiza mediante la integración de la velocidad angular del robot (La unidad inercial solo detecta aceleraciones y velocidades angulares en los tres ejes principales), para determinar variaciones de posición angular se usó la siguiente expresión

$$\phi_i = \phi_{i-1} + \frac{1}{2}(\omega_i + \omega_{i-1})dt \quad (20)$$

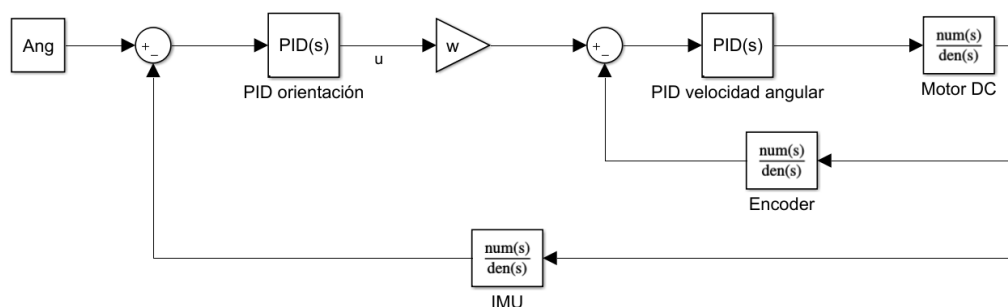
De la ecuación 20 observamos que el segundo término corresponde a la integración de la velocidad angular por el método del trapecio, de esta manera se puede obtener una realimentación para el diseño del PID.

Como se observa en la Figura 21 el robot debe girar un ángulo ϕ para llegar a (x_f, y_f) por lo que se requiere de un bloque a parte para controlar el giro del robot sobre su propio eje, por lo que se ha propuesto otro controlador de tipo cascada, pero a diferencia del anterior es

que ahora el *setpoint* para cada motor será positivo y otro negativo, pero de igual magnitud para ambos y esto variará dependiendo si el robot tiene que girar en sentido horario o anti horario.

Figura 22

PID de orientación sobre el centro de giro



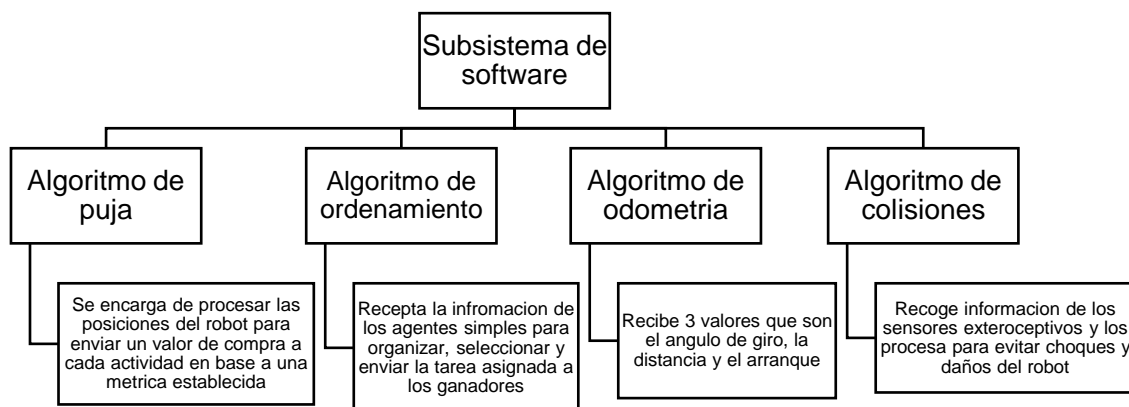
De la Figura 22 se observa que de entrada es una constante llamada *Ang* esta constante es el ángulo de giro requerido para llegar a (x_f, y_f) desde (x_i, y_i) , mediante el primer bloque de PID se obtiene una señal u que multiplica por una ganancia w , w se encuentra configurada a 70 revoluciones por minuto por lo que, al entrar al bloque sumador se tiene el *setpoint* de velocidad angular de un motor DC. La señal u solo puede tomar valores entre 0 y 1.5, por lo que la velocidad angular podría variar desde 0 a 112.5 rpm, de esta forma es posible detener los motores y obtener un giro muy preciso.

Diseño del subsistema de software

Dentro del subsistema de software se plantea el desarrollo de diversos algoritmos con el fin de obtener resultados óptimos para el funcionamiento colectivo e individual de los robots, dentro de esta subsección se plantean varios algoritmos y cada uno tiene sus propios requerimientos que más adelante se los detallará (Figura 23). Así mismo se representará con diagramas de flujo el funcionamiento en conjunto e individual dependiendo el caso.

Figura 23

Resumen del subsistema de software



Adquisición y procesamiento de información.

Jetson Nano: Esta tarjeta de control está encargada de realizar el enlace de comunicación con la Reina y sus homólogos, posee funciones esenciales para el funcionamiento individual, además de ser la encargada del control del movimiento a partir del envío de datos para el subsistema de odometría.

- **Puja de tareas:** Entregar un valor de “compra” para cada actividad disponible para pujar a partir de su posición inicial.
- **Comunicación con agentes externos:** Enviar y recibir la información requerida al agente Reina o datos de movimiento al agente simple siguiente en la jerarquía.
- **Comunicación con subsistema odometría:** Usado como enlace con el subsistema de odometría en la tarjeta ESP32, se envía información de distancia y ángulo para el movimiento.

- **Reconocimiento de objetos:** Debido a su potencia de procesamiento se acopla una cámara para realizar reconocimiento del entorno, información utilizada para decidir sobre si es posible continuar la trayectoria sin accidentes.

ESP32: Usada como tarjeta de adquisición de datos recibe la información de los sensores relacionados con el movimiento como IMU, *encoder* de los motores y sensor laser. Se almacenan las funciones para realizar el movimiento del robot diferencial y corregir su trayectoria.

- **Adquisición de información:** Obtiene la información de los *encoders* de efecto hall acoplados a los motores del robot diferencial, la unidad inercial (MPU 6050) y el sensor laser
- **Procesamiento de datos:** Con el uso de los paquetes desarrollados se realiza la conversión de variables análogas a digitales como ángulo de giro distancia recorrida o distancia frontal con objetos en el entorno.
- **Control de variables:** Con los datos obtenidos del sensor inercial se verifica el seguimiento de la trayectoria y se corrige en caso de ser necesario modificando la velocidad de los motores la cual esta monitoreada por sus *encoders*.

PC: Al ser un sistema tipo enjambre necesitamos un agente que sirva como enlace para realizar la comunicación con los agentes simples, este computador sirve para envió de coordenadas para realizar movimiento, además de actuar como intermediario para escuchar las ofertas de cada actividad y asignar cada una al mejor postor.

- **Ordenamiento:** Recibe las ofertas por las actividades de los agentes activos actuando como anunciador, mediante subasta se selecciona a la oferta adecuada para cada ítem, finalmente se comunica a los ganadores la tarea asignada.

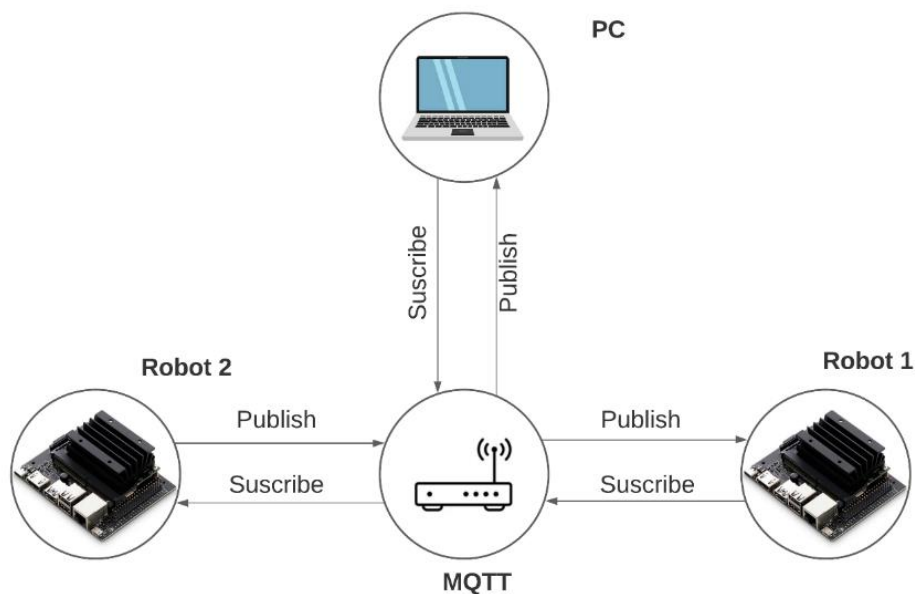
- **Enlace Humano-maquina:** Para realizar el movimiento como problema de juego se requiere que el operador introduzca las coordenadas de movimiento para el tren de robots diferenciales.
- **Comunicación con los agentes:** Sirve como enlace con los agentes para pedir la información requerida para las funciones internas.

Comunicación entre los dispositivos.

Comunicación PC – Jetson Nano El enjambre cuenta inicialmente con 3 agentes participantes los agentes simples dotados de una tarjeta de control Jetson nano, la cual además de encargarse de la comunicación con el subsistema de odometría esta enlazada con la Reina para enviar y recibir información para la organización de los agentes simple y su movimiento. El principal requerimiento para realizar la comunicación es que debe ser inalámbrica por lo cual se opta por el uso del protocolo MQTT el cual se inicia en los dispositivos cuando es requerido el intercambio de información. En conjunto con la plataforma de ROS se lleva registro los tópicos para realizar publicaciones y suscripciones para obtener un mensaje de respuesta a cada solicitud realizada, en la Figura 24 se representa la comunicación entre agentes.

Figura 24

Comunicación entre los agentes participantes



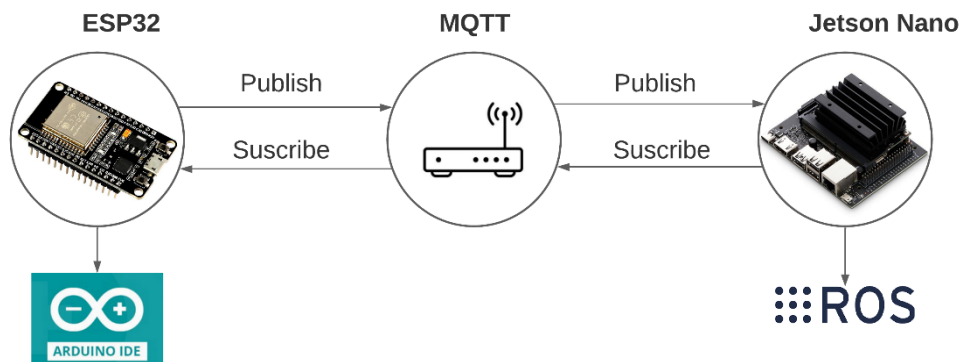
Comunicación Jetson Nano y ESP32 Debido a la búsqueda de modularidad en los sistemas que componen a los agentes se establece una separación entre la tarjeta de control dedicada para cálculos complejos como reconocimiento de objetos aprovechando así sus características detalladas en la Tabla 11, y la tarjeta ESP32 es usada para adquisición de datos.

El protocolo MQTT presenta ventajas como escalabilidad e independencia de funcionamiento, por lo cual es elegida para realizar la comunicación entre los dispositivos.

Para la configuración de cada dispositivo nos valemos de las herramientas que se acoplen mejor a cada uno, en el caso de la ESP32 se realiza la conexión programada con la ayuda del Arduino IDE, para la Jetson Nano hacemos uso de Python la cual permite la colaboración de la plataforma ROS con MQTT.

Figura 25

Comunicación entre tarjeta de control y DAQ



Selección de método de subasta. En base a los parámetros mostrados en la Tabla 18 y como se revisó en el estado del arte poseemos varios métodos de organización de los cuales describiremos a continuación,

Tabla 18

Variables para selección del ordenamiento

Variable	Tipo de ordenamiento	Descripción
A	Secuencial	Para este algoritmo se vende únicamente una tarea por cada ronda, la selección de una tarea se la puede realizar por diversos métodos.
B	Paralelo	En una única ronda de subasta se asignan todas las actividades simultáneamente.

Variable	Tipo de ordenamiento	Descripción
C	G-prime	Se subasta una única tarea por ronda, aunque se ofertan varias, durante cada rondase puja por la tarea no vendida de mayor valor.
D	Optimización	Para encontrar las posibles soluciones se aplican soluciones matemáticas restringidas por criterios y constantes los cuales ayudan a seleccionar la solución optima
E	Combinado	Para cada oferta de los compradores el anunciante calcula la forma óptima de división de tareas por cada equipo

Para el análisis y selección del método nos ayudaremos de la tabla con los criterios y variables en donde cada criterio tendrá un peso en la selectividad de las opciones presentadas anteriormente.

Finalmente, después del análisis realizado en la Tabla 19 se opta por implementar el método de subasta paralela pues cumple ampliamente los requisitos planteados para el proyecto, es importante remarcar que el método por optimización debido a la complejidad para obtener una ecuación que represente el problema planteado dificulta su aplicación, motivo por el cual es descartada.

Tabla 19

Selección del método para organización

Criterios	Peso [%]	A		B		C		D		E	
		Ca	Ep	Ca	Ep	Ca	Ep	Ca	Ep	Ca	Ep
Rapidez relativa en el procesamiento	15	3	0.45	4	0.6	3	0.45	5	0.75	4	0.6
Menor número de ronda de anuncio posibles	20	1	0.2	5	1	2	0.4	4	0.8	4	0.8
Mayor número de asignaciones por ronda	30	3	0.9	4	1.2	3	0.9	5	1.5	5	1.5
Simplicidad en su aplicación	35	4	1.4	4	1.4	4	1.4	2	0.7	3	1.05
Total			3.3		4.2		3.15		3.75		3.95

Algoritmo de ordenamiento. Para abordar la asignación de tareas se realiza un acercamiento a través de un problema de juego de seguimiento, por lo tanto, es necesario tomar en cuenta las ubicaciones iniciales y las posiciones de las tareas predefinidas a donde se moverán los agentes para iniciar el seguimiento entre los agentes para poder generar la ruta de movimiento, la subasta consta fundamentalmente de 2 etapas:

- La primera consiste en anunciar la tarea y recibir las ofertas de los agentes.
- La segunda etapa involucra al anunciador el cual recoge las ofertas, decide la mejor opción de venta y finalmente comunica al ganador el cual no podrá participar en futuras subastas.

En la Tabla 20 se mencionan los criterios considerados para la selección del ordenamiento.

Tabla 20

Criterios de selección para el ordenamiento

Criterios para el ordenamiento
Rapidez relativa en el procesamiento
Menor número de ronda de anuncio posibles
Mayor número de asignaciones por ronda
Simplicidad en su aplicación

El método de ordenamiento como se describe en la Tabla 20 debe cumplir los criterios para que en el funcionamiento práctico al momento de generar las trayectorias de movimiento no se demoren en llegar a sus posiciones de partida para iniciar el seguimiento, por lo tanto debe haber una rapidez relativa en la organización, lo que con lleva en disminuir el número de rondas de subastas en las cuales sea posible asignar el mayor número de tareas disponibles, finalmente debido a la naturaleza de nuestra investigación se plantea encontrar el método de organización con una simplicidad moderada para aplicarlo pues es un campo de estudio emergente.

Integración del subsistema de ordenamiento. El método paralelo es seleccionado entre varias ventajas por su una única ronda de subasta, en esta ronda el anunciador recibe las ofertas por todas las tareas de cada agente, mediante una matriz se organiza las columnas para decidir el ganador el cual ya no podrá ser candidato para las siguientes actividades, finalmente se anuncia a los ganadores su tarea asignada.

Para la etapa de anuncio de las actividades el agente Reina actúa como intermediario por lo cual da a conocer a los participantes de las actividades disponibles y espera la respuesta

de estos para llevar a cabo la segunda etapa, los agentes simples para poder realizar la puja envían un valor, el cual se calcula en base a la distancia.

Para poder asignar un valor de compra existen diversas formas, para el proyecto se analizará 3 métodos para asignar un valor de puja a cada actividad disponible para la compra, en la Tabla 21 se presentan las opciones para realizar el cálculo.

Tabla 21

Posibles métricas para valores de puja

Variable	Tipo de calculo	Descripción	Ecuación
A	Ángulos	Utilizando la variación de distancias en los ejes establecemos el ángulo al punto.	$\theta = \arctan\left(\frac{\Delta y}{\Delta x}\right)$
B	Variación en eje x	El valor se establece entre la distancia horizontal entre los la posición de partida y las tareas disponibles	$\Delta x = x_f - x_i$
C	Distancia	Mediante la ecuación de distancia de dos puntos se establece el valor de compra para la actividad	$\sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$

En base a la Tabla 21 para seleccionar el mejor método para realizar la puja de las actividades, tendremos en cuenta los siguientes criterios:

- Escalabilidad
- Cantidad de variables
- Facilidad de calculo

En la Tabla 22 se exponen la valoración de los criterios seleccionados y el método ganador para realizar la compra de las tareas.

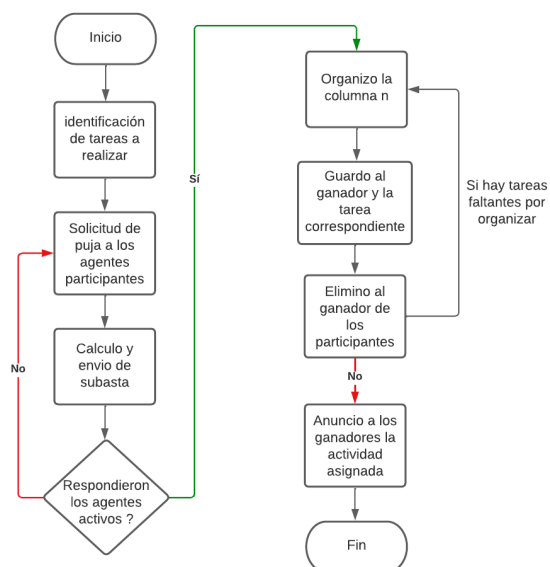
Tabla 22

Matriz de selección de métrica

Criterios	Peso [%]	A		B		C	
		Ca	Ep	Ca	Ep	Ca	Ep
Escalabilidad	50	3	1.5	2	1.0	4	2.0
Cantidad de variables	30	4	1.2	5	1.5	4	1.2
Facilidad de calculo	20	4	0.8	5	1.0	4	0.8
Total			3.4		3.5		3.9

Figura 26

Algoritmo de ordenamiento



El método seleccionado será el C el cual es la distancia, pues la escalabilidad de los métodos descartados no es la deseada, en caso de los ángulos como para la variación de distancia en el eje x si las posiciones de partida están en vertical y las posiciones de llegada de forma vertical con variación en la distancia en el eje vertical obtendremos valores iguales en todas las etapas y no consiguiendo optimizar el ordenamiento, caso contrario a la distancia que no posee dicho inconveniente. La cantidad de variables es similar en las opciones presentadas, aunque por la simpleza de la opción B es la que tiene el mayor puntaje, finalmente la factibilidad del cálculo se ve influenciada por las variables relacionadas a cada uno, mientras menos datos son utilizados menos recursos computacionales se requieren.

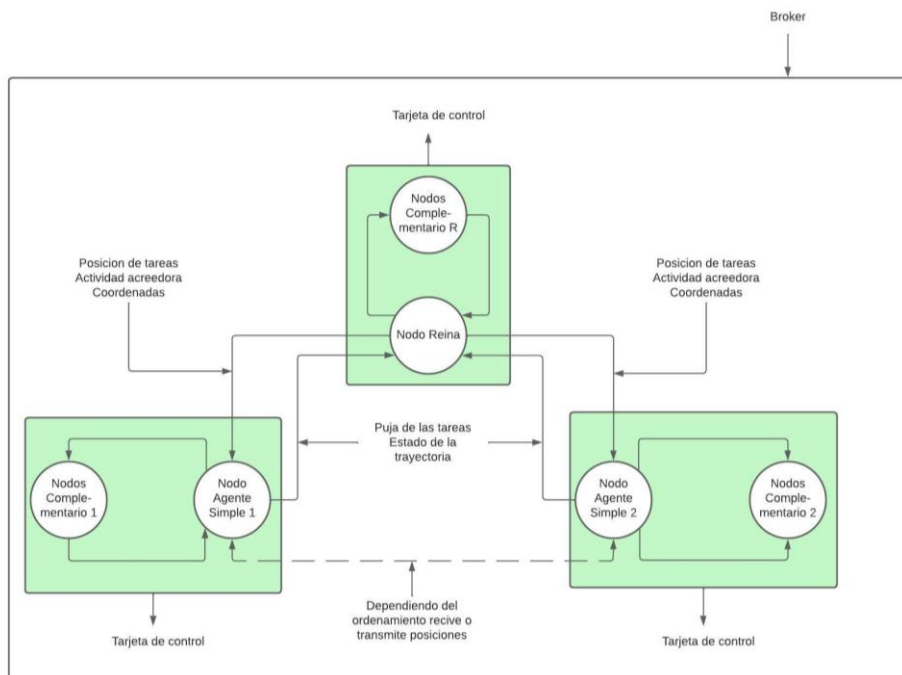
El cálculo se realiza para cada tarea anuncia la cual será devuelta en un vector al anunciador, una vez obtenida la respuesta de los participantes entramos al algoritmo de ordenamiento basado en subastas, se realiza el ordenamiento por columnas para poder identificar al ganador de cada tarea, el algoritmo se muestra en la Figura 26.

Algoritmo de funcionamiento global. Como se muestra en la Figura 27 la arquitectura del proyecto es soportada por el protocolo MQTT, siendo la columna vertebral para la comunicación entre los agentes participantes y el nodo reina, ya que el objetivo es buscar modularidad en los distintos sistemas del enjambre, se observa la independencia de cada agente a través de su tarjeta de control, aunque en el conjunto solo puede alcanzar su máximo potencial al interactuar entre sí.

Los nodos complementarios surgen debido a la heterogeneidad de los agentes pues cada rol ha sido asignado aprovechando los recursos disponibles por lo cual en las siguientes figuras se expondrán las arquitecturas para el caso correspondiente.

Figura 27

Comunicación y control entre los agentes participantes

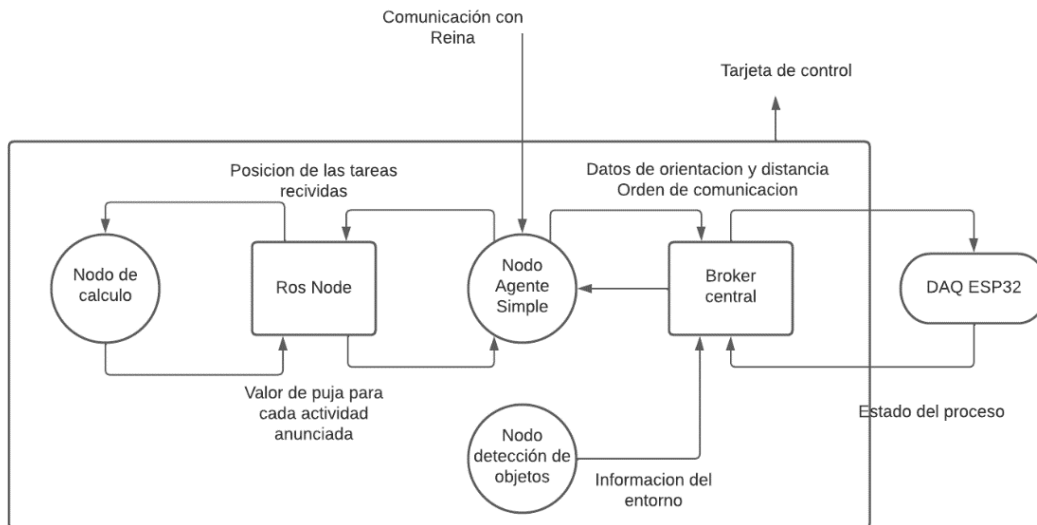


En el caso de los agentes simples como se observa en la Figura 28 se implementa ROS para dar respuesta al sistema de subastas implementado, reconocimiento de su tarea asignada y finalmente para la comunicación entre los agentes.

Por otro lado, la tarjeta de adquisición de datos ESP32 está destinada para el movimiento del individuo, por lo tanto, recibe información proporcionada por su unidad de control a través del protocolo MQTT, estos son datos de movimiento como ángulo, distancia y condiciones del entorno.

Figura 28

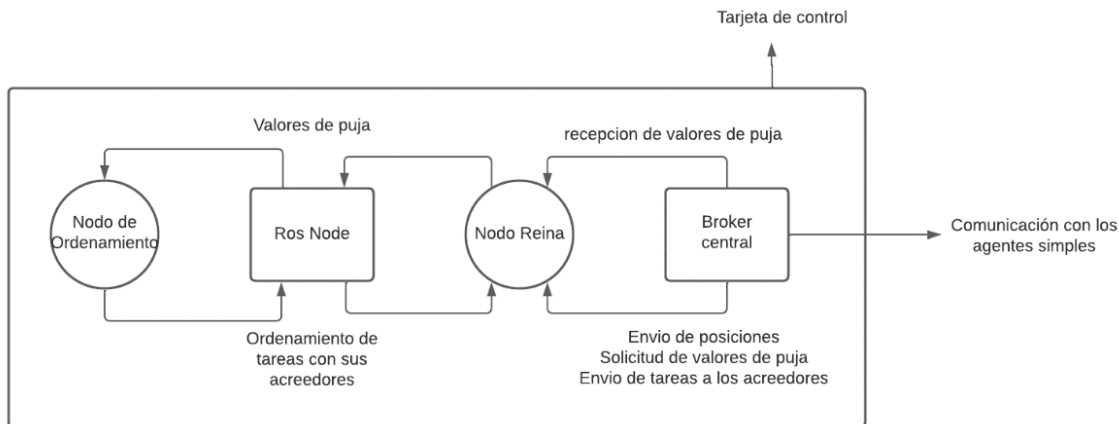
Estructura de los agentes simples



La Figura 29 muestra la estructura del agente Reina el cual está encargado de procesar la información mediante nodos de ROS, su nodo ordenamiento actúa como subastador pues escucha las ofertas y elige el valor adecuado de puja, finalmente mediante protocolo MQTT se anuncia al ganador de cada tarea.

Figura 29

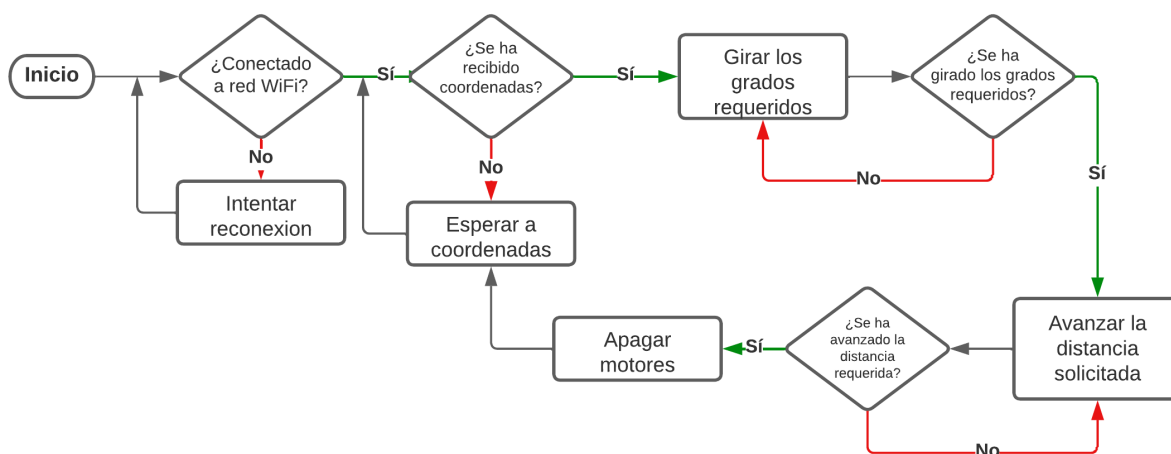
Estructura del agente reina



Algoritmo de odometría. Este algoritmo se encuentra implementado en la tarjeta ESP32 puesto que es más sencillo realizar la lectura de sensores, a continuación, se muestra el diagrama de flujo para este algoritmo

Figura 30

Diagrama de flujo de odometría



Durante el proceso de la Figura 30 podemos notar dos bloques que dicen “Girar los grados requeridos” y “Avanzar la distancia solicitada”, dentro de estos bloques se encuentran los controladores PID tanto de giro sobre el centro del robot, como se habló en la página 76 y el control de avance en línea recta detallado en la página 75.

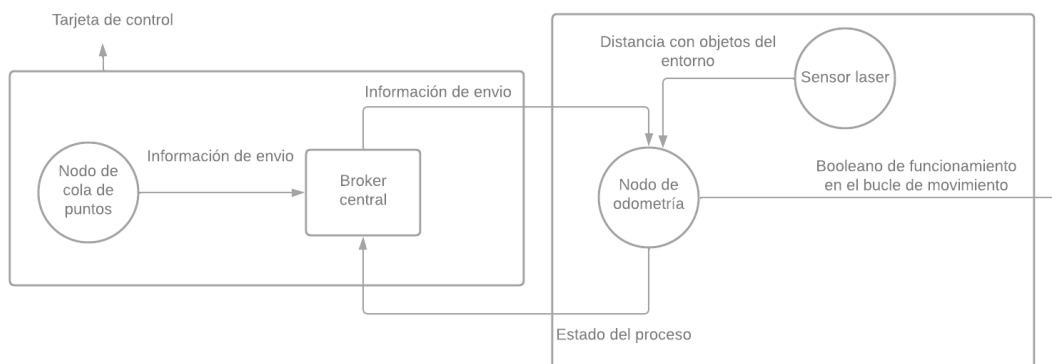
Algoritmo de prevención de colisiones. Como se muestra en la Figura 31 la prevención de colisiones depende de sensores, la forma de evitar colisiones es validar la distancia entre los objetos con un sensor infrarrojo SHARP, en caso de no cumplir la condición de estar dentro de una distancia segura (15cm) entonces se establece un falso para evitar la colisión y detener el agente simple.

Para evitar la mayor cantidad de fallas posibles y disminuir el uso del sistema de percepción se optó por implementar un nodo en el cual un agente n llega a su destino, da la

orden de salida al agente $n+1$ (en base a su orden jerárquico) de esta forma se realiza un seguimiento más ordenado y fácil de monitorear.

Figura 31

Estructura de colisiones



Resumen

El capítulo III presenta el dimensionamiento y diseño de todo el robot en base a características deseadas para el correcto funcionamiento del mismo, todos los criterios se basaron mediante métricas y ponderaciones lo cual permitió seleccionar los mejores componentes, materiales, sensores, y algoritmos.

Capítulo IV

Validación y análisis de resultados

A continuación, se presenta el capítulo IV donde en base a lo establecido en el capítulo III se procede a validar la construcción de los robots mediante análisis de elementos finitos, pruebas de odometría, validaciones de los diversos algoritmos descritos anteriormente y el proceso de construcción de los robots.

Validación del subsistema mecánico

Para validar que el subsistema mecánico pueda cumplir con los objetivos propuestos en el capítulo anterior, se propone realizar un análisis de elementos finitos con el fin de determinar el factor de seguridad de las piezas que son críticas, en base a esto se seleccionan dos piezas que se detallan a continuación:

- Soporte de motor
- Placa inferior

Todo el conjunto estructural se encuentra fabricado en acrílico, a continuación, se muestran las propiedades mecánicas.

Tabla 23

Propiedades mecánicas del acrílico

Propiedad	Valor	Unidades
Modulo elástico	3000	$\frac{N}{mm^2}$
Coefficiente Poisson	0.35	-
Densidad	1200	$\frac{kg}{m^3}$

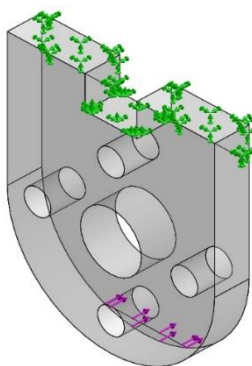
Propiedad	Valor	Unidades
Límite elástico	45	$\frac{N}{mm^2}$
Límite de tracción	73	$\frac{N}{mm^2}$

Soporte de motor

El soporte de motor es una pieza en la cual va montada los motores DC al chasis del robot, se estima cada uno de estos soportes va a estar cargado con un torque de $97.12[N \cdot mm]$

Figura 32

Puntos de apoyo y fuerza equivalente del soporte de motor



De la Figura 32 se pueden observar flechas verdes que indican los puntos de fijación del elemento y las flechas moradas indican la carga del elemento.

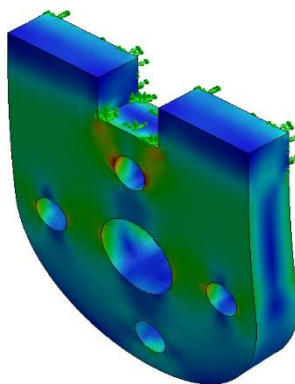
Tabla 24

Análisis de tensiones en soporte de motor

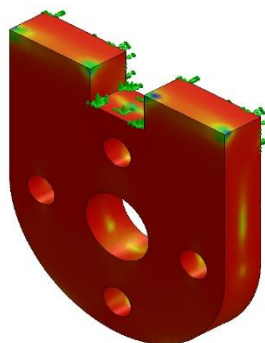
Nombre	Tipo	Min.	Máx.
Tensiones	Tensiones de Von Mises	13.34 [KPa]	26.07 [KPa]
Factor de seguridad	-	17.2	3300

Figura 33

Análisis de tensiones en soporte de motor

**Figura 34**

Análisis de tensiones en soporte de motor



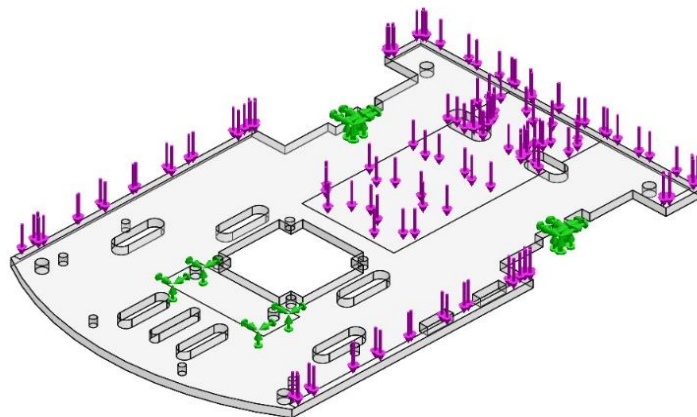
En base a la Tabla 24 se puede decir que el elemento no ha excedido ningún esfuerzo permisible y que el mínimo factor de seguridad de la pieza es de 17, por lo que el soporte del motor se encuentra correctamente diseñado y será capaz de soportar la carga del robot.

Placa inferior

La placa inferior es una pieza que se diseñó con el fin de soportar todos los elementos que se detallaron en la página, por lo que la placa inferior estará sometida a flexión debido a que solamente se apoya en tres puntos.

Figura 35

Puntos de apoyo y fuerza aplicada en la placa inferior



De la Figura 35 se pueden observar flechas verdes que indican los puntos de fijación del elemento y las flechas moradas indican la carga del elemento.

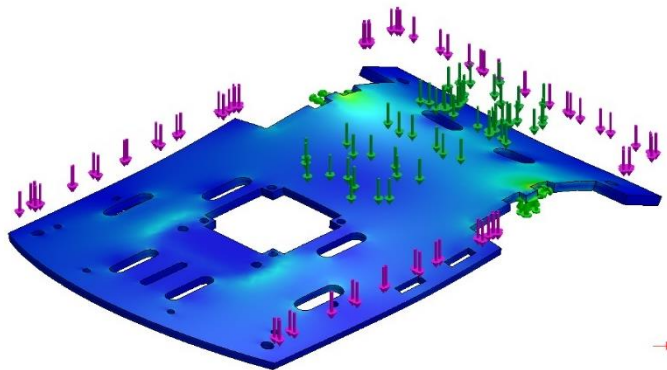
Tabla 25

Análisis de tensiones en placa inferior

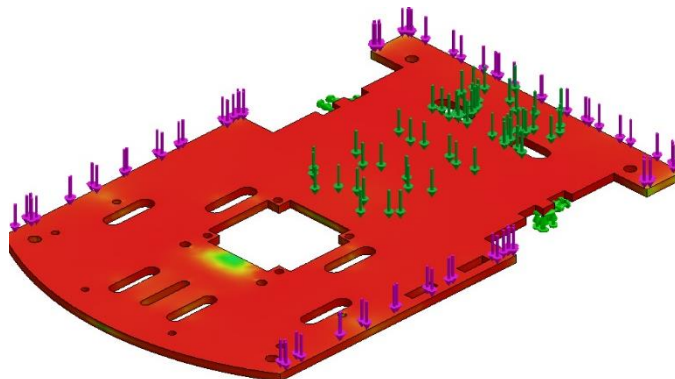
Nombre	Tipo	Min.	Máx.
Tensiones	Tensiones de Von Mises	1.85 [KPa]	3.13 [MPa]
Factor de seguridad	-	14.3	23870

Figura 36

Análisis de tensiones en placa inferior

**Figura 37**

Análisis de tensiones en placa inferior



En base a la Tabla 25 se puede decir que el elemento no ha excedido ningún esfuerzo permisible y que el mínimo factor de seguridad de la pieza es de 14, por lo que la placa inferior

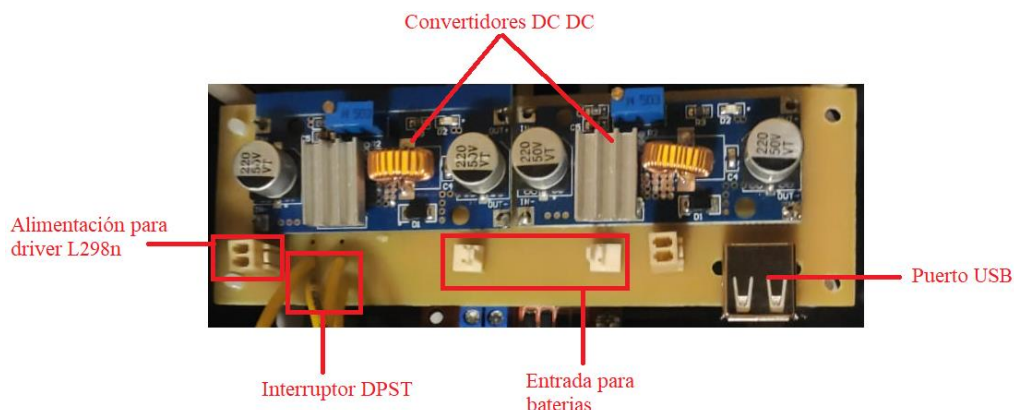
se encuentra correctamente diseñado y será capaz de soportar la carga de todos los elementos.

Validación del sistema electrónico

Placa de potencia

Figura 38

Placa de potencia ensamblada

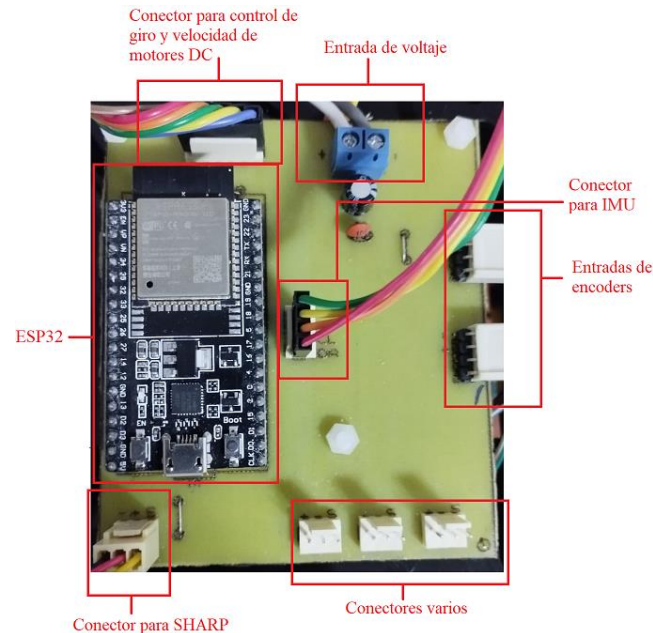


Como se puede observar en la Figura 38 se encuentran montados los convertidores DC DC, y los diferentes elementos para la distribución de voltajes como se habló en el capítulo III. Durante el proceso de pruebas del robot la placa se desempeñó de acuerdo a los parámetros establecidos previamente puesto que la corriente que fluye por la placa no supera las corrientes nominales de los convertidores.

Placa de control

Figura 39

Placa de control ensamblada



Se observa en la Figura 39 se obtiene una placa totalmente modular en la que si se requiere cambiar de placa o elemento es posible desconectar fácilmente los elementos que la componen.

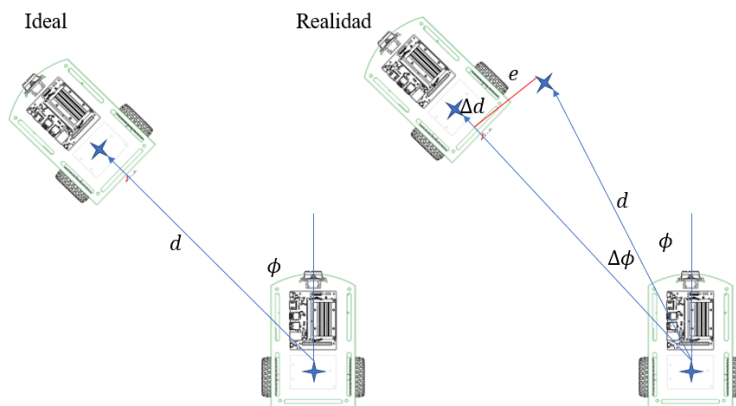
Validación de odometría

Prueba de giro, seguimiento en línea recta y distancia (GSD)

Para la prueba GSD se propone colocar al robot sobre un punto de control al cual se le va a otorgar un ángulo de giro y una distancia con lo cual el robot deberá llegar al final siguiendo una trayectoria recta, de esta forma es posible validar si el robot es capaz de llegar a un punto con el mínimo error posible

Figura 40

Prueba GSD



Como se observa en la Figura 40 se busca determinar el error del robot al momento de su llegada realizando varias pruebas con diferentes ángulos y distancias de prueba. Para la prueba de 0° el robot deberá recorrer 100cm, a 30° deberá recorrer 115cm, a 45° deberá recorrer 142cm y para 60° grados deberá recorrer 115cm. Estas distancias fueron calculadas para que el robot parta desde el vértice de un cuadrado de 100x100cm hacia su periferia.

A continuación, se muestran los resultados correspondientes a los dos robots construidos de acuerdo al alcance del proyecto.

Tabla 26

Datos de prueba GSD para J1

J1							
0°		30°		45°		60°	
$\Delta\phi$	Δd	$\Delta\phi$	Δd	$\Delta\phi$	Δd	$\Delta\phi$	Δd
1.40	2	2.45	2	0.81	0	2.41	4
1.40	3	0.26	3	0.79	3	0.49	3
1.11	3	0.50	0	0.79	3	0.00	2
1.39	3	1.00	0	0.20	2	1.47	2

J1							
0°		30°		45°		60°	
0.57	1	0.75	0	1.18	4	3.42	2
2.22	3	0.50	0	0.80	2	1.96	2
0.84	2	0.00	1	0.79	3	0.49	2
0.28	2	0.49	1	0.50	3	1.46	3
3.37	2	3.39	3	1.25	3	0.72	4
4.40	4	1.97	2	1.99	1	1.47	2

Tabla 27

Datos de prueba GSD para J2

J2							
0°		30°		45°		60°	
$\Delta\phi$	Δd	$\Delta\phi$	Δd	$\Delta\phi$	Δd	$\Delta\phi$	Δd
1.42	1	2.76	-1	0.61	0	2.51	-1
2.55	1	3.54	-2	1.99	2	1.00	0
1.13	1	3.51	-1	1.59	2	0.25	1
0.28	2	4.40	2	1.39	2	0.49	1
5.71	1	2.49	0	3.55	3	0.99	1
1.42	1	2.76	-1	1.01	2	0.98	2
2.27	1	4.97	0	2.80	1	2.49	0
1.13	1	1.51	-1	2.37	3	1.97	1
1.43	0	2.51	-1	1.19	2	4.23	0
0.56	2	4.51	-1	0.80	1	3.48	0

Para la validación de las pruebas GSD se plantea realizar una prueba Z con el fin de determinar que los robots no van a superar más de 1.8° de error en la desviación y que en distancias mayores a 1 metro el robot no diferirá de esta en más de 1.8cm.

Prueba para J1

Se muestran los estadísticos descriptivos tanto de $\Delta\phi$ y Δd en la Tabla 26,

Tabla 28

Estadísticos descriptivos para J1

Variable	μ	σ	Z_{obs}
Δd	2,175	1,152	-1.784
$\Delta\phi$	1,281	1,026	-1.966

Se plantean las siguientes hipótesis para $\Delta\phi$,

$$H_0: \mu > 1.8 \quad (21)$$

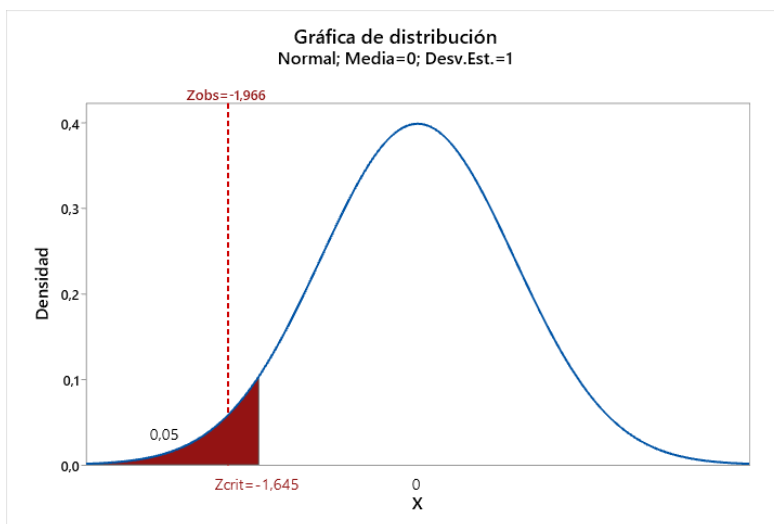
$$H_a: \mu < 1.8 \quad (22)$$

Donde:

- H_0 : Es la hipótesis nula
- H_a : Es la hipótesis nula
- μ : Es la media hipotética del error cometido

Figura 41

Prueba Z para $\Delta\phi$



Como se observa en la Figura 41 que $Z_{obs} < Z_{crit}$ se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa, por lo cual se asegura que el robot J1 es capaz de realizar la prueba GSD sin desviarse más de 1.8° en su trayectoria en ángulos mayores a 30° con un nivel de significancia del 95%.

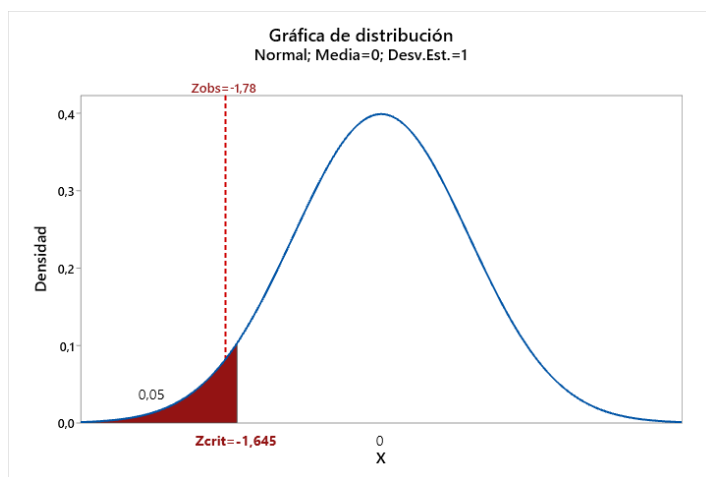
Se plantean las siguientes hipótesis para Δd ,

$$H_0: \mu > 2.5 \quad (22)$$

$$H_a: \mu < 2.5 \quad (23)$$

Figura 42

Prueba Z para Δd



Como se observa en la Figura 42 que $Z_{obs} < Z_{crit}$ se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa, por lo cual se asegura que el robot J1 es capaz de realizar la prueba GSD sin diferir el valor de distancia en más de 2.5cm con un nivel de significancia del 95%.

Prueba para J2

Se muestran los estadísticos descriptivos tanto de $\Delta\phi$ y Δd de la Tabla 27,

Tabla 29

Estadísticos descriptivos para J2

Variable	μ	σ	Z_{obs}
Δd	1.15	0.802	-1.97
$\Delta\phi$	2.164	1,367	-2.01

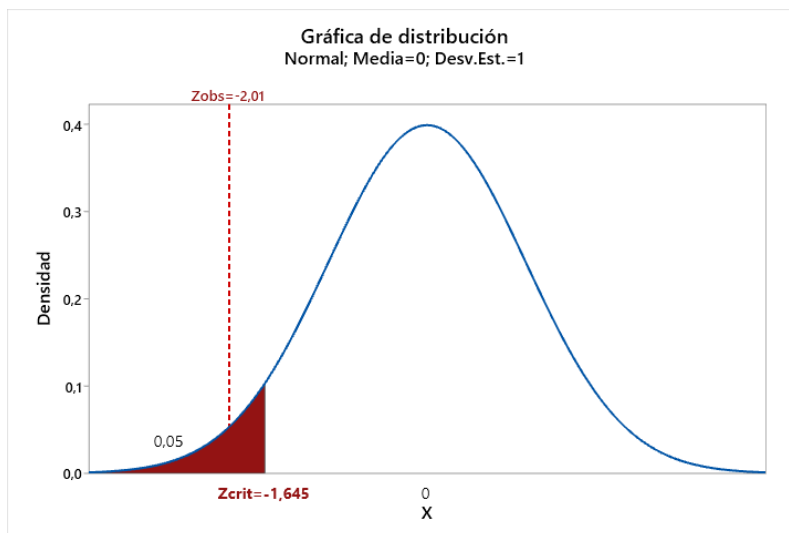
Se plantean las siguientes hipótesis para $\Delta\phi$,

$$H_0: \mu > 2.6 \quad (20)$$

$$H_a: \mu < 2.6 \quad (21)$$

Figura 43

Prueba Z para $\Delta\phi$



Como se observa en la Figura 43 que $Z_{obs} < Z_{crit}$ se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa, por lo cual se asegura que el robot J2 es capaz de realizar la prueba GSD sin desviarse más de 2.6° en su trayectoria en ángulos mayores a 30° con un nivel de significancia del 95%.

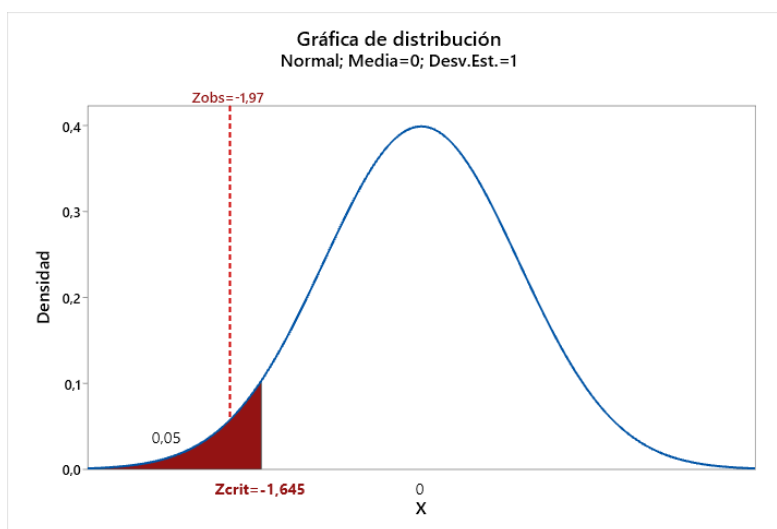
Se plantean las siguientes hipótesis para Δd ,

$$H_0: \mu > 1.4 \quad (22)$$

$$H_a: \mu < 1.4 \quad (23)$$

Figura 44

Prueba Z para Δd



Como se observa en la Figura 44 que $Z_{obs} < Z_{crit}$ se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa, por lo cual se asegura que el robot J1 es capaz de realizar la prueba GSD sin diferir el valor de distancia en más de 1.4cm con un nivel de significancia del 95%.

Prueba de trayectoria de un cuadrado

La prueba de trayectoria de un cuadrado sirve para determinar la repetitividad del algoritmo sin presentar errores significativos.

Para dicha prueba se establece un cuadrado de 1x1[m], en el que se marcan cuatro puntos en una superficie plana, se determinó la desviación tanto del ángulo de giro como la distancia recorrida. Para cada robot realizaron 3 vueltas en la periferia de un cuadrado.

Tabla 30

Prueba de trayectoria cerrada para J1

J1				
	X	Y	Δd	$\Delta\phi$
	-1	100	0.00	0.57
1	-103	105	2.12	-2.81
	-107	7	-1.92	2.34
	-10.5	-7	-2.49	-8.25
	1	100	0.00	0.57
2	-102	102	3.02	-1.11
	-103	4	-1.99	0.58
	-10	-10	-5.95	-8.56
	-2	100	0.02	1.15
3	-102	106	0.18	-3.43
	-111	7	-0.59	5.19
	-12	-10	0.45	-9.74

Tabla 31

Prueba de trayectoria cerrada para J2

J2				
	X	Y	Δd	$\Delta\phi$
	2	101	1.02	1.13
1	-100	98	2.04	1.68
	-98	0	-1.98	1.17

		J2		
	X	Y	Δd	$\Delta \phi$
	3	1	1.00	0.57
	3.5	101	1.06	1.98
2	-97	105	0.58	-2.28
	-103	7	-1.82	3.50
	-2	2	1.12	-2.83
	2	102	2.02	1.12
3	-100	101	2.00	0.56
	-101	3	-1.99	0.58
	-1	1	0.02	-1.15

A continuación, se muestra el resumen de medias y desviaciones estándar para ambos robots, después se procederá a mostrar el valor-p y corroborar que los parámetros y cumulo de errores se encuentran dentro de lo establecido en la sección de prueba GSD.

Tabla 32

Resumen de estadísticos descriptivos para ambos robots

Variable	$ \mu $	σ
Δd_{J1}	0,596	2,321
$\Delta \phi_{J1}$	1,96	4,73
Δd_{J2}	0,424	1,540
$\Delta \phi_{J2}$	0,505	1,795

A continuación, se muestran los valores-p para todas las variables de la Tabla 32,

Variable	Valor-p
Δd_{J1}	0,002
$\Delta \phi_{J1}$	0.547
Δd_{J2}	0,014
$\Delta \phi_{J2}$	0,000

En base a la literatura el valor-p debe ser menor que la significancia de las pruebas realizadas que para este caso es de 95%, es decir que el valor-p deberá ser menor que 0.05, y se observa que $\Delta \phi_{J1}$ acumulo más errores en los giros, de todas formas, en cuatro giros acumulo en promedio aproximadamente 2° que pueden ser directamente asociados a fallas en el sensor por la calidad de fabricación del mismo.

Validación de algoritmos de asignación de tareas

Comparación de algoritmo de asignación de tarea paralelo y secuencial

Los utilizados para seleccionar el tipo de asignación de tareas a implementar en resumen se puede describir como: la mayor cantidad de tareas asignadas en el menor número de rondas posibles obteniendo así el menor tiempo posible en el cálculo y ordenamiento.

En la Figura 45 tenemos un ordenamiento secuencial pues se observa que se oferta por cada tarea, al reunir la información de todos los participantes buscamos la mejor oferta y su ganador, este proceso se realiza para cada tarea, el algoritmo se ejecutó en un tiempo de 1 minuto con 36 segundos.

Para la subasta paralela se solicita las ofertas por todas las tareas disponibles, una vez se recibe la información de todos los agentes simple, se procede a realizar la asignación de tareas como se muestra en la Figura 46 ,al tener todas las ofertas se asignan las tareas en una sola ronda, lo cual nos da un tiempo de ejecución de 1 minuto con 13 segundos.

Figura 45

Asignación de tareas por método secuencial

```
[40.0, 3.0]
-----
(3.0, ) Primer ganador
-----
Recibiendo valor de subasta
Respuesta_subasta 2
[44.72135925292969, 2.0]
Respuesta_subasta
Respuesta_subasta 1
[82.46211242675781, 1.0]
Recibiendo valor de subasta
Respuesta_subasta 3
[20.0, 3.0]
-----
(3.0, 2.0) Se añade al segundo ganador
-----
Recibiendo valor de subasta
Respuesta_subasta 2
[40.31128692626953, 2.0]
Recibiendo valor de subasta
Respuesta_subasta 3
[5.0, 3.0]
Respuesta_subasta
Respuesta_subasta 1
[80.15609741210938, 1.0]
-----
(3.0, 2.0, 1.0) Se obtiene la asignacion total de las tareas
-----
```

Figura 46

Asignación de tareas por método paralelo

```
[100.0, 50.0, 5.0, 1.0]
..... Menu de acciones .....
1: Selecccion de figura a realizar
2: Realizar la subasta
3: Realizar el ordenamiento
4: Moverse a las posiciones
5: Realizar seguimeinto al lider
6: Operamiento manual de movimiento R1
7: Operamiento manual de movimiento R2
8: Operamiento manual de movimiento R3
.....
Ingrese la seleccion:
3
-----
Enviando los valores de puja
resultado: [1.0, 2.0, 3.0]
tarea: [1.0, 2.0, 3.0]
```

La diferencia de tiempos de ejecución se debe por limitaciones de implementación en el método secuencial, ya que es necesario agregar un tiempo de espera para que puedan llegar las respuestas de todos los agentes simples para poder realizar el ordenamiento y asignación, es decir mientras más tareas para ofertar mayor será el tiempo de ejecución, por lo cual el

sistema de asignación de tareas a pesar de ser escalable no es la mejor opción, demostrando así la correcta elección del algoritmo de ordenamiento.

Ejemplo de cálculo de una subasta paralela para n agentes

Como se presenta en la Figura 45 observamos el resultado de la organización por el método de subasta paralela para 6 agentes simples, para comprobar la veracidad de los datos y estructura obtenida a continuación presentamos un ejemplo de cálculo.

Las posiciones de las coordenadas iniciales de cada robot serán:

$$C_{ix} = [0, 40, 80, 118, 158, 198]$$

$$C_{iy} = [0, 0, 0, 0, 0, 0]$$

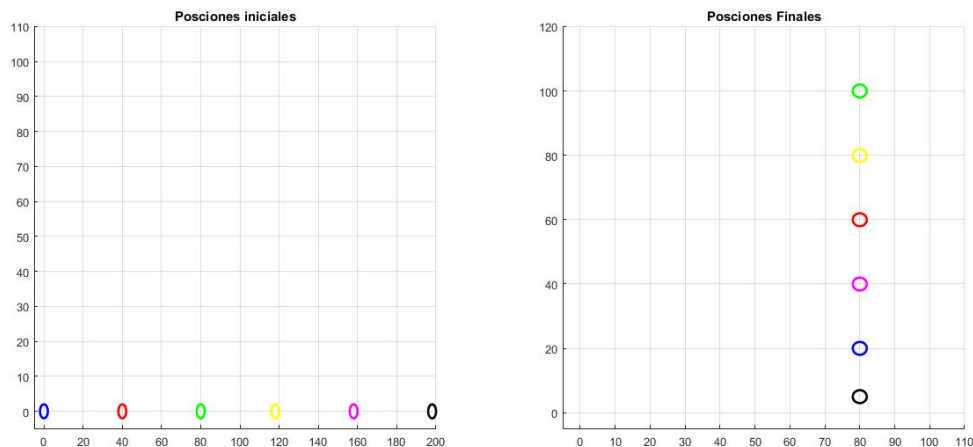
Las coordenadas de las actividades por asignar serán:

$$C_{fx} = [80, 80, 80, 80, 80, 80]$$

$$C_{fy} = [100, 80, 60, 40, 20, 5]$$

Figura 47

Simulación de ordenamiento



El cálculo de puja del agente simple ubicado en la posición del origen será:

$$d_{ag1} = \sqrt{(C_{fx_n} - C_{ix_n})^2 + (C_{fy_n} - C_{iy_n})^2}$$

En base a la ecuación anterior obtenemos el siguiente vector de oferta para cada tarea del agente simple $n=1,2, 3\dots,6$.

$$d_{ag1} = [128.06, 113.14, 100, 89.44, 82.46, 80.16]$$

La posición del valor de compra representa la tarea a la cual pertenece la oferta. El proceso se repite para cada agente simple obteniendo así la siguiente matriz (Figura 48).

Figura 48

Ejemplo da cálculo de matriz para asignación de tareas

<i>Comprador/Tarea</i>							
1	128.06	113.14	100.00	89.44	82.46	80.16	
2	107.70	89.44	72.11	56.568	44.74	40.31	
3	100.00	80.00	60.00	40.00	20.00	5.00	
4	106.98	88.57	71.02	55.17	42.94	38.33	
5	126.82	111.73	98.41	87.66	80.52	78.16	
6	154.67	142.56	132.38	124.60	119.68	118.11	

La matriz obtenida es una representación de los valores que posee el anunciador para poder establecer al ganador de la tarea el cual posea menor recorrido y no tenga ya una tarea asignada, para la primera tarea el valor menor será: 100.00 por lo tanto al robot diferencial numero 3 le corresponde la primera tarea.

Para la segunda tarea conocemos que el agente simple 3 ya tienen asignada una tarea por lo tanto lo eliminamos de la matriz para no considerarlo en las siguientes pujas por lo tanto tenemos la siguiente matriz:

Figura 49

Matriz de cálculo - eliminación de agente ganador

<i>Comprador/Tarea</i>							
1	128.06	113.14	100.00	89.44	82.46	80.16	
2	107.70	89.44	72.11	56.568	44.74	40.31	
4	106.98	88.57	71.02	55.17	42.94	38.33	
5	126.82	111.73	98.41	87.66	80.52	78.16	
6	154.67	142.56	132.38	124.60	119.68	118.11	

Al encontrar un ganador para la primera tarea avanzamos a la siguiente actividad en la cual se establece al agente 4 como ganador al poseer la menor distancia, realizamos el proceso hasta no tener más tareas por asignar, finalmente obtenemos un vector con el resultado del ordenamiento y las actividades que les corresponden a los ganadores los cuales serán informados en el siguiente paso.

$$resultado = [3 , 4 , 2 , 5 , 1 , 6]$$

$$tarea = [1 , 2 , 3 , 4 , 5 , 6]$$

Validación del ordenamiento cuando existen distancias iguales

Para el inicio del ordenamiento se ha establecido que las posiciones iniciales de cada agente simple serán:

$$C_{ix} = [0, 40, 80]$$

$$C_{iy} = [0, 0, 0]$$

Para comprobar la capacidad de asignación de tareas establecemos que las coordenadas de las tareas a realizar se ubicarán al frente del segundo agente simple por lo tanto las coordenadas de tareas serán:

$$C_{fx} = [40, 40, 40]$$

$$C_{fy} = [100, 50, 5]$$

En el agente reina enviamos el vector de datos que contienen las posiciones de las tareas a pujar, una vez enviado el mensaje solicitamos los vectores de puja a cada agente simple, se envía al nodo de subasta que tienen como datos de recepción posición inicial del agente y la ubicación de las actividades, una vez realizado los cálculos se envía la respuesta al agente reina, en la

Figura 50 se visualiza que las ofertas para las actividades entre el agente simple 1 y 3 son idénticas.

Figura 50

Vectores de compra a cada actividad

```

2: triangulo
Ingrese la seleccion:
1
40,100,40,50,40,5
Enviando coordenadas de las ubicaciones a pujar

..... Menu de acciones .....
1: Seleccion de figura a realizar
2: Realizar la subasta
3: Realizar el ordenamiento
4: Moverse a las posiciones
5: Realizar seguimiento al lider
6: Operamiento manual de movimiento R1
7: Operamiento manual de movimiento R2
8: Operamiento manual de movimiento R3
.....
Ingrese la seleccion:
2

solicitar los valores de puja
Respuesta_subasta
Respuesta_subasta 1
[107.70329284667969, 64.03124237060547, 40.31128692626953, 1.0]
Recibiendo valor de subasta
Respuesta_subasta 2
[100.0, 50.0, 5.0, 2.0]
Recibiendo valor de subasta
Respuesta_subasta 3
[107.70329284667969, 64.03124237060547, 40.31128692626953, 3.0]

..... Menu de acciones .....

```

```

Agente Reina

```

```

Agente Simple 1
coordenadas recibidas
[40.0, 100.0, 40.0, 50.0, 40.0, 5.0] <class 'list'>
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 107.70329284667969,64.03124237060547,40.31128692626953
-----
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 100.0,50.0,5.0
-----
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 107.70329284667969,64.03124237060547,40.31128692626953
-----
#####
#####

```

```

Agente Simple 2
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 100.0,50.0,5.0
-----
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 107.70329284667969,64.03124237060547,40.31128692626953
-----
#####
#####

```

```

Agente Simple 3
#####
#####
enviando valor de subasta
#####
-----
<class 'str'> 107.70329284667969,64.03124237060547,40.31128692626953
-----
#####
#####

```

Para el ordenamiento nos valemos de la función *where* de la librería Numpy, en sus parámetros establecemos la condición de búsqueda, la cual devuelve una matriz de filas y columnas que cumple la condición especificada, la búsqueda en la matriz comienza desde la fila superior a la inferior por lo que nuestro ganador será el que este en las filas superiores, en la

Figura 51 resaltado en un rectángulo rojo se observa el resultado de la búsqueda y la asignación.

Figura 51

Ordenamiento a distancias iguales

```

edisona@edison-a-linux:~$ rosrn central Ordenamiento.py
-----
recibido GUI
(107.70329284667969, 64.03124237060547, 40.31128692626953, 3.0)
[[107.7  64.03  40.31  1.  ]
 [100.   50.    5.    2.  ]
 [107.7  64.03  40.31  3.  ]]
(array([0]),)
-----
100.0
[107.7, 100.0, 107.7]
2.0
[1] [0]
-----
(array([], dtype=int64),)
-----
64.03
[64.03, 64.03]
1
[0 1] [1 1] Posiciones de la matriz que cumplen la condicion
-----
(array([], dtype=int64),)
-----
40.31
[40.31]
3.0
[0] [2]
-----
-----
[2.0, 1, 3.0] Resultado del ordenamiento (Agente simple)
[1.0, 2.0, 3.0] Tarea correspondiente (Actividad asignada)

```

Validación de sistema descentralizado

Un sistema descentralizado es aquel que puede seguir operando a pesar de interrupciones espontaneas en cualquiera de sus individuos por lo tanto se validara la desconexión de un agente simple y el agente reina.

Desconexión de un agente simple

Partimos del ejemplo de ordenamiento cuando tenemos distancias iguales y por una eventualidad como descarga de baterías o desconexión de red de algún agente, su estatus se verá reflejado en el nodo reina el cual informará de la desconexión a los agentes restantes como se muestra en la Figura 52.

El recuadro verde indica el momento de la desconexión del segundo agente simple, posterior en el recuadro verde se muestra la información enviada a los agentes restantes para permitir el desarrollo normal de las operaciones.

Figura 52

Desconexión del segundo agente simple

```

..... Menu de acciones .....
1: Selección de figura a realizar
2: Realizar la subasta
3: Realizar el ordenamiento
4: Moverse a las posiciones
5: Realizar seguimiento al lider
6: Operamiento manual de movimiento R1
7: Operamiento manual de movimiento R2
8: Operamiento manual de movimiento R3
.....
Ingrese la seleccion:                               Cambio de estado del agente simple 2
El dispositivo Jetson/Status/V2 tiene un status de Apagado retained 0
4

Antes del dispositivo 2 se apague
(2.0, 1.0, 3.0)
(1.0, 2.0, 3.0)
despues
1
3
['1', '3']

En camino a los puntos asignados
['1', '3']
(1.0, 2.0, 3.0)
-----
['posicion 1', 'posicion 3']
1,0,0,movimiento 3
2,0,0
3,0,0
-----

```

Envío de información a los agentes

Desconexión del agente reina

Cuando el agente reina falla el enjambre debe seguir funcionando, por lo cual cada agente simple tiene la capacidad de tomar el papel como agente reina y seguir cumpliendo sus funciones como agente simple, en la Figura 53 se muestra la continuación del ejemplo de desconexión de un agente simple, se observa en el recuadro verde la espera y puesta en funcionamiento cuando la reina falla, además el recuadro rojo muestra la capacidad de continuar con el estado de jerarquía asignado inclusive si existe la desconexión de agentes simples. Al final se despliega el menú de acciones para continuar con las operaciones deseadas en el enjambre.

Figura 53

Desconexión del agente reina

```

waiting
waiting
waiting
waiting
waiting
waiting
waiting
waiting
Estado del agente reina y funcionamiento en caso de fallo
La central ha fallado
suscribiendome para realizar control
El dispositivo Jetson/Status/V1 tiene un status de Activo retained 1
El dispositivo Jetson/Status/V2 tiene un status de Activo retained 1
El dispositivo Jetson/Status/V3 tiene un status de Activo retained 1
El dispositivo Jetson/Jerarquia/V2 tiene una jerarquia de 0
El dispositivo Jetson/Jerarquia/V1 tiene una jerarquia de 1
El dispositivo Jetson/Jerarquia/V3 tiene una jerarquia de 2
['1', '0', '2']
Desea continuar con la jerarquia para el movimiento:
1: Si
2: No
seleccione la opcion:
1
['movimiento 1', 'movimiento 3'] Continuation de funcionamiento del sistema

..... Menu de acciones .....
1: Seleccion de figura a realizar
2: Realizar la subasta
3: Realizar el ordenamiento
4: Moverse a las posiciones
5: Realizar seguimeinto al lider
6: Operamiento manual de movimiento R1
7: Operamiento manual de movimiento R2
8: Operamiento manual de movimiento R3
.....

```

Resumen

Durante todo este capítulo se validó todos los subsistemas ya sean mediante simulaciones o pruebas físicas obteniéndose resultados aceptables tanto en la odometría como en la comunicación y posibles fallas que se puedan desarrollar a lo largo del funcionamiento del seguimiento al líder ya sea por eventualidades espontaneas o colisiones entre agentes.

Conclusiones, recomendaciones y trabajos futuros

Conclusiones

Debido a la forma en la que se desarrollaron los algoritmos de comunicación se obtuvo un sistema robusto debido a que se tiene un sistema de respaldo en cada agente simple en el caso de que el agente reina sufra un eventual fallo, además debido a la modularidad tanto de los agentes simples como del algoritmo se obtiene un sistema escalable y flexible puesto que resulta sencillo incorporar más agentes para realizar, esto es comprobado ya que en el alcance se planteó emplear dos agentes, mientras que para las pruebas de validación de algoritmos se emplearon tres agentes obteniéndose así un sistema multi-robot tipo enjambre.

Durante la fase de diseño de cada agente simple se pensó en la modularidad de su construcción para que en el caso de tener fallas en los sensores o actuadores sea posible intercambiarlos de forma sencilla y rápida, además con el uso de la tarjeta ESP32 y la unidad de control Jetson Nano fue posible almacenar diferentes rutinas para el correcto desempeño de los agentes simples.

La primera validación para evitar colisiones en el sistema es en la cola de movimiento generada por la comunicación entre agentes simples, pues para informar a su seguidor es necesario llegar al punto asignado, en caso de obstáculos como un líder desconectado de la red se utiliza el sistema de percepción (SHARP) con el objetivo de evitar daños y colisiones.

El método de ordenamiento basado en mercado paralelo cumple con la asignación de tres tareas y tres agentes en un tiempo de 73 segundos, comparado a la subasta secuencial la cual incrementa su tiempo en ejecutar 96 segundos debido a limitaciones de implementación en la comunicación de los agentes participantes, al ser necesario un *delay* en la recepción del mensaje por parte del agente reina, por lo cual si buscamos rapidez en la asignación de tareas la subasta secuencial es la mejor opción.

Las pruebas de odometría establecen que los agentes simples pueden recorrer una distancia de un metro con una variación máxima en su recorrido de 2.5 cm, lo cual representa un error de 2.5%, además en los giros mayores a 30° se desvía como máximo 2.6° de la trayectoria indicada.

Recomendaciones

Para integrar los subsistemas que componen el proyecto es importante revisar la existencia de conflictos en la comunicación entre dispositivos, pues el protocolo de comunicación I2C de la MPU 6050 no permite enlazar la tarjeta de adquisición de datos ESP32 con la tarjeta de control Jetson Nano.

La selección de sensores, actuadores y elementos mecánicos resulta importante al momento de obtener mejores resultados puesto que durante la fase de pruebas se pudo evidenciar que lo más óptimo es implementar ruedas con mayor adherencia al piso, por otra parte, el uso de un giroscopio de mejor calidad también resulta importante puesto que de este depende la orientación del robot.

Es necesario tener en cuenta para el envío y recepción de mensajes en la plataforma MQTT el uso de tiempos de espera en la ejecución de la tarea, ya que provoca desincronización para escuchar los mensajes enviados por otro dispositivo perdiendo información y afectando al movimiento conjunto del enjambre.

Trabajos futuros

Actualmente el posicionamiento de los robots se basa en odometría, esta se encuentra calibrada mediante bloques de PIDs, pero esta puede mejorar mediante el uso de cámaras cenitales con el fin de corregir trayectorias y las posiciones finales.

Además, es posible mejorar el sistema de percepción de los robots aumentando los sensores infrarrojos SHARP (actualmente tiene uno, se podría emplear dos más) y para

realizarlo de forma más eficiente se pueden emplear comparadores para usar únicamente una entrada digital en la tarjeta de adquisición de datos ESP32.

Finalmente, para emplear toda la capacidad de procesamiento de las unidades de control Jetson Nano, es posible implementar algoritmos de navegación SLAM para que los agentes tengan más autonomía al momento de la navegación, puesto que actualmente los robots llegan a un punto y esperan otro punto para seguir avanzando.

Bibliografía

- Baturone, A. O. (2005). *Robótica: Manipuladores y Robots Móviles*. Marcombo.
- Beltran Tuquerres, C. E. (2019). *Desarrollo de un controlador centralizado de seguimiento de trayectoria para un sistema multi robot*.
- Burgard, W., Moors, M., Stachniss, C., & Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3), 376–386.
<https://doi.org/10.1109/TRO.2004.839232>
- Campbell, A., Wu, A. S., & Shumaker, R. (2008). Multi-agent task allocation: Learning when to say no. *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation - GECCO '08*, 201. <https://doi.org/10.1145/1389095.1389128>
- Efrain, G. B. D., del Carmen, V. C. A., Fernando, E. C. L., & Cesar, L. M. D. (2018). Implementation of an IoT Architecture based on MQTT for a Multi-Robot System. *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, 1–6.
<https://doi.org/10.1109/ETCM.2018.8580321>
- Enríquez, F. J., Sifuentes, E., Bravo, G., & Castro, A. (2016). Sistema Embebido para Validar el Funcionamiento de la Tarjeta de Adquisición de Datos USB-6009 de National Instruments. *Información Tecnológica*, 27(5), 191–200. <https://doi.org/10.4067/S0718-07642016000500021>
- Erickson, R. W., & Maksimović, D. (2001). *Fundamentals of power electronics* (Second edition, softcover reprint of the hardcover 2nd edition 2001). Springer Science+Business Media, LLC.
- Escobar, L., Moyano, C., Aguirre, G., Guerra, G., Allauca, L., & Loza, D. (2020). Multi-Robot platform with features of Cyber-physical systems for education applications. *2020 IEEE ANDESCON*, 1–6. <https://doi.org/10.1109/ANDESCON50619.2020.9272030>
- Garces, L. A. G., & Gaviria, J. L. R. (2014). *CONTROL PID PARA EL CONTROL DE VELOCIDAD DE UN MOTOR DC*.

- Gil, A., Aguilar, J., Dapena, E., & Rivas, R. (2018). Verification of the emergence in an architecture for multi-robot systems (AMEB). *Latin-American Journal of Computing*, 5(2), Art. 2.
- Graessler, I., & Hentze, J. (2020). The new V-Model of VDI 2206 and its validation. *At - Automatisierungstechnik*, 68(5), 312–324. <https://doi.org/10.1515/auto-2020-0015>
- Janchiv, A., Batsaikhan, D., Kim, G. hwan, & Lee, S.-G. (2011). Complete coverage path planning for multi-robots based on. *2011 11th International Conference on Control, Automation and Systems*, 824–827.
- Jose, K., & Pratihari, D. K. (2016). Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, 80, 34–42. <https://doi.org/10.1016/j.robot.2016.02.003>
- Khamis, A., Hussein, A., & Elmogy, A. (2015). Multi-robot Task Allocation: A Review of the State-of-the-Art. En A. Koubâa & J. R. Martínez-de Dios (Eds.), *Cooperative Robots and Sensor Networks 2015* (Vol. 604, pp. 31–51). Springer International Publishing. https://doi.org/10.1007/978-3-319-18299-5_2
- Knudsen, M., & KaiVo-Oja, J. (2020). Collaborative Robots: Frontiers of Current Literature. *Journal of Intelligent Systems: Theory and Applications*, 13–20. <https://doi.org/10.38016/jista.682479>
- Koes, M., Nourbakhsh, I., & Sycara, K. (2005). Heterogeneous multirobot coordination with spatial and temporal constraints. *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, 1292–1297.
- López, B., & Suárez, S. (2003). *Task allocation in rescue operations using combinatorial auctions*. 11.
- Mosquera, R. E. (2017). *Diseño de un algoritmo de enjambre para el trabajo colaborativo de mini robots para recoger y clasificar piezas de diferentes formas y colores. Propuesta de semillero de investigación SYNAP*. 6.

- Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación.
- Otte, M., Kuhlman, M. J., & Sofge, D. (2020). Auctions for multi-robot task allocation in communication limited environments. *Autonomous Robots*, 44(3), 547–584.
<https://doi.org/10.1007/s10514-019-09828-5>
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '87*, 25–34. <https://doi.org/10.1145/37401.37406>
- Robin, C., & Lacroix, S. (2016). Multi-robot target detection and tracking: Taxonomy and survey. *Autonomous Robots*, 40(4), 729–760. <https://doi.org/10.1007/s10514-015-9491-7>
- Rocha, R., Dias, J., & Carvalho, A. (2005). Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems*, 53(3), 282–311. <https://doi.org/10.1016/j.robot.2005.09.008>
- Sastre Guerrero, J. (2011). *Nuevas metodologías para la asignación de tareas y formación de coaliciones en sistemas multi-robot*.
- Sempere-Tortosa, M. (2014). *Agentes y enjambres artificiales: Modelado y comportamientos para sistemas de enjambre robóticos*. <http://rua.ua.es/dspace/handle/10045/36619>
- Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots* (2nd ed). MIT Press.
- Smith, R. G. (1980). Communication and Control in a Distributed Problem Solver. *IEEE TRANSACTIONS ON COMPUTERS*, 29, 10.
- Smith, S. L., & Bullo, F. (2009). The dynamic team forming problem: Throughput and delay for unbiased policies. *Systems & Control Letters*, 58(10–11), 709–715.
<https://doi.org/10.1016/j.sysconle.2009.07.001>
- Tzafestas, S. G. (2014). *Introduction to mobile robot control* (First edition). Elsevier.

Apéndices