



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

PROYECTO TÉCNICO PREVIO A LA OBTENCIÓN DEL TÍTULO

DE:

INGENIERO ELECTRÓNICO

PROYECTO TÉCNICO:

**“REPOTENCIACIÓN Y ANÁLISIS DE ESTRATEGIAS
APLICADAS A UN ROBOT SUMO AUTONOMO”**

AUTORES:

**JOSÉ FABRICIO PAREDES FRANCO
MICHAEL ENRIQUE BACA GRANDA**

TUTOR:

ING. BYRON XAVIER LIMA CEDILLO, Msc.

GUAYAQUIL – ECUADOR

2020

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Nosotros, JOSÈ FABRICIO PAREDES FRANCO y MICHAEL ENRIQUE BACA GRANDA autorizamos a la **Universidad Politécnica Salesiana** la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, se declara que los conceptos y análisis desarrollados y conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.



José Fabricio Paredes Franco
Cédula: 1206453712



Michael Enrique Baca Granda
Cédula: 0705127181

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Nosotros, **JOSÉ FABRICIO PAREDES FRANCO** y **MICHAEL ENRIQUE BACA GRANDA**, con documentos de identificación N° **1206453712** y **0705127181**, manifestamos nuestra voluntad y cedemos a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado: **“REPOTENCIACIÓN Y ANÁLISIS DE ESTRATEGIAS APLICADAS A UN ROBOT SUMO AUTONOMO”** mismo que ha sido desarrollado para optar por el título de **INGENIEROS ELECTRÓNICOS**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscritos este documento en el momento que se realiza la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



José Fabricio Paredes Franco
Cédula: 1206453712

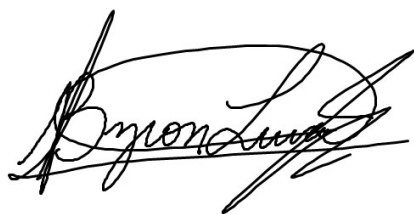


Michael Enrique Baca Granda
Cédula: 0705127181

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Ing. Byron Xavier Lima Cedillo, MSc.** Director del proyecto de titulación denominado: **“REPOTENCIACIÓN Y ANÁLISIS DE ESTRATEGIAS APLICADAS A UN ROBOT SUMO AUTONOMO”** realizado por los estudiantes **JOSÈ FABRICIO PAREDES FRANCO** y **MICHAEL ENRIQUE BACA GRANDA**, certifico que han sido orientados y revisado durante su desarrollo, por cuanto se aprueba la presentación de este ante las autoridades pertinentes.

Guayaquil, agosto del 2020

A handwritten signature in black ink, appearing to read 'Byron Lima', with a large, stylized flourish at the end.

Ing. Byron Xavier Lima Cedillo, MSc.
Cedula: 0921971768

DEDICATORIA

Este Trabajo está dedicado principalmente a Dios por darme fuerzas, salud y ánimos en momentos difíciles de mi vida, demostrando que con fe, devoción y trabajo duro se puede cumplir con los objetivos de vida.

A mis Padres Julio Ricardo Paredes Cervantes y Lourdes Narcisa Franco Pin que siempre fueron mi inspiración para lograr esta meta, gracias por siempre esforzarse en darme lo mejor y enseñarme a ser un hombre de bien.

A mis Profesores que en poco tiempo dejaron de ser mis profesores y se volvieron grandes amigos en especial al Ingeniero Byron Lima C. que siempre supo cómo sacar nuestro mejor potencial en cada clase, examen y competencia.

A todas las buenas amistades que coseche durante mi formación académica algunas de amistades que se volvieron familia.

Finalmente, y no menos Importante a mi querido Club de Robótica, donde cada experiencia fue de gran aprendizaje y conocimos la robótica desde otro punto de vista cosechando logros tanto a nivel nacional como internacional.

José Fabricio Paredes Franco

AGRADECIMIENTO

Parto desde la gratitud siempre a mis padres que estuvieron en todo momento levantándose cuando me sentía derrotado, dando ánimos cuando el día era malo o cuando surgía algún traspie en el trayecto del proyecto de titulación, a mis amados abuelos que desde su lugar de paz y tranquilidad están cuidándome y guiándome.

Agradecimiento también a mi compañero de titulación Fabricio Paredes, amigo y hermano de tantas historias, que estuvo ahí siempre y en el trayecto más difícil que fue sacar el proyecto adelante tuvo el valor de tomar las riendas cuando debía y avanzar en los momentos que no pude estar.

Agradecimiento a mis profesores, pilares fundamentales en mi formación como Ingeniero si lo puedo decir ahora, en especial a los que se convirtieron en mis amigos, que de la docencia subieron un nivel más y apoyaban en los momentos difícil y daban algún que otro consejo para lograr el objetivo, Ing. Byron Lima y el Ing. Gino Alvarado, grandes amigos.

Gratitud al club de robótica, club que me abrió las puertas y me regalo un espacio donde poder desarrollarme en lo práctico y comprender la verdadera razón de “hay que dañar para aprender”. Club donde me relacione con grandes personas, amigos que durante las competencias apoyaban y trataban de hacer menos dura la misma, grandes persona y grandes enseñanzas que cada uno de ellos te regala con el pasar de los días.

Michael Enrique Baca Granda

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO TÉCNICO	TEMA DE PROYECTO TÉCNICO
2020	JOSÉ FABRICIO PAREDES FRANCO MICHAEL ENRIQUE BACA GRANDA	ING. BYRON XAVIER LIMA CEDILLO, MSc.	REPOTENCIACIÓN Y ANÁLISIS DE ESTRATEGIAS APLICADAS A UN ROBOT SUMO AUTÓNOMO

El presente proyecto técnico tiene como objetivo el análisis de las diferentes estrategias de combate de un robot sumo autónomo, mediante la comparación de las diferentes magnitudes tales como lo son el voltaje, amperaje y posición del robot sobre el Dohyo que se emplean durante un combate de robots sumos.

El objetivo principal es elaborar y seleccionar un algoritmo optimizado en la detección e identificación de los objetos móviles dentro del perímetro de juego y finalmente se procede a verificar el desempeño del sistema y se obtendrá algunos resultados en base a los diferentes algoritmos de control empleados en el autómata y también analizar el desplazamiento del prototipo en el Dohyo adicionalmente se da la facilidad de un entorno de programación de los embebidos usando un núcleo basado en el ATMEGA 328 para el sistema de control a implementar.

Palabras claves: Estrategias, Dohyo, Voltaje, Amperaje

ABSTRACT

YEAR	STUDENTS	PROJECT DIRECTOR	PROJECT THEME
2020	JOSÉ FABRICIO PAREDES FRANCO MICHAEL ENRIQUE BACA GRANDA	ENG. BYRON XAVIER LIMA CEDILLO, MSc.	ANALYSIS AND ACONDITIONING OF STRATEGIES APPLIED TO AN AUTONOMOUS SUMO ROBOT.

The objective of this technical project is to analyze the different combat strategies of an autonomous sumo robot, by comparing the different magnitudes such as the voltage, amperage and position of the robot on the Dohyo that are used during a combat of sumo robots.

The main objective is to elaborate and select an optimized algorithm in the detection and identification of mobile objects within the perimeter of the game and finally we proceed to verify the performance of the system and obtain some results based on the different control algorithms used in the automaton and also analyze the displacement of the prototype in the Dohyo.

Additionally, there is the ease of an embedded programming environment using a core based on the ATMEL 32UA for the control system to be implemented.

Keywords: Strategies, Dohyo, Voltage, Amperage

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORIA	III
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR.....	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN.....	VII
ABSTRACT	VIII
ÍNDICE GENERAL	- 1 -
ÍNDICE DE FIGURAS.....	- 3 -
ÍNDICE DE TABLAS.....	- 3 -
INTRODUCCIÓN.....	- 5 -
1.EL PROBLEMA.	- 6 -
1.1.Descripción del Problema.....	- 6 -
1.2.Importancia y Alcance.	- 6 -
1.3.Delimitación del Problema.	- 6 -
1.3.1 Delimitación Temporal:	- 6 -
1.3.2 Delimitacion Espacial:.....	- 6 -
1.3.3 Delimitación Academica:	- 6 -
1.4 Objetivos.	- 7 -
1.4.1 Objetivo General.....	- 7 -
1.4.2 Objetivos Específicos.....	- 8 -
2.FUNDAMENTACIÓN TEÓRICA.	- 8 -
2.1.Robotica Movil.	- 6 -
2.1.1 De estatica a movil: La robotica en el transcurso del tiempo.....	- 6 -
2.1.2 Caracteristicas de la robotica movil.	- 6 -
2.1.3 Sensores y actores: pilares de la robotica movil.....	- 6 -
2.1.4 La robotica movil y sus ambitos de aplicación.	- 6 -
2.2.Descripcion del robot-sumo.	- 6 -
2.2.1 Proposito del robot-sumo.....	- 8 -
2.2.2 Historia de All Japan Robot Sumo Tournament.....	- 8 -
2.3.Estrategias de Control.	- 6 -
2.3.1 Slow Search.	- 8 -
2.3.2 WoodPecker.....	- 8 -
2.3.3 Tornado.....	- 8 -
2.4.Sensores y Actuadores.....	- 6 -
2.4.1 Sensores de Distancia.	- 7 -

2.4.2 Alimentacion del Prototipo.	7 -
2.4.3 Controlador de Potencia.	7 -
2.4.4 Motores.	7 -
2.4.5 Microcontrolador ATMEGA328-SMD.	6 -
2.4.6 Sensor de Corriente ACS709.	7 -
2.4.7 Sensor de Voltaje.	7 -
3.Marco Teorico.....	8 -
3.1.Diseño Mecanico.	6 -
3.2.Diseño de la tarjeta de control.	6 -
3.3.Arduino.	6 -
3.5.Topografia de la tarjeta de control.	6 -
4.Resultados.....	8 -
4.1.Analisis Estructural.	6 -
4.1.2. Analisis Lateral.	6 -
4.1.3. Analisis Base del Sumo.	6 -
4.1.4. Analisis de soporte de maxon.	6 -
4.2.Analisis de movimiento.....	8 -
4.1.2. Ejecucion del programa.	6 -
4.1.2. Selección de Color.	8 -
4.1.3. Grafico de Trayectoria.	8 -
4.1.4. Nueva Grafica.	8 -
4.1.5. Obtencion de Datos.	8 -
4.3.Datos resultantes de las Estrategias.....	8 -
4.3.1 Estrategia numero uno:	6 -
4.3.2 Estrategia numero dos:	6 -
4.3.3 Estrategia numero tres:	6 -
4.3.4 Estrategia numero cuatro:.....	6 -
CONCLUSIONES.....	40-
RECOMENDACIONES.....	40-
REFERENCIAS BIBLIOGRAFICAS.....	41-
ANEXOS	42-

ÍNDICE DE FIGURAS

Figura 1.	Delimitación espacial del prototipo del Proyecto Técnico	- 7 -
Figura 2.	Prototipo de Robot-Sumo.....	-10-
Figura 3.	Inicios de All Japan robot sumo Tournament	-11-
Figura 4.	Desarrollo de estrategia uno	-12-
Figura 5.	Desarrollo de estrategia dos	-13-
Figura 6.	Desarrollo de la estrategia tres	-14-
Figura 7.	Referencia sensor AMBA 2409	-15-
Figura 8.	Referencia sensor AMBA 345912	-15-
Figura 9.	Sensor QRT-1A	-15-
Figura 10.	Batería LIPO 3s 850mAh	-16-
Figura 11.	Roboclaw 2x45st.....	-16-
Figura 12.	Motor Maxon	-17-
Figura 13.	ATMEGA328-SMD microcontrolador	-17-
Figura 14.	Sensor de Corriente ACS709.....	-18-
Figura 15.	Sensor de Voltaje FZ0430	-19-
Figura 16.	Plano de la base en 2D y 3D	-20-
Figura 17.	Pared lateral con vista lateral acotada	-21-
Figura 18.	Pared Lateral con vista superior	-21-
Figura 19.	Modelado 3D.....	-21-
Figura 20.	Pared trasera vista Frontal Modelado 2D.....	-22-
Figura 21.	Pared Trasera vista superior acotada	-22-
Figura 22.	Pared trasera Modelado 3D	-22-
Figura 23.	Base vista superior.....	-23-
Figura 24.	Base vista frontal.....	-23-
Figura 25.	Base Modelada 3D.....	-23-
Figura 26.	Prototipo Ensamblado.....	-23-
Figura 27.	Topografía General de la tarjeta de control.....	-24-
Figura 28.	Topografía de los sensores.....	-25-
Figura 29.	Topografía del Programador	-25-
Figura 30.	Topografía del Sensor de Corriente y Encoders	-25-
Figura 31.	Topografía Modulo SD	-26-
Figura 32.	Topografía Modulo de Inicio.....	-26-
Figura 33.	Topografía de Pulsadores, comunicación serial con roboclaw	-27-
Figura 34.	Topografía de alimentación de energía.....	-27-
Figura 35.	Topografía LED's	-28-
Figura 36.	Topografía Microcontrolador	-28-
Figura 37.	Vista de las conexiones internas de la tarjeta controladora	-29-
Figura 38.	Vista de manufactura de la tarjeta controladora.....	-29-
Figura 39.	Tarjeta Controladora Implementada	-30-
Figura 40.	Modelado del Módulo de Arranque	-30-
Figura 41.	Análisis pared posterior.....	-31-
Figura 42.	Análisis pared lateral.....	-31-
Figura 43.	Análisis de la base del robot-sumo	-32-
Figura 44.	Análisis del soporte del motor	-32-

Figura 45. Menu de comandos.....	-33-
Figura 46. Ejecutado en el programa	-34-
Figura 47. Selección del color para impresión de trayectoria.....	-34-
Figura 48. Gráfico de trayectoria por el robot-sumo.....	-35-
Figura 49. Salto y giro para mapear la pista en busca del contrincante....	-35-
Figura 50. Detección del contrincante.....	-36-
Figura 51. Datos obtenidos por los encoders del robot-sumo.....	-36-
Figura 52. Datos usados para creación de graficas en el programa	-37-
Figura 53. Datos obtenidos después de cada estrategia	-37-
Figura 54. Graficas de voltaje, corriente y desplazamiento.....	-38-
Figura 55. Graficas de voltaje, corriente y desplazamiento.....	-40-
Figura 56. Graficas de voltaje, corriente y desplazamiento.....	-41-

ÍNDICE DE TABLAS

Tabla 1. Estrategia Número 1.....	-38-
Tabla 2. Estrategia Número 2.....	-39-
Tabla 3. Estrategia Número 3.....	-40-

INTRODUCCIÓN

El presente documento está enfocado en los estudiantes de la Universidad Politécnica Salesiana sede Guayaquil en donde ellos puedan ver la aplicación de materias de la malla curricular tales como Electiva I (Robótica), Teoría de control (I y II), Dinámica (I y II) y Programación como el enfoque a la robótica móvil de competencia, configurar diferentes estrategias de combate y analizar la mejor de ellas enfocado al consumo de variables tales como son el voltaje y el amperaje. En este proyecto técnico será entre el control de motores mediante drivers de potencia en conjunto con una gama de sensores de presencia como también la adquisición de datos para posteriormente analizar las magnitudes antes mencionadas.

Con la implementación del prototipo robótico planteado en este documento, se pueden demostrar varios temas del pensum académico, como tiene que ver en temas de programación de Microcontroladores, comunicación de drivers de potencia con el microcontrolador, adquisición y proceso de señales provenientes de los sensores, varias demostraciones de la implementación de las materias antes mencionadas como son el diseño mecánico de la caja de reducción, el diseño de la tarjeta de control, algoritmos de control y programación de diferentes estrategias de combate.

1. EL PROBLEMA.

1.1. Descripción del Problema.

En la Universidad Politécnica Salesiana sede Guayaquil específicamente en el Club de Robótica, se implementan diferentes prototipos de competencia de las cuales una de las más importantes y competitivas es la categoría de los robots sumos autónomos. En la actualidad la categoría robot sumo autónomo es una de las más competitivas tanto a nivel local como internacional, tanto así que se ha llegado a celebrar un mundial de esta categoría de robótica de competencia.

En la Universidad se cuenta con 2 prototipos de la categoría robot sumo, pero bajo la modalidad radio control (RC), lo cual limita a participar en una sola modalidad siendo así la Universidad Politécnica Salesiana una de las Universidades más competitivas a nivel nacional se debe mantener a la vanguardia de estos desarrollos tecnológicos y poner a prueba tanto a nivel local como internacional

El prototipo de Robot Sumo pone a prueba los conocimientos teóricos adquiridos con el paso de las materias antes mencionadas, para que los estudiantes tengan una visión más cercana a la implementación de un autómatas.

1.2. Importancia y Alcance.

El prototipo del proyecto técnico es de mucha importancia para el club de robótica de nuestra sede dada la implementación de diseños mecánicos y electrónicos ya que esta es la implementación de materias impartidas y directamente ligadas a la electrónica y automatización.

1.3 Delimitación del Problema.

Dentro de la larga trayectoria que tenemos como club de robótica de la universidad politécnica salesiana tanto en torneos a nivel local como a nivel internacional se ha logrado identificar cuáles son los algoritmos de programación que dan muy buenos resultados a esto debemos implementar mejoras en base a los algoritmos y respuesta mecánica para poder mejorar el desempeño de nuestros prototipos.

1.3.1 Delimitación Temporal:

La puesta en práctica del proyecto técnico duró en un tiempo de 17 meses y completó en el mes de marzo del 2020.

1.3.2 Delimitación Espacial.

El prototipo del proyecto va a ser aprovechado en el Laboratorio del Club de Robótica de universidad politécnica salesiana de la Sede Guayaquil.



Figura 1. Delimitación espacial del prototipo del Proyecto Técnico. [1]

1.3.3 Delimitación Académica.

El proyecto provee de un alcance académico que se sustenta mediante su empleo en las diferentes competencias de robótica tanto a nivel nacional como internacional que engloba conocimientos obtenidos a lo largo de la carrera.

Innovación: El prototipo y su aplicación de ser un robot de competencia, también proporciona una opción en la industria de sensar objetos en una banda transportadora con una alta velocidad de respuesta a los objetos sin necesidad de tener un operario encargado de verificar el estado de los objetos en la banda transportadora, y en el área de la robótica servirá para muchas funciones como motivo de módulo de práctica.

Impacto: En el ambiente industrial este tipo de prototipos y su aplicación de identificación de objetos es muy importante gracias a su alta respuesta de identificación de objetos ya sea en una banda transportadora o una línea de ensamblaje.

1.4 Objetivos.

1.4.1 Objetivo General.

Repotenciar un robot sumo comandado a distancia implementando rutinas de control para su funcionalidad en modo autónomo.

1.4.2 Objetivos Específicos.

- Dimensionar, diseñar e implementar las conexiones respectivas para control y fuerza del prototipo.
- Programar 3 estrategias para el modo autónomo en el robot sumo.
- Monitorear y analizar los parámetros eléctricos después de cada combate.
- Obtener trayectoria de desplazamiento después de cada combate.
- Analizar las trayectorias después de cada combate según la estrategia programada.

2. FUNDAMENTACIÓN TEÓRICA.

En esta sección se podrán obtener conceptos iniciales basados al enfoque del proyecto y cómo influyen en nuestra sociedad hoy en día.

2.1 Robótica Móvil

Como rama de la industria que se desarrolla rápidamente y que está orientada a encontrar soluciones, la robótica móvil reúne ciencias del ámbito de la ingeniería e informática, a la vez que ciencias cognitivas, inteligencia artificial y otras disciplinas. Solo mediante la cooperación entre todos estos ámbitos especializados es posible abarcar la complejidad de los robots móviles. Estos robots son capaces de moverse autónomamente y de ejecutar determinadas acciones. Junto a la movilidad, por tanto, también la autonomía, esto es, la independencia del robot respecto de la intervención humana es uno de los aspectos esenciales de la robótica móvil. En este contexto, se suele distinguir entre aparatos semiautónomos y totalmente autónomos, si bien la autonomía se define sobre todo en términos de abastecimiento de energía de los robots. Si, por ejemplo, se requiere un abastecimiento externo como, por ejemplo, una estación de carga, se considera que la autonomía es restringida.

A pesar de que la robótica móvil actualmente aún se encuentra en un estadio relativamente temprano de su desarrollo, hace varios años que se utilizan prototipos y productos en serie en diferentes sectores. Con aspiradoras casi totalmente autónomas, la robótica móvil ya ha dado un paso para introducirse en la vida cotidiana. Las innovaciones en el ámbito de la inteligencia artificial y su interconexión harán crecer enormemente la relevancia y la utilización de robots móviles autónomos en el futuro.

2.1.1 De estática a móvil: la robótica en el transcurso del tiempo

Si bien a lo largo de muchos años la robótica se centraba en aspectos tales como la capacidad de rendimiento, la precisión de repetición y la velocidad, desde hace algún tiempo se han introducido criterios que responden a la transformación industrial y que han adquirido también una gran relevancia. En casi todos los sectores, sobre todo en el sector productor, entre las condiciones básicas de los robots modernos se encuentran la flexibilidad, la adaptabilidad y la actuación autónoma. Mediante las

características mencionadas queda manifiesto que: el desarrollo y el progreso de la industria se basan en gran medida en la robótica móvil y en técnicas de automatización relacionadas. Pero no solo la rama industrial se ve beneficiada: las innovaciones de la robótica influirán en el futuro en todas las facetas de nuestra vida.

2.1.2 Características de la robótica móvil:

- Percepción mediante sistemas sensores
- Capacidad de adaptación ante circunstancias espaciales modificadas
- Navegación, planificación y actuación autónomas
- Software/programación orientados a tareas

2.1.3 Sensores y actores: pilares de la robótica móvil

Junto al software y a la electrónica de control, los sensores y actores componen la estructura básica de cada robot móvil. Los sensores sirven para obtener informaciones y datos de muy diferentes tipos, los cuales se necesitan para realizar la localización y para planear movimientos. El sistema sensor en el interior del aparato (por ejemplo, aparatos de medición de tensión, termómetros y radiorreceptores) proporciona información sobre el estado actual del sistema. Los sensores externos integrados en la robótica móvil, tales como teclas, lectores de códigos de barras, sensores de par motor y de velocidad suministran al sistema los datos que se requieren para la secuencia de movimientos.

De llevar a la práctica los comandos de movimiento se encargan los actores. También ellos actúan o bien interna o bien externamente. Los actores internos modifican el estado del sistema; así, entre otros, el ventilador y la calefacción regulan la temperatura interior. Los actores externos tales como motores y bombas sirven, por su parte, para la locomoción del robot o para la ejecución de actividades mediante manipuladores tales como el brazo del robot.

2.1.4 La robótica móvil y sus ámbitos de aplicación

El espectro de aplicación de la robótica móvil es casi ilimitado ya que, gracias a la tecnología inteligente y al modo de trabajo independiente, se pueden utilizar aparatos respectivos de muy diversas maneras. Su utilización no se reduce a las naves de producción: también en situaciones cotidianas encuentra su lugar la robótica móvil. Un ejemplo de ello son las aspiradoras semiautónomas. Al igual que sus predecesores tradicionales, limpian suelos domésticos mediante un ventilador que genera presión negativa, asimilando así partículas de suciedad. Al contrario que los modelos de manejo manual, las aspiradoras autosuficientes no necesitan que las guíe una persona, sino que siguen patrones de movimiento preprogramados y,

gracias al sistema sensor de suelo, pueden reaccionar de forma dinámica a impedimentos tales como escalones. Los modelos mencionados solo están limitados desde el punto de vista de su abastecimiento de energía.

En el marco de fabricaciones industriales, la robótica móvil se utiliza sobre todo en el ámbito del transporte. Cuando se deben transportar materiales necesarios para la producción a una determinada máquina para su procesamiento posterior o piezas para ser almacenadas, robots especiales se encargan del transporte. La modernización de estructuras logísticas y operaciones eleva la eficiencia, reduce a largo plazo los gastos y contribuye a la seguridad del entorno de trabajo. Como elemento conector entre los pasos de trabajo particulares, la robótica móvil supone, ya hoy, un componente importante de la denominada fábrica inteligente.

Entre los ámbitos de aplicación de la robótica móvil se encuentran, además, la prevención del terrorismo, la protección civil (contra catástrofes) y el ámbito militar. Ya sea en el fondo del mar, al desactivar y desalojar un artefacto explosivo o como dron en el aire: los robots semiautomáticos asumen tareas en las que un ser humano podría herirse. Otros campos de aplicación de la robótica móvil son, entre otros, la agricultura y el tráfico por vías públicas, incluyendo vehículos autónomos (sin conductor). [2]

2.2 Descripción del Robot Sumo

Prototipo con una serie de sensores que imita el deporte japonés sumo sobre un dohyo. El objetivo de la contienda es lograr que el robot oponente se salga del área de combate.

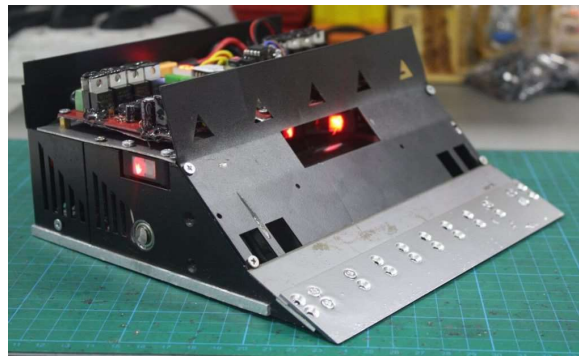


Figura 2. Prototipo de robot-sumo. [3]

2.2.1 Propósito del robot-sumo

El robot sumo se ha diseñado para poner a prueba conocimientos tanto de electrónica, diseño mecánico y lenguajes de programación.

2.2.2 Historia del All Japan Robot Sumo Tournament

Se inició como una celebración del 20 aniversario de Fuji Soft Co., Ltd., comenzó por iniciativa de Hiroshi Nozawa presidente del fundador de la Compañía.

Nozawa muestra el mismo como presidente el tamaño del robot y el anillo de juego o dohyo, el tamaño del campo de juego es aproximadamente un tercio del campo real de juego de 15 pies (455cm) a un diámetro de 154 cm, altura de 5 cm. El robot de peso es del 3 kg estándar, altura 20 cm × 20 cm y total libertad en la altura del robot.

El presidente de Nozawa también tuvo dificultades para decidir las reglas del juego. En ese momento, no había ningún precedente para la competencia mundial de robots de tipo batalla, y no había ninguna referencia en absoluto así que se obtuvo la sabiduría del profesor Katsuya Nagata de la Universidad de Waseda, el profesor Kasuga Wisdom del Instituto de Tecnología de Shibaura y todos los estudiantes, las reglas se crearon como esqueletos de prueba y error. La regla de los 5 segundos autónomos también se decidió en este punto. Reclutamos principalmente universidades y celebramos una pelea experimental en agosto de 1989.

Se obtuvieron muchos datos de rendimiento de esta competencia y nos convertimos en la regla actual del juego. Bueno, se examinó una gran cantidad de material de superficie del anillo, en ese momento seleccionamos un material de cloruro de vinilo que es fácil de procesar y duradero, además del color, el ancho, la posición, etc. de las balas / línea divisoria, la influencia del sensor, la apariencia, sumo, una variedad de tácticas divide la victoria y la derrota su atractivo es como si siguieras el gran sumo real.

Seguiremos evolucionando de ahora en adelante como nombre y realidad, como la mejor competencia de robots de batalla del mundo.



Figura.3 Inicios de All Japan robot sumo Tournament [4]

2.3 Estrategias de control

2.3.1 Slow Search

La primera estrategia se la conoce como “SlowSearch” y trata de que el autómeta se coloque en uno de los extremos de la pista con el frente hacia el centro de la pista este avanzará dando un “salto” de 500ms y al mismo tiempo estará leyendo la información que envían los sensores de distancia, una vez censado y ubicado el objetivo el autómeta se dirigirá hacia su objetivo a máxima velocidad para poder desplazar su objetivo fuera de la pista.

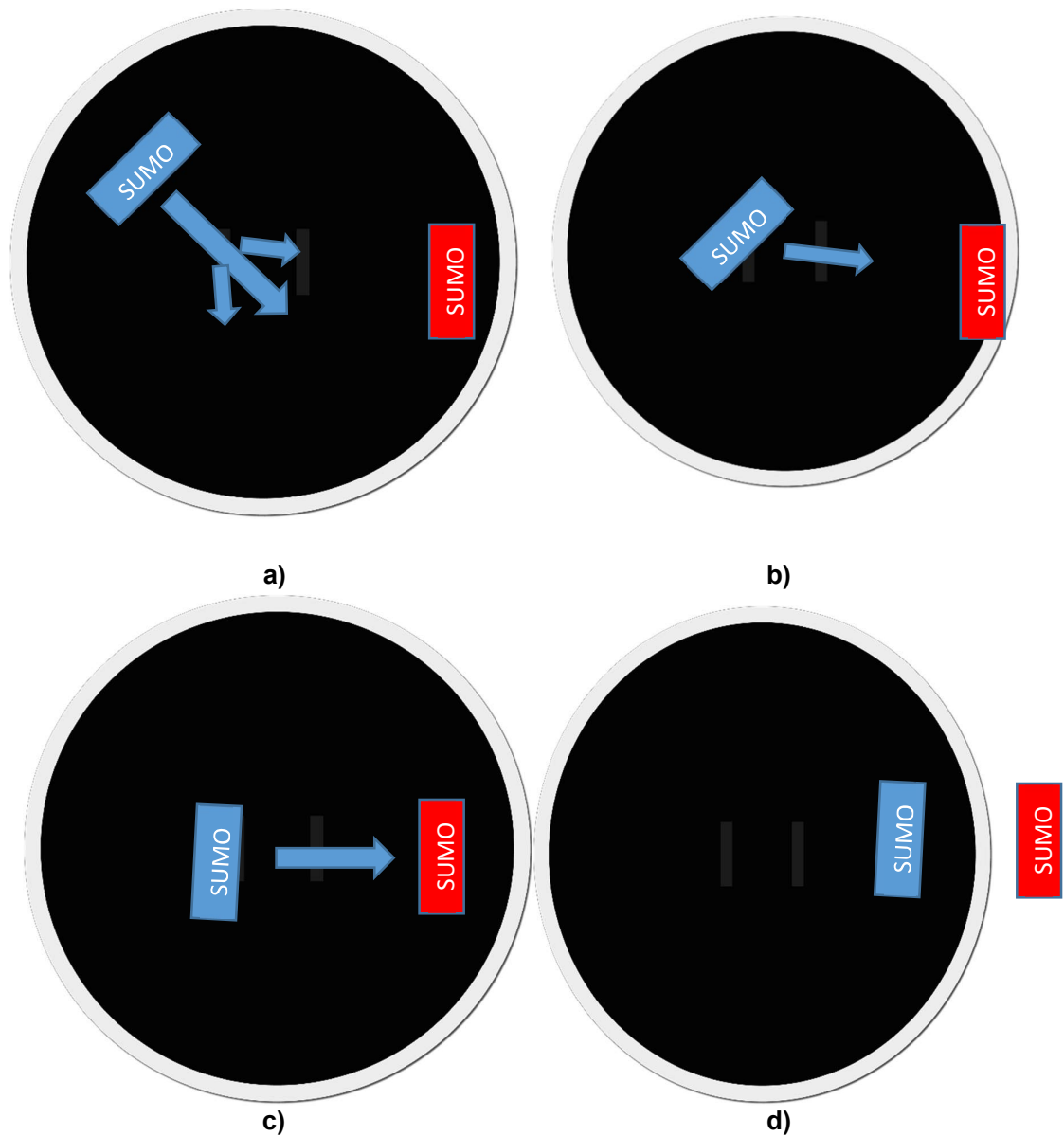


Figura 4. a; b; c; d; Desarrollo de estrategia uno

2.3.2 WoodPecker

La segunda estrategia se la conoce como "WoodPecker" y trata de que el autómata se dirija hacia adelante dando saltos y se detenga cada 200ms, cada vez que se detenga debe tomar lectura de los sensores de distancia para obtener información del objetivo tanto distancia y posición dentro de la pista, de esta misma manera cada vez que el autómata se desplace hacia adelante debe corregir la posición hacia su objetivo y girar un poco hacia la derecha o hacia la izquierda dependiendo de la posición del objetivo, una vez que el autómata este a una distancia optima del objetivo (aproximadamente 60 cm) el autómata atacara suministrando todo el voltaje a los motores para que el mismo valla a todas sus rpm y poder desplazar el objetivo fuera de la pista.

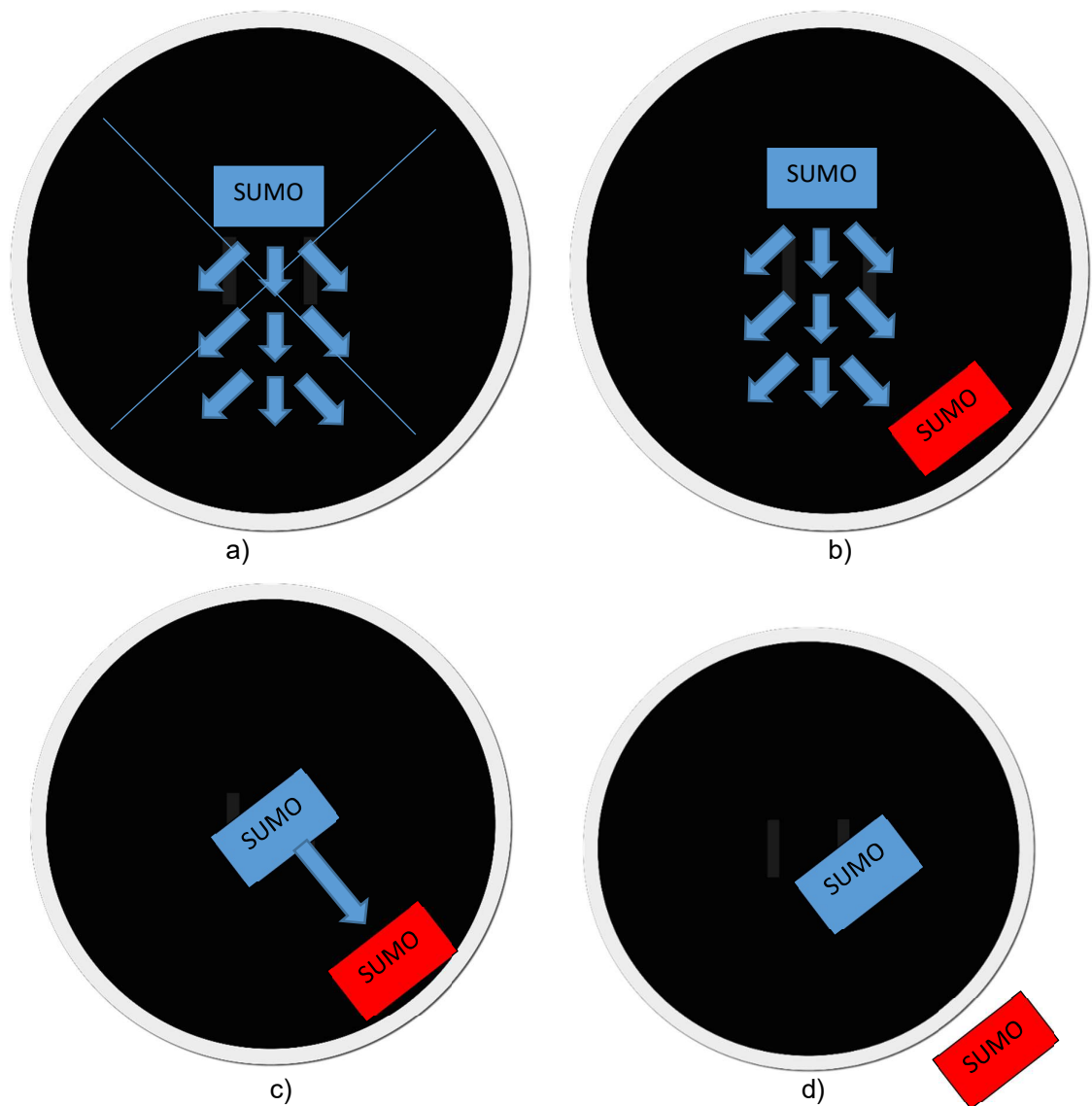


Figura 5. a; b; c; d; Estrategia en funcionamiento

2.3.3 Tornado

La tercera estrategia se la conoce como “Tornado” y trata de que el autómeta empiece a girar a una velocidad normal sobre su propio eje al mismo tiempo está obteniendo la lectura de los sensores (esto hace analogía a un radar de 360°) y una vez que encuentra el objetivo se lanza hacia el con su máxima velocidad y así poder lograr su objetivo que consiste en sacarlo fuera de la pista.

Una vez obtenidos los datos tanto de voltajes y corrientes se pasará a hacer una comparativa de cada una de las magnitudes antes mencionadas y se obtendrá con cuál de las estrategias el robot tiene un uso óptimo de su fuente de energía.

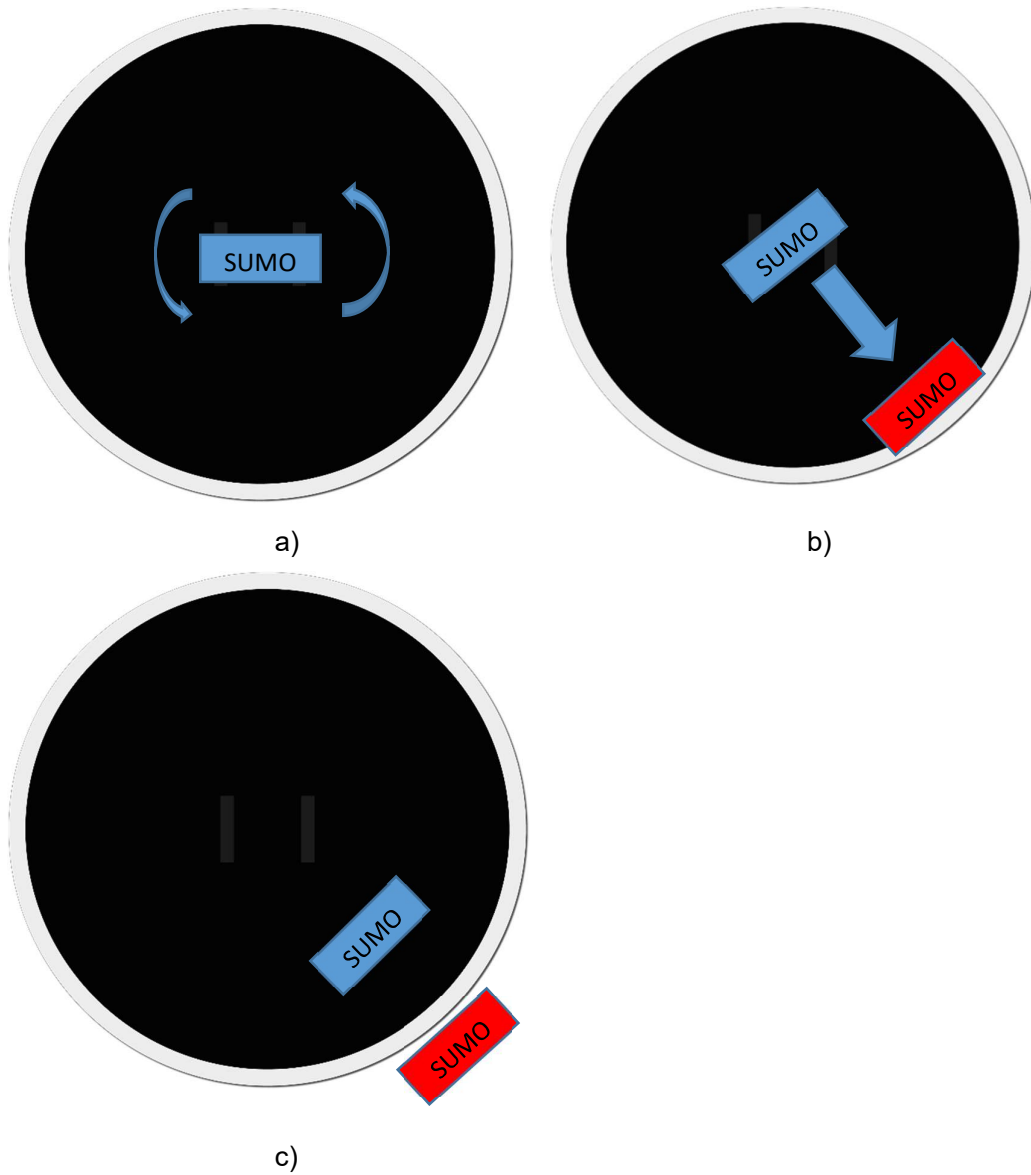


Figura 6. a; b; c; Desarrollo de la estrategia tres

2.4 Sensores y Actuadores

2.4.1 Sensores de distancia

Para que el prototipo detecte a sus contrincantes usa 2 tipos de sensores de distancia de la marca Panasonic 3 del modelo AMBA 345912 con una distancia de censado de hasta 120cm y tiempo de respuesta mínimo de 500ms y 2 del modelo AMBA2409 con distancia de censado de hasta 80 cm y tiempo de respuesta de 500ms



Figura 7. Referencia sensor AMBA 2409 [5]



Figura 8. Referencia sensor AMBA 345912



Figura 9. Sensor QRT-1A [6]

2.4.2 Alimentación del Prototipo

El prototipo se alimentará de 2 baterías de litio y polímero (lipo) de 4 celdas y 860 mAh con un factor de descarga de 100c de la marca maxamps ambas conectadas en serie. Esto significa que nuestro prototipo se alimentara con un aproximado de 30 a 32 voltios.

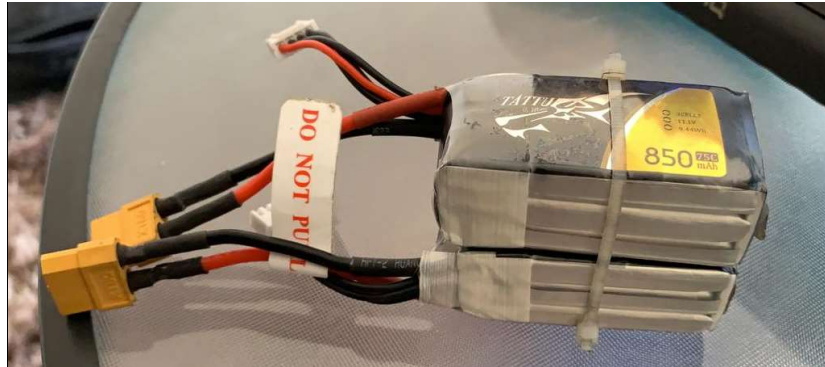


Figura 10. Batería LIPO 3s 850 mAh

2.4.3 Controlador de Potencia

El controlador que usa el prototipo es llamado Roboclaw 2x45st, fabricado por la compañía pololu y tiene las siguientes prestaciones:

- Doble canal
- Voltaje máximo de entrada 32V.
- 45 amperios por canal continuos
- 60 amperios por canal pico
- Además de esto este controlador de potencia ofrece una función para trabajar en modo radiocontrol (RC) otorgándole mucha más diversidad a sus funciones.

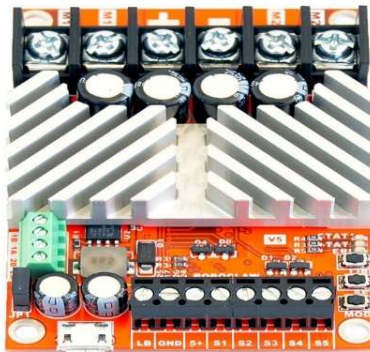


Figura 11. Roboclaw 2x45st [7]

2.4.4 Motores

Los motores que se emplearan en el desarrollo del prototipo son los Maxon 148867 trabajan desde 12 hasta 24 voltios con una potencia de 150 watts a máximo voltaje, una velocidad de 7580 rpm trabajando al vacío y un torque constante de 30.2 mNm/A.



Figura 12. Motor Maxon [8]

2.4.5 Microcontrolador ATMEGA328-SMD

El ATMEGA328-SMD es un microcontrolador de montaje superficial con muy altas prestaciones para este tipo de prototipos de competencia gracias a su minúsculo tamaño y alto desempeño en función al procesamiento de datos y ejecución de comandos previamente programados, para poder lograr ese desempeño este microcontrolador posee las siguientes características técnicas:

- Chip: ATMEGA328P-AU
- Formato SMD: TQFP
- Pines: 32
- Memoria FLASH: 32KB
- Memoria RAM: 2KB
- Máxima frecuencia de funcionamiento: 20Mhz
- CPU: 8-bit AVR
- Pines de entrada/salida: 23
- Entradas Analógicas (ADC): 8 [9]

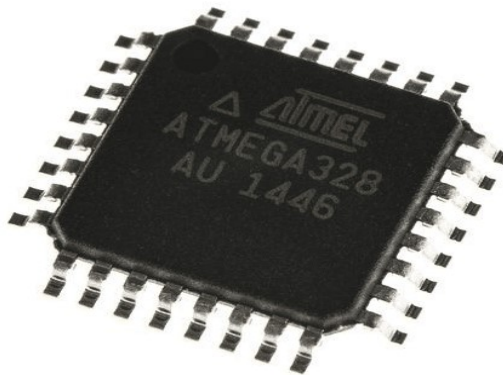


Figura 13. ATMEGA328-SMD microcontrolador de nuestra tarjeta de control [9]

2.4.6 Sensor de corriente ACS709

Este sensor de corriente es una placa portadora o placa de conexión para el sensor de corriente lineal basado en el efecto Hall ACS709LLFTR-35BB-T de Allegro con salida de falla por sobrecorriente,

El sensor tiene un voltaje de funcionamiento de 3 V a 5.5 V y una sensibilidad de salida de 18.5 mV / A cuando Vcc es 3.3 V (o 28 mV / A cuando Vcc es 5 V)

La siguiente lista detalla algunas de las características clave del sensor:

Precisión optimizada para corriente de entrada bidireccional de -37.5 A 37.5 A, con un rango de detección lineal de -75 A 75 A

La resistencia interna del camino conductivo es típicamente de 1.1 mΩ, y la PCB está hecha con cobre de 2 onzas, por lo que se pierde muy poca energía en la placa.

El blindaje integrado reduce en gran medida el acoplamiento capacitivo del conductor de corriente a la matriz y evita la deriva desplazada en aplicaciones de lado alto.

El uso de un sensor de efecto Hall significa que el IC puede aislar eléctricamente la ruta de corriente de la electrónica del sensor (hasta 2.1 kV RMS), lo que permite que el sensor se inserte en cualquier lugar a lo largo de la ruta de corriente y se use en aplicaciones que requieren electricidad aislamiento.

Alta precisión y fiabilidad: error de salida total típico del 2% a temperatura ambiente con calibración de fábrica, un voltaje de salida inactivo extremadamente estable
Rango de temperatura de funcionamiento de grado automotriz de -40 ° C a 150 ° C.
Umbral de sobrecorriente configurable por el usuario.

La salida de pin de FALLO se bloquea cuando la corriente excede el umbral configurado durante un tiempo que se puede configurar mediante la adición de un condensador externo. [10]

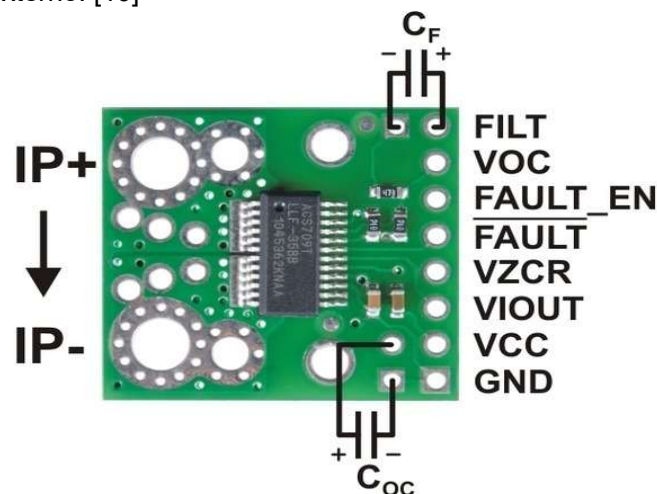


Figura 14. Sensor de Corriente ACS709 [10]

2.4.7 Sensor de Voltaje FZ0430

El módulo sensor de voltaje FZ0430 es un módulo bastante simple, el cual consiste en un divisor de tensión conformado por dos resistores uno de 7.5K y uno de 30K, con el cual podremos ser capaces de reducir el voltaje medido en una razón de 0.2 y por lo tanto seremos capaces de medir una tensión máxima de 25 V con procesador de 5V. [11]

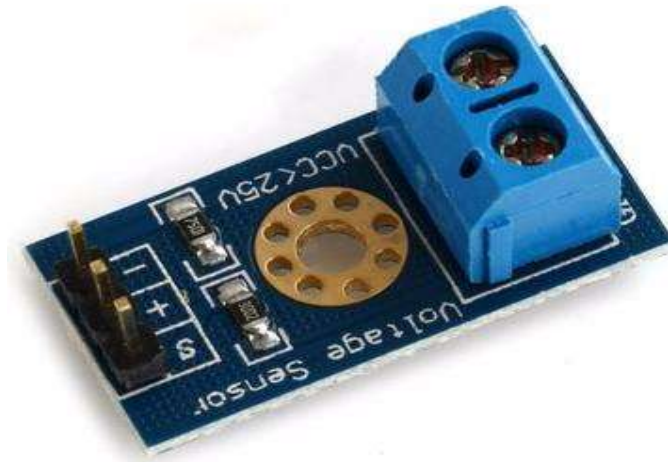


Figura 15. Sensor de voltaje FZ0430 [11]

3 Marco Metodológico

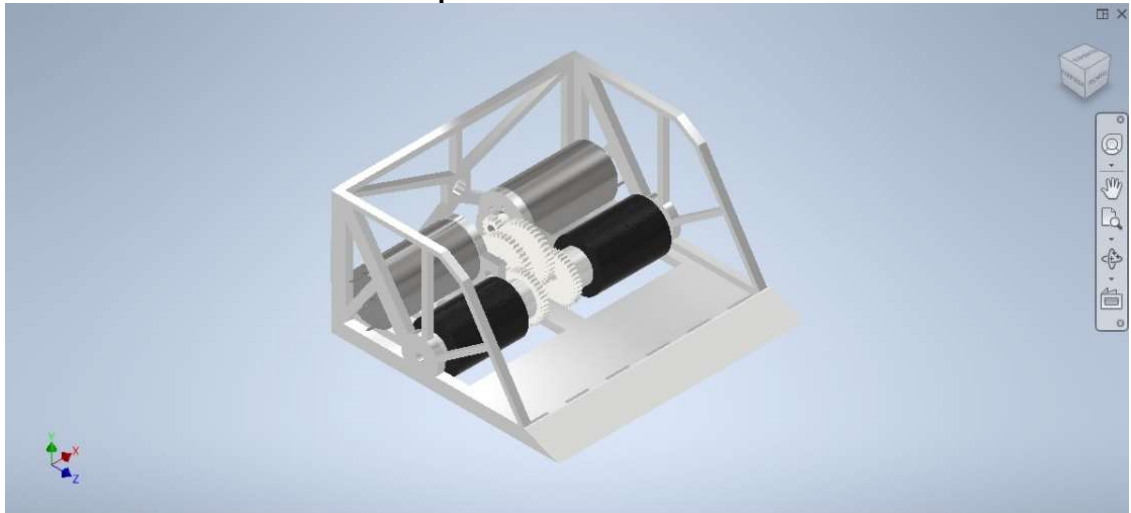
En esta etapa veremos y conoceremos un poco más a detalles los diferentes módulos que emplearemos a lo largo del desarrollo de este proyecto técnico.

3.1 Diseño Mecánico

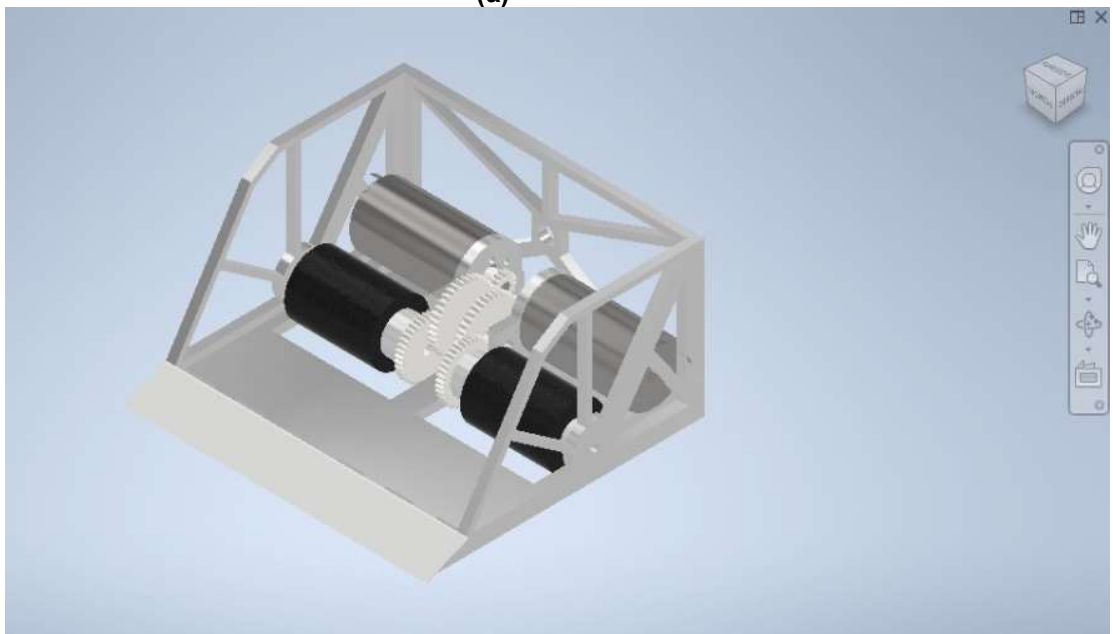
Para el diseño mecánico de nuestro prototipo hemos utilizado el software de Autodesk Inventor el mismo que es utilizado en diferentes ámbitos y campos del diseño tanto mecánico como estructural.

Esta parte es muy importante para el desarrollo de nuestro prototipo a futuro ya que convertirá las altas revoluciones de los motores a torque. Esto permitirá que nuestro prototipo sea mucho más eficiente y robusto en cuestión fuerza. Adicional a esto sin un correcto desempeño mecánico no podremos cumplir efectivamente con las estrategias propuestas y posteriormente no obtendremos resultados positivos en la implementación del proyecto.

Prototipo modelado en 3D



(a)



(b)

Figura 16. Plano de la base en 2D(a) Plano de la base en 3D (b)

Paredes Laterales

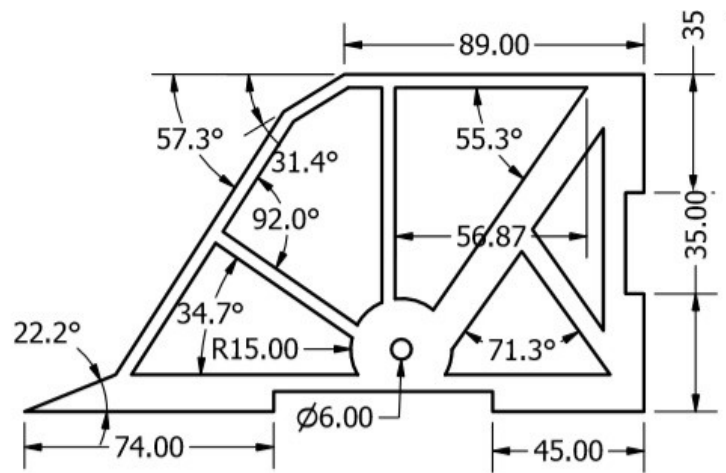


Figura 17. Pared Lateral con vista Lateral Acotada

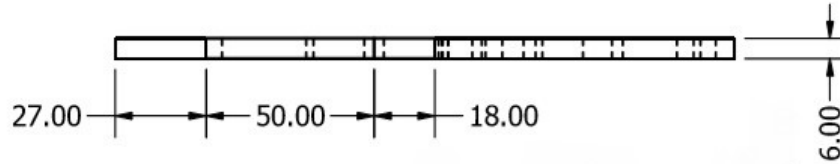


Figura 18. Pared Lateral con vista Superior

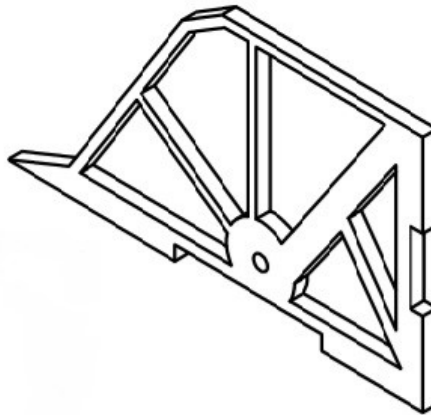


Figura 19. Modelado 3D

Pared Trasera

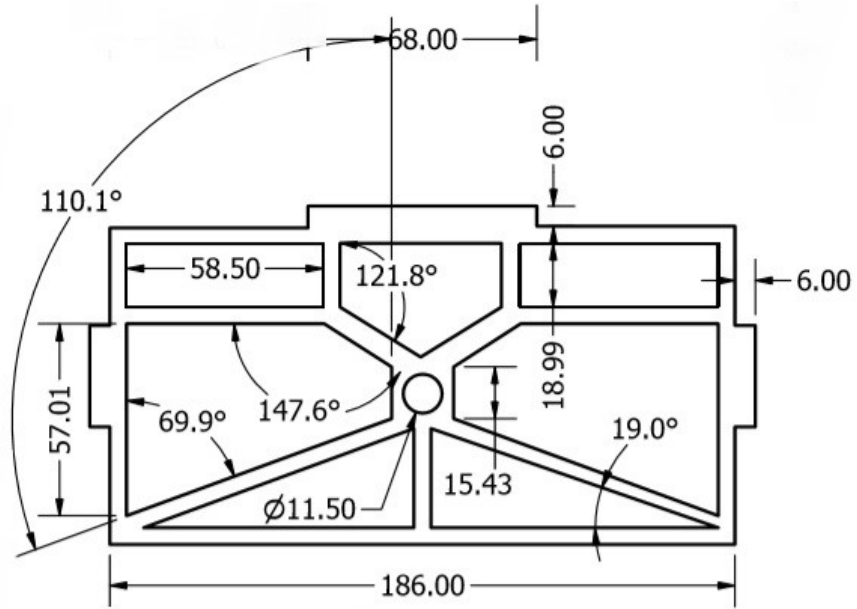


Figura 20. Pared Trasera vista Frontal Modelado 2D

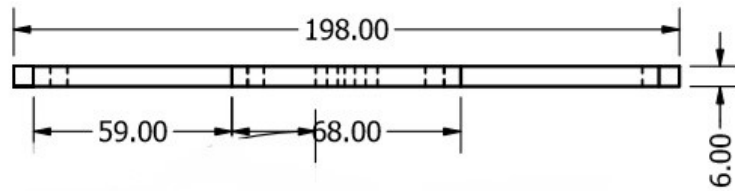


Figura 21. Pared Trasera Vista Superior Acotada

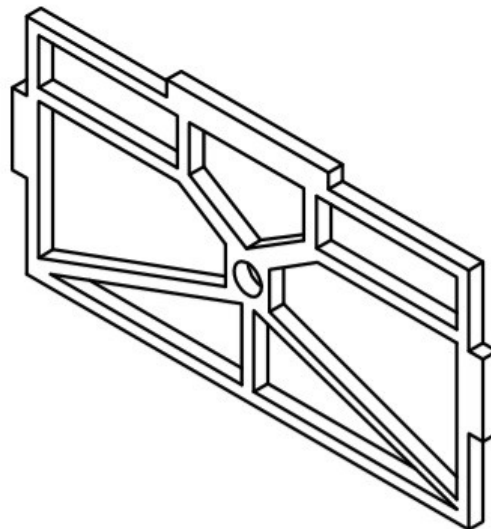


Figura 22. Pared Trasera Modelado en 3D

Base

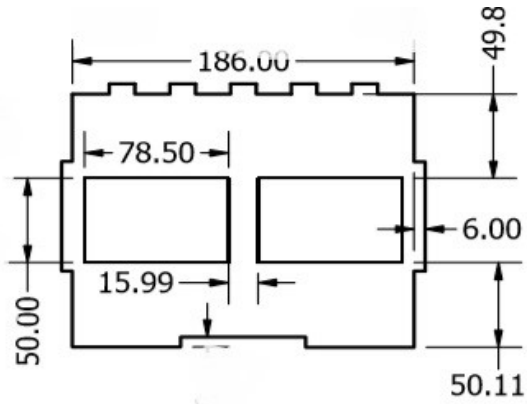


Figura 23. Base vista superior

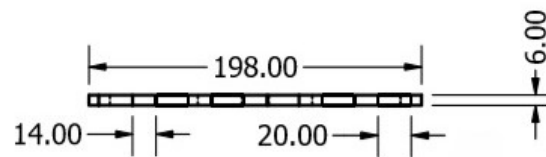


Figura 24. Base vista Frontal

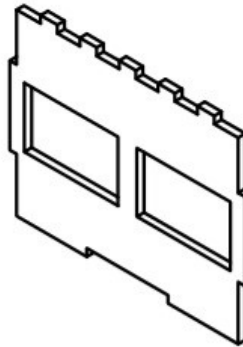


Figura 25. Base Modelada 3D



Figura 26. Prototipo Ensamblado

3.2 Diseño de la tarjeta de control

Esta es una de las partes más importantes de nuestro proyecto ya que esta es la parte donde asignaremos la programación además de eso es donde obtendremos y procesaremos las diferentes señales obtenidas por la diferente gama de sensores que utilizaremos entre ellos la distancia a el objetivo, también podremos analizar el voltaje y el amperaje de nuestro prototipo con cada una de las estrategias asignadas, además es el complemento de nuestra parte mecánica una vez que obtengamos un equilibrio entre la parte mecánica y la tarjeta de control de nuestro prototipo obtendremos resultados positivos para nuestro proyecto. Esta tarjeta de control la diseñaremos en el software EAGLE AUTODESK que es una herramienta muy completa al momento del diseño electrónico

3.3 Arduino

Arduino en la actualidad es uno de los softwares mas utilizados en el mercado gracias a su flexibilidad y adaptacion a varios tipos de proyectos, esto sumado a la gran cantidad de informacion disponible en el internet hacen de este software

3.5 Topografía de la tarjeta de control

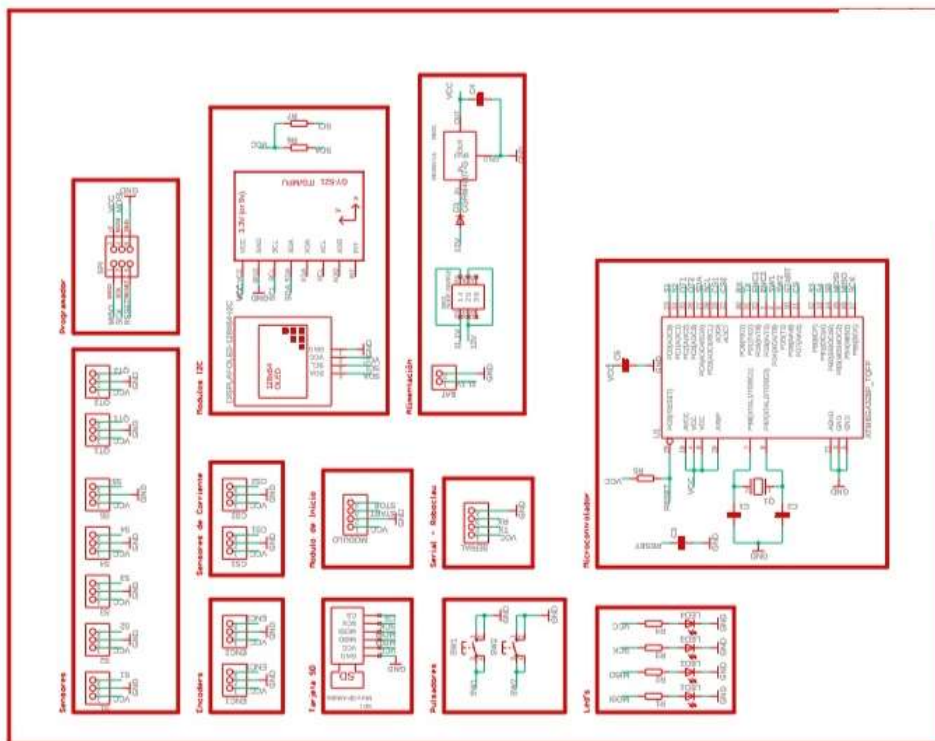


Figura 27. Topografía General de la tarjeta de control

Sensores

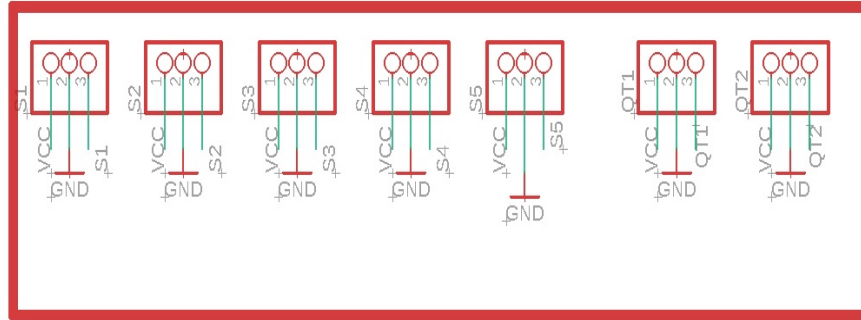


Figura 28. Topografía de los sensores

Se observa el diagrama de conexiones que llevarán los sensores AMBA y QRT respecto a la tarjeta controladora.

Programador

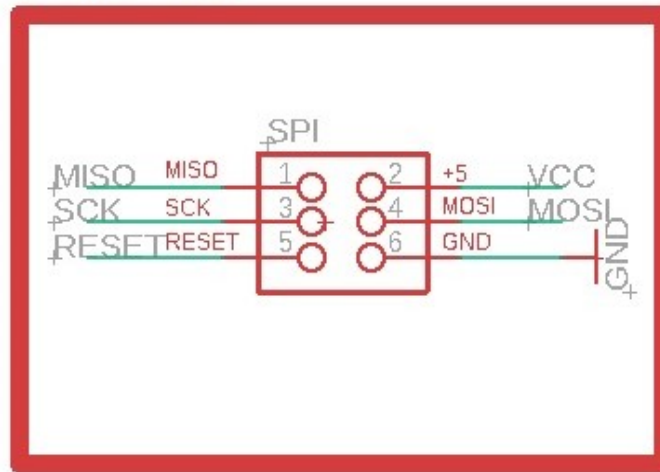
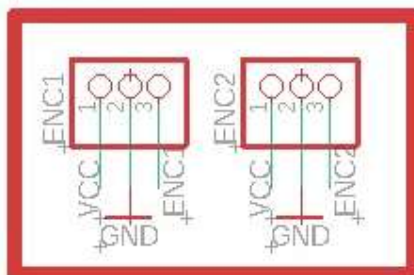


Figura 29. Topografía del Programador

Conexión del Programador en comunicación SPI, mismo que nos permitirá cargarle nueva programación a la tarjeta de control.

Encoders



Sensores de Corriente

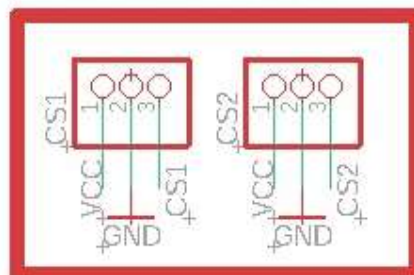


Figura 30. Topografía del Sensor de Corriente y Encoders

Se observa la conexión que usa tanto los encoders y el sensor de corriente que ayudaran almacenando datos de movilidad y corriente respectivamente

Tarjeta SD

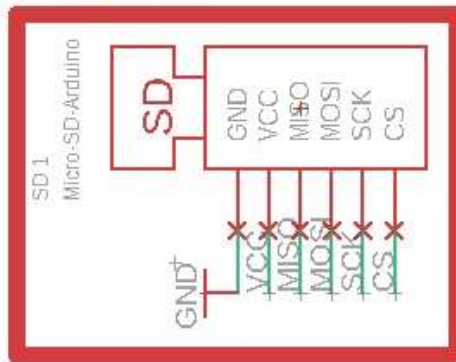


Figura 31. Topografía Modulo SD

Se muestra la conexión necesaria para poder utilizar el modulo de almacenamiento de datos en una micro SD, posteriormente analizar los datos obtenidos de las diferentes constantes electricas.

Modulo de Inicio

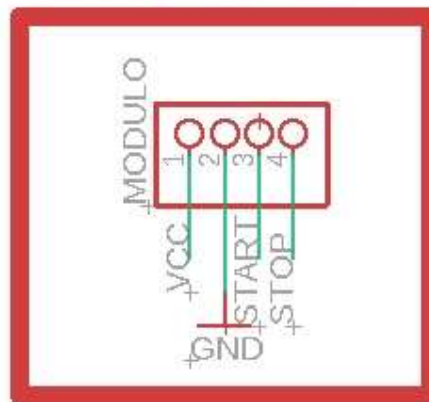
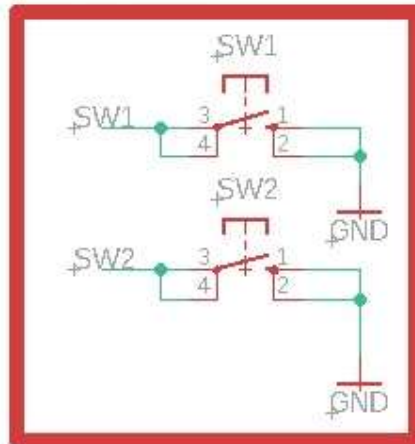


Figura 32. Topografía Modulo de Inicio

Se muestra la conexión necesaria para el modulo de arranque, que nos permitira encender y apagar el prototipo.

Pulsadores



Serial - Roboclaw

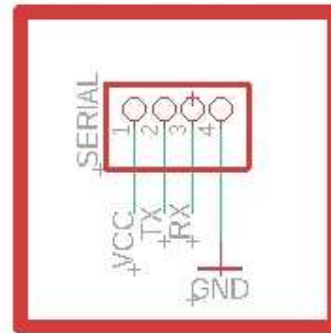


Figura 33. Topografía de Pulsadores y comunicación Serial con la Roboclaw
Se muestra la conexión de los pulsadores que permiten la selección de las diferentes estrategias cargadas en el prototipo, también la conexión de comunicación serial en la roboclaw.

Alimentación

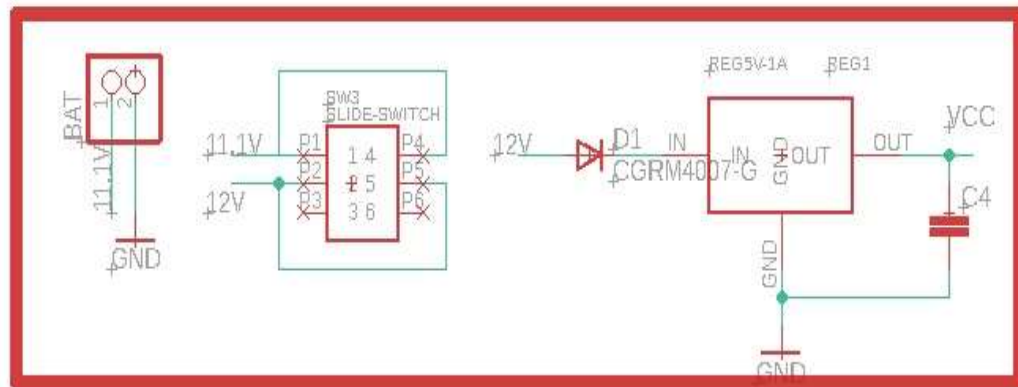


Figura 34. Topografía de la Alimentación de energía
Diagrama de conexiones de la batería hacia la tarjeta de control.

Led's

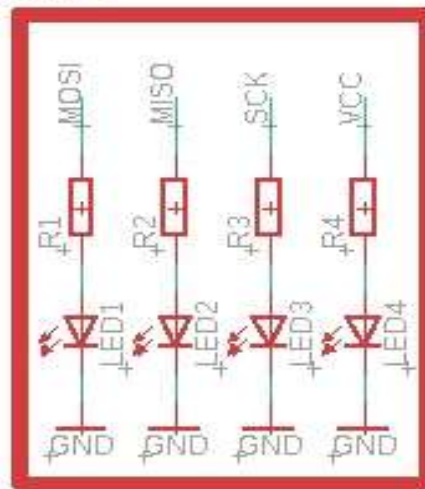


Figura 35. Topografía LEDS

Conexión de Leds y resistores hacia la tarjeta controladora.

Microcontrolador

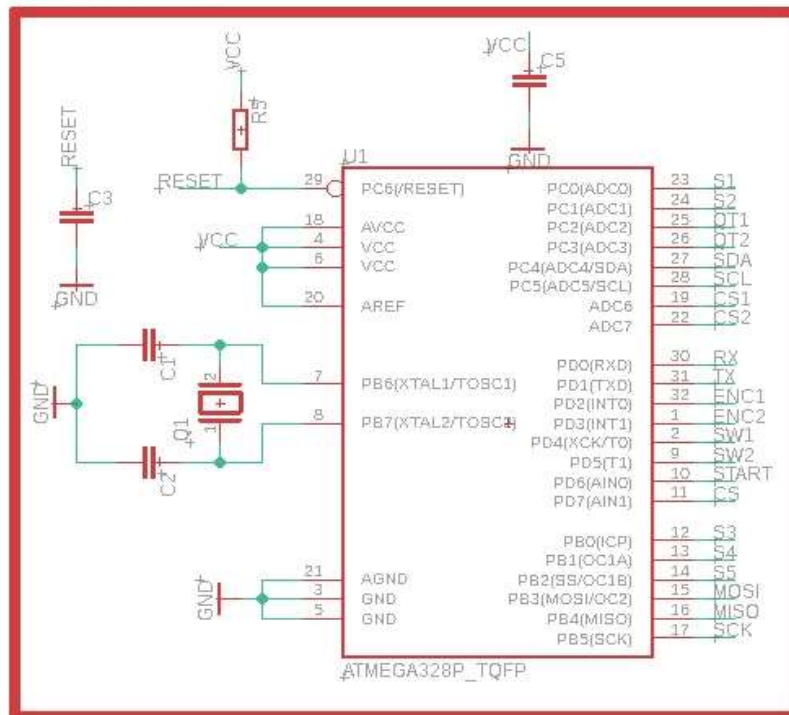


Figura 36. Topografía Microcontrolador

Conexiones del microcontrolador hacia cada componente de nuestro prototipo entre ellos sensores,switch,leds,spi entre otros.

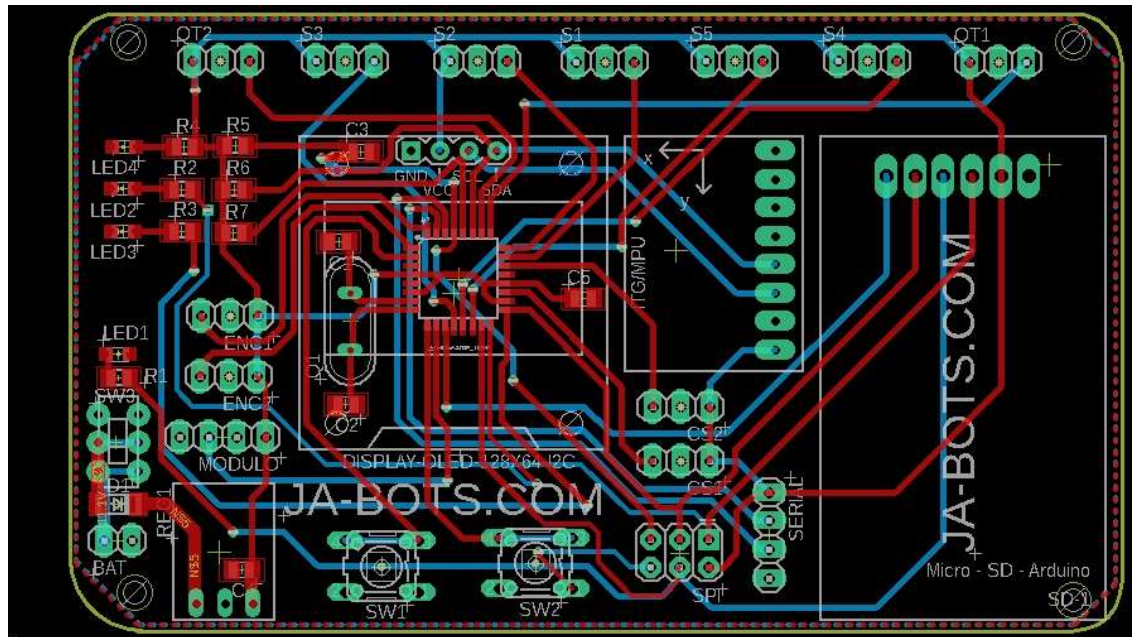


Figura 37. Vista de las conexiones internas de la tarjeta controladora
Se muestra todas las conexiones internas de la tarjeta controladora.

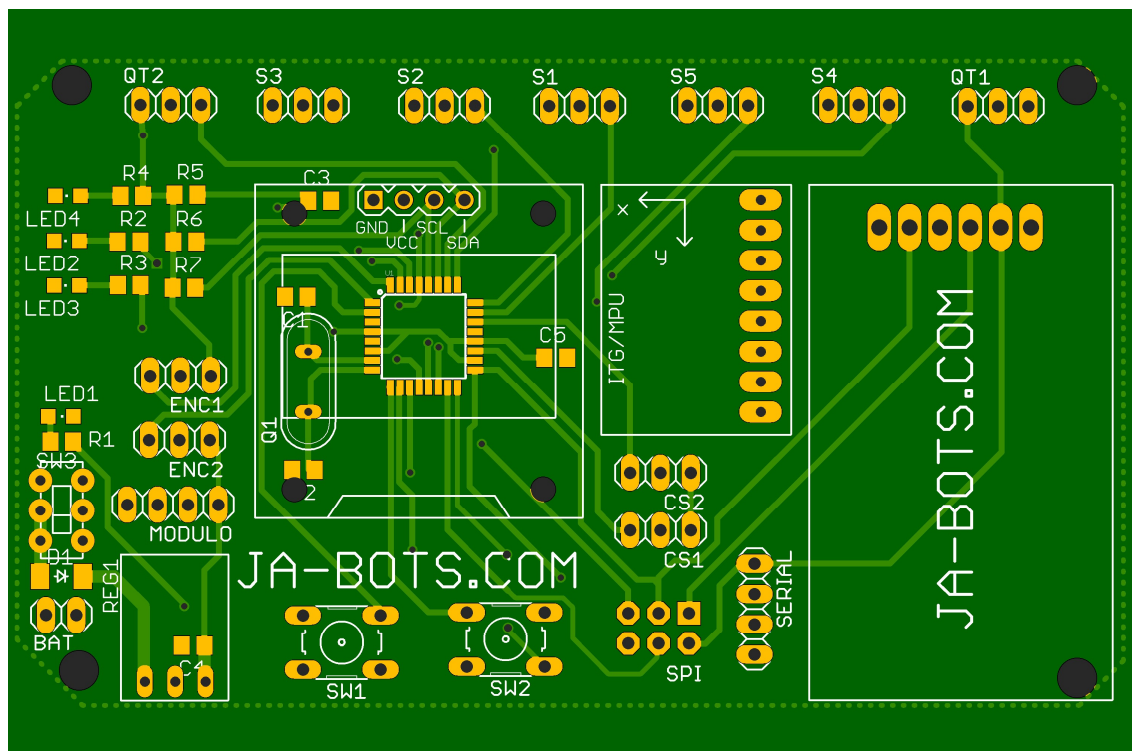


Figura 38. Vista de manufactura de la tarjeta controladora

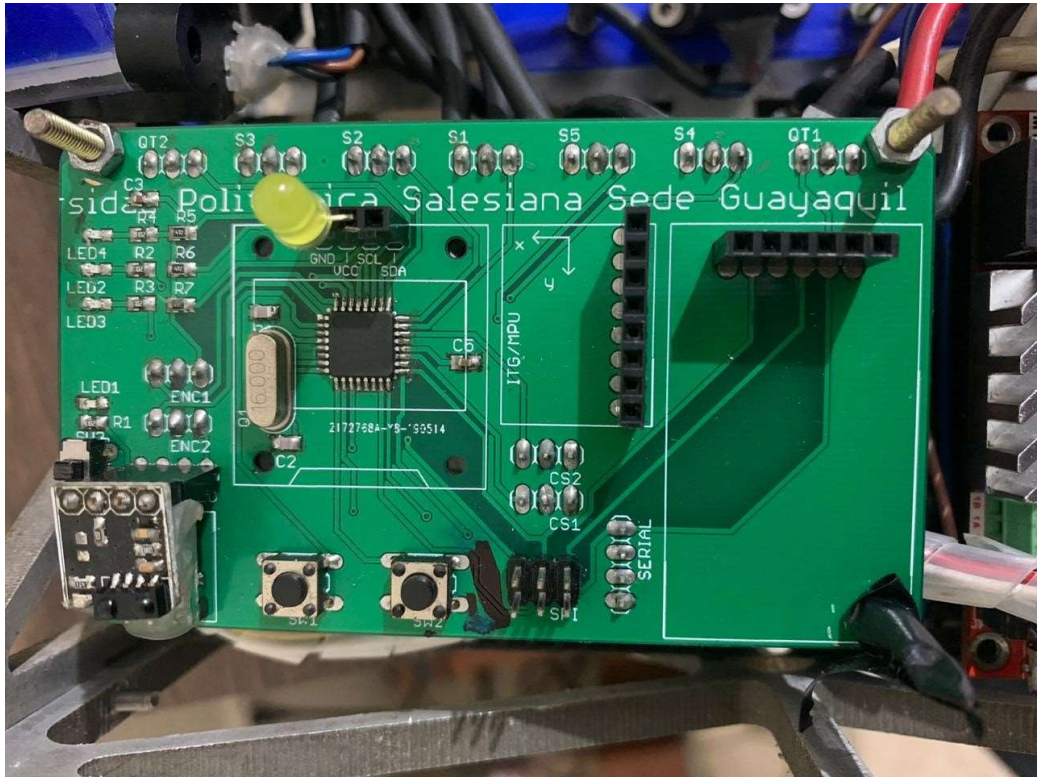


Figura 39. Tarjeta Controladora Implementada.

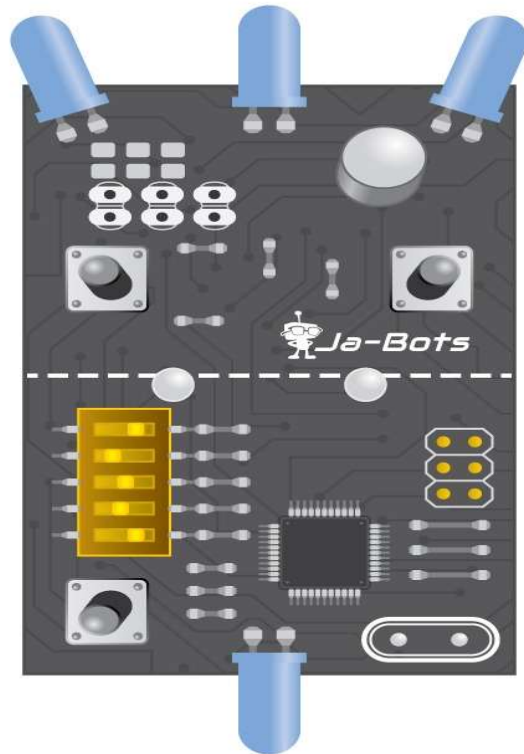


Figura 40. Modelado del Modulo de Arranque [10]

4. Resultados

En esta sección analizaremos los resultados obtenidos en las pruebas hechas durante el desarrollo de este proyecto técnico.

4.1 Analisis Estructural

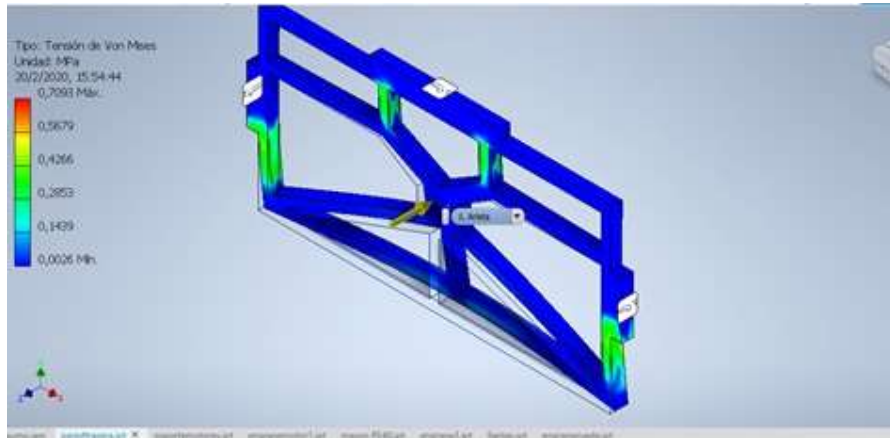


Figura 41. Análisis pared posterior

En esta grafica pudimos observar el analisis estructural donde se aplica una fuerza ortogonal en referencia a la grafica mostrada, donde observamos que la tension maxima que resiste la estructura es 0,7093 MPa(Megapascal) antes que la misma pierda su rigidez y tienda a quebrarse, recordemos que un Megapascal es equivalente a 10 bar y en este caso nuestro valor en Bares es 7.09 .Usando el programa APP Inventor.

4.1.2 Analisis Lateral

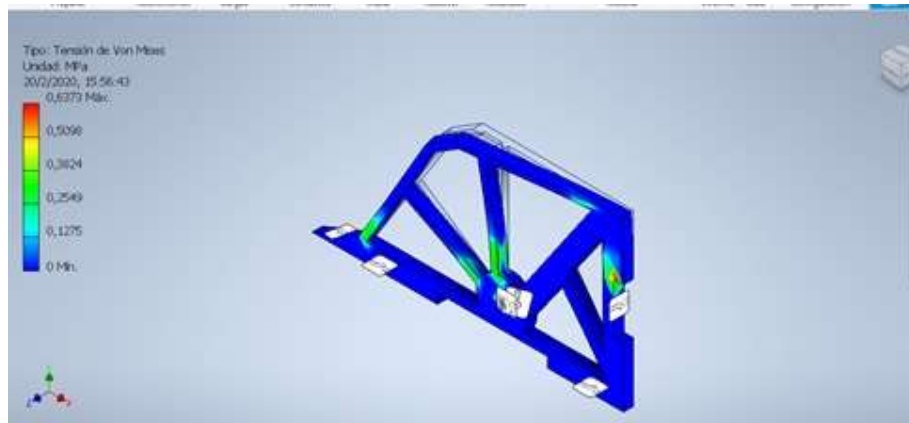


Figura 42. Análisis pared lateral

En esta grafica pudimos observar el analisis estructural donde se aplica una fuerza ortogonal en referencia a la grafica mostrada, donde observamos que la tension maxima que resiste la estructura es 0,6373 MPa(Megapascal) antes que la misma

pierda su rigidez y tienda a quebrarse, recordemos que un Megapascal es equivalente a 10 bar y en este caso nuestro valor en Bares es 6.37 .

4.1.3 Analisis Base del Sumo

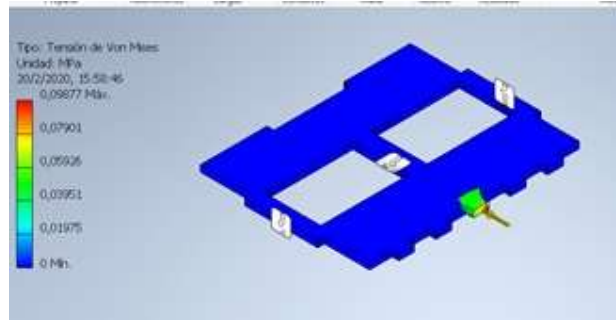


Figura 43. Análisis de la base del robot-sumo

En esta grafica pudimos observar el analisis estructural donde se aplica una fuerza ortogonal en referencia a la grafica mostrada, donde observamos que la tension maxima que resiste la estructura es 0,09877 MPa(Megapascal) antes que la misma pierda su rigidez y tienda a quebrarse, recordemos que un Megapascal es equivalente a 10 bar y en este caso nuestro valor en Bares es 0.98 .

4.1.4 Analisis de soporte maxon

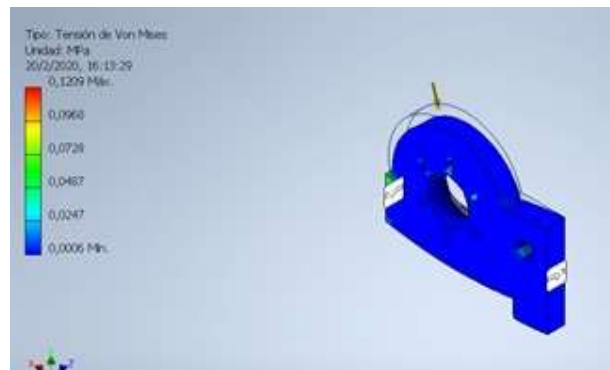


Figura 44. Análisis del soporte para el motor

En esta grafica pudimos observar el analisis estructural donde se aplica una fuerza ortogonal en referencia a la grafica mostrada, donde observamos que la tension maxima que resiste la estructura es 0,1209 MPa(Megapascal) antes que la misma pierda su rigidez y tienda a quebrarse, recordemos que un Megapascal es equivalente a 10 bar y en este caso nuestro valor en Bares es 1.20 .

4.2 Análisis de movimiento.

En esta sección analizaremos los resultados obtenidos en las pruebas hechas durante el desarrollo de este proyecto técnico.

4.2.1 Ejecución del programa

Como primer punto para analizar los resultados obtenidos realizamos un código generado en Matlab.



Figura 45. Menú de comandos

Paso siguiente generamos un GUI en Matlab mismos que posee varias opciones como por ejemplo "Abrir Video"

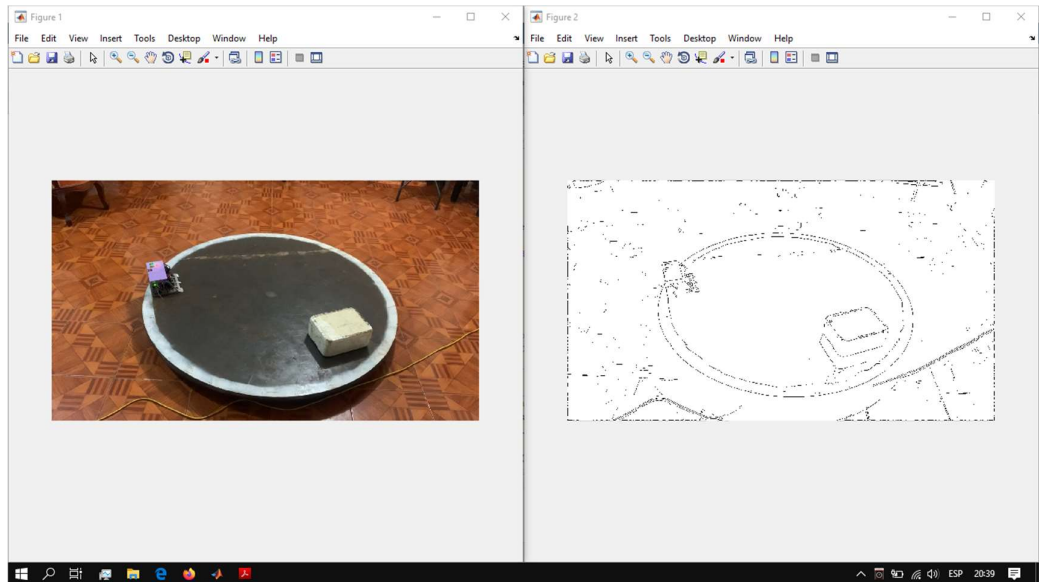


Figura 46. Ejecutado en el programa

Este nos permite seleccionar el video que deseamos analizar en función a la gráfica de la trayectoria, de la misma manera tenemos la opción “siguiente” mismo que nos ayudara adelantando nuestro video un cuadro hacia adelante esto es muy útil cuando deseamos capturar el color (RGB) que nuestro código va a graficar “Anterior”: esta opción también nos ayuda a movernos en los diferentes cuadros del video para obtener una muestra optima del tono RGB de la misma manera contamos con una opción que la hemos denominado “Limpiar” misma que nos ayuda eliminar objetos no deseados en el análisis del video.

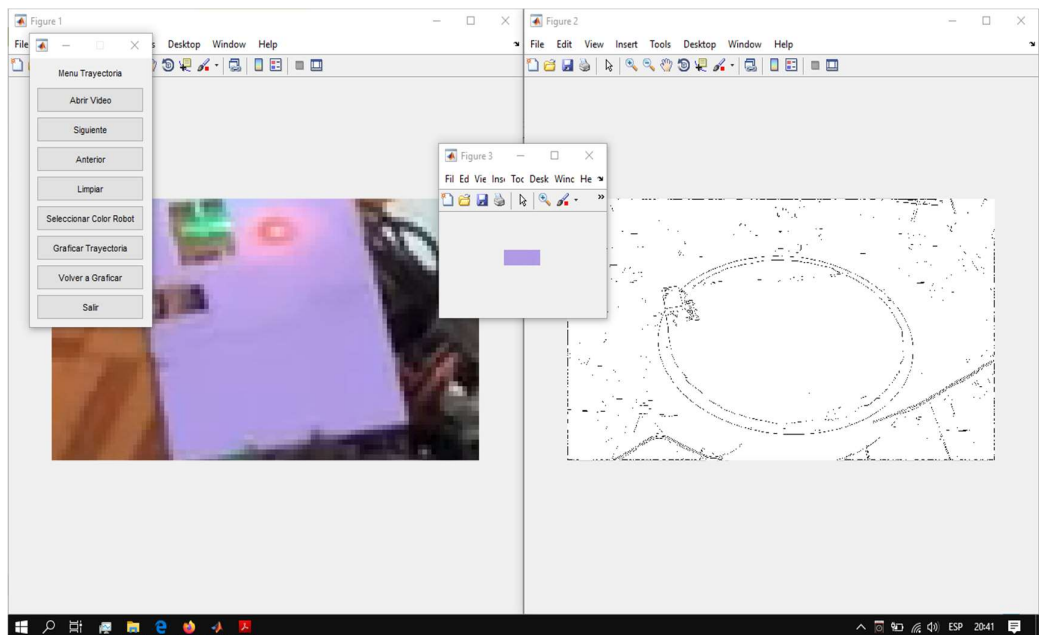


Figura 47. Selección del color para impresión de trayectoria

4.2.2 Seleccionar Color

“Seleccionar color”: es una función nos permite capturar el color en (RGB) para que nuestro código pueda graficar la trayectoria en base al desplazamiento de los pixeles capturados en el anterior paso.

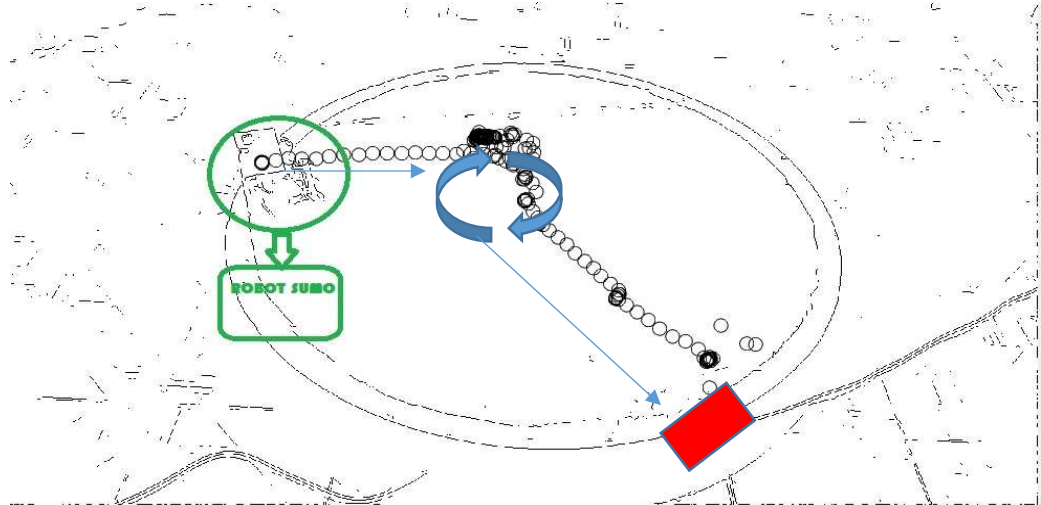


Figura 48. Gráfico de trayectoria por el robot-sumo

4.2.3 Grafico de Trayectoria

“Graficar Trayectoria” esta opción nos permite trazar la trayectoria que se desplazó el prototipo mediante la captura del color en RGB que realizamos el anterior paso, a continuación, veremos una sucesión de imágenes.

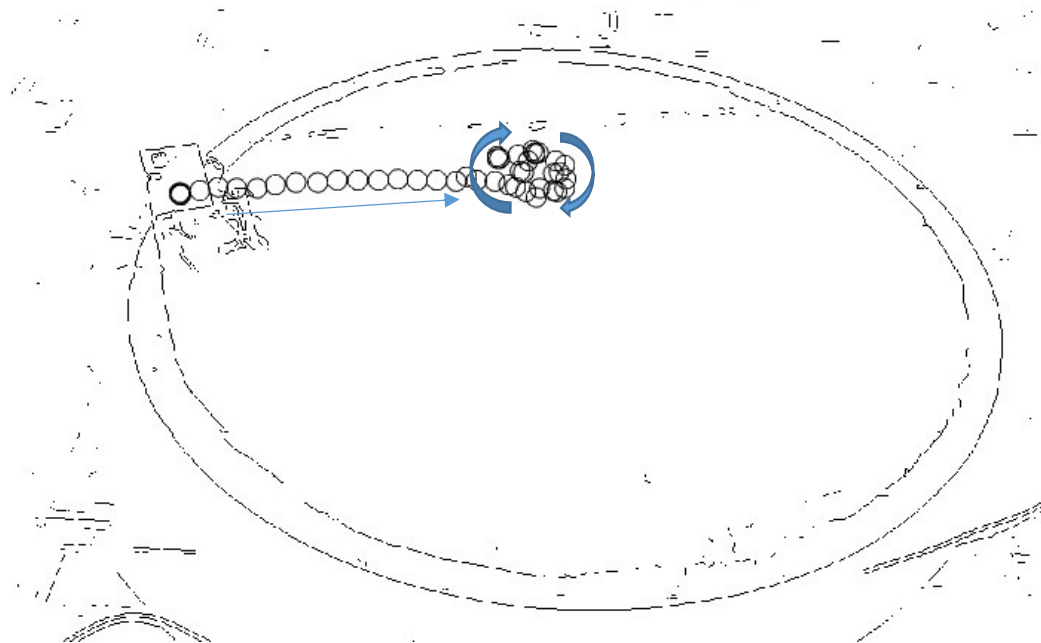


Figura 49. Salto y giro para mapear la pista en busca del contrincante

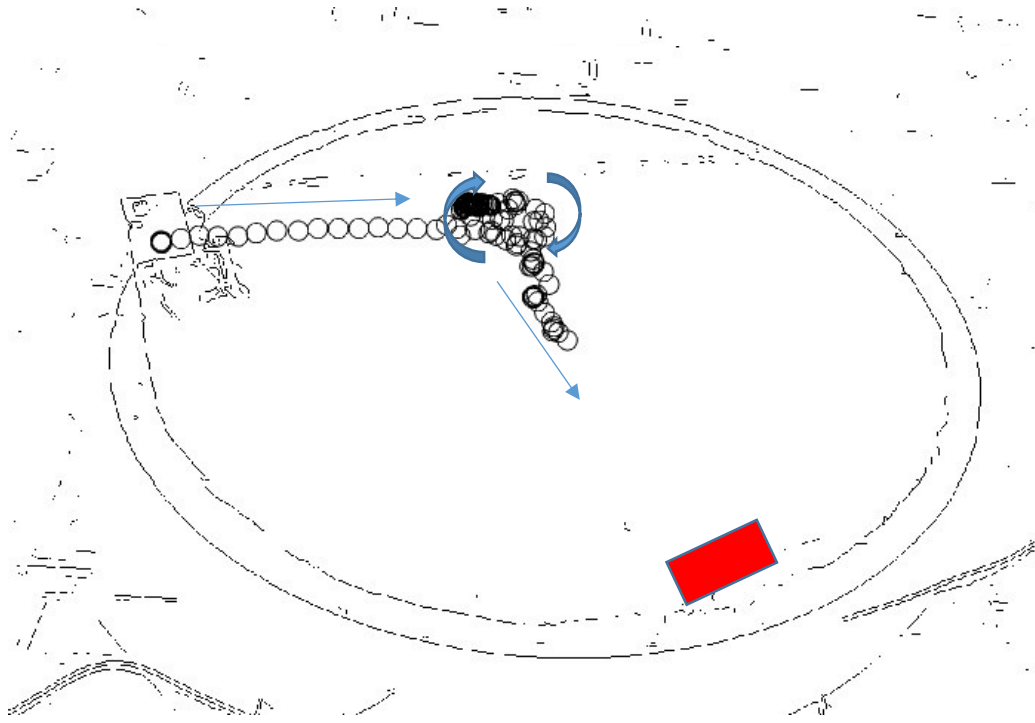


Figura 50. Detección del contrincante

4.2.4 Nueva Grafica

“Volver a Graficar” esta opción nos permite volver a graficar con una secuencia más rápida la trayectoria de nuestro prototipo, una vez terminado la rutina de la estrategia y posteriormente adquiridos los datos se genera un archivo .txt por cada tipo de dato adquirido como lo son VOLTAJE, CORRIENTE, DESPLAZAMIENTO EN LAS RUEDAS.

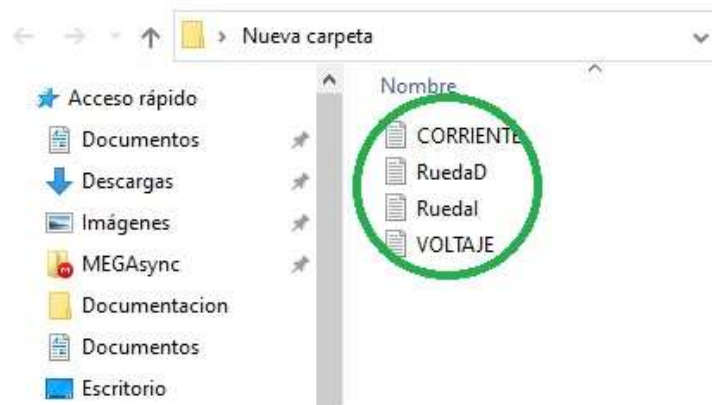


Figura 51. Datos obtenidos por los encoders del robot-sumo

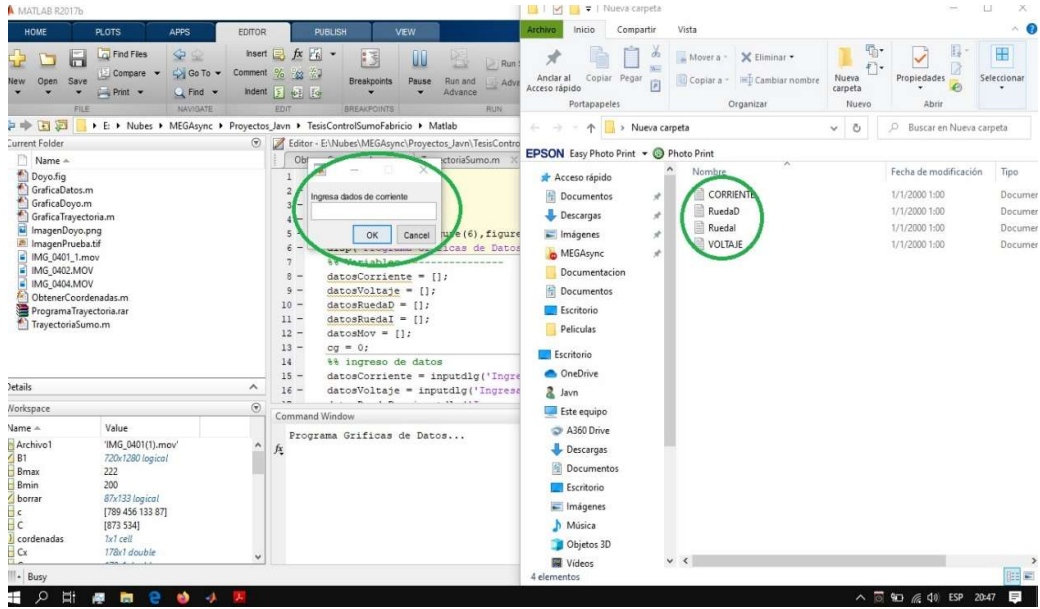


Figura 52. Datos usados para creación de graficas en el programa

4.2.5 Obtención de Datos

Una vez que obtuvimos los datos en el archivo “.txt” encontramos los valores ADC que van desde 0 hasta 1023

```

CORRIENTE: Bloc de notas
Archivo Edición Formato Ver Ayuda
< ===== Nueva Recopilación ===== > Vbase 2.51 ==> Modo 1
527
527
532
526
523
532
533
527
530
526
524
530
530
521
531
526
533
532
523
528
535
532
535

```

Figura 53. Datos obtenidos después de cada estrategia

4.3 Datos resultantes de las Estrategias

4.3.1 Estrategia número uno

Estrategia Numero 1

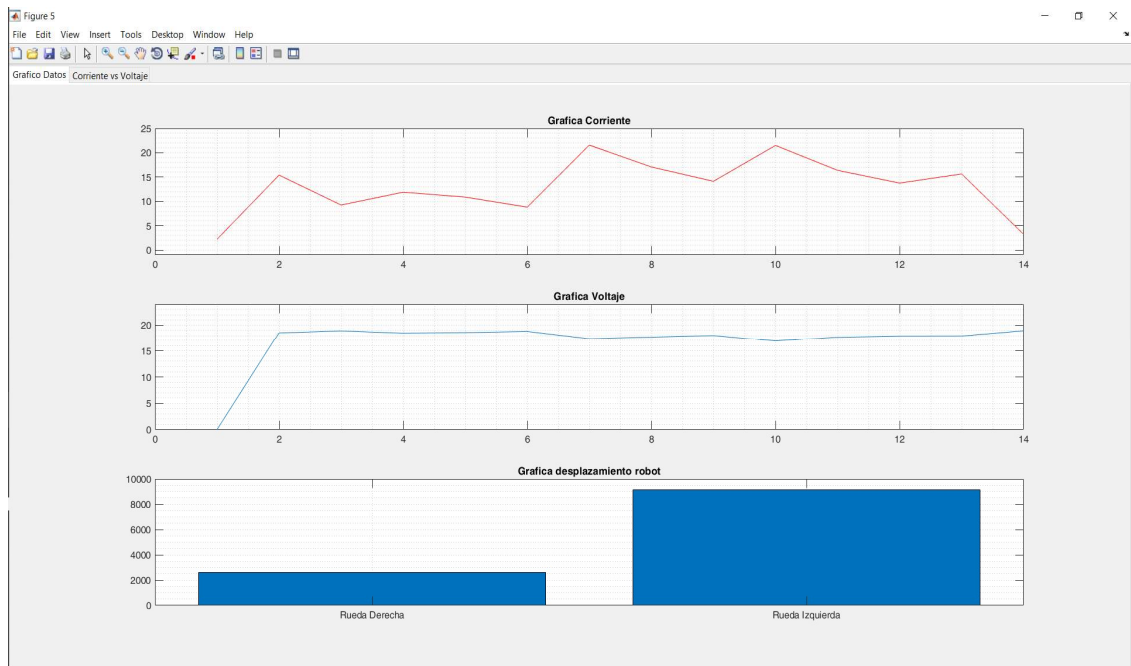
LECTURAS ADC ESTRATEGIA Nro. 1			
CORRIENTE	VOLTAJE	RUEDA IZQ	RUEDA DER
524	853	0	0
603	753	25	15
562	775	16	13
580	755	18	14
573	761	16	15
561	770	22	13
638	709	49	13
607	726	191	14
591	738	71	14
636	689	195	24
603	726	38	17
589	733	18	13
601	734	18	13
526	777	50	28

Tabla 1. Estrategia número uno

En este caso tomamos 4 muestras con la condición de que sean las lecturas ADC más altas de la tabla y obtuvimos un promedio de 619,5 y 793,5 en las tablas de corriente y voltaje respectivamente, esto nos demuestra que en cuestión de constantes eléctricas es la estrategia que más consume estos recursos.

Para más detalles de la estrategia empleada verificar en el capítulo 2 sección análisis de estrategias página número 12 del presente documento

Ahora mostraremos las gráficas de la estrategia numero 1



**Figura 54. Graficas de voltaje, corriente y desplazamiento de ambas ruedas
4.3.2 Estrategia numero dos**

Estrategia número 2

LECTURAS ADC ESTRATEGIA Nro. 2			
CORRIENTE	VOLTAJE	RUEDA IZQ	RUEDA DER
515	858	1	0
546	787	28	9
579	766	19	15
620	773	80	12
610	771	161	2
584	750	147	3
627	767	168	1
610	769	32	2
620	764	21	2
617	762	107	1
617	760	48	3
588	750	100	4
601	758	35	4
607	758	23	4
604	747	78	6
570	744	142	6
564	744	18	14
591	725	25	14
597	724	18	16
580	736	21	14
565	733	34	24
600	721	21	15
587	729	35	15
534	763	21	19

Tabla 2. Estrategia numero dos

En este caso tomamos 4 muestras con la condición que sean las lecturas ADC más altas de la tabla y obtuvimos un promedio de 609 y 765 en las tablas de corriente y voltaje respectivamente, esto nos demuestra que en cuestión de constantes eléctricas es la estrategia que mantiene un equilibrio entre consumo de recursos y tiempo de funcionamiento ya que de las tres estrategias planteadas es la que más tiempo operacional usa, pero notamos que su consumo de recursos no es tan elevado comparado con los otros dos.

Para más detalles de la estrategia empleada verificar en el capítulo 2 sección análisis de estrategias página número 13 del presente documento.

Ahora mostraremos las gráficas de la estrategia numero 2

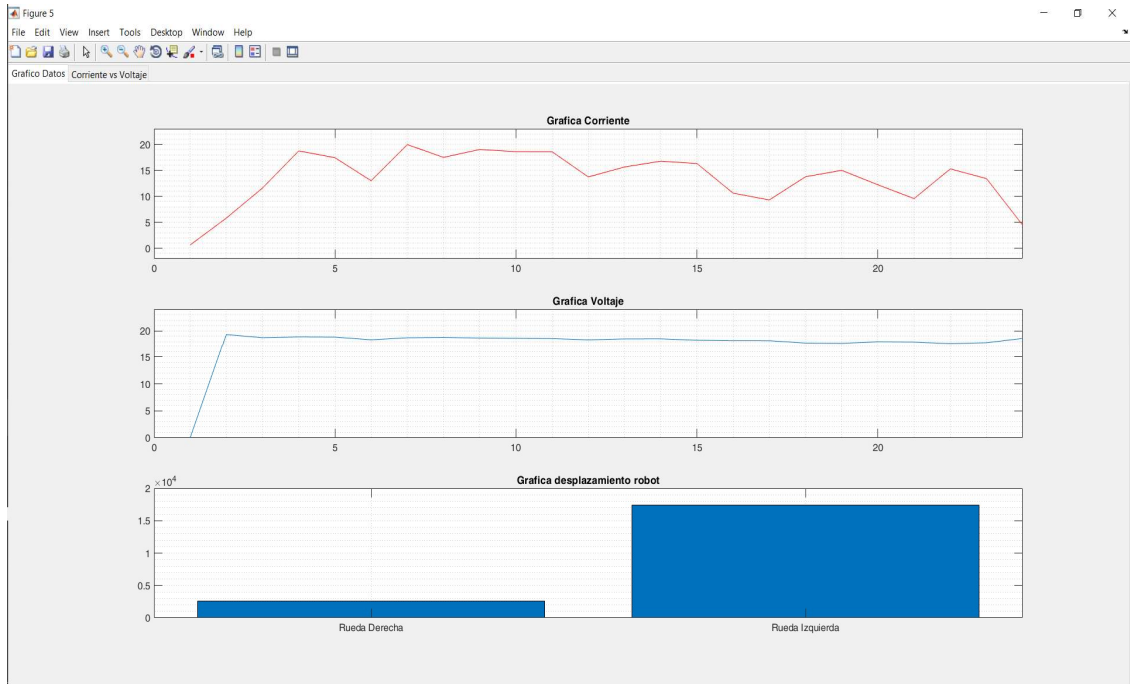


Figura 55. Graficas de voltaje, corriente y desplazamiento de ambas ruedas 4.3.3 Estrategia número tres

Estrategia número 3

LECTURAS ADC ESTRATEGIA Nro. 3			
CORRIENTE	VOLTAJE	RUEDA DER	RUEDA IZQ
529	844	0	0
595	746	6	112
605	733	10	54
650	733	8	108
621	694	12	140
625	687	12	21
605	695	9	122
608	691	7	7
632	676	8	47
582	701	9	15
573	702	10	9
596	695	11	10
580	699	7	16
580	714	10	5
579	686	2	42
619	728	0	91
570	720	17	21
556	716	12	37
552	699	0	14
546	709	0	23
559	706	6	9
570	712	16	12
594	701	15	15
569	716	16	10

529	745	19	20
-----	-----	----	----

Tabla 3. Estrategia número tres

En este caso tomamos 4 muestras con la condición de que sean las lecturas ADC más altas de la tabla y obtuvimos un promedio de 591,5 y 714 en las tablas de corriente y voltaje respectivamente, esto nos demuestra que en cuestión de constantes eléctricas es la estrategia que menos consume estos recursos y en conclusión nos resulta mucho mejor usar esta estrategia en combates de larga duración como por ejemplo torneos donde existan varias fases eliminatorias.

Para más detalles de la estrategia empleada verificar en el capítulo 2 sección análisis de estrategias página número 14 del presente documento.

Ahora mostraremos las gráficas de la estrategia numero 3

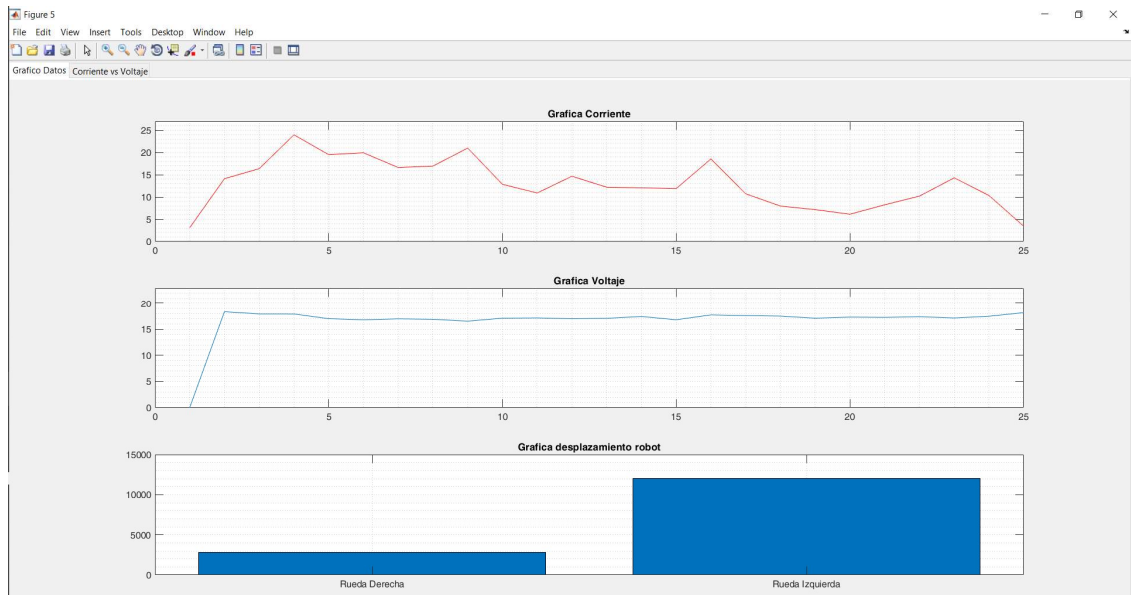


Figura 56. Graficas de voltaje, corriente y desplazamiento de ambas ruedas

Conclusiones

Luego del análisis de las estrategias se puede concluir que al momento de implementar fueron muy importantes los pasos previos como el dimensionamiento, ya que en este paso se da por sentado las características de los componentes de nuestro robot-sumo en el sentido largo, ancho y profundidad para poder ubicarlos al momento del ensamblaje e implementación.

En función a los parámetros eléctricos la mejor estrategia que puede utilizar es la numero 2 o también denominada "WOODPECKER", ya que otorga un equilibrio entre el tiempo de operación y los recursos consumidos. Pero cada combate es diferente y se debe analizar al contrincante desde su posición en la pista hasta la velocidad de desplazamiento de este para escoger una estrategia de acuerdo con el oponente.

De las estrategias pensadas y programadas para la implementación del robot-sumo se puede finalizar que cada estrategia al ser puesta en práctica da como resultado distintos parámetros de voltaje, trayectoria en pista y corriente utilizada tanto para desplazarse y para arrojar al contrincante fuera de la pista.

Recomendaciones

Luego de implementar varias estrategias se puede verificar que cada una tendrá mejor desempeño dependiendo de las características del contrincante respecto a la velocidad y diseño mecánico del mismo.

Se determina que para poder Graficar la Trayectoria del prototipo es necesario un color fuerte y diferente al entorno al momento de hacer el video para que realice un contraste y Matlab pueda identificar y graficar el tono RGB con mayor exactitud.

Se determina que cuando el prototipo termine un combate se recomienda limpiar tanto el Doyho como las llantas del robot, ya que las impurezas sobre la pista generan que el prototipo pierda tracción y tienda a resbalar o deslizarse.

Optimizar la cuchilla frontal con un ángulo aproximado de entre 25 a 30 grados, esto asegura que el prototipo estará lo más pegado a la superficie de la pista y así al momento de chocar con el oponente la misma ingresa por debajo de él y poder hacer que el adversario pierda contacto con la superficie de la pista y lograr sacarlo de la misma.

Distribuir de una manera correcta los imanes en la base del prototipo, la mayoría de ellos deben estar ubicados en la parte frontal del prototipo en donde está ubicada la cuchilla y unos cuantos al nivel de las ruedas.

Los Imanes que se recomiendan son los N52 de neodimio, mismo que deben traer una perforación en el centro para poder emperrar a la base del prototipo y así aumentar la adherencia de este.

Cada 2 enfrentamientos es recomendable cambiar las cuchillas finas, hay que recordar que el prototipo lleva 2 cuchillas una gruesa usada de base y una fina en la parte superficial, misma que debido a los enfrentamientos se golpea y pierde tanto su forma como su filo.

Siempre tener el receptor del robot en un lugar visible donde le pueda llegar la orden de detenerse ya que esta clase de prototipos son muy peligrosos cuando están fuera de control.

Al momento de operar y/o competir con el prototipo se debe usar tanto guante anticorte como gafas de protección.

Bibliográfica

- [1] «Google Maps,» [En línea]. Available: <https://www.google.com/maps/place/Universidad+Polit%C3%A9cnica+Salesiana+-+Guayaquil/@-2.220144,-79.8877792,18z/data=!4m5!3m4!1s0x902d6e4fced73235:0xb76f5008ec6c4345!8m2!3d-2.2201494!4d-79.8866849>. [Último acceso: 24 Marzo 2019].
- [2] Harmonic Drive SE, «Harmonic Drive,» [En línea]. Available: <https://harmonicdrive.de/es/glosario/robotica-movil>. [Último acceso: 27 Marzo 2020].
- [3] Jsumo, [En línea]. Available: <http://blog.jsumo.com/gzero-robot-sumo-meb-turkey-robot-competition-2016-1st-place-winner/>. [Último acceso: 04 Septiembre 2018].
- [4] Fuji Soft., «All Japan Robot Sumo Sournament,» [En línea]. Available: <https://www.fsi.co.jp/sumo/history/index.html>. [Último acceso: 24 Marzo 2019].
- [5] Panasonic, «digi-key,» [En línea]. Available: https://media.digikey.com/pdf/Data%20Sheets/Panasonic%20Sensors%20PDFs/M_A_Motion_Sensor.pdf. [Último acceso: 25 Marzo 2019].
- [6] Pololu, «pololu robotics & electronics,» [En línea]. Available: <https://www.pololu.com/product/958#lightbox-picture0J624;main-pictures>. [Último acceso: 25 Marzo 2019].
- [7] Pololu, «Pololu Robotics & Electronics,» [En línea]. Available: <https://www.pololu.com/product/2398#lightbox-picture0J6617;main-pictures>. [Último acceso: 25 Marzo 2019].
- [8] Maxon Motor, «Maxon Motor Online Shop,» [En línea]. Available: <https://www.maxonmotor.com/maxon/view/product/148867>. [Último acceso: 28 Marzo 2019].
- [9] CDMX ELECTRONICA, «CDMX ELECTRONICA,» 29 AGOSTO 2019. [En línea]. Available: <https://www.cdmxelectronica.com/producto/microcontrolador-atmega328p-au-smd/>.
- [10] POLOLU, «POLOLU STORE,» [En línea]. Available: <https://www.pololu.com/product/2199>. [Último acceso: 31 MARZO 2020].
- [11] ArcaElectronica, «ArcaElectronica,» [En línea]. Available: <http://arcaelectronica.blogspot.com/2017/12/sensor-de-voltaje-fz0430.html>. [Último acceso: 31 MARZO 2020].
- [12] Ja-bots, [En línea]. Available: <https://ja-bots.com/producto/control-profesional/>.
- [13] Arduino, «Arduino,» 17 Febrero 2020. [En línea]. Available: <https://arduino.cl/que-es-arduino/>.
- [14] Arduino, [En línea]. Available: <https://arduino.cl/que-es-arduino/>.

ANEXOS

Anexo 1: Código Arduino

```
////////////////////////////////////
Control Principal
#include "Librerias.h"
/*****
*****/
void setup() {
  configurarPines(); // configurar pines
  iniciarRoboClaw(); // iniciar comunicación con RoboClaw
  iniciarSD(); // iniciar comunicación con SD
  iniciarTimer(10);
  iniciarEncoder();
  modo = 1;
  contarP = 0;
  seleccionarModo();
}
void loop() {
// ---- leer sensores -----
  leerSensores();
// ---- inicio competicion -----
  if(digitalRead(btnStart))
  {
    if(contarP == 0){contarP = 1; tiempoC = 0;}
//   if(!digitalRead(sw1)){atacarRobot();}
//   else if(!digitalRead(sw2)){retroRobot(100);}
//   else{detenerRobot();}
// modo de competencia -----
    switch (modo)
    {
      case 1:
        estrategia1();
        break;
      case 2:
        estrategia2();
        break;
      case 3:
        estrategia3();
        break;
      case 4:
        estrategia4();
        break;
      default:
        break;
    }
  }
// *****/
```

```

    if(tiempoLed >= 100)
    { grabarDatos();}
} // fin if competicion
// fin de competicion -----
else
{
if(!digitalRead(sw1))
{
retroRobot(25);
}
else if (!digitalRead(sw2))
{
adelanteRobot(25);
}
else{
detenerRobot();
}
digitalWrite(ledY,0);
}
} // ----- fin de while loop
////////////////////////////////////
"Librerias"
#include <MsTimer2.h>
#include "RoboClaw.h"
#include <SoftwareSerial.h>
#include <SPI.h>
#include "SdFat.h"
SdFat SD;
// ----- Funciones -----
#include "controlSensores.h"
#include "lectorEncoder.h"
#include "controlRoboclave.h"
#include "controlSd.h"
#include "timerDesborde.h"
#include "estrategia_1.h"
#include "estrategia_2.h"
#include "estrategia_3.h"
#include "estrategia_4.h"
////////////////////////////////////
"Control Roboclave"
SoftwareSerial serial(0,1);
RoboClaw roboclaw(&serial,10000);
#define address 0x80 // variable direccion de roboclaw
void iniciarRoboclave(void)
{
roboclaw.begin(38400); //Open roboclaw serial ports
roboclaw.ForwardM1(address,0); //start Motor1 forward at Stop 0 - 127

```

```

    roboclaw.ForwardM2(address,0); //start Motor2 forward at Stop 0 - 127
}
// M1 motor derecho
// M2 motor izquierdo
void atacarRobot(void)
{
    roboclaw.ForwardM1(address,127); //start Motor1 forward at Stop 0 - 127
    roboclaw.ForwardM2(address,127); //start Motor2 forward at Stop 0 - 127
    dir = 1;
}
void detenerRobot(void)
{
    roboclaw.ForwardBackwardM1(address,64); //64 stop
    roboclaw.ForwardBackwardM2(address,64); //64 stop
    dir = 0;
}
void atacarBloqueo()
{
    while((s1)&&(digitalRead(btnStart))){atacarRobot();}
    detenerRobot();
}
void adelanteRobot(int v)
{
    v = map(v,1,100,5,126);
    roboclaw.ForwardM1(address,v); //start Motor1 forward at Stop 0 - 127
    roboclaw.ForwardM2(address,v); //start Motor2 forward at Stop 0 - 127
    dir = 1;
}
void retroRobot(int v)
{
    v = map(v,1,100,5,63);
    roboclaw.ForwardBackwardM1(address,64-v); //64 stop
    roboclaw.ForwardBackwardM2(address,64-v); //64 stop
    dir = 2;
}
void girarDerecha(int v)
{
    v = map(v,1,100,5,63);
    roboclaw.ForwardBackwardM1(address,64+v); //64 stop
    roboclaw.ForwardBackwardM2(address,64-v); //64 stop
    dir = 4;
}
void girarIzquierda(int v)
{
    v = map(v,1,100,5,63);
    roboclaw.ForwardBackwardM1(address,64-v); //64 stop
    roboclaw.ForwardBackwardM2(address,64+v); //64 stop
}

```

```

    dir = 3;
}
void girarI90(void)
{
    tiempoG = 0;
    while(tiempoG < 180)
    {
        girarIzquierda(50);
        if(s1)
        {
            girarDerecha(60);delay(25);
            //atacarBloqueo();
            tiempoG = 250;
        }
    }
    detenerRobot();
}
void girarD90(void)
{
    tiempoG = 0;
    while(tiempoG < 180)
    {
        girarDerecha(50);
        if(s1)
        {
            girarIzquierda(60);delay(25);
            //atacarBloqueo();
            tiempoG = 250;
        }
    }
    detenerRobot();
}
void girarI30(void)
{
    tiempoG = 0;
    while(tiempoG < 80)
    {
        girarIzquierda(50);
        if(s1)
        {
            girarDerecha(60);delay(25);
            //atacarBloqueo();
            tiempoG = 120;
        }
    }
    detenerRobot();
}

```



```

void iniciarSD(void)
{
  SD.begin(SSpin);
}
// ----- GRABAR DATOS -----
void nuevosDatos(void)
{
  Voltaje = SD.open("VOLTAJE.txt", FILE_WRITE);
  if (Voltaje)
  {
    Voltaje.print("< ===== Nueva Recopilacion ===== > Modo ");
    Voltaje.println(modos);
    Voltaje.close();
  }
  Corriente = SD.open("CORRIENTE.txt", FILE_WRITE);
  if (Corriente)
  {
    Corriente.print("< ===== Nueva Recopilacion ===== > Vbase ");
    Corriente.print(corriente_V);
    Corriente.print(" ==> Modo ");
    Corriente.println(modos);
    Corriente.close();
  }
  Movimiento = SD.open("MOVIMIENTO.txt", FILE_WRITE);
  if (Movimiento)
  {
    Movimiento.println("-- detenido: 0 -- adelante: 1 -- retroceder: 2 -- derecha: 3 -
- izquierda: 4 ");
    Movimiento.print("< ===== Nueva Recopilacion ===== > Modo ");
    Movimiento.println(modos);
    Movimiento.println("Direccion");
    Movimiento.close();
  } // */
  RuedaD = SD.open("RuedaD.txt", FILE_WRITE);
  if (RuedaD)
  {
    RuedaD.println("-- Rotacion Rueda Derecha ");
    RuedaD.print("< ===== Nueva Recopilacion ===== > Modo ");
    RuedaD.println(modos);
    RuedaD.println("Revoluciones ");
    RuedaD.close();
  } // */
  Ruedal = SD.open("Ruedal.txt", FILE_WRITE);
  if (Ruedal)
  {
    Ruedal.println("-- Rotacion Rueda Izquierda ");
    Ruedal.print("< ===== Nueva Recopilacion ===== > Modo ");

```



```

Control de sensores
// distribucion de pines
int ledW = A4;
int ledY = A5;
int sensor1 = A0;
int sensor2 = A1;
int sensor3 = 8;
int sensor4 = 9;
int sensor5 = 10;
int qt1 = A2;
int qt2 = A3;
int encl = 3;
int encD = 2;
int sw1 = 4;
int sw2 = 5;
int btnStart = 6;
int cs1 = A6;
int cs2 = A7;
// ----- Variables -----
//***** sensores *****
int s1 = 0;
int s2 = 0;
int s3 = 0;
int s4 = 0;
int s5 = 0;
int modo = 0;
int controlF = 0;
int corriente_mA = 0;
float corriente_V = 0;
int nivelV = 0;
int tiempoLed = 0;
int tiempoTitilar = 0;
int tiempoC = 0;
int contarP = 0;
int tiempoG = 0;
int led11 = 0;
int led12 = 0;
int titilar = 0;
int encoderi = 0;
int encoderd = 0;
float vueltaL = 0;
float vueltaD = 0;
int dir = 0;
// ----- leer voltaje base sensor de corriente-----
void leerVoltajeB(void)
{
    int cont = 0;

```



```

else if((tiempoC > 200)&&(tiempoC <= 700))
{
    girarIzquierda(40);
    buscar1();
}
// ----- buscar, girar derecha y atacar -----
else if((tiempoC > 700)&&(tiempoC <= 1700))
{
    girarDerecha(40);
    buscar1();
}
else if((tiempoC > 1700)&&(tiempoC <= 2200))
{
    girarIzquierda(40);
    buscar1();
}
// ----- sensor y esperar -----
else if((tiempoC > 2200)&&(tiempoC <= 3200))
{
    buscar();
}
// ++++++ salto 2
+++++
// ----- avance recto -----
if((tiempoC >= 3200)&&(tiempoC <= 3400)){ atacarRobot(); }
// ----- buscar, girar izquierda y atacar -----
else if((tiempoC > 3400)&&(tiempoC <= 3900))
{
    girarIzquierda(40);
    buscar1();
}
// ----- buscar, girar derecha y atacar -----
else if((tiempoC > 3900)&&(tiempoC <= 4900))
{
    girarDerecha(40);
    buscar1();
}
else if((tiempoC > 4900)&&(tiempoC <= 5400))
{
    girarIzquierda(40);
    buscar1();
}
// ----- sensor y esperar -----
else if((tiempoC > 5400)&&(tiempoC <= 6400))
{
    buscar();
}

```

```

// ++++++salto 3 ++++++
// ----- avance recto -----
if((tiempoC >= 6400)&&(tiempoC <= 6600)){ atacarRobot(); }
// ----- buscar, girar izquierda y atacar -----
else if((tiempoC > 6600)&&(tiempoC <= 7100))
{
    girarIzquierda(40);
    buscar1();
}
// ----- buscar, girar derecha y atacar -----
else if((tiempoC > 7100)&&(tiempoC <= 8100))
{
    girarDerecha(40);
    buscar1();
}
else if((tiempoC > 8100)&&(tiempoC <= 8600))
{
    girarIzquierda(40);
    buscar1();
}
// ----- sensor y esperar -----
else if((tiempoC > 8600)&&(tiempoC <= 9600))
{
    buscar();
}
// ----- reiniciar -----
else if(tiempoC > 9700){tiempoC = 6700;}
// */
}
////////////////////////////////////////////////////////////////////////////////////////////////////
Estrategia 3
void estrategia3(void)
{
    if((s1==1)||(s2==1)|(s5==1))
    {
        buscar1();
    }
    else
    {
        girarIzquierda(80);
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////
estrategia 4
void estrategia4(void)
{
    // ----- avance recto -----

```



```

attachInterrupt(digitalPinToInterrupt(encD), interruptCount1, HIGH);
attachInterrupt(digitalPinToInterrupt(encI), interruptCount2, HIGH);
}
//////////////////////////////////////
Desborde Timer
//-----
void seleccionarModo(void)
{
  while(digitalRead(sw1))
  {
    if(!digitalRead(sw2))
    {
      delay(100);
      while(!digitalRead(sw2)){
        modo++;
        if(modo >4){modo =1;}
        titilar = modo * 2;
      } // fin if
    } // fin while
    nuevosDatos();
    digitalWrite(ledY,1);
    while(!digitalRead(btnStart)){
  }
}
// -----
void desborde(void)
{
  // crear variable contadora para combate "tiempoC"
  tiempoC = tiempoC + 10;
  tiempoG = tiempoG + 10;
  // *****
  tiempoTitilar = tiempoTitilar + 10;
  if(tiempoTitilar >= 200)
  {
    tiempoTitilar = 0;
    if(titilar >= 2)
    {
      titilar--;
      led12 = !led12;
      digitalWrite(ledY,led12);
    }
    else if(titilar >=1 )
    {
      titilar--;
      led12 = 0;
      digitalWrite(ledY,led12);
    }
  }
} // if > 200

```

```

// *****
tiempoLed = tiempoLed + 10;
if(tiempoLed >= 500)
{
    led11 = !led11;
    digitalWrite(ledW,led11);
    tiempoLed = 0;
}
}
//-----
void iniciarTimer(int x)
{
    MsTimer2::set(x, desborde); // periodo [ms]
    MsTimer2::start();
}

```

Anexo 2: Código Matlab

```

////////////////////////////////////
////////////////////////////////////
GUI Opciones
clc
clear all
warning off;
close.figure(1),figure(2),figure(3)
disp('Programa Trayectoria iniciado...')
opcion = 0;
sc = 0;
scB1 = 0;
vg = 0;
while(opcion ~= 8)
opcion = menu('Menu Trayectoria','Abrir
Video','Siguiete','Anterior','Limpiar',...
'Seleccionar Color Robot','Graficar Trayectoria','Volver a
Graficar','Salir');
switch opcion
%% abrir video
case 1
% Obtener ruta de la Video
[Archivo1,Ruta1] = uigetfile({'*.mp4';*.*.wmv'},'Seleccione Video');
dreal = strcat(Ruta1,Archivo1);
video = VideoReader(dreal);
if(~isempty(video))
N = video.NumberOfFrames;
ni = 1;
imagen = read(video,ni);
H1 = fspecial('average');
imagen1 = imfilter(imagen,H1);

```

```

imagengray = rgb2gray(imagen1);
B1 = not(edge(imagengray,'Prewitt'));
figure(1)
imshow(imagen)
grid off
figure(2)
imshow(B1)
grid off
sc = 0;
scB1 = 0;
disp('Video abierto...')
end % if verificar video
%% siguiente imagen
case 2
if(~isempty(video))
ni = ni +1;
if(ni > N)
ni = 1;
end % end if(ni )
imagen = read(video,ni);
H1 = fspecial('average');
imagen1 = imfilter(imagen,H1);
imagengray = rgb2gray(imagen1);
B1 = not(edge(imagengray,'Prewitt'));
figure(1)
imshow(imagen)
grid off
figure(2)
imshow(B1)
grid off
end % if verificar video
%% anterior imagen
case 3
if(~isempty(video))
ni = ni -1;
if(ni < 1)
ni = N;
end % end if(ni )
imagen = read(video,ni);
H1 = fspecial('average');
imagen1 = imfilter(imagen,H1);
imagengray = rgb2gray(imagen1);
B1 = not(edge(imagengray,'Prewitt'));
figure(1)
imshow(imagen)
grid off
figure(2)

```

```

imshow(B1)
grid off
end % if verificar video
%% Limpiar doyo
case 4
if(~isempty(video))
%[J, rect] = imcrop(imagen);
figure(2)
imshow(B1)
borrar = imcrop(B1);
cordenadas = inputdlg('Ingrese coordenadas');
c= round(str2num(cordenadas{1}));
B1(c(2):c(2)+c(4),c(1):c(1)+c(3)) = 1;
figure(2)
imshow(B1)
scB1 = 1;
disp('Limpieza realizada...')
end % if verificar video
%% seleccionar color robot
case 5
if(~isempty(video))
%[J, rect] = imcrop(imagen);
figure(1)
imshow(imagen)
selecColor = imcrop(imagen);
figure(3)
imshow(selecColor)
grid off
scR = double(selecColor(:,1));
scG = double(selecColor(:,2));
scB = double(selecColor(:,3));
Rmin = min(min(scR));
Rmax = max(max(scR));
Gmin = min(min(scG));
Gmax = max(max(scG));
Bmin = min(min(scB));
Bmax = max(max(scB));
sc = 1;
disp('Color del robot seleccionado...')
end % if verificar video
%% graficar trayectoria
case 6
if(sc ==1)
if(scB1 ==1)
disp('Iniciado proceso de Graficar Trayectoria...')
Cx = [];
Cy = [];

```

```

% buscar coordenadas Robot
disp('Buscando coordenadas de trayectoria...')
for i = 1:N
    imagen = read(video,i);
    C = ObtenerCoordenadas(imagen,Rmin,Rmax,Gmin,Gmax,Bmin,Bmax);
    if(~isempty(C))
        C = round(C);
        Cx = [Cx; C(:,1)];
        Cy = [Cy; C(:,2)];
    end% end if(~isempty(C))
end % end for i
H = video.Height;
W = video.Width;
X = Cx;
%Y = H - Cy;
Y =Cy;
ti = video.Duration;
Nc = length(X);
delay = ti/Nc;
figure(3)
imshow(B1)
hold on
for i = 1:Nc;
    figure(3)
    plot(X(i),Y(i),'ko','MarkerSize',10);
    hold on
    axis([0 W 0 H])
    grid off
    axis off
    pause(delay);
end%end for grafica
plot(X,Y,'ko','MarkerSize',10);
hold off
vg = 1;
end
end%end if ~isempty(video)
disp('Proceso de grafica de trayectoria concluido...')
%% volver a graficar
case 7
if(vg ==1)
    figure(3)
    imshow(B1)
    hold on
    for i = 1:Nc;
        figure(3)
        plot(X(i),Y(i),'ko','MarkerSize',10);
        hold on

```

```

axis([0 W 0 H])
grid off
axis off
pause(delay/2);
end%end for grafica
plot(X,Y,'ko','MarkerSize',10);
hold off
end%end if ~isempty(video)
%% salir
case 8
%close all
otherwise
end% end switch
end
%% end while
disp('Programa Tractoria finalizado')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Obtener coordenadas del video
function[centroides] = ObtenerCoordenadas(imagen,Tr1,Tr2,Tg1,Tg2,Tb1,Tb2)
% clc
% clear all
% imagen = imread('ImagenPrueba.tif');
imR = double(imagen(:,:,1));
imG = double(imagen(:,:,2));
imB = double(imagen(:,:,3));
[N, M,Z] = size(imagen);
Y = ones(N,M);
% 20 - 145 - 229
% Tr1 = 180;
% Tr2 = 220;
% Tg1 = 220;
% Tg2 = 255;
% Tb1 = 85;
% Tb2 = 110;
for i= 1:N;
for j= 1:M;
if((imR(i,j) > Tr1 && imR(i,j) < Tr2)...
&&(imG(i,j) > Tg1 && imG(i,j) < Tg2)...
&&(imB(i,j) > Tb1 && imB(i,j) < Tb2))
Y(i,j) = 1;
else
Y(i,j) = 0;
end%end if
end%end for j
end%end for i
se = strel('disk',10);

```

```

imagenY1 = imclose(Y,se);
imagen_bw = bwareaopen(imagenY1,30);
s = regionprops(imagen_bw,'centroid');
centroides = cat(1,s.Centroid);
%imshow(imagen_bw)
End
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Graficar Datos
% inicializacion
clc
clear all
warning off;
close(gcf(5),gcf(6),gcf(7))
disp('Programa Grificas de Datos...')
%% Variables -----
datosCorriente = [];
datosVoltaje = [];
datosRuedaD = [];
datosRuedaI = [];
datosMov = [];
cg = 0;
%% ingreso de datos
datosCorriente = inputdlg('Ingresa datos de corriente');
datosVoltaje = inputdlg('Ingresa datos de voltaje');
datosRuedaD = inputdlg('Ingresa datos de rueda derecho');
datosRuedaI = inputdlg('Ingresa datos de rueda izquierdo');
% datosMov = inputdlg('Ingresa datos de rotacion');
%% Constantes
vbase = 2.52;
vcc = 4.96;
alphaC = 0.95;
%% procesamiento de informacion
%% isempty(datosCorriente)-----
if(~isempty(datosCorriente))
datosCorriente = round(str2num(datosCorriente{1}));
dn = length(datosCorriente);
if(dn > 0)
cg(2) = 1;
sc(1) = datosCorriente(1,1);
sc(1) = (sc(1)*5)/1023;
corriente(1) = ((sc(1)/(vcc/5))- vbase)/0.028;
for i=2:dn
sc(i) = datosCorriente(1,i)*alphaC + ((1-alphaC)*datosCorriente(1,i-1));
sc(i) = (sc(i)*5)/1023;
corriente(i) = ((sc(i)/(vcc/5))- vbase)/0.028;
end %end for
end % if dn

```



```

end% if corriente
%% -----
% isempty(datosVoltaje)
if(~isempty(datosVoltaje))
datosVoltaje = round(str2num(datosVoltaje{1}));
dn = length(datosVoltaje);
if (dn > 0)
cg(3) = 1;
sv(1) = datosVoltaje(1,1);
sv(1) = (sv(1)*5)/1023;
for i=2:dn
sv(i) = datosVoltaje(1,i)*alphaC + ((1-alphaC)*datosVoltaje(1,i-1));
voltaje(i) = (sv(i)*25)/1023;
end %end for
end % end if dn
end% if voltaje
%% Datos rueda derecha -----
if(~isempty(datosRuedaD))
datosRuedaD = round(str2num(datosRuedaD{1}));
dn = length(datosRuedaD);
if (dn > 0)
cg(4) = 1;
rRD = datosRuedaD*12.566;
distanciaRD = sum(rRD);
end % end if dn
end% if rueda derecho
%% Datos rueda Izquierda -----
if(~isempty(datosRuedal))
datosRuedal = round(str2num(datosRuedal{1}));
dn = length(datosRuedal);
if (dn > 0)
cg(5) = 1;
rRI = datosRuedal*12.566;
distanciaRI = sum(rRI);
end % end if dn
end% if motor Izquierdo
%% sentido de movimientos
if(~isempty(datosMov))
datosMov = round(str2num(datosMov{1}));
dn = length(datosMov);
end% if Motor movimiento
%% graficas de datos-----
if (length(cg) > 1)
np = sum(cg);
figure(5);
tab1 = uitab('Title','Grafico Datos');
ax1 = axes(tab1); % selecciona la pestaña 1 para

```

```

graficar
if(np > 3)
np = 3;
end
ax1 = axes(tab1);
subplot(np,1,1)
if (cg(2) == 1)
plot(corriente,'color','r');title('Grafica Corriente')
axis([0 length(corriente) round(min(min(corriente)))-3 3+round(max(max
(corriente)))]])
grid minor
% ax2 = axes(tab2);
% plot(corriente,'color','r');
% hold on
% grid minor
ax1 = axes(tab1);
subplot(np,1,2)
end% if np
if (cg(3) == 1)
plot(10*voltaje);title('Grafica Voltaje')
axis([0 length(voltaje) 140 10+10*round(max(max(voltaje)))]])
grid minor
tab2 = uitab('Title','Corriente vs Voltaje');
ax2 = axes(tab2);
plot(corriente,'color','r');
hold on
plot(voltaje,'color','g');
hold off
legend('Corriente','Voltaje')
axis([0 length(voltaje) 0 50])
grid minor
end% if np
ruedas = 0;
c = '';
if (cg(4) == 1)
ruedas = [ruedas ; distanciaRD];
c = categorical({' ' 'Rueda Derecha'});
end
if (cg(5) == 1)
ruedas = [ruedas ; distanciaRI];
c = categorical({' ' 'Rueda Derecha' 'Rueda Izquierda'});
end
if (np >= 3)
ax1 = axes(tab1);
subplot(np,1,3)
bar(c,ruedas);title('Grafica desplazamiento robot')
grid minor

```

end
end% if length
Anexos 3: Foto galería



Realizando prueba de movilidad



Recibiendo reconocimiento por cumplir con la presidencia del RAS de la rama IEEE de nuestra Universidad



La versión anterior del prototipo Striker participando entre los 8 mejores de Robochallenge México



Evento de clausura del robochallenge México desarrollado en la Universidad aeronáutica de Querétaro (UNAQ)