



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA**

CARRERA INGENIERÍA AUTOMOTRIZ

**“DESARROLLO DE UNA PLATAFORMA MÓVIL TIPO
ACKERMANN PARA LA VALIDACIÓN DE ALGORITMOS DE
NAVEGACIÓN AUTÓNOMA DENTRO DEL CAMPUS ESPOCH”**

Trabajo De Integración Curricular
Tipo: Proyecto de Investigación

Presentado para optar el grado académico de:
INGENIERO AUTOMOTRIZ

AUTORES:

ANDRÉS FABRICIO BARAHONA ANDINO

KLEBER RAMIRO RÍOS SIGCHO

Riobamba – Ecuador

2021



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA**

CARRERA INGENIERÍA AUTOMOTRIZ

**“DESARROLLO DE UNA PLATAFORMA MÓVIL TIPO
ACKERMANN PARA LA VALIDACIÓN DE ALGORITMOS DE
NAVEGACIÓN AUTÓNOMA DENTRO DEL CAMPUS ESPOCH”**

Trabajo De Integración Curricular
Tipo: Proyecto de Investigación

Presentado para optar el grado académico de:

INGENIERO AUTOMOTRIZ

AUTORES: ANDRÉS FABRICIO BARAHONA ANDINO

KLEBER RAMIRO RÍOS SIGCHO

DIRECTOR: Ing. JAVIER JOSÉ GAVILANES CARRIÓN

Riobamba – Ecuador

2021

©2021, Andrés Fabricio Barahona Andino; & Kleber Ramiro Ríos Sigcho

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

Nosotros, Andrés Fabricio Barahona Andino y Kleber Ramiro Ríos Sigcho, declaramos que el presente trabajo de titulación es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación: el patrimonio intelectual pertenece a la escuela superior politécnica de Chimborazo.

Riobamba, 1 de diciembre de 2021



Andrés Fabricio Barahona Andino
C.I.:172535378-1



Kleber Ramiro Ríos Sigcho
C.I.:092676591-8

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA AUTOMOTRIZ

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular: Tipo Proyecto de Investigación: “**DESARROLLO DE UNA PLATAFORMA MÓVIL TIPO ACKERMANN PARA LA VALIDACIÓN DE ALGORITMOS DE NAVEGACIÓN AUTÓNOMA DENTRO DEL CAMPUS ESPOCH**”, realizado por los señores: **ANDRÉS FABRICIO BARAHONA ANDINO; KLEBER RAMIRO RIOS SIGCHO**, ha sido minuciosamente revisado por los miembros del tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el tribunal autoriza su presentación

Ing. Juan Carlos Rocha Hoyos _____ 2021-12-01
PRESIDENTE

Ing. Javier José Gavilanes Carrión _____ 2021-12-01
DIRECTOR

Ing. Johnny Marcelo Pancha Ramos _____ 2021-12-01
MIEMBRO

DEDICATORIA

El presente trabajo de titulación se lo dedico en primer lugar a mi madre Ruth de los Ángeles Andino Carrasco quien con su lucha y sacrificio ha permitido cumplir un sueño más en mi formación como profesional, por ser la motivación y el apoyo constante en cada paso de mi vida, por su amor y cariño incondicional. A mi padrastro William Taco por todo su apoyo incondicional, emocional para mi formación personal y profesional en el campo laboral. A mi hermano Erick que con su cariño y esfuerzos me llevaron a tomar buenas decisiones en lo académico y personal, A mi novia Carolina Guamán que con su apoyo incondicional y amor creyó en mí desde un principio para conseguir mi sueño y seguir adelante pese a las diferentes adversidades. A mi abuelita Juana Andino, que siempre estuvo para brindarme palabras de aliento. A mis familiares que han sabido brindarme su apoyo y consejo.

Dedicado también a Luis Ases gracias por todas tus palabras de aliento aunque a pesar de esta pandemia del 2020 te nos marchaste pronto, siempre valorare tus enseñanzas de no rendirse jamás y Roberto Taco gracias por tu cariño y apoyo que siempre creer en mí, Aunque físicamente no los pueda ver sé que desde el cielo comparten la felicidad de este logro.

Andrés Fabricio Barahona Andino

El presente trabajo de titulación se lo dedico a mis padres Ramiro Ríos y Jessmine Sigcho que, a base de esfuerzo, sacrificio y lucha me han permitido cumplir un logro más, por ser esa inspiración de constante trabajo duro y apoyo que día a día me inculcaron con un cariño incondicional. A mi hermana Cristhina Ríos por ser esa consejera y ser una persona muy importante en mi vida, A mi novia Mishel Alpusig por ser ese apoyo incondicional que con su amor diario siempre creyó en mí y me apoyo para cumplir mi sueño, A mis abuelitos Luis Sigcho y Fanny Madero quienes me enseñaron y aconsejaron a crecer como persona y me apoyaron a cumplir mi anhelo de ser Ingeniero Automotriz. A mis familiares que con sus palabras de aliento y consejos fueron guías para ser la persona que hoy soy, siendo este logro no solo mío sino de todos ellos.

Kleber Ramiro Ríos Sigcho

AGRADECIMIENTO

Agradezco a Dios por haberme dado la vida, por encaminarme en cada una de sus voluntades y desarrollarme como un ser humano y profesional de provecho para la sociedad. A mi Madre Ruth Andino y mi Padrastro William Taco que fueron pilares fundamentales en mi vida que con su apoyo y amor incondicional estuvieron en momentos importantes de mi vida, así mismo a mi hermano Erick Taco que con su cariño y esfuerzos me llevaron a tomar buenas decisiones en lo académico y personal, A mi novia Carolina Guamán que con su apoyo incondicional y amor creyó en mi desde un principio para conseguir mi sueño y seguir adelante pese a las diferentes adversidades, a mi Abuelita Juana Andino me apoyaron y me inculcaron valores muy importantes y a nunca rendirse ante un sueño porque el que persevera alcanza lo que uno desea sin importar cuán difícil sea. A la persona con quien compartí buenos y malos momentos durante varios años de mi vida. Al Ingeniero José Gavilanes, quien con su apoyo incondicional han hecho posible la culminación del presente trabajo de titulación.

Andrés Fabricio Barahona Andino

Agradezco antes que todo a Dios que me dio las fuerzas para encaminar todo el trayecto de este ciclo de mi vida para mi formación académica, A mi Madre Jessmine Sigcho y mi Padre Ramiro Ríos que fueron pilares fundamentales en mi vida que con su apoyo y amor incondicional estuvieron en momentos importantes de mi vida, así mismo a mi hermana Cristhina Ríos que con su cariño y consejos me llevaron a tomar buenas decisiones en lo académico y personal, A mi novia Mishel Alpusig que con su apoyo incondicional y amor creyó en mi desde un principio para conseguir mi sueño y seguir adelante pese a las diferentes adversidades, a mi Abuelitos Luis Sigcho y Fanny Madero que me apoyaron y me inculcaron valores muy importantes y a nunca rendirse ante un sueño porque el que persevera alcanza lo que uno desea sin importar cuán difícil sea. Al Ingeniero José Gavilanes, quien con su apoyo incondicional han hecho posible la culminación del presente trabajo de titulación

Kleber Ramiro Ríos Sigcho

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE GRÁFICOS.....	xii
INDICE DE ANEXOS	xiii
RESUMEN.....	xiv
SUMMARY	xv
INTRODUCCIÓN	1

CAPÍTULO I

1. MARCO REFERENCIAL TEÓRICO.....	2
1.1. Antecedentes	2
1.2. Justificación.....	3
1.3. Hipótesis	3
1.4. Planteamiento del problema.....	3
1.4.1. <i>Formulación del Problema</i>	4
1.4.2. <i>Sistematización del problema</i>	4
1.5. Objetivos.....	4
1.5.1. <i>Objetivo General</i>	4
1.5.2. <i>Objetivo Especifico</i>	4
1.6. Estado del Arte	5
1.6.1. <i>Navegación autónoma</i>	5
1.6.1.1. <i>Conducción autónoma</i>	6
1.6.1.2. <i>Componentes de la navegación autónoma en vehículos.</i>	6
1.6.1.3. <i>Sistemas Inteligencia De Transporte</i>	8
1.6.2. <i>Dirección Ackermann</i>	9
1.6.2.1. <i>Condición de Ackermann</i>	10
1.6.2.2. <i>Cálculo de las condiciones de Ackermann</i>	10
1.6.2.3. <i>Trabajos Referentes acerca de la geometría en plataformas tipo Ackermann</i>	11
1.6.3. <i>Plataformas móviles con visión artificial</i>	13
1.6.4. <i>Análisis del software implementado en el diseño.</i>	15
1.6.4.1. <i>Diseño del Prototipo</i>	15
1.6.4.2. <i>Programación</i>	16

CAPITULO II

2.	MARCO METODOLÓGICO	18
2.1.	Tipo de investigación.....	18
2.2.	Diagrama de Flujo	19
2.3.	Concepción de la plataforma móvil de navegación autónoma.	21
2.4.	Diseño de la plataforma móvil en SolidWorks.....	21
2.4.1	<i>Chasis plataforma.....</i>	22
2.4.2	<i>Bases tipo C.....</i>	22
2.4.3	<i>Caja reductora</i>	23
2.4.4	<i>Servomotor</i>	23
2.4.5	<i>Diseño de la rueda en CAD.....</i>	24
2.5.	Análisis Estructural.....	24
2.5.1	<i>Selección del material.....</i>	25
2.5.2	<i>Análisis de deformaciones (ASTM-A36)</i>	26
2.5.3	<i>Análisis de deformaciones (PLASTICO).....</i>	28
2.5.4	<i>Estudio comparativo de los materiales.....</i>	30
2.6.	Selección del hardware que conforma el prototipo.	30
2.6.1.	<i>Raspberry Pi 2 Modelo B.....</i>	31
2.6.2.	<i>Cámara Web</i>	31
2.6.3.	<i>Modem Nano NAS Wireless Router.....</i>	32
2.6.4.	<i>Motor de corriente continua.....</i>	32
2.6.5.	<i>Servomotor</i>	33
2.6.6.	<i>Batería 6v /Alimentación autónoma</i>	33
2.6.7.	<i>USB WIFI (D-link DWA-125)</i>	34
2.7.	Diagrama de conexión	34
2.8.	Códigos de Navegación Autónoma.....	35
2.8.1.	<i>Código STOP</i>	35
2.8.1.1	<i>Entrenamiento por computadora.</i>	35
2.8.1.2	<i>Detector de objetos HAAR CASCADE.....</i>	35
2.8.1.3	<i>Creación del archivo XML.....</i>	36
2.8.1.4	<i>Prueba en computadora.....</i>	37
2.8.1.5	<i>Ejecución de códigos en el prototipo (Raspberry pi).....</i>	38
2.8.2.	<i>Código Detección de línea</i>	39
2.8.3.	<i>Código señal velocidad.</i>	43
2.8.4.	<i>Código de semáforo</i>	44

CAPITULO III

3.	ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	46
3.1.	Validación señal de stop.....	46
<i>3.1.1</i>	<i>Mañana (9-10 am).....</i>	<i>46</i>
<i>3.1.2</i>	<i>Medio Dia (13-14 Pm).....</i>	<i>48</i>
<i>3.1.3</i>	<i>Tarde (17-18 pm).....</i>	<i>51</i>
<i>3.1.4</i>	<i>Análisis de varianza- señal de stop.....</i>	<i>54</i>
3.2.	Detección de carriles.....	57
<i>3.2.1.</i>	<i>Entornos para pruebas del prototipo.</i>	<i>57</i>
<i>3.2.2.</i>	<i>Detección de Carriles y Corrección de giro.</i>	<i>59</i>
<i>3.2.3.</i>	<i>Corrección giro de dirección.</i>	<i>59</i>
<i>3.2.3.1</i>	<i>Escenario A.....</i>	<i>59</i>
<i>3.2.3.2</i>	<i>Escenario B.....</i>	<i>60</i>
<i>3.2.3.3</i>	<i>Escenario C.....</i>	<i>61</i>
3.3	Discusión.....	62
	CONCLUSIONES.....	66
	RECOMENDACIONES.....	67
	GLOSARIO	
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-3: Datos del enfoque focal a las (9-10).....	46
Tabla 2-3: Datos del enfoque focal a las (13-14 pm).....	49
Tabla 3-3: Datos del enfoque focal a las (13-14 pm).....	52
Tabla 4-3: Datos del enfoque focal a diferentes horas del día mediante condiciones.....	55

ÍNDICE DE FIGURAS

Figura 1-1: Sistema y detección de Cámaras.....	7
Figura 2-1: Sistema de Detección de Sensores ultrasónicos.....	7
Figura 3-1: Distribución de los SIT	9
Figura 4-1: Condición de Ackermann	10
Figura 5-1: Esquema de Distribución Ackerman.....	12
Figura 6-1: Prototipo tipo Ackermann con cámaras de visión artificial	13
Figura 7-1: Prototipo de visión artificial con Kinect 360, Arduino y Open CV	14
Figura 8-1: Visión Artificial para prototipo móvil en seguimiento de carril	15
Figura 9-1: Python3 y Open CV	16
Figura 10-1: Arduino y Servomotor	17
Figura 1-2: Carril ciudad Riobamba	21
Figura 2-2: Escala del carril vs ciclovía.....	22
Figura 3-2: Diseño del chasis en CAD	22
Figura 4-2: Diseño en CAD de las C	23
Figura 5-2: Caja reductora en CAD.....	23
Figura 6-2: Diseño del Servomotor en CAD	23
Figura 7-2: Rueda en CAD	24
Figura 8-2: Diseño en CAD de las C	24
Figura 9-2: Propiedades Chasis	25
Figura 10-2: Análisis de propiedades físicas del prototipo (plástico).....	26
Figura 12-2: Mallado Chasis.....	27
Figura 13-2: Análisis Estructural Chasis	27
Figura 14-2: Factor de Seguridad Chasis.....	28
Figura 15-2: Detalles Malla Chasis (plástico)	28
Figura 16-2: Mallado Chasis (plástico)	29
Figura 17-2: Análisis deformación unitaria	29
Figura 18-2: Factor de Seguridad Chasis.....	30
Figura 19-2: Raspberry pi 2.....	31
Figura 20-2: Cámara Web.....	32
Figura 21-2: NAS Wireless	32
Figura 22-2: Motor de corriente continua.....	32
Figura 23-2: Servomotor.....	33
Figura 24-2: Alimentación de la fuente de 5V.....	33
Figura 25-2: USB Wifi	34
Figura 26-2: Diagrama de conexión	34
Figura 27-2: Fotografías de imágenes positivas.	36
Figura 29-2: Entrenamiento de señal de Stop con un error (6.35783e-07).....	37
Figura 30-2: Detección de la señal de STOP por medio de la cámara de la computadora.	37

Figura 31-2: Prototipo detectando señal de Stop.....	38
Figura 32-2: Señal de Stop Detectado por la Raspberry y una cámara web.....	38
Figura 33-2: Detección en tiempo real de señales de Tránsito con el algoritmo	39
Figura 34-2: Filtro Canny	39
Figura 35-2: Filtro rango de interés	40
Figura 36-2: Transformada de Hough	40
Figura 37-2: Detección en tiempo real del Carril	41
Figura 38-2: Código detección de Línea.....	42
Figura 39-2: Detección de señal de velocidad mediante la Raspberry.	43
Figura 40-2: Detección se señal de velocidad con el ordenador.....	43
Figura 41-2: Reconociendo del semáforo (ROJO)	44
Figura 42-2: Reconociendo del semáforo (VERDE).....	44
Figura 43-2: Código de semáforo.	45
Figura 1-3: Z en una muestra: enfoque focal (9-10 am)	48
Figura 2-3: Z en una muestra: enfoque focal (13-14 pm)	51
Figura 3-3: Z en una muestra: enfoque focal (17-18 pm)	54
Figura 4-3: Interior de la Espoch.....	58
Figura 5-3: Exterior de la Espoch.	58
Figura 6-3: Detección de Carriles.....	58
Figura 7-3: Filtros en la detección de carriles.....	59
Figura 8-3: Prototipo Ackerman, cámara LIDAR.	63
Figura 9-3: Prototipo tipo Ackermann en el campus ESPOCH.....	63
Figura 10-3: Pruebas en el Campus ESPOCH.....	63
Figura 11-3: Detección del semáforo.....	64
Figura 12-3: Filtros de visión artificial.	65

ÍNDICE DE GRÁFICOS

Gráfico 1-2: Diagrama de flujo de Proceso de Validación Prototipo	20
Gráfico 1-3: Cálculos estadísticos del enfoque focal (9-10 am)	47
Gráfico 2-3: Histograma de enfoque focal.....	47
Gráfico 3-3: Cálculos estadísticos del enfoque focal (13-14 pm).....	50
Gráfico 4-3: Histograma de enfoque focal.....	50
Gráfico 5-3: Cálculos estadísticos del enfoque focal (17-18 pm).....	53
Gráfico 6-3: Histograma de enfoque focal (17-18).....	53
Gráfico 7-3: Grafico de puntos, algoritmo de stop	56
Gráfico 8-3: Análisis de Varianzas	57
Gráfico 9-3: Comportamiento del giro de corrección.	60
Gráfico 10-3: Comportamiento del giro de corrección B.	61
Gráfico 11-3: Comportamiento del giro de corrección C.	62

INDICE DE ANEXOS

ANEXO A: Código de Stop.

ANEXO B: Código Detección de línea

ANEXO C: Código de detección del semáforo.

ANEXO D: Datasheet Puente H L298N

ANEXO E: Diagrama de conexión con Arduino

ANEXO F: Datasheet Raspberry Pi II

ANEXO G: Datasheet motor DC

ANEXO H: Datasheet ServoMotor

ANEXO I: Diagrama de conexión prototipo ESPOCH01

ANEXO J: Diagrama de flujo Procesamiento de la imagen del prototipo ESPOCH01

ANEXO K: Prototipo Vehículo ESPOCH01

RESUMEN

El objetivo del presente estudio fue desarrollar una plataforma móvil tipo Ackerman equipada con visión artificial para validar algoritmos de navegación autónoma dentro del campus ESPOCH. Para lo cual se obtuvo la información en vivo mediante una cámara pi, la cual envía la información a una Raspberry pi que posteriormente se encargó de procesar la información y enviar a los actuadores. La plataforma se desarrolló en conjunto con un diseño y construcción del mismo; la validación de los algoritmos se realizó en diferentes entornos y situaciones meteorológicas cubriendo etapas de visión artificial para la detección de señales de tránsito y detección de carretera, la primera etapa dedicada al algoritmo de detección a través de una cámara pi desarrollado en OpenCV. La segunda etapa se encargó de la localización de las señales de tránsito y detección de carretera a través de filtros como Canny, Haar Cascade y la Transformada de Hough, resultando en una precisión arriba del 85% en todas las pruebas en condiciones favorables. Para el análisis estadístico se utilizó un modelo de recolección de datos de entre 20 a 30 datos en 3 etapas del día para un posterior análisis en Minitab dando un porcentaje mayor de efectividad en las tomas del medio día. Se concluye que la validación de los algoritmos de detección de carril, señales de tránsito y semáforos actúan de forma correcta en un entorno del medio día donde la luz favorece a la cámara. Se recomienda para una mejor lectura y rapidez adaptar una cámara más potente como una Raspberry de mayor capacidad y velocidad para no tener retrasos en las lecturas en vivo.

Palabras clave: <PLATAFORMA MÓVIL> <VISIÓN ARTIFICIAL> <NAVEGACIÓN AUTÓNOMA> <RASBPERRY PI 2> <TRANSFORMADA DE HOUGH>.



Firmado electrónicamente
por:

**HOLGER
GERMAN
RAMOS
UVIDIA**

0643-DBRA-UPT-2022

2022-04-08

SUMMARY

The objective of the present study was to develop an Ackerman-type mobile platform equipped with the artificial vision to validate autonomous navigation algorithms within the ESPOCH campus. The live information was obtained through a pi camera, which sends information to a Raspberry pi, in charge of processing information and sending it to the actuators. The platform was developed in conjunction with a design and construction of the same; the validation of the algorithms was carried out in different environments and meteorological situations; covering stages of artificial vision for the detection of traffic signs and road detection. The first stage dealt with the detection algorithm through a pi camera developed in OpenCV. The second stage was responsible for the location of traffic signs and road detection through filters such as Canny, Haar Cascade, and the Hough Transform, resulting in an accuracy of over 85% in all tests under favorable conditions. For the statistical analysis, a data collection model of between 20 to 30 data in 3 stages of the day was used for a subsequent analysis, in Minitab, giving a higher percentage of effectiveness in the noon meals. It is concluded that the validation of the lane detection algorithms, traffic signals and traffic lights act correctly in a noon environment where the light favors the camera. It is recommended for a better reading and speed to adapt a more powerful camera such as a Raspberry with higher capacity and speed to avoid delays in live readings.

Keywords: <MOBILE PLATFORM> <ARTIFICIAL VISION>
<AUTONOMOUS NAVIGATION> <RASBPERRY PI 2>
<HOUGH TRANSFORMATION>.



Firmado electrónicamente por:

**PATRICIA
PILAR MOYOTA
AMAGUAYA**

INTRODUCCIÓN

Hoy en día, la tecnología forma parte de nuestras vidas de formas que nadie podría haber previsto hace unos años. Cada vez es más frecuente la aparición de nuevos dispositivos con el objetivo de mejorar o ayudar de alguna forma a nuestra vida diaria. Gracias a la aparición de nuevas tecnologías, muchos sectores industriales ahora están innovando y, por lo tanto, desarrollándose gracias al progreso tecnológico.

Una de las áreas que crece y mejora gracias al avance tecnológico es el sector de la automoción, o incluso el sector de las redes sociales gracias a la tecnología de visión artificial. Cada vez más, vemos que las grandes empresas tecnológicas y automotrices apuestan por los vehículos autónomos como el transporte del futuro sin la necesidad de un conductor humano. Como se puede imaginar, esto no es fácil de hacer, ya que hay infinidad de variables que no son fáciles de predecir, como un accidente, pero, aun así, la visión por computadora se puede reducir mucho.

En el presente proyecto de investigación tomamos la iniciativa para implementar este nuevo prototipo a la Escuela de Ingeniería Automotriz a través de un prototipo tipo Ackermann para la validación de algoritmo de navegación autónoma para futuras generaciones. El prototipo de plataforma móvil tipo Ackermann se realizará junto a sus algoritmos a validar en diferentes entornos y situaciones meteorológicas cubriendo etapas de visión artificial para la detección de señales básicas de tránsito y detección de carretera.

En el estado del arte se hará una investigación bibliográfica de varios conceptos referentes al tema como que es navegación autónoma, los diferentes programas a utilizar para la ejecución de los algoritmos de visión artificial, equipo en la construcción de nuestra plataforma tipo Ackermann y las diferentes codificaciones de los algoritmos ejecutados en los diferentes softwares.

El sistema de control consta de tres partes, la primera parte dedicada al algoritmo de detección por medio de visión artificial para recolectar información del entorno a través de una cámara web, el entorno de desarrollo en Python con OpenCV. La segunda etapa se encarga de la localización de las señales de tránsito y detección de carretera a través de filtros como Canny, Haar Cascade y la Transformada de Hough.

Finalmente se realiza un análisis de validación de cada uno de los algoritmos propuestos en nuestro prototipo, obteniendo como resultado de la recolección de datos una precisión del 95.00% en la detección de señales de tránsito y un 85% en detección del carril en condiciones favorables teniendo como resultado un funcionamiento aceptable en las diferentes situaciones meteorológicas del entorno.

CAPÍTULO I

1. MARCO REFERENCIAL TEÓRICO

1.1. Antecedentes

La regla de Ackermann es la principal razón para el avance de la investigación, este mecanismo es clave para conseguir una geometría óptima de giro con respecto a la trayectoria del vehículo. El teorema enuncia: “En el punto en el que un vehículo gira en una curva, los ejes de todas las ruedas deben encontrarse en un punto llamado centro permanente de la dirección”. Aquel sistema se aplicará en el prototipo al tener como una condición favorable la baja aceleración lateral, asegurando bajos ángulos de deslizamiento para un manejo estable y una fricción constante al suelo proporcionando un comportamiento al coche de mucha maniobrabilidad (Cova Rocamora R. S, 2019, p.15).

Desde su creación en 1886 con la aparición del primer motor de combustión interna por Karl Friedrich Benz y Gottlieb Daimler, fueron los pioneros en ventas de automóviles que ofrecían rendimientos aceptables. En la actualidad la empresa TESLA es una de las más importantes a nivel mundial por su alta gama de coches eléctricos, la cual fue fundada en el 2003 pero estuvo a punto de desaparecer en 2008 debido a problemas económicos.

Actualmente esta compañía propone incorporar un sistema de navegación totalmente autónoma en donde el auto sea capaz de imitar las capacidades humanas de manejo, así como percibir cada uno de los elementos y señaléticas que les rodean en su entorno. Esta nueva promesa de vehículos será idónea para eliminar los accidentes de tránsito provocados por errores humanos al volante, y a su vez reducir la contaminación ambiental dentro de nuestro planeta. (Cancela, et al, 2020)

El AutoPilot o Piloto automático es una de las dificultades extraordinarias para los fabricantes de vehículos, llevando consigo objetivos de crear un sistema computarizado que controle al carro y sea capaz de sí mismo simular las habilidades que tiene una persona en el automóvil. La inteligencia artificial controla funciones como la comunicación entre vehículos (V2V) captura y gestión de datos en tiempo real, aplicaciones dinámicas para la movilidad, monitoreo de flota, controles de carretera y tráfico dando al usuario una facilidad en la conducción. (Cancela & Sardina, 2020)

1.2. Justificación

La presente investigación tiene como finalidad adaptar una plataforma móvil tipo Ackermann y controlarla para que mediante algoritmos de visión artificial pueda realizar una navegación autónoma en el Campus ESPOCH estableciendo una visión sobre el comportamiento técnico en nuestra región, otorgando a futuras generaciones una línea base de conocimiento y enseñanza para el desarrollo de emprendimientos acerca de vehículos autónomos para estudios académicos, por lo cual tomamos la iniciativa para implementar este nuevo prototipo a la Escuela de Ingeniería Automotriz a través de la validación de algoritmos de navegación autónoma para futuras generaciones.

Para poder basarnos en el tipo de dirección utilizada en nuestro prototipo se utilizó las referencias según “Se implementa la geometría de Ackerman, y su configuración tradicional de cuatro ruedas proporciona grados de libertad y no permite desplazamiento el lateral, limitando así las posibles trayectorias de estos movimientos, generando así una dirección de dirección adecuada para la respuesta del servomotor acoplado al sistema de dirección.” (Urquiza, 2017)

Debido que el autor menciona que los grados de libertad no permiten el desplazamiento lateral y nuestro prototipo en su mayor parte de la trayectoria es línea recta, este tipo de dirección se acopla mejor en el campus ESPOCH, los vehículos convencionales implementan este tipo de dirección Ackermann como principio fundamental para sus sistemas.

1.3. Hipótesis

Es posible validar algoritmos de navegación autónoma mediante reconocimiento de imágenes para la movilización dentro del Campus ESPOCH por medio del desarrollo de un prototipo tipo Ackerman a bajo costo.

1.4. Planteamiento del problema

Hoy en día la navegación autónoma es una realidad en donde operan en ambientes naturales no modificados y son capaces de tomar decisiones propias en función de una meta dada, desarrollando así una tendencia a crear algoritmos que otorguen cada vez una autonomía a un sistema.

Actualmente no existe información suficiente sobre la navegación autónoma en la región debido a sus limitaciones tecnológicas, en la investigación se pretende incorporar un prototipo tipo Ackermann para validar dichos algoritmos de sistemas autónomos e impulsar a la generación sobre nuevas técnicas dentro del campus ESPOCH y poseer una línea base de información para desarrollos dentro del área Automotriz.

1.4.1. Formulación del Problema

¿Es posible realizar un control electrónico de un prototipo de plataforma tipo Ackermann para la validación de un algoritmo de navegación autónomo que se pueda desplazar dentro del Campus ESPOCH? ¿Cómo influye el algoritmo en la navegación en el desarrollo de la plataforma móvil? ¿Qué posibles aplicaciones tiene este prototipo dentro de las instalaciones de la ESPOCH?

1.4.2. Sistematización del problema

- **Objeto de estudio:** Validación de algoritmos para Navegación Autónoma Campo de acción: Automotriz
- **Límite espacial:** La presente investigación será realizada en la Carrera de Ingeniería Automotriz, de la ESPOCH, en la provincia de Chimborazo.
- **Delimitación temporal:** La presente investigación se llevará en el periodo académico 2020-2021.

1.5. Objetivos

1.5.1. Objetivo General

- Desarrollar una plataforma móvil tipo Ackermann equipada con visión artificial para validar algoritmos de navegación autónoma dentro del Campus ESPOCH

1.5.2. Objetivo Especifico

- Seleccionar la plataforma adecuada y sus componentes para el control del sistema de navegación autónoma.
- Analizar los algoritmos para el control electrónico del vehículo y el procesamiento visual de las imágenes

- Implementar y Validar los algoritmos de navegación autónoma en la plataforma móvil tipo Ackermann
- Recolectar información visual por medio de prototipo tipo Ackermann dentro del campus Epoch, de acuerdo con una ruta asignada detectar señalética y reconocimiento de carril en condiciones meteorológicas.

1.6. Estado del Arte

1.6.1. Navegación autónoma

Según (Rendón, 2015) menciona que “La navegación autónoma es particularmente útil en entornos dinámicos y desconocidos. Tiene en cuenta los cambios del entorno en tiempo real”. Por lo cual nos menciona el autor que los diferentes sistemas de navegación autónoma requieren de parámetros meteorológicos recolectados en tiempo real para un rápido procesamiento de información.

La navegación autónoma es un sistema complejo el cual va evolucionando día a día, las áreas de trabajo que se debe desarrollar para tener un óptimo funcionamiento del sistema son específicamente tres características que según (Camps, P. 2016) afirma:

“Detección del entorno: Mediante visión artificial, es necesario analizar lo que está sucediendo en el entorno y las condiciones que nos genera. Por tanto, obtenemos información de los semáforos y señales de tránsito, entendemos nuestra posición relativa a la calle o carril de la autopista, y obtenemos información sobre el tráfico que nos rodea analizando el comportamiento y las condiciones de los vehículos que circulan junto a nosotros. Ajustar los parámetros de conducción del vehículo según las condiciones obtenidas del entorno, como aumentar o disminuir la velocidad, cambiar el ángulo de giro del volante o simplemente cambiar el modo de conducción e indicar un conjunto de intersecciones desde los puntos inicial y final”.

El autor nos trata de mencionar en su investigación realizada acerca de navegación autónoma, nos menciona algunas características y condiciones que deben estar presentes en el estudio para una recolección de datos adecuada.

1.6.1.1. *Conducción autónoma*

Los vehículos autónomos cuentan con un sistema de conducción autónoma que optimiza varias funciones humanas, lo cual los ingenieros automotrices han decidido dividirles en varios niveles.

Según el (Cea. S.A., 2021) menciona estos que “Hay 5 niveles de conducción autónoma divididos en: conductor puro, conducción asistida, automatización parcial, alta automatización y automatización total; a su vez, hay tres roles principales: el conductor como factor humano, el sistema de conducción automático y los que tienen los sistemas convencionales el coche en sí.”

Se dividen en los siguientes niveles de conducción:

- **Nivel 0:** Los sistemas que no poseen ninguna intervención robótica son netamente mecánicos y controlados únicamente por intervención humana.
- **Nivel 1:** Es solo una guía de conducción que mejora la comodidad. Puede un control de viaje o una advertencia de cambio de ruta.
- **Nivel 2:** El vehículo puede mantener la velocidad y girar el volante, sin embargo, no tendrá la opción de identificar impactos o predecir impedimentos.
- **Nivel 3:** En este nivel, puede reconocer diferentes vehículos y obstáculos, el vehículo que tiene la opción de pasar a otro carril autogobernado del tráfico así lo requiere.
- **Nivel 4:** El vehículo cuenta con autonomía para terminar todo el recorrido solo. En cualquier caso, tiene controles de regateo directo con el objetivo de que esté muy bien preparado para el control manual en caso de una decepción del marco autosuficiente.
- **Nivel 5:** El vehículo no cuenta con volante y ningún control manual humana, este sistema será controlado únicamente por un sistema autónomo. En caso de fallo este vehículo se comunicará con otro vehículo autónomo y lo llevaría a un centro de reparación automatizada.

1.6.1.2. *Componentes de la navegación autónoma en vehículos.*

Los principales componentes de un sistema de navegación autónoma en transporte terrestre, tomaremos como referencia uno de los vehículos pilares en estos sistemas como son los Testas. Que su mayor ambición es llegar a un nivel 5 de conducción autónoma y poder reducir la contaminación ambiental debido que todos sus prototipos son totalmente eléctricos ecológicos que no hacen daño a nuestro planeta. Según la empresa (Tesla Motors, 2020) nos menciona algunos de los componentes necesarios para tener un óptimo funcionamiento de vehículos autónomos.

Hardware: Es importante para una conducción totalmente autónoma en casi todas las condiciones en las que decidirá el curso ideal, explorará las carreteras urbanas e incluso sin marcas en el camino, supervisará puntos de cruce complejos con semáforos, señales de alto y rotondas, y conducirá por avenidas muy pobladas con autos que se mueven a gran velocidad. Apareciendo en tu objetivo, con suprema autosuficiencia

Cobertura Avanzada del Sensor: Alrededor de con 8 cámaras en el automóvil que otorga 360 grados de visibilidad del entorno del automóvil con un alcance de hasta 250 metros



Figura 1-1: Sistema y detección de Cámaras

Fuente: (Tesla Motors, 2020)

Radar: posee una longitud de onda que atraviesa la niebla, el polvo, la lluvia, la nieve y debajo de los automóviles, el radar juega un papel esencial en la detección y respuesta a los objetos que avanzan. (Tesla Motors, 2020)

Sensores Ultrasónicos: un rango con una sensibilidad mejorada utilizando señales codificadas de forma única. Estos sensores son útiles para detectar automóviles cercanos, especialmente cuando invaden su carril, y brindan orientación al estacionar. (Tesla Motors, 2020)

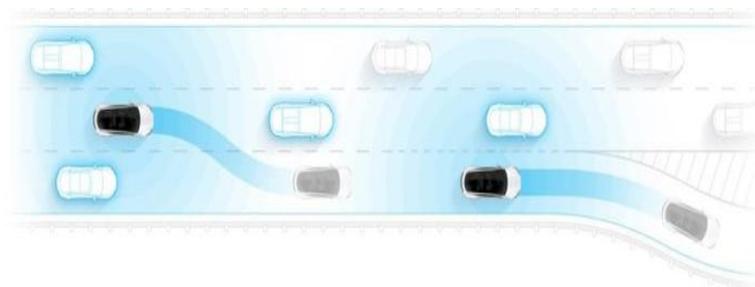


Figura 2-1: Sistema de Detección de Sensores ultrasónicos

Fuente: (Tesla Motors, 2020)

Fuente de Procesamiento: contiene una potencia de cálculo que ejecuta la nueva red neuronal desarrollada por Tesla para el software de procesamiento de visión, sonar y radar. (Tesla Motors, 2020).

1.6.1.3. Sistemas Inteligencia De Transporte

Con respecto a los Sistemas Inteligentes de Transporte basado en (Milanés et al, 2008) como nos dice en su artículo Sistema de Posicionamiento para vehículos autónomos “Su base principal es mejorar la seguridad, eficiencia y comodidad de cualquier medio de transporte que SIT establece una interfaz entre vehículos y carreteras (diseñada para facilitar la conducción)”.

El autor menciona que el sistema sea capaz de manejar los actuadores del vehículo de tal forma que lo haría un ser humano con eficacia enfocándose en una de las principales fuentes de información.

Un ejemplo claro nos indica (León, J. M. 2013) el cual en su tesis: Implantación de Nuevas Tecnologías de los Sistemas Inteligentes de Transporte en un Vehículo Eléctrico nos define como: “Los ITS pueden ser un conjunto de aplicaciones avanzadas en tecnología informática, electrónica y de las comunicaciones, desde el punto de vista social, económico y ambiental, tienen como objetivo mejorar la movilidad, seguridad y productividad del transporte y optimizar el uso de la información. Eficiencia del consumo de energía y mejora de la capacidad del sistema de transporte para reducir las emisiones”.

Siendo una gran ayuda en lo que respecta a la protección del medio ambiente asociado al entorno automotriz dando nuevas idea e ideologías para un desarrollo factible en un futuro dado con sistemas desarrollados en lo que respecta a la comunicación de varias tareas las cuales entre ellas están:

- Mejora de la vigilancia, videovigilancia y auto vigilancia
- Sistemas de seguridad avanzada Vehículo a Vehículo y Entre vehículo e infraestructura
- Mejora de las condiciones de seguridad en el transporte por carretera
- Mejora de la captura de datos y servicios de monitorización
- Mejora de la difusión de información
- Coordinación entre sistemas ITS
- Mejora de la vialidad urbana y Gestión de determinadas mercancías y de los terminales modales
- Administración electrónica y E-movilidad

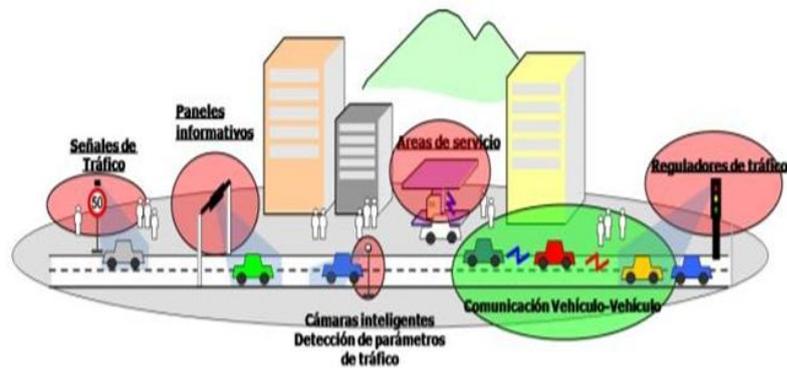


Figura 1-1: Distribución de los SIT

Fuente: (León, J. M. 2013)

Entonces para el desarrollo de un ITS óptimo se debe obtener un entorno como se ve en la figura 3-1. En la cual el conjunto de subsistemas tales como los paneles, señales y reguladores de tráfico y la comunicación entre vehículos dan información a los sensores a bordo del automóvil. Procediendo así al análisis de estos factores por medio de un procesador, algoritmos los cuales leerán dichas señales y enviarán la acción que corresponda al actuador del automóvil llevando consigo un bucle cerrado el cual va a estar en constante monitoreo hasta llegar a su destino.

1.6.2. Dirección Ackermann

La dirección tipo de Ackermann es uno de los diseños más utilizados en vehículos debido a su principio de funcionamiento que es muy eficiente para calcular la dirección que va a tomar el vehículo en una curvatura, solo se le han realizado pequeñas modificaciones al sistema para mejorar la eficiencia de estos.

Podemos mencionar que según el autor (Hernández, 2017) menciona que “Una de las configuraciones más comunes en los automóviles es la configuración de Ackerman, que puede coordinar mecánicamente los ángulos de las dos ruedas delanteras. Las dos ruedas delanteras están conectadas a un eje común para la dirección, mientras que las dos ruedas traseras están conectadas a la otra. el eje de la unidad.” p.13.

El autor nos menciona cual es el principio de funcionamiento de la dirección Ackermann por lo cual vemos su eficiencia en el funcionamiento que lo implantaremos en nuestro prototipo para una dirección más precisa en nuestros diseños.

1.6.2.1. Condición de Ackermann

Las condiciones de Ackermann se dan por medio de una igualación entre la resta de los ángulos de los radios interiores y exteriores con respecto a la distancia de las llantas sobre la de los ejes por lo cual el autor (Teodoro, J.2014) nos menciona que:

“Para cumplir con esta condición cinemática, existen varios factores, uno de los cuales es considerar un vehículo de cuatro ruedas cuya dirección sea hacia adelante, permitiendo una curva entre la rueda interior y la rueda exterior sin ser girada con respecto al mismo centro de rotación. y patinaje. O, llamándose a sí mismo una condición designada de Ackerman”

Por lo cual la mayoría de los vehículos en circulación poseen estas condiciones o podríamos llamarlos a todos los que poseen los sistemas vehículos con sistemas Ackermann.

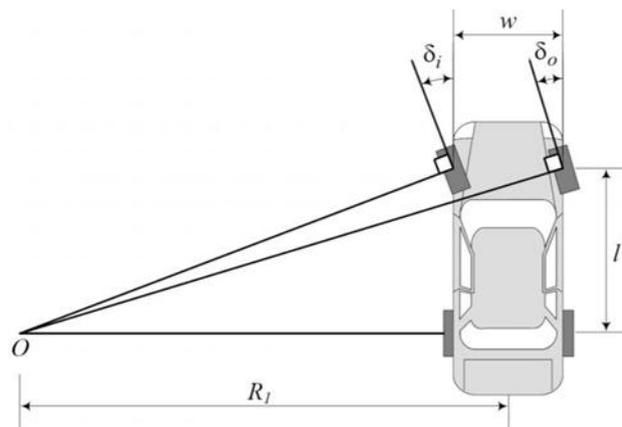


Figura 2-1: Condición de Ackermann

Fuente: (Teodoro, J.2014)

En la figura 4-1 podemos observar cómo poder detectar cada uno de los elementos para calcular el Angulo de Ackermann y donde tenemos nuestro punto en común de las cuatro ruedas.

1.6.2.2. Cálculo de las condiciones de Ackermann

Una forma de corroborar datos es que la condición existente se valide por medio de fórmulas a partir de los datos obtenidos con instrumentos de medición.

Esta condición se expresa con la siguiente fórmula

$$\cot \delta_o - \cot \delta_i = \frac{w}{l}$$

Siendo:

- δ_0 : Angulo de la rueda exterior del prototipo.
- δ_i : Angulo de la rueda interior del prototipo
- W : Distancia entre llantas delanteras
- l : Distancia entre los ejes delanteros

Según la fórmula para la condición de Ackermann nos menciona que dicha ecuación [1] se obtiene de las relaciones trigonométricas existentes en la geometría con el centro O y la distancia recta del eje trasero con las llantas delanteras interior y exterior al punto O otra forma de obtener los ángulos de inclinación puede ser

$$\tan\delta_i = \frac{1}{R_1 + 2} [2]$$

$$\tan\delta_0 = \frac{1}{R_1 + 2} [3]$$

En donde la ecuación [2] y [3] igualando a R_1 obtenemos el valor de la ecuación [1] demostrando la correcta obtención de la formula la cual se puede usar para la correcta corroboración de datos de la distancia y Angulo de giro

1.6.2.3. *Trabajos Referentes acerca de la geometría en plataformas tipo Ackermann*

Dependiendo de las restricciones geométricas de la plataforma móvil se tendrá que aplicar un esquema u otro. Hay que tener en cuenta que un buen modelo cinemático y dinámico es esencial para realizar el control sobre el vehículo dando una estabilidad de giro y un agarre provocando una eficiencia de giro para una correcta conducción

Según (Hernández. J, et al. 2016) en su publicación nos muestra el diseño e implementación de un robot con estructura Ackerman Steering usando una plataforma NIOS II en donde para la etapa del mecanismo este nos propone que la configuración creada evita el derrape de las ruedas haciendo que la rueda delantera interior posea un Angulo ligeramente mayor al Angulo de la rueda exterior cuando el sistema se encuentra girando.

Por otra parte. Un módulo para la navegación autónoma de máquinas Ackermann (Cova-Rocamora et al., 2018) en el congreso nacional XXXIX Jornadas de automática. En este artículo se describe

recta, el CIR aparecerá en el infinito, por lo que todas las ruedas del vehículo estarán girando a la misma velocidad.

1.6.3. Plataformas móviles con visión artificial

Las plataformas móviles son vehículos a control remoto que poseen diferentes sistemas de dirección que pueden ser: tipo diferencial y tipo Ackermann. Es el más común encontraremos sistemas con tipo Ackermann que es una de las opciones por la cual elegimos ese sistema. Realizamos un estudio de y recopilaremos información acerca de los diferentes prototipos que se han realizado para sacar nuestras conclusiones y tener un mejor enfoque hacia dónde va encaminado nuestro proyecto de investigación.

Uno de los aspectos más relevantes para el estudio de la visión artificial en prototipos nos menciona el autor (Benavent Pla, 2017) describe que:

“Una de las principales mejoras puede ser el uso de la nueva Raspberry Pi 3 u Odroid-XU4, que tiene cuatro núcleos y puede soportar mejor la carga intensiva de la CPU debido al uso de la visión por computadora, que consume una gran cantidad de recursos. El uso de un Arduino u otra placa que pueda impulsar varios motores a través del control PWM, así como el método anterior, puede tener una mejora más poderosa que el método actual, que puede generar una señal PWM más detallada y precisa que PWM. Raspberry Pi 2 y la señal PWM actual generada por software.” P.79.

El autor menciona cita algunas mejoras para futuras generaciones que toman como pilar nuestro tema de estudio por lo cual vamos a tener en cuenta para nuestro estudio la validación de algoritmos y la estructura como tal, para poder ir superando y dejar a el pilar fundamental de estudio a futuras generaciones de estudio de vehículos autónomos



Figura 4-1: Prototipo tipo Ackermann con cámaras de visión artificial

Fuente: (Benavent Pla, 2017)

El uso de una cámara es indispensable para la recolección de datos por medio de visión artificial y el procesamiento de datos posteriormente con los componentes adecuados, en este caso la Kinect 360, la cual cuenta con características que la hacen una buena opción al momento de escoger para el desarrollo del proyecto de investigación como los autores (Diaz & Romero, 2012) nos explican:

“El uso de Kinect como sensor de visión estéreo en este trabajo de investigación es una herramienta con un rendimiento excelente porque proporciona un mapa de profundidad, es fácil de usar y es adecuada para el objetivo del proyecto, es decir, la navegación estéreo, ignorando los problemas típicos o la visión estéreo típica. Como la alineación y sincronización de dos cámaras RGB independientes.” (Diaz & Romero, 2012, p. 77)



Figura 5-1: Prototipo de visión artificial con Kinect 360, Arduino y Open CV

Fuente: (Diaz & Romero, 2012)

Características indispensables como el rango dinámico de visión y los lentes con sensores de color y profundidad lo hacen una herramienta potencialmente utilizable para el proyecto de investigación ya que se adapta a las prestaciones que se necesitan para la recolección de datos de acuerdo con los objetivos establecidos

Una de las funciones de la visión artificial relacionado a la navegación autónoma es mantenerse en el carril de acuerdo con un guiado de las líneas de la carretera para lo cual por medio de actuadores se recolecta datos procesándolos y reenviando para la posible acción, en este caso la solución que se le da para mantenerse en el carril y obtener datos correctos como nos menciona (Rodríguez et al, 2020) en su artículo:

“Soluciones de seguimiento de carril en curvas, rectas y circuitos cerrados. La propuesta incluye el uso de controladores y algoritmos de detección de carril para guiar al vehículo a permanecer en el carril. Los resultados muestran que el algoritmo propuesto es eficaz tanto en entornos luminosos como sin barreras. Siempre que haya iluminación natural o artificial, el algoritmo de detección puede predecir eficazmente el error de posicionamiento del vehículo. El controlador realiza ajustes favorables a la trayectoria, mientras no exista problema de iluminación en la detección, el vehículo no vibrará en un estado estable y se adaptará al carril con pequeños cambios.” (Rodríguez et al, 2020)

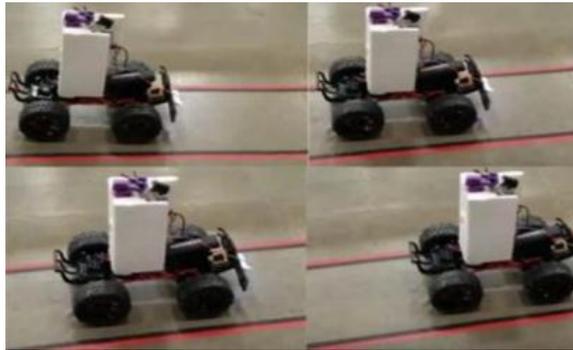


Figura 6-1: Visión Artificial para prototipo móvil en seguimiento de carril

Fuente: (Rodríguez et al, 2020)

1.6.4. Análisis del software implementado en el diseño.

Para realizar el proceso de visión artificial y navegación autónoma del prototipo, tomados como base artículos científicos y tesis relacionadas al tema, llegamos a la conclusión de implementar los programas: para el diseño Solidwork 2020 y para la programación las herramientas de Python y Open CV. Son los más indicados, debido a que se encuentran orientados en el manejo de objetos y procesamiento de imágenes de visión artificial en tiempo real para enviar la información a los actuadores.

1.6.4.1. Diseño del Prototipo

La construcción de un modelo en SolidWorks comienza generalmente con un boceto 2D. El dibujo se compone de la geometría como puntos, líneas, arcos, las cónicas (excepto la hipérbola), y las estrías. Las dimensiones se añaden al boceto para definir el tamaño y la ubicación de la geometría. Relaciones se utilizan para definir atributos tales como tangencia, paralelismo, perpendicularidad y concetricidad (Cuatitlán, 2015)

1.6.4.2. Programación

Debido a su procesamiento de imágenes y librerías de visión artificial se utilizó Python 3 y Open CV en nuestras computadoras para un preentrenamiento de cada uno de los algoritmos antes de utilizar en nuestra Raspberry pi 2. Se utilizó estas herramientas debido que dentro de la Raspberry pi 2 viene incluida u instalada de n software libre con estas herramientas para el control de cada una de sus salidas hacia los actuadores.

- **Python y Open CV**

Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Es una biblioteca de software de visión artificial y aprendizaje automático de código abierto.

La cual consta de una biblioteca tiene más de 2500 algoritmos optimizados, que incluye un conjunto completo de algoritmos de aprendizaje automático y visión por computadora clásicos y de última generación. (Yim, et al, 2003)



Figura 7-1: Python3 y Open CV

Fuente: (Yim, et al, 2003)

- **Arduino**

Es una placa microcontroladora que permite recibir señales y ejecutarlas a distintos actuadores. En el proyecto el servomotor es el actuador que este microcontrolador va a mandar los grados de inclinación que debe realizar con relación a las condiciones de la ciclovia. (Jain, 2018)

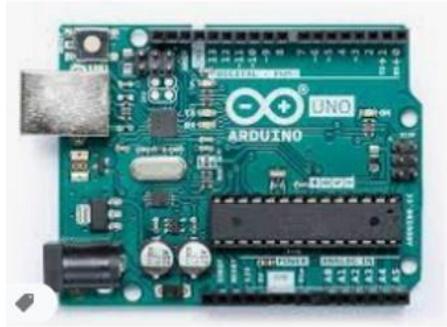


Figura 8-1: Arduino y Servomotor

Fuente: (Jain, 2018)

CAPITULO II

2. MARCO METODOLÓGICO

Este capítulo se centrará en los requisitos para implementar un sistema de visión artificial para una plataforma móvil. Se introducirá el diseño y montaje de una estructura mecánica necesaria para un prototipo tipo Ackermann y cámaras para inspección de imágenes. Posteriormente se revisará el diseño y montaje electrónico para el mencionado control y comunicación del sistema de visión artificial. Además, se introducirán en general las características de los elementos utilizados para implementar el proyecto de investigación y desarrollo tecnológico.

2.1. Tipo de investigación

Con el fin de poder demostrar la validación de algoritmos de navegación autónoma a un prototipo móvil tipo Ackermann se considera a la siguiente investigación como Cuasiexperimental – Descriptivo con un enfoque Cuantitativo puesto que en este documento la identificación, selección y evaluación objetiva y experimental de la plataforma se desarrollará en conjunto con un diseño y construcción del mismo, para posteriormente mediante algoritmos validarlos teniendo en cuenta posible variables a desarrollarse en el transcurso de la investigación dentro Campus de la Epoch.

- **Cuasi Experimental:** al tener variables en este caso como lo es el rango dinámico visual de la cámara web hay que tener los parámetros de distancia de una forma óptima para la lectura de los obstáculos o líneas de carretera para la correcta validación de los algoritmos al momento de las pruebas.
- **Descriptivo:** se considera como una forma de investigación dentro de este documento ya que los resultados que se obtendrán van a estar sometidos a una interpretación posterior la cual va a depender de muchos factores como potencia del motor, visión de la cámara, procesamiento de datos en posibles condiciones meteorológicas cambiantes.
- **Cuantitativa:** la recolección de datos es parte fundamental en este apartado y se lo consigue de la forma visual, puesto que la cámara web será la encargada de recolectar los datos visuales que posteriormente serán procesados por un Raspberry pi y enviados para el correcto funcionamiento de los actuadores y cumplir con los objetivos de la investigación

2.2. Diagrama de Flujo

El diagrama de flujo nos da las técnicas y procedimientos a seguir durante el desarrollo del proyecto llevando un orden de procesos a ejecutar conforme se va avanzando, el flujograma esta especificado en regiones para distinguir las partes de trabajo a desarrollarse durante un periodo de tiempo de forma didáctica y controlando posibles problemas o variables, como podemos observar en nuestro grafico 1-2 del flujograma.

2.3. Concepción de la plataforma móvil de navegación autónoma.

- Con la ayuda de herramientas integradas, se realiza un sistema de movimiento seguro que puede detectar señales de tránsito y los carriles de la vía.
- Configurar los parámetros necesarios en la dinámica y receptores móviles para tomar las mejores decisiones en el procesamiento de imágenes de visión artificial.
- Calibrar el rango de visión por medio de filtros de imágenes para eliminar los ruidos existentes en el procesamiento de imágenes por medio de visión artificial.
- Por medio de nuestra ventana de interfaz (tkinter) podemos observar en tiempo real el comportamiento de las etapas de adquisición de imágenes para el envío y recepción constante del monitoreo.

2.4. Diseño de la plataforma móvil en SolidWorks.

Para el diseño de la plataforma móvil se realizó mediante una simulación de elementos CAD para denotar errores antes de la construcción. Por lo cual se tomó como referencia las mediciones de un vehículo promedio en un carril de la ciudad de Riobamba, realizamos a una escala apropiada con relación a una ciclovía.



Figura 1-2: Carril ciudad Riobamba

Fuente: (Pérez, A. 2018)

Tomando como referencia la avenida canónico ramos de los exteriores de la politécnica de Riobamba. Que podemos verificar el carril principal con relación a la ciclovía y calcular una escala apropiada para nuestro prototipo. Como podemos observar en la figura.2-2.

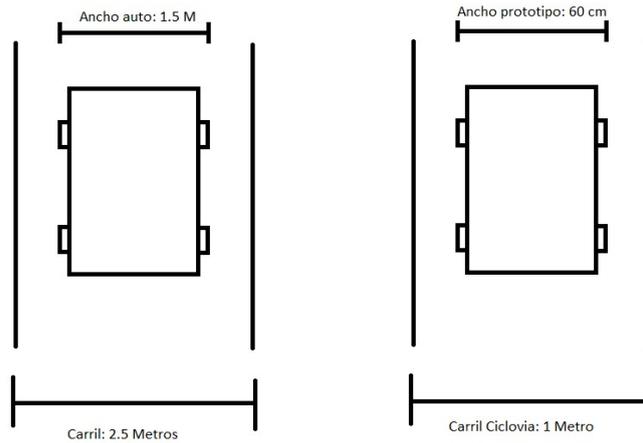


Figura 9: Escala del carril vs ciclovía.

Fuente: (Barahona & Ríos, 2021)

En la figura 2-2 podemos verificar la correcta relación que existe entre nuestro prototipo con un vehículo promedio en la ciudad de Riobamba.

2.4.1 Chasis plataforma

Para el desarrollo del chasis se utilizó medidas basadas en la ciclovía de la Espoch.

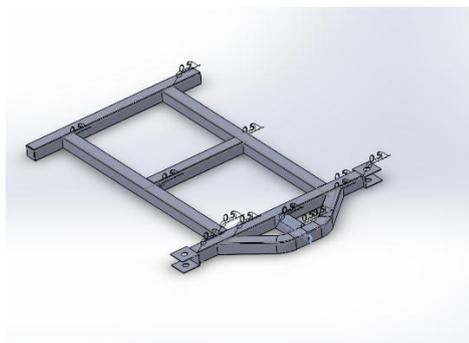


Figura 3-2: Diseño del chasis en CAD

Fuente: (Barahona & Ríos, 2021)

2.4.2 Bases tipo C

Las bases en forma de C son utilizadas para el montaje y soporte de la dirección tipo Ackermann.

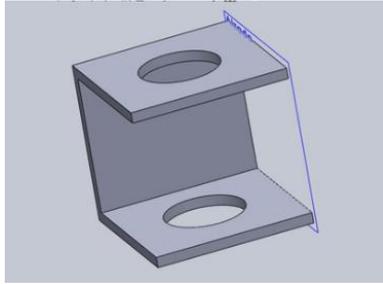


Figura 10-2: Diseño en CAD de las C

Fuente: (Barahona & Ríos, 2021)

2.4.3 *Caja reductora*

Caja reductora utilizada para el motor que va a generar el movimiento tanto delantero como de retro mediante el programa CAD.

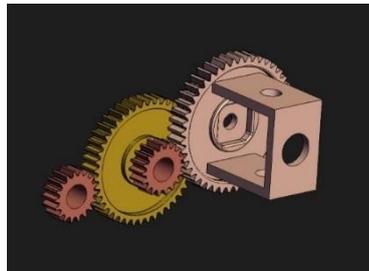


Figura 5-2: Caja reductora en CAD

Fuente: (Barahona & Ríos, 2021)

2.4.4 *Servomotor*

El servo motor va a controlar la dirección del prototipo con relación de procesamiento de imágenes, este servo va a corregir los grados hacia dónde va a proyectarse el prototipo.



Figura 6-2: Diseño del Servomotor en CAD

Fuente: (Barahona & Ríos, 2021)

2.4.5 *Diseño de la rueda en CAD*

Se utilizó una rueda de diámetro 24 cm para no tener mucha altura con respecto al piso. Se utilizó un material plástico para aligerar pesos.



Figura 7-2: Rueda en CAD

Fuente: (Barahona & Ríos, 2021)

2.5. **Análisis Estructural**

Para el análisis estructural tomamos varios factores para verificar qué tan fuerte es el diseño que tenemos a escala y proporcionalmente el aguante a la carga propuesta, por lo cual se toman como referencias análisis estructurales de prototipos realizados en varias ciudades de Latinoamérica. Se ha considerado fuerzas externas no mayor al peso aproximado de un niño, así mismo la gravedad distribuidos en la cara superior del chasis con los anclajes fijos que serían los puntos de apoyo de las llantas, descartando los pesos insignificantes como lo es la parte electrónica.

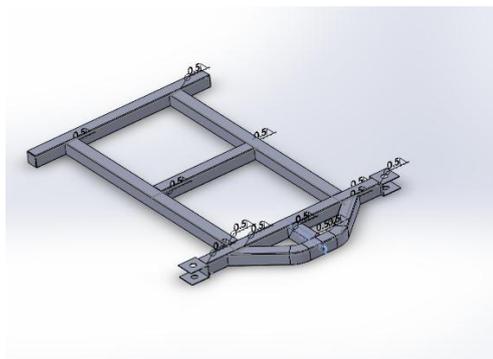


Figura 8-2: Diseño en CAD de las C

Fuente: (Barahona & Ríos, 2021)

Utilizando la herramienta “Solid Simulation” para el análisis estructural de la plataforma móvil tipo Ackermann, se evaluó el comportamiento que presentaría el diseño. Teniendo en cuenta las diferentes cargas posibles que pueda tener. Sobre el cual se debe tener muy en claro los lugares

en los cuales existirán fuerzas o presiones en este modelo, así como los puntos de sujeción que tendrán los distintos elementos del prototipo.

2.5.1 Selección del material

Se ha planificado la utilización de acero ASTM A36 para lo que será la fabricación del Chasis en donde este acero es capaz de resistir muy bien las fuerzas y pesos aplicados, entre las propiedades más destacables del prototipo se detallan a continuación en la figura 9-2.

Propiedades de masa de CHASIS ESTRUCTURA		
Configuración: Predeterminado<Como mecanizada>		
Sistema de coordenadas: -- predeterminado --		
Densidad = 0.01 gramos por milímetro cúbico		
Masa = 2090.64 gramos		
Volumen = 266324.16 milímetros cúbicos		
Área de superficie = 269111.81 milímetros cuadrados		
Centro de masa: (milímetros)		
X = 150.00		
Y = 0.00		
Z = -241.93		
Ejes principales de inercia y momentos principales de inercia: (gramos * milímet		
Medido desde el centro de masa.		
lx = (0.00, 0.00, 1.00)	Px = 14643330.09	
ly = (1.00, 0.00, 0.00)	Py = 55848921.12	
lz = (0.00, 1.00, 0.00)	Pz = 70263433.41	
Momentos de inercia: (gramos * milímetros cuadrados)		
Obtenidos en el centro de masa y alineados con el sistema de coordenadas de r		
Lxx = 55848921.12	Lxy = 0.06	Lxz = 0.00
Lyx = 0.06	Lyy = 70263433.41	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 14643330.0
Momentos de inercia: (gramos * milímetros cuadrados)		
Medido desde el sistema de coordenadas de salida.		
lxx = 178213702.95	lxy = 0.06	lxz = -75868166.3
lyx = 0.06	lyy = 239667719.42	lyz = 0.00
lzx = -75868166.36	lzy = 0.00	lzz = 61682834.2

Figura 9-2: Propiedades Chasis

Fuente: (Barahona & Ríos, 2021)

Otra opción de material el cual se analizó para el diseño del prototipo fue un plástico PVC, debido a su peso más ligero y fácil manipulación en la construcción del mismo como podemos observar en la figura 10-2.

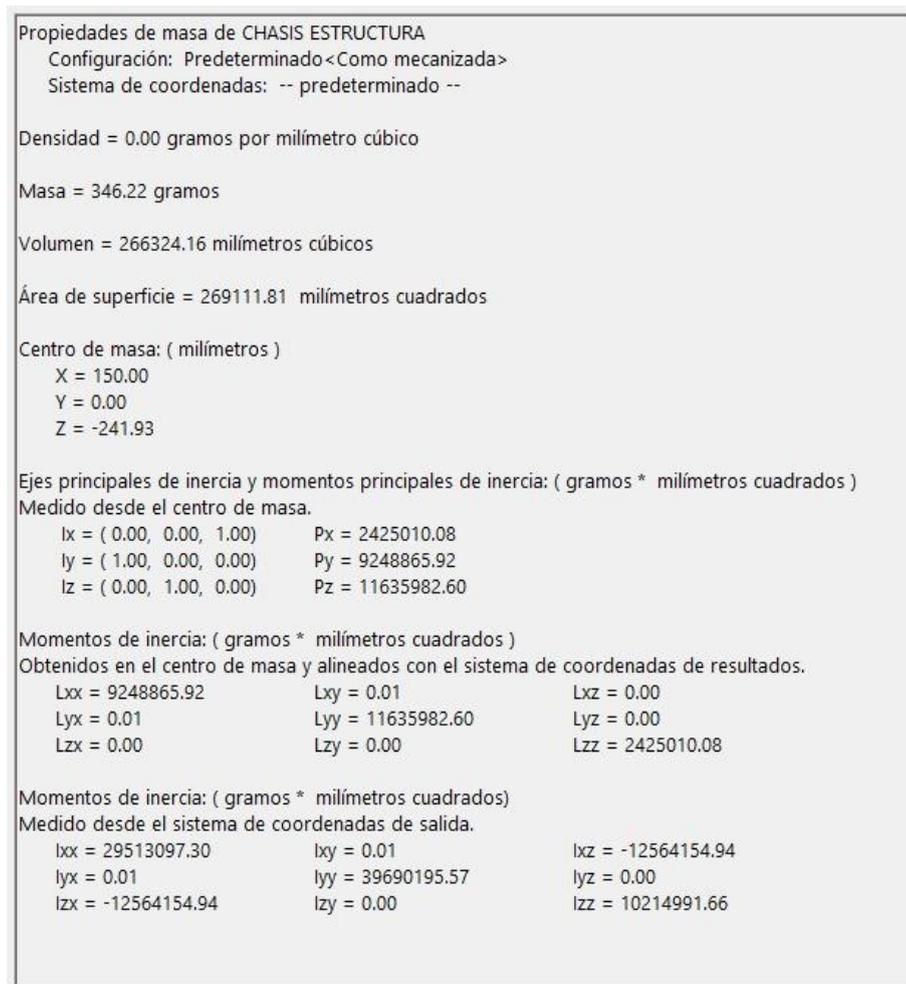


Figura 10-2: Análisis de propiedades físicas del prototipo (plástico)

Fuente: (Barahona & Ríos, 2021)

Como podemos observaren las figuras 9-2 y 10-2 las diferentes propiedades físicas de los distintos materiales propuestos son: la estructura de acero, con una masa de 2kg aproximadamente lo suficiente compacto para soportar el peso. A diferencia del diseño con el material de plástico que dio como resultado 0.346 kg de masa que es un diseño menos resistente.

2.5.2 Análisis de deformaciones (ASTM-A36)

El software realiza una división del solido en pequeños segmentos homogéneos, en un proceso denominado mallado. En donde se considera que las divisiones son de 2.65 mm de tamaño por elemento con una tolerancia de 0.1327 mm, siendo el software capaz de representar sin ningún problema el análisis, obteniendo datos como el total de nodos de 17541 con un total de 16 puntos jacobianos para una malla de alta calidad, como se representa en la figura 11-2 y la figura 12-2

Malla Detalles	
Nombre de estudio	Análisis estático 1 (-Predeterminado<Como mecanizada>-)
Tipo de malla	Malla mixta
Mallador utilizado	Malla estándar
Transición automática	Desactivar
Incluir bucles automáticos de malla	Desactivar
Puntos jacobianos para malla de alta calidad	16 puntos
Verificación jacobiana para el vaciado	Activar
Tamaño de elemento	2,65597 mm
Tolerancia	0,132798 mm
Calidad de malla	Elementos cuadráticos de alto orden
Número total de nodos	17541
Número total de elementos	8380
Tiempo para completar la malla (hh:mm:ss)	00:00:10
Nombre de computadora	

Figura 11-2: Detalles Malla Chasis

Fuente: (Barahona & Ríos, 2021)

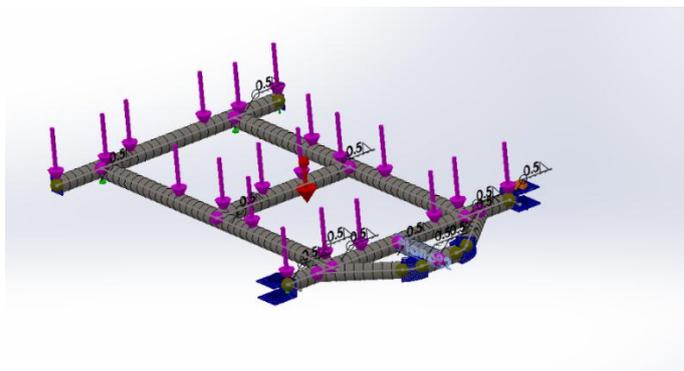


Figura 12-2: Mallado Chasis

Fuente: (Barahona & Ríos, 2021)

Cada elemento de esta malla será sometido a un análisis individual, teniendo así resultados como el Factor de seguridad y su deformación unitaria las cuales se muestran a continuación

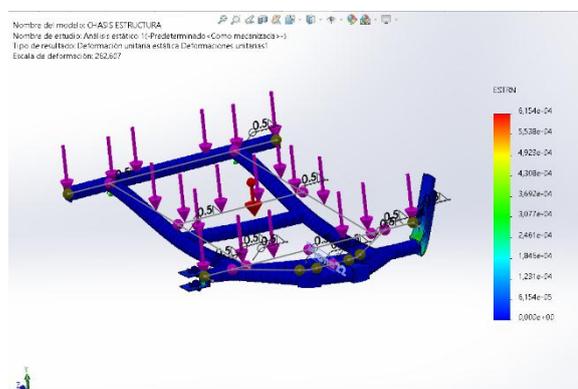


Figura 12: Análisis Estructural Chasis

Fuente: (Barahona & Ríos, 2021)

La presión ejercida es alrededor de 200N teniendo como puntos fijos las esquinas en donde se asentarán las llantas tanto delanteras como traseras, a su vez vamos a tener una deformación casi insignificante puesto que el mayor desbalance se dará en una de las C con un valor máximo de 0.000307mm muy pequeño para notararlo dando los requerimientos óptimos.

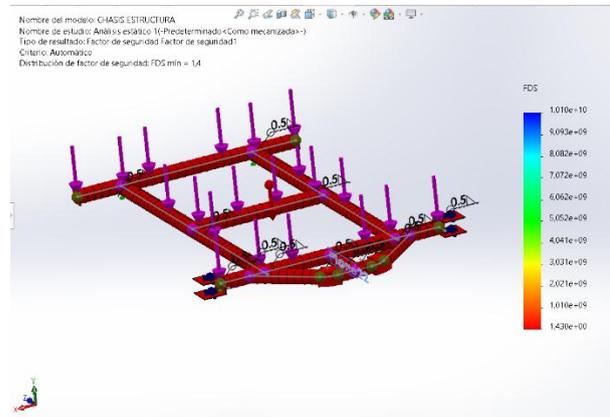


Figura 13-2: Factor de Seguridad Chasis

Fuente: (Barahona & Ríos, 2021)

Así mismo en la figura 14-2 observamos un factor de seguridad de 1.4 en donde va a estar por encima de los estándares globales que enmarcan como 1 el factor mínimo para un chasis rígido, teniendo un chasis compacto y lo suficientemente estable para soportar los pesos impuestos

2.5.3 Análisis de deformaciones (PLÁSTICO)

Como podemos verificar el análisis del prototipo, el diseño va a poseer los mismos valores para le mallado debido a que el software realiza cálculos de los valores automáticamente de acuerdo con la superficie de la estructura, como observamos en la figura 15-2.

Malla Detalles	
Nombre de estudio	Análisis estático 1 -(Predeterminado<Como mecanizado>-)
Tipo de malla	Malla mixta
Mallador utilizado	Malla estándar
Transición automática	Desactivar
Incluir bucles automáticos de malla	Desactivar
Puntos jacobianos para malla de alta calidad	16 puntos
Verificación jacobiana para el vaciado	Activar
Tamaño de elemento	2,65597 mm
Tolerancia	0,132798 mm
Calidad de malla	Elementos cuadráticos de alto orden
Número total de nodos	17541
Número total de elementos	8380
Tiempo para completar la malla (hh:mm:ss)	00:00:10
Nombre de computadora	

Figura 15-2: Detalles Malla Chasis (plástico)

Fuente: (Barahona & Ríos, 2021)

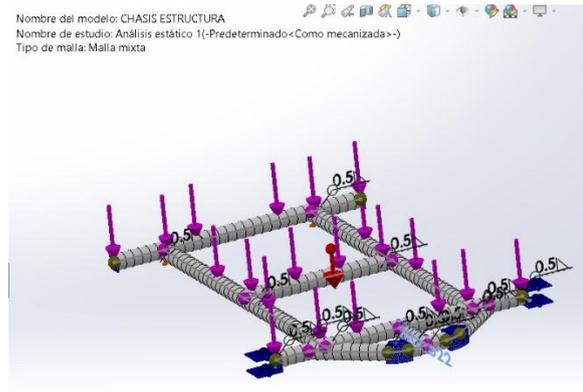


Figura 14-2: Mallado Chasis (plástico)

Fuente: (Barahona & Ríos, 2021)

Cada elemento de esta malla será sometido a un análisis individual, teniendo así resultados su deformación unitaria y el factor de seguridad.

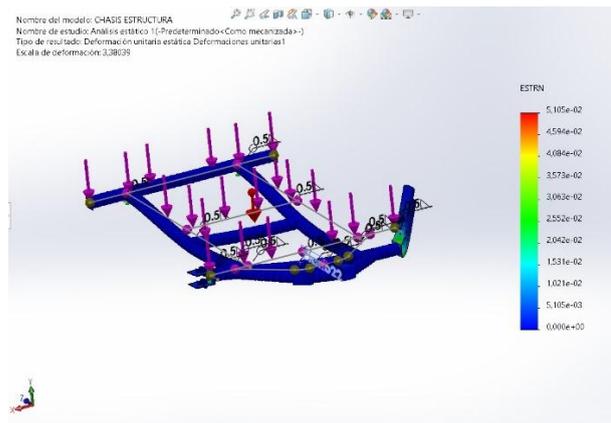


Figura 17-2: Análisis deformación unitaria

Fuente: (Barahona & Ríos, 2021)

La presión ejercida es alrededor de 200N teniendo como puntos fijos las esquinas en donde se asentarán las llantas tanto delanteras como traseras, observamos que la deformación es mucho más considerable con el material de plástico PVC, así mismo se verifica que la estructura se va a trasladar alrededor de 0,255mm.

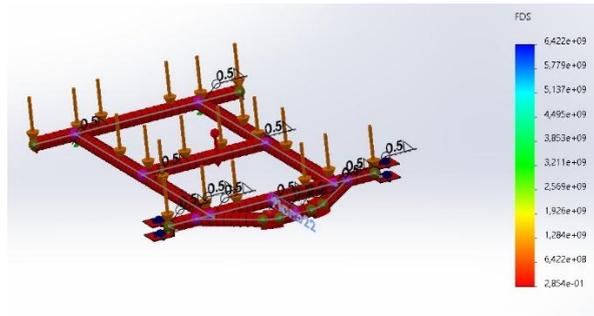


Figura 18-2: Factor de Seguridad Chasis

Fuente: (Barahona & Ríos, 2021)

Así mismo en la figura 18-2 observamos un factor de seguridad de 0.29 en donde va a estar por debajo de los estándares globales que enmarcan como 1 el factor mínimo para un chasis rígido, llevando a no ser una opción viable para soportar los pesos establecidos

2.5.4 Estudio comparativo de los materiales.

El análisis de cada uno de los materiales a utilizar en el prototipo se lo realizo para un estudio preliminar de construcción mediante el software para la plataforma. Por lo cual estos materiales estudiados fueron sometidos a una comparativa sobre la calidad del diseño y material a escoger, después de varios análisis estructurales (deformación, factor de seguridad) llegamos a concluir que el material ASTM A36 es la mejor opción, aunque por varios factores nos favorece el plástico PVC debido a que es más liviano, aligerando la carga del motor debido a su factor de seguridad por debajo del 1.

Es decir, la tendencia a un estiramiento o sobrepeso en donde el material se deforme no bastara tanto para que se rompa, a diferencia de ASTM A36 el cual con un módulo de elasticidad mayor y un factor de seguridad mayor a 1, soportara cargas más pesadas sin temer al rompimiento de este.

2.6. Selección del hardware que conforma el prototipo.

Para el correcto funcionamiento de nuestro prototipo, es necesario el uso de procesador para controlar todo el sistema tanto de procesamiento de visión artificial como cada una de las salidas de sus actuadores.

Los elementos utilizados en la plataforma son:

- Raspberry pi 2
- 1 servo motor 180°
- 1 cámara web
- 1 motor DC
- 1 batería de 6V
- 1 Puente H (LN298n)
- 1 Switch de encendido.
- Modem Nano NAS Wireless Router.
- USB WIFI (D-link DWA-125)
- Batería 6v /Alimentación autónoma

2.6.1. Raspberry Pi 2 Modelo B

Se decidió utilizar una Raspberry Pi en nuestra respuesta y no utilizar Arduino, ya que Raspberry Pi es una microcomputadora y Arduino es un microcontrolador. Por lo tanto, fue Raspberry Pi es más razonable para capturar y diseccionar imágenes, ya que tiene un poder de cálculo más notable y está más centrado en este tipo de disposición que el Arduino, que podría servir como ayuda más adelante. Se eligió la Raspberry Pi 2 Modelo B ya que anteriormente la tenía y no era importante conseguirla, a pesar de que ahora mismo está disponible la Raspberry Pi 3, que podría utilizarse más adelante para otra variante de nuestra respuesta.



Figura 15-2: Raspberry pi 2

Fuente: (Barahona & Ríos, 2021)

2.6.2. Cámara Web

En un principio se pensó en utilizar la Kinect360, pero como no se necesitaba de una gran resolución y a su vez los problemas que conllevan al tener un sistema de visión diferente a nuestras necesidades se optó como solución utilizar la cámara Web de la Raspberry pi II , debido a su amplia visión y versatilidad



Figura 20-2: Cámara Web

Fuente: (Barahona & Ríos, 2021)

La elección de la cámara fue definida porque tiene un precio asequible y además es compatible con la Raspberry Pi 2 como se puede observar en a Figura 20-2.

2.6.3. Modem Nano NAS Wireless Router

Para poder tener una conexión debemos tener un punto de acceso por eso escogimos un nano modem NAS Wireless Router para conectarnos a la Raspberry pi y tener una comunicación estable de acuerdo con el pin, como se puede observar en la figura 21-2



Figura 21-2: NAS Wireless

Fuente: (Barahona & Ríos, 2021)

2.6.4. Motor de corriente continua.

La decisión de utilizar un motor de CC para impulsar las ruedas traseras del vehículo se basó en que son motores didácticos y fáciles de controlar.



Figura 162-2: Motor de corriente continua

Fuente: (Barahona & Ríos, 2021)

Para esta situación, no se ha utilizado ningún control de velocidad para este motor; sin embargo, funciona al límite total, por lo que una mejora potencial es utilizar una señal de PWM para controlar la velocidad del motor como como.

2.6.5. Servomotor

Para mover la dirección tipo Ackermann del prototipo se decidió por utilizar el servomotor MG-995, por medio del servomotor y con una señal PWM, realizaremos el control más exhaustivo del movimiento de estas ruedas. No se utilizó un motor bidireccional para controlar la dirección tipo Ackermann por su movimiento muy brusco y poco preciso.



Figura 23-2: Servomotor

Fuente: (Barahona & Ríos, 2021)

2.6.6. Batería 6v /Alimentación autónoma

En el tema investigación describe un dispositivo para la validación autónoma por lo cual el prototipo no debe estar conectado a ningún tipo de cable o accesorio que le impida funcionar solo. Una de las soluciones más dables que proponemos es implementar un cargador portátil de 5mA, además una batería de litio para la alimentación impediendo del servomotor por su consumo excesivo de batería.



Figura 24-2: Alimentación de la fuente de 5V

Fuente: (Barahona & Ríos, 2021)

2.6.7. USB WIFI (D-link DWA-125)

Con el pin Wifi USB lo que logramos es que la Raspberry Pi 2 se asocie a una organización similar a la que está asociada la PC, que a través de VNC Viewer tendrá la opción de llegar al área de trabajo PÍXEL del clima de la Raspberry Pi 2 y este De manera que será posible modificar, actualizar, agregar y ejecutar el programa creado para el arreglo remoto.



Figura 25-2: USB Wifi

Fuente: (Barahona & Ríos, 2021)

Para dar a nuestra respuesta a la conexión de internet, era importante utilizar un Wifi USB, como podemos observare en la Figura 25-2.

2.7. Diagrama de conexión

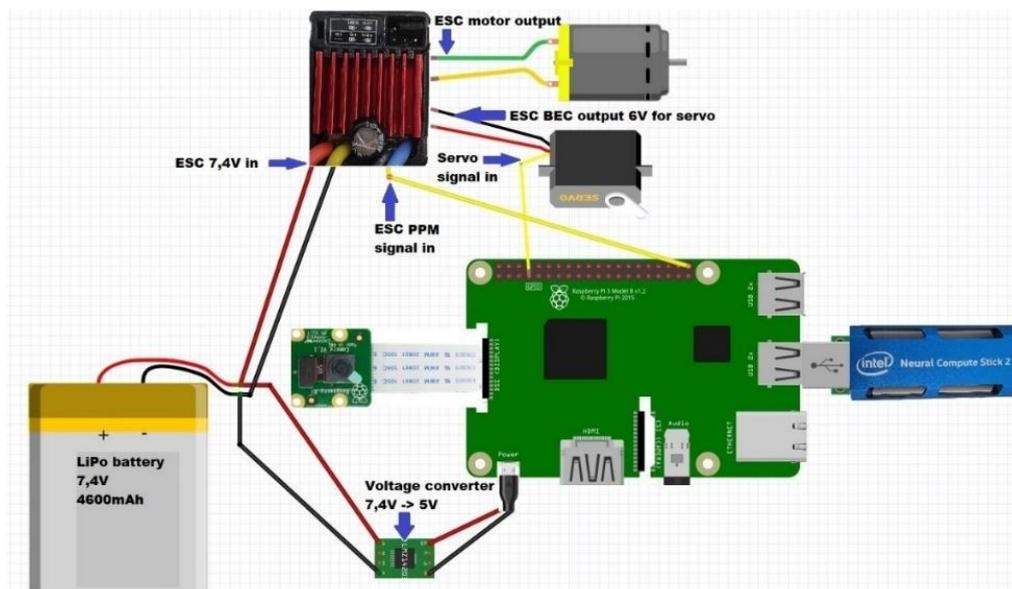


Figura 26-2: Diagrama de conexión

Fuente: (Barahona & Ríos, 2021)

A continuación, como se muestra en la figura 26-2 observamos el diagrama de conexión el cual consta de nuestro Raspberry pi II el cual estará conectado a servo y será el responsable de la dirección, así como el motor estará en la parte trasera junto con una caja reductora para mayor potencia, observamos también un puente H el cual dará paso de energía al motor cuando se lo ordene la raspberry.

La batería para usar será una de 7.4 V LiPO con un convertidor de voltaje, todo comandado por la cámara pi la cual receptara la información y la Raspberry la procesará para posteriormente enviar a los actuadores que funcionen de acuerdo con los algoritmos establecidos de navegación autónoma

2.8. Códigos de Navegación Autónoma

La implementación de los diferentes algoritmos en nuestro prototipo tipo Ackermann, los ejecutaremos en diferentes medios como son: en la computadora invalidando Python 3 y después ejecutar cada uno de los algoritmos en nuestra Raspberry para validar los algoritmos. Después de varias pruebas realizar una prueba de ruta dentro y fuera del campus Espoch.

2.8.1. Código STOP

Para la detección del código de señalética de STOP el cual detectara en función a la cámara pi una señal de STOP previo a un entrenamiento de algoritmos por medio de HAAR CASCADE para la detección de la señal y posteriormente al corte de energía del motor para el frenado.

2.8.1.1 Entrenamiento por computadora.

Previo a la ejecución del código en la Raspberry pi, se buscaron varias alternativas de algoritmos para verificar cual es el más apropiado para un buen procesamiento de nuestra detección de señal de stop.

2.8.1.2 Detector de objetos HAAR CASCADE

Para crear nuestro detector de la señal de stop, se utilizó varios pasos a seguir y conseguir nuestro HAAR CASCADE debido que todos estos pasos a continuación. Obtendremos un archivo con extensión (XML) que nos permitirá el reconocimiento de cada uno de los objetos.

- Crear una carpeta con imágenes positivas (fotografías de las señales de STOP).
- Crear una carpeta con imágenes negativas (fotografías del lugar donde no exista la señal de STOP).
- Utilización del programa CASCADE TRAINER GUI, para crear la extensión XML.

2.8.1.3 Creación del archivo XML.

Como podemos observar para realizar una recopilación de todo el procedimiento para el entrenamiento del HAAR CASCADE y obtener nuestra extensión XML.

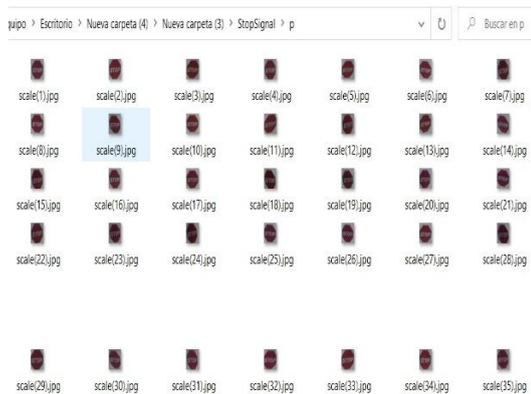


Figura 27-2: Fotografías de imágenes positivas.

Fuente: (Barahona & Ríos ,2021)

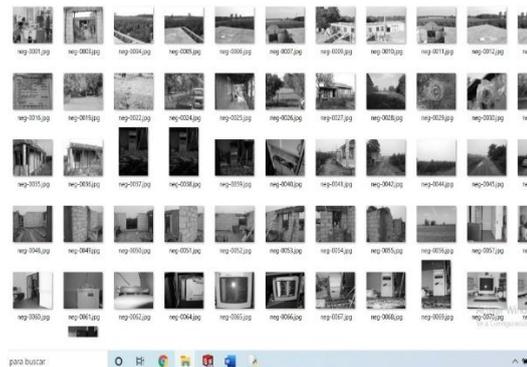


Figura 28-2: Fotografías de imágenes negativas.

Fuente: (Barahona & Ríos, 2021)

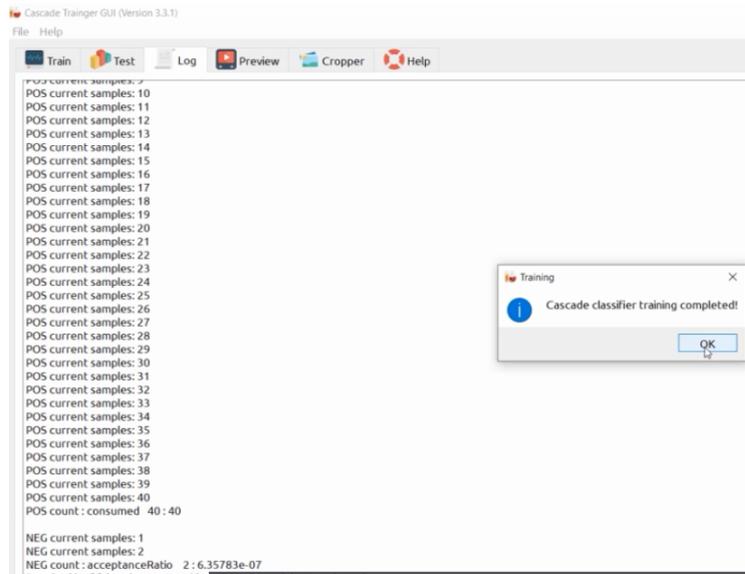


Figura 29-2: Entrenamiento de señal de Stop con un error ($6.35783e-07$)

Fuente: (Barahona & Ríos, 2020)

Como podemos observar en las figuras 29-2, seguimos los pasos para la creación de un detector de objetos (señal de stop).

2.8.1.4 Prueba en computadora

Para el proceso de prueba mediante la computadora se cómo como referencia el código de Anexo A, para el reconociendo mediante un HAAR CASCADE de la señal de tránsito Stop.

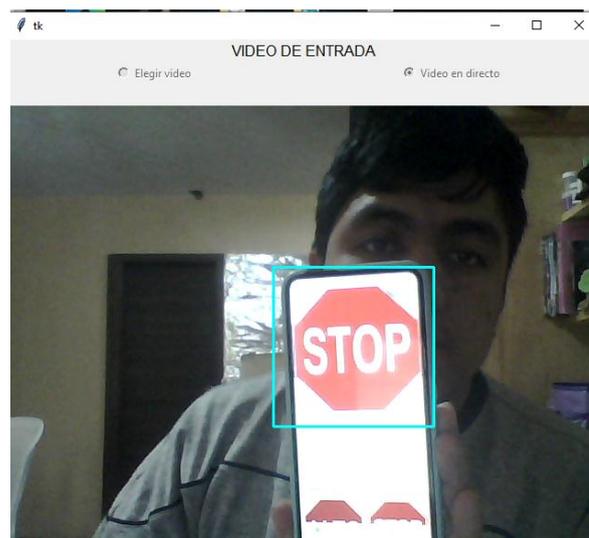


Figura 30-217: Detección de la señal de STOP por medio de la cámara de la computadora.

Fuente: (Barahona & Ríos ,2021)

2.8.1.5 Ejecución de códigos en el prototipo (Raspberry pi).

Previo al entrenamiento de la señal de Stop en nuestra computadora, procedemos a ejecutar el mismo código para verificar si el funcionamiento del algoritmo es óptimo tanto para computadores con procesadores de alta velocidad. Debido que nuestra Raspberry tiene un procesador lento pero muy efectivo para programar las salidas de control y verificar el tiempo de respuesta de detección del enfoque focal.



Figura 31-2: Prototipo detectando señal de Stop.

Fuente: (Barahona & Ríos, 2021)

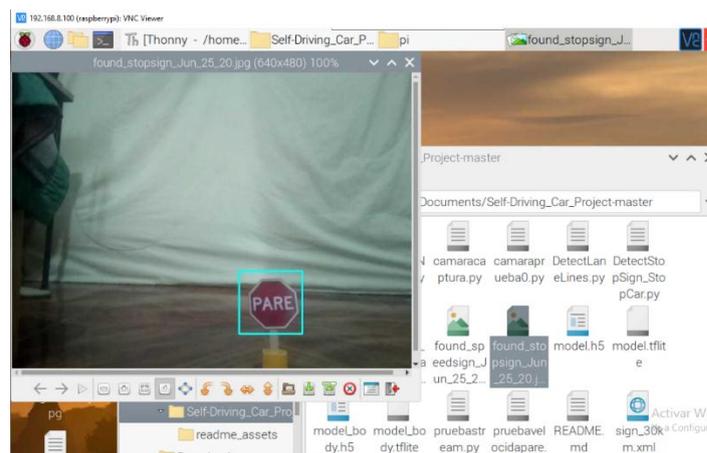


Figura 32-2: Señal de Stop Detectado por la Raspberry y una cámara web.

Fuente: (Barahona & Ríos. 2021)



Figura 33-2: Detección en tiempo real de señales de Tránsito con el algoritmo

Fuente: (Barahona & Ríos, 2021)

Como podemos verificar en la figura 33-2, se observa como el prototipo ya este acoplado con todos los elementos electrónicos y la detección de la señal por medio de nuestros elementos. Después de la detección adecuada se realizó pruebas de enfoque focal activando el motor y los servos que vayan corrigiendo la velocidad.

2.8.2. Código Detección de línea

Con las imágenes capturadas y obtenidas mediante una cámara web la misma que debe de estar instalada en el prototipo se procederá a trabajar adecuando el ambiente para que la detección sea correcta, esto se realiza mediante una conversión de la imagen RGB a escala de grises en donde posteriormente se aplicara un filtro Canny el cual resaltara las líneas de grupo de colores de dos tonalidades diferentes es decir el contorno de esas regiones, así como se muestra en la figura 34-2



Figura 34-2: Filtro Canny

Fuente: (Barahona & Ríos, 2021)

Posteriormente se aplica un filtro para determinar la región de interés que vamos a usar, en este caso usamos un triángulo con dimensiones $[(-100, 382), (660, 382), (280, 250)]$ el cual nos ayuda a obtener información dentro del polígono para cálculos de las posibles líneas a seguir y eliminando todo ruido existente afuera del mismo que no intervenga en lo absoluto

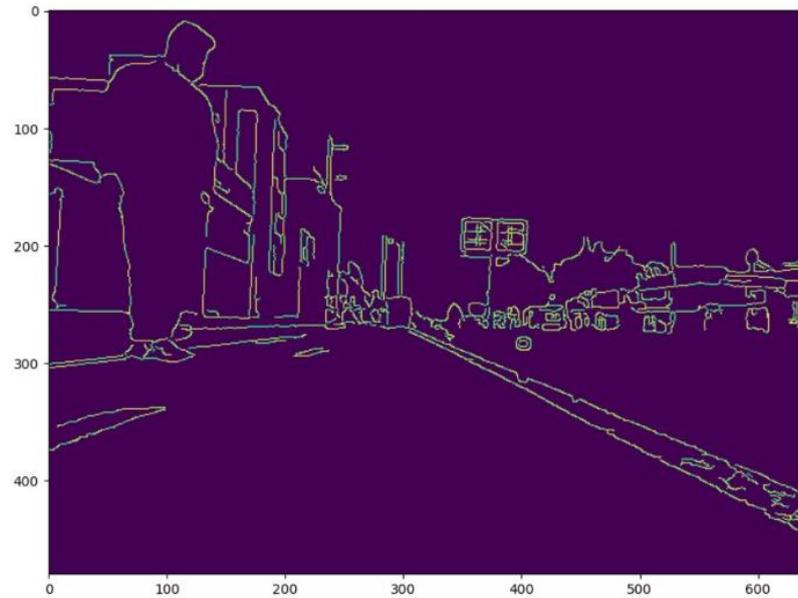


Figura 35-2: Filtro rango de interés

Fuente: (Barahona & Ríos, 2021)

Finalmente se aplicó el filtro de la transformada de Hough en donde se recoge información sobre el entorno ya preestablecido por los puntos dentro del triángulo ya mencionado y transforma en posibles líneas al conjunto de puntos dentro del mapa, dándonos, así como resultado la figura 35-2 marcándose las líneas y posterior mente optimizando el mismo código para alargar las líneas de detección y no haya errores de espacios vacíos



Figura 36-2: Transformada de Hough

Fuente: (Barahona & Ríos, 2021)

Como se observa en la figura 36-2, se tiene unas líneas ya marcadas la cual es la detección de la línea ya usando la transformada de Hough la cual mediante el algoritmo y su fórmula se calcula e interpreta la línea

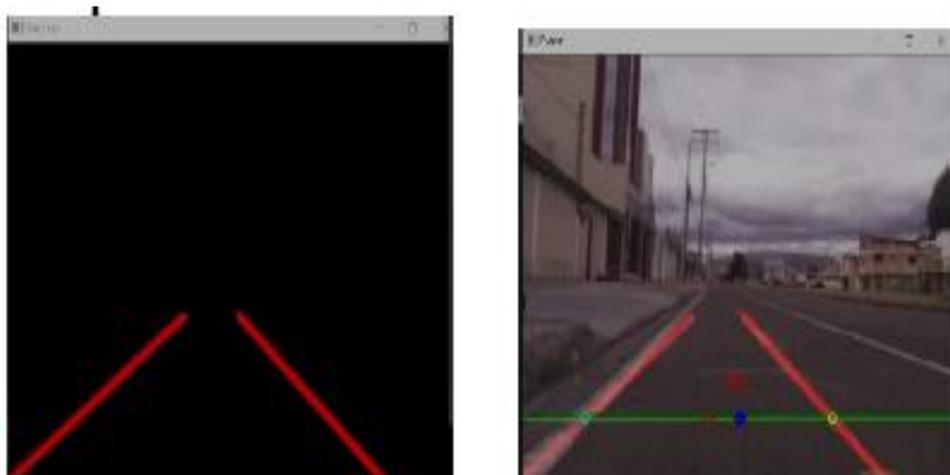


Figura 37-2: Detección en tiempo real del Carril

Fuente: (Barahona & Ríos, 2021)

En la figura 37-2, se tiene ya en si la detección del carril, en donde se saca el punto medio del frame como el punto medio de las líneas ya detectadas para una comparación y dar la señal al servo para su corrección

```

#####
### LEER DETECCION DE VIA ###
###          ###
### OBJETIVO          ###
### > conducir dentro  ###
### de las lineas hasta ###
### llegar a un punto B| ###
#####
from picamera.array import PiRGBArray
from picamera import PiCamera
import RPi.GPIO as GPIO
from time import sleep
import numpy as np
import cv2

# Inializations: #
#Camera fps/size
camera=PiCamera()
camera.resolution=(640,480)
camera.framerate=20
rawCapture=PiRGBArray(camera,size=(640,480))
sleep(0.1)
#Car PIN setups
enA=14
in1=15
in2=18
temp1=1
servo=22
angle=45
GPIO.setmode(GPIO.BCM)
GPIO.setup(enA,GPIO.OUT)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(servo,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
pwm_servo=GPIO.PWM(servo,100)
pwm_motor=GPIO.PWM(enA,1000)
pwm_servo.start(12.5)
pwm_motor.start(25)

def make_coordinate(img,line_parameters):
    slope,intercept=line_parameters
    y1=img.shape[0]
    y2=int(y1*(3/5))
    x1=int((y1-intercept)/slope)
    x2=int((y2-intercept)/slope)
    return x1,y1,x2,y2

```

Figura 38-2: Código detección de Línea

Fuente: (Barahona & Ríos, 2021)

Se observa el código de detección de línea el cual básicamente por medio de funciones algorítmicas como lo son el filtro Canny y la transformada de Hough corregirá los grados de giro de la dirección Ackermann comandada por el servo la cual recibirá ordenes de la Raspberry a través de la cámara pi uniendo la visión artificial y la navegación autónoma como lo indica la fig. 48

2.8.3. Código señal velocidad.

Este código tiene el mismo principio de funcionamiento de la señal de Stop explicada anteriormente con la diferencia que va a activar mayor o menor PWM del motor respectivamente sea su señal de tránsito.



Figura 39-2: Detección de señal de velocidad mediante la Raspberry.

Fuente: (Barahona & Ríos, 2021)



Figura 40-2: Detección se señal de velocidad con el ordenador

Fuente: (Barahona & Ríos, 2021)

Como podemos verificar en las imágenes y videos recolectados, al momento de la detección de la señal de velocidad el motor va a variar su velocidad dándole un impulso más alto de PWM

por lo cual se detiene siempre y cuando detecte la señal de Stop indicada anteriormente el proceso.

2.8.4. Código de semáforo

Para la detección del semáforo se implementó un método diferente para el procesador y reconcomiendo de imágenes, debido que al momento del funcionamiento le reconoce a las dos imágenes del semáforo (rojo y verde) como si fueran una sola señal de Stop o de Pare. Por lo cual para solucionar este error se empleó un reconocimiento de colores para mayor eficiencia del sistema como en la Figura 41-2 y Figura 42-2

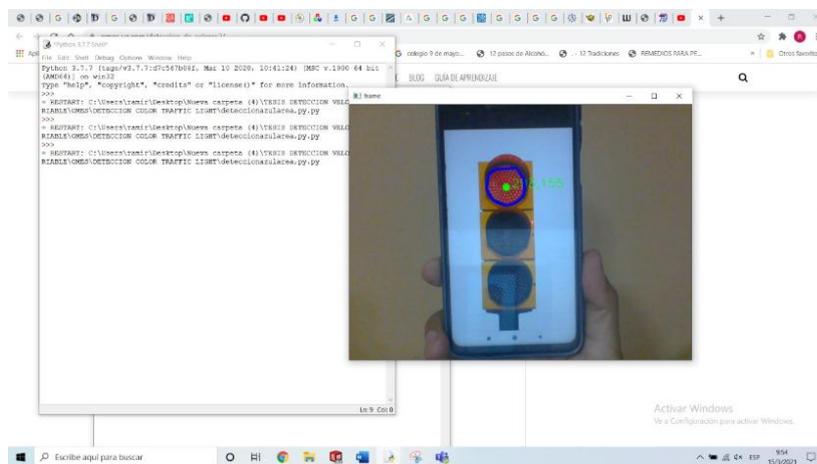


Figura 41-2: Reconociendo del semáforo (ROJO)

Fuente: (Barahona & Ríos, 2021)

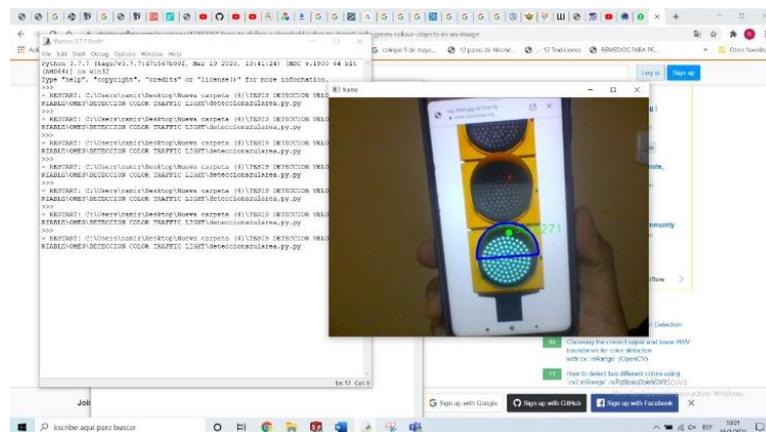


Figura 42-2: Reconociendo del semáforo (VERDE)

Fuente: (Barahona & Ríos, 2021)

Verificamos que tiene un buen reconocimiento de colores en los diferentes semaforos para tener una mejor procesamiento y respuesta del sistema. Con el semaforo rojo vamos a tener el mismo funcionamiento de la señal de Stop en cambio con el semaforo verde va a activar el motor a la velocidad mas baja y despues de varios minutos hacerle detectar a señal de velocidad que aumente al motor.

```
import numpy as np
import matplotlib.pyplot as plt
# change this to your caffe root dir
caffe_root = '/home/jfg/Documents/work/caffe'
import os
import sys
sys.path.insert(0, caffe_root + '/python')
import caffe
from google.protobuf import text_format
from caffe.proto import caffe_pb2
import cv2

def get_labelname(labelmap, labels):
    num_labels = len(labelmap.item)
    labelnames = []
    if type(labels) is not list:
        labels = [labels]
    for label in labels:
        found = False
        for i in range(0, num_labels):
            if label == labelmap.item[i].label:
                found = True
                labelnames.append(labelmap.item[i].display_name)
                break
        assert found
    return labelnames

def ssd_detection(detect_image, map_label, is_show=True):
    model_def = 'models/VGGNet/LISA/SSD_414x125/deploy.prototxt'
    model_weights = 'models/VGGNet/LISA/SSD_414x125/VGG_LISA_SSD_414x125_iter_120000.caffemodel'
    if not os.path.exists('./predict_result'):
        os.mkdir('./predict_result')
    if len(detect_image.split('/')) != 0:
        image_save_path = './predict_result/' + detect_image.split('/')[-1]
    else:
        image_save_path = './predict_result/' + detect_image

    net = caffe.Net(model_def,
                    model_weights,
                    caffe.TEST)
    transformer = caffe.io.Transformer({'data': net.blobs['data'].data.shape})
    transformer.set_transpose('data', (2, 0, 1))
    transformer.set_mean('data', np.array([104, 117, 123]))
    transformer.set_raw_scale('data', 255)
    transformer.set_channel_swap('data', (2, 1, 0))
```

Figura 43-2: Código de semáforo.

Fuente: (Barahona & Ríos, 2021)

Aquí observamos el algoritmo del tráfico de luz el cual va a detectar por medio del HAAR CASCADE la diferencia de colores por medio de un modelo entrenado preestablecido, diferenciando así los colores Verde, Amarillo y Rojo actuando de manera correlativa en función de estos.

CAPITULO III

3. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

3.1. Validación señal de stop.

Para la toma de datos se realizó 3 pruebas en distintos horarios del día, en una trayectoria de 5m para detectar cual es la distancia de detección del enfoque focal de la señal de STOP con un rango de aprobación a los 15 cm con un nivel de significancia del 0,05 de confiabilidad en cada uno de ellos horarios del día. Todos los cálculos se realizaron con el programa estadístico Minitab a partir de los datos de la tabla 1-3.

3.1.1 Mañana (9-10 am)

Ho: $\mu = 15$

Hi: $\mu \neq 15$

$\alpha = 0.05$

Tabla 1-3: Datos del enfoque focal a las (9-10)

PRUEBAS	ENFOQUE FOCAL (CM)	PRUEBAS	ENFOQUE FOCAL (CM)
1	16	16	14
2	14	17	16
3	16	18	14
4	15	19	15
5	14	20	15
6	15	21	15
7	14	22	14
8	15	23	16
9	14	24	14
10	14	25	14
11	15	26	16
12	14	27	15
13	16	28	16
14	16	29	15
15	14	30	15

Nota: Esta tabla 1-3 muestra la distancia del enfoque focal detecta la señal de stop en cada una de las pruebas.

A partir de los datos obtenidos en la Tabla 1-3. Podemos calcular la desviación estándar para poder detectar mediante el **T-STUDENT**, si podemos validar el algoritmo de stop.

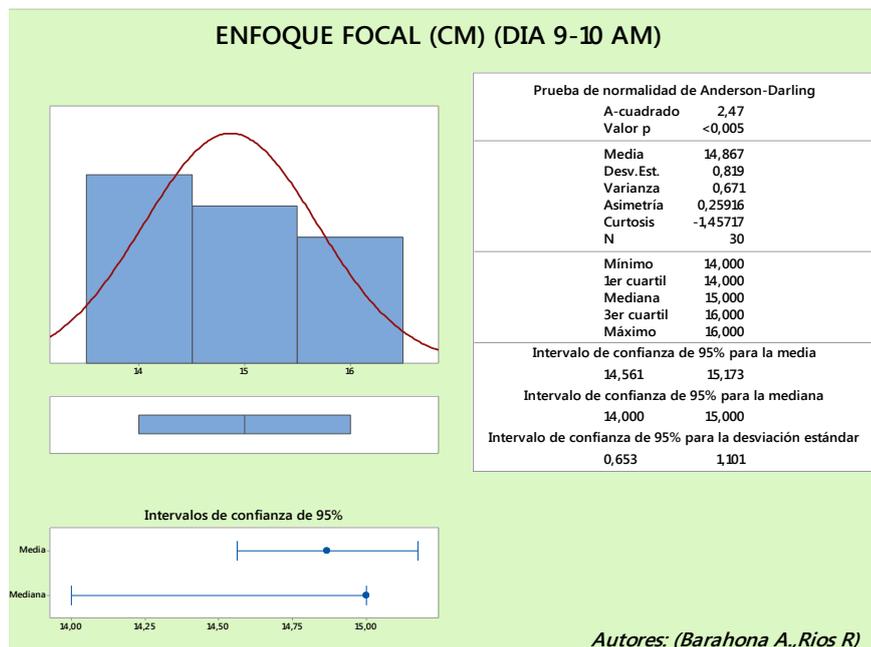


Gráfico 1-3: Cálculos estadísticos del enfoque focal (9-10 am)

Fuente: (Barahona & Ríos, 2021)

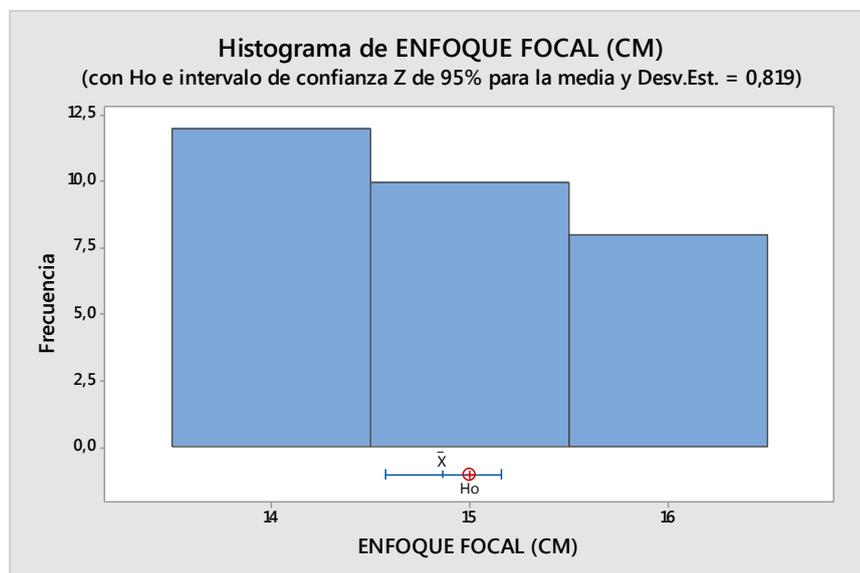


Gráfico 2-3: Histograma de enfoque focal.

Fuente: (Barahona & Ríos, 2021)

En el Grafico 1-3. Tenemos un análisis estadístico de la detección de la señalética de stop, en el mañana en un horario de (9-10 am). Por lo cual tenemos *una desviación estándar de 0.819*, con este valor calculamos la **T-STUDEN** con una confiabilidad del 0,05 para validar el algoritmo.

Z de una muestra: ENFOQUE FOCAL (CM)

Prueba de $\mu = 15$ vs. $\neq 15$

La desviación estándar supuesta = 0,819

Variable	N	Media	Desv.Est.	Error estándar de la media	IC de 95%	Z	P
ENFOQUE FOCAL (CM)	30	14,867	0,819	0,150	(14,574; 15,160)	-0,89	0,373

Figura 1-3: Z en una muestra: enfoque focal (9-10 am)

Fuente: (Barahona & Ríos, 2021)

En la Figura 1-3 calculamos el valor de $P = 0,373$, el periodo de tiempo de (9-10 am) con un nivel de significancia del $\alpha = 0.05$. Como verificamos que $P > \alpha$, podemos concluir que nuestra H_0 es aceptable, el Algoritmo de Stop es aceptable en el horario de (9-10 am).

3.1.2 Medio Dia (13-14 Pm)

$H_0: \mu = 15$

$H_1: \mu \neq 15$

$\alpha = 0.05$

Tabla 2-3: Datos del enfoque focal a las (13-14 pm)

PRUEBAS	ENFOQUE FOCAL (13-14 PM)	PRUEBAS	ENFOQUE FOCAL (13-14 PM)
1	16	16	17
2	15	17	17
3	12	18	14
4	16	19	14
5	12	20	14
6	16	21	12
7	14	22	15
8	14	23	17
9	14	24	12
10	15	25	15
11	15	26	15
12	16	27	13
13	17	28	14
14	14	29	15
15	15	30	12

Nota: Esta tabla 2-3 muestra la distancia del enfoque focal detecta la señal de stop en cada una de las pruebas al medio día.

A partir de los datos obtenidos en la Tabla 2-3. Podemos calcular la desviación estándar para poder detectar mediante el **T-STUDENT**, si podemos validar el algoritmo de stop.

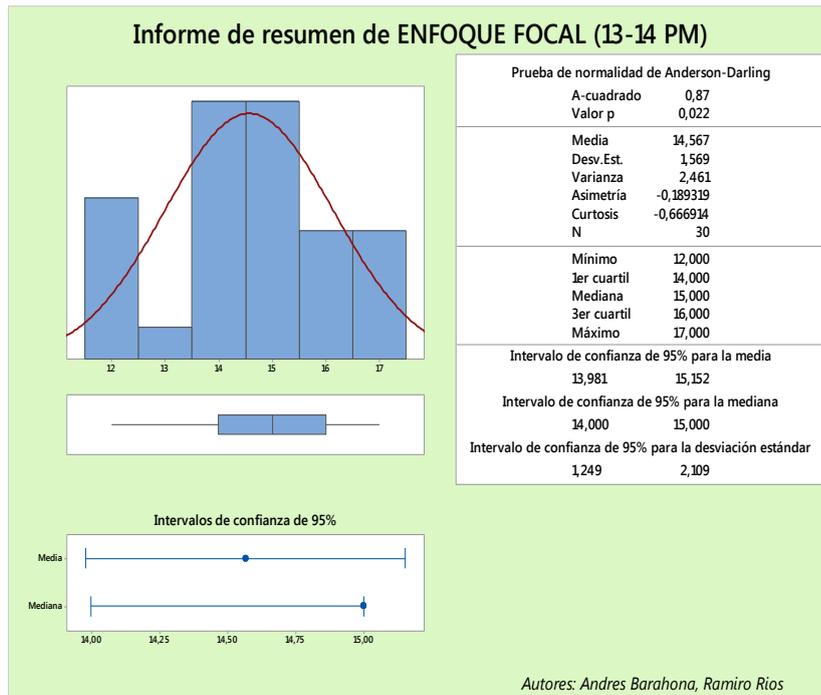


Gráfico 3-3: Cálculos estadísticos del enfoque focal (13-14 pm)

Fuente: (Barahona & Ríos, 2021)

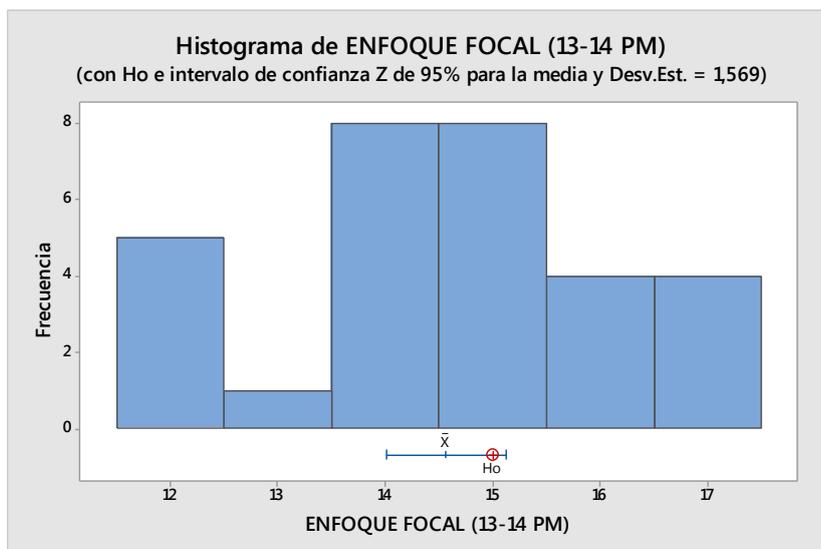


Gráfico 4-3: Histograma de enfoque focal.

Fuente: (Barahona & Ríos, 2021)

En el Grafico 3-3. Tenemos un análisis estadístico de la detección de la señalética de stop, en el mañana en un horario de (9-10 am). Por lo cual tenemos *una desviación estándar de 1,569*, con este valor calculamos la **T-STUDEN** con una confiabilidad del 0,05 para validar el algoritmo.

Z de una muestra: ENFOQUE FOCAL (13-14 PM)

Prueba de $\mu = 15$ vs. $\neq 15$

La desviación estándar supuesta = 1,569

Variable	N	Media	Desv.Est.	Error estándar de la media	IC de 95%	Z	P
ENFOQUE FOCAL (13-14 PM)	30	14,567	1,569	0,286	(14,005; 15,128)	-1,51	0,130

Figura 2-3: Z en una muestra: enfoque focal (13-14 pm)

Fuente: (Barahona & Ríos, 2021)

En la Figura 2-3 calculamos el valor de $P = 0.130$, el periodo de tiempo de (13-14 pm) con un nivel de significancia del $\alpha = 0.05$. Como verificamos que $P > \alpha$, podemos concluir que nuestra H_0 es aceptable, el Algoritmo de Stop es aceptable en el horario de (13-14 am).

3.1.3 Tarde (17-18 pm)

$H_0: \mu = 15$

$H_1: \mu \neq 15$

$\alpha = 0.05$

Tabla 3-3: Datos del enfoque focal a las (13-14 pm)

PRUEBAS	ENFOQUE FOCAL (17- 18)	PRUEBAS	ENFOQUE FOCAL (17- 18)
1	17	16	17
2	15	17	15
3	14	18	16
4	15	19	15
5	17	20	17
6	17	21	15
7	14	22	16
8	14	23	14
9	16	24	17
10	15	25	17
11	14	26	15
12	16	27	17
13	15	28	16
14	14	29	15
15	15	30	17

Nota: Esta tabla 3-3 muestra la distancia del enfoque focal detecta la señal de stop en cada una de las pruebas al medio día.

A partir de los datos obtenidos en la Tabla 3-3. Podemos calcular la desviación estándar para poder detectar mediante el **T-STUDENT**, si podemos validar el algoritmo de stop.

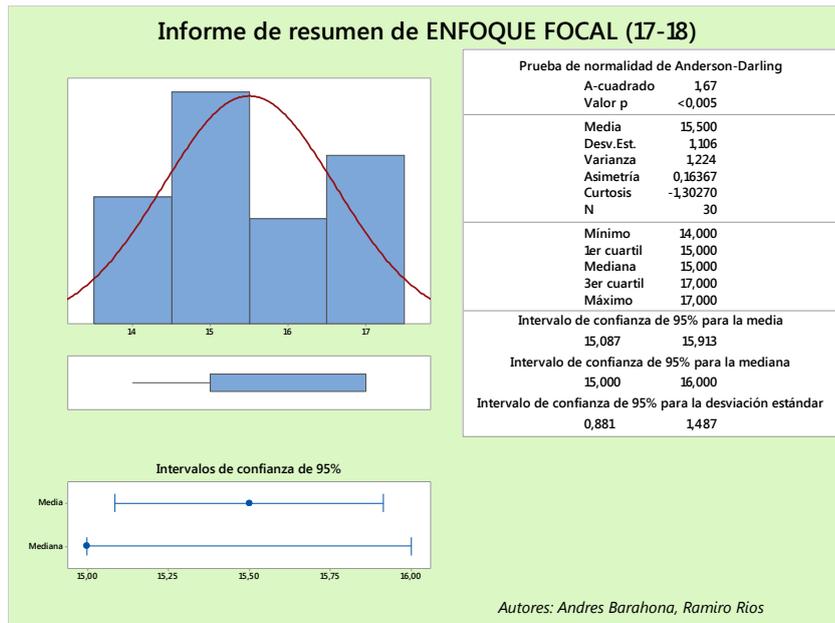


Gráfico 5-3: Cálculos estadísticos del enfoque focal (17-18 pm)

Fuente: (Barahona & Ríos, 2021)

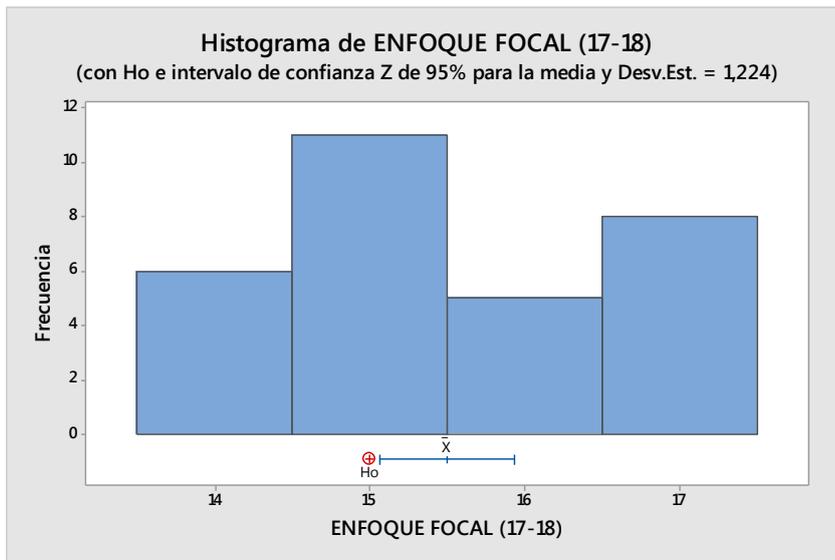


Gráfico 6-3: Histograma de enfoque focal (17-18)

Fuente: (Barahona & Ríos, 2021)

En la Grafico 5-3. Tenemos un análisis estadístico de la detección de la señalética de stop, en la tarde (17-18 pm). Por lo cual tenemos *una desviación estándar de 1,224*, con este valor calculamos la **T-STUDEN** con una confiabilidad del 0,05 para validar el algoritmo.

Prueba de $\mu = 15$ vs. $\neq 15$
 La desviación estándar supuesta = 1,224

Variable	N	Media	Desv.Est.	Error estándar de la media	IC de 95%	Z	P
ENFOQUE FOCAL (17-18)	30	15,500	1,106	0,223	(15,062; 15,938)	2,24	0,025

Figura 3-318: Z en una muestra: enfoque focal (17-18 pm)

Fuente: (Barahona & Ríos, 2021)

En la Figura 3-3 calculamos el valor de $P = 0.025$, el periodo de tiempo de (13-14 pm) con un nivel de significancia del $\alpha = 0.05$. Como verificamos que la H_0 es rechazada $P < \alpha$, podemos concluir que el Algoritmo de Stop NO es aceptable en el rango (17-18 pm).

3.1.4 Análisis de varianza- señal de stop

$$H_0: \mu_1 = \mu_2 = \mu_3$$

$$H_i: \mu_1 \neq \mu_2 \neq \mu_3$$

$$\alpha = 0.05$$

Tabla 4-3: Datos del enfoque focal a diferentes horas del día mediante condiciones meteorológicas favorables

PRUEBAS	ENFOQUE FOCAL (CM)	ENFOQUE FOCAL (13- 14 PM)	ENFOQUE FOCAL (17-18)
1	16	16	17
2	14	15	15
3	16	12	14
4	15	16	15
5	14	12	17
6	15	16	17
7	14	14	14
8	15	14	14
9	14	14	16
10	14	15	15
11	15	15	14
12	14	16	16
13	16	17	15
14	16	14	14
15	14	15	15
16	14	17	17
17	16	17	15
18	14	14	16
19	15	14	15
20	15	14	17
21	15	12	15
22	14	15	16
23	16	17	14
24	14	12	17
25	14	15	17
26	16	15	15
27	15	13	17
28	16	14	16
29	15	15	15
30	15	12	17

Nota: Esta tabla 4-3 muestra la distancia del enfoque focal detecta la señal de stop en cada una de las pruebas al medio día.

A partir de los datos obtenidos en la Tabla 4-3. Realizaremos un análisis de varianza de las desviaciones estándar de los tres casos de condiciones del día (MAÑANA, MEDIO DIA Y TARDE). Podemos calcular la desviación estándar para poder detectar mediante el **T-STUDENT**, si podemos validar el algoritmo de stop. Por lo cual utilizaremos el método ANOVA para la validación total del algoritmo STOP.

Mediante un análisis de los datos estadísticos podemos verificar en el rango promedio esta cada una de nuestras distancias focales en cada uno de los casos y tener un rango de precisión, debido que cada uno de ellos varia de acuerdo a varias circunstancias del entorno o procesamiento de datos y hasta batería insuficiente en el sistema.

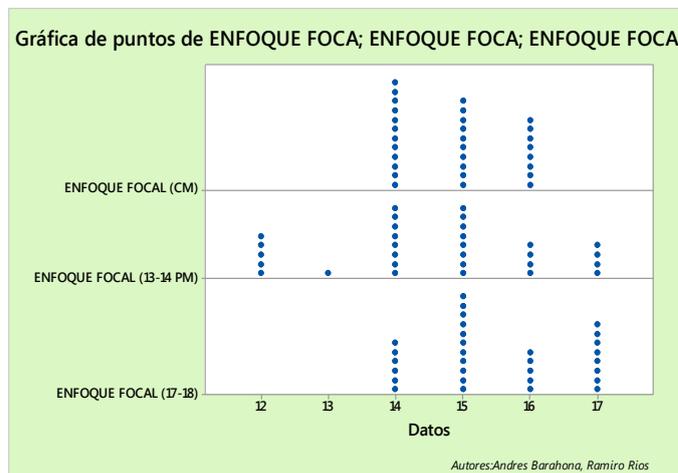


Gráfico 7-3: Grafico de puntos, algoritmo de stop

Fuente: (Barahona & Ríos, 2021)

En el Grafico 7-3. podemos observar como el algoritmo de STOP trabaja con normalidad en las mayoría de los caso, pese podemos observar que al MEDIO DIA a partir de las 13 pm tenemos varios errores causados por una mala calidad de procesamiento de imágenes, por demasiada iluminación, en otro caso en la Tarde podemos verificar que el procesamiento es bueno ya que la diferencia es muy baja en el rango de error pero con respecto al análisis estadístico de la desviación estándar no va a ser factible por la mucha diferencia de datos.

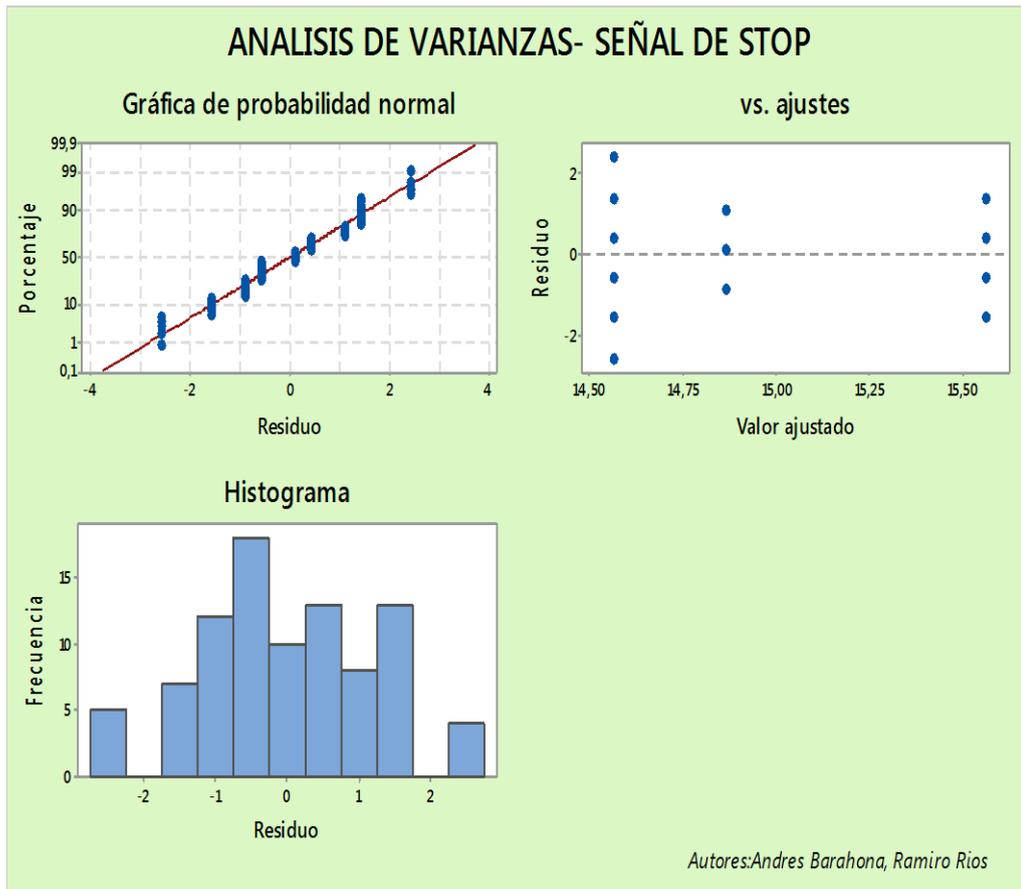


Gráfico 8-3: Análisis de Varianzas

Fuente: (Barahona & Ríos, 2021)

Según el Grafico 8-3, podemos observar que los datos con respecto a cada una de las gráficas son muy precisas, debido que el error es muy corto. Grafica de probabilidad normal podemos verificar que los datos no se alejan a los de recta numérica, eso nos menciona que es muy aceptable el algoritmo en las diferentes situaciones de funcionamiento. La grafica de valor ajustado, al igual que las respuestas anteriores podemos observar que su rango de error es muy corto además su residuo está en un rango muy aceptable por lo cual sus desviaciones estándar en esta grafica validan el algoritmo de STOP.

3.2. Detección de carriles

3.2.1. Entornos para pruebas del prototipo.

El servomotor es el que va a controlar la dirección tipo Ackermann, por lo cual se tendrá que sacar distintos escenarios para poder tener un resultado más preciso de la detección de carriles. Ya sea al exterior, interior de la ESPOCH y una curva al ingresar a la ESPOCH para poder apreciar mejor el funcionamiento del servomotor.



Figura 4-3: Interior de la Espoch

Fuente: (Barahona & Ríos, 2021)

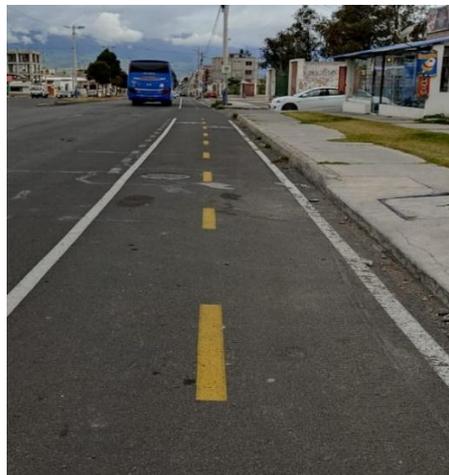


Figura 5-3: Exterior de la Espoch

Fuente: (Barahona & Ríos, 2021)

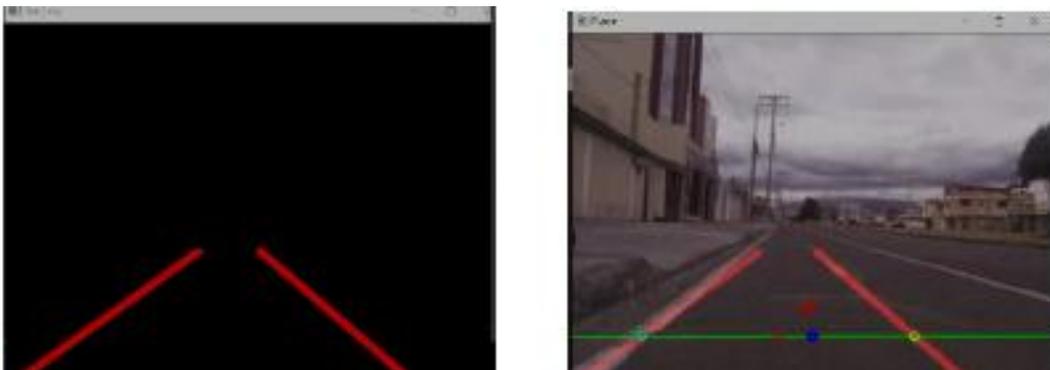


Figura 6-3: Detección de Carriles

Fuente: (Barahona & Ríos, 2021)

En las Figuras 6-3, podemos observar los distintos escenarios donde se tomó como referencia puntos estratégicos para la corrección constante del servomotor al pasar por su trayectoria.

3.2.2. *Detección de Carriles y Corrección de giro.*

Para el funcionamiento correcto de la detección de los carriles se utiliza varios filtros que procesen y eliminen el entorno como se observa en la Figura 7-3, para que los elementos que no sean las líneas de carril deben eliminarse de la imagen debido que una mala lectura provocaría un procesamiento erróneo de la imagen por medio de la cámara.

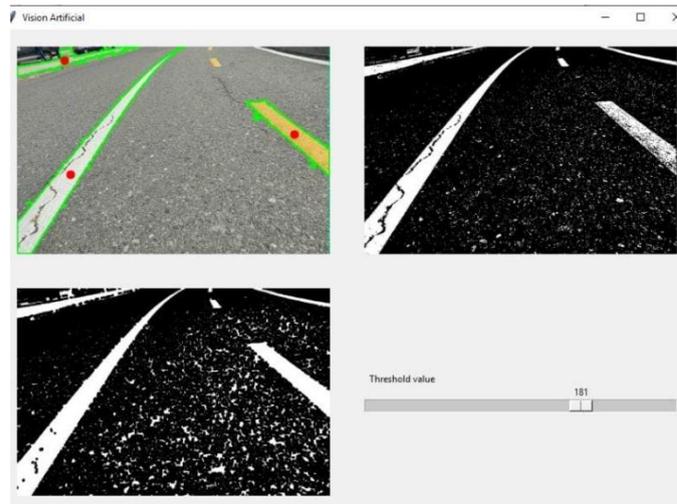


Figura 7-3: Filtros en la detección de carriles

Fuente: (BARAHONA & RÍOS, 2021)

En la figura 7-3 de nuestra observamos la transformada de HOUGH como podemos ver las líneas del carril están encerrada con un contorno verde, para poder localizar las ares de las líneas de los carriles y visualizar cada una de las líneas según vaya avanzando el vehículo. Así mismo las otras dos imágenes se implementan varios filtros como son CANNY (eliminan la parte superior de la imagen como son nube, paisaje alrededor) y el de contraste debido que tenemos que variar la iluminación para eliminar los objetos que perjudiquen el funcionamiento del servomotor.

3.2.3. *Corrección giro de dirección.*

3.2.3.1 *Escenario A*

Al momento del funcionamiento de nuestro prototipo, el servomotor debe variar la dirección de las ruedas con respecto hacia donde se va direccionado nuestro vehículo autónomo. Por lo cual tomamos varios escenarios de funcionamiento con respecto a los ángulos dados por nuestra

visión arterial siguiendo los diferentes filtros. Para validar nuestro algoritmo de dirección (detección de carriles).

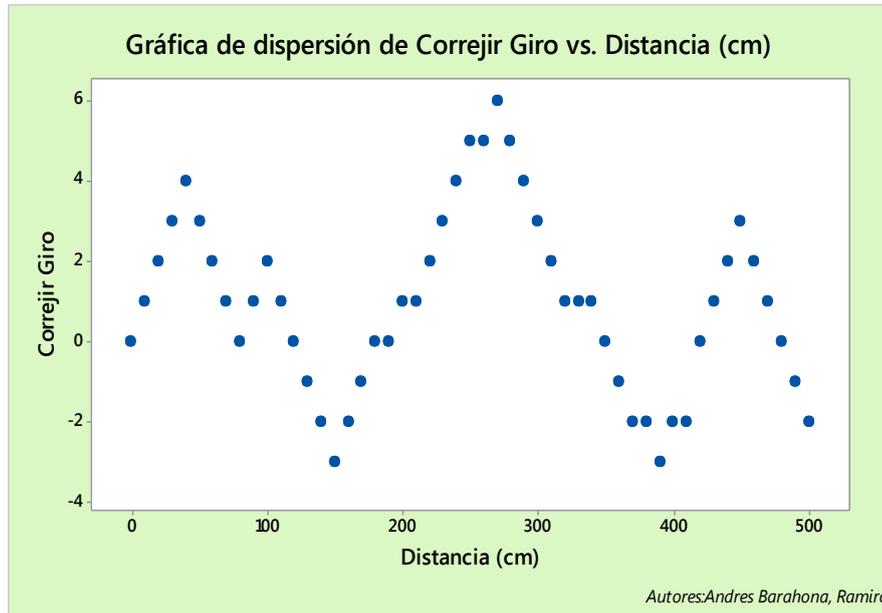


Gráfico 9-3: Comportamiento del giro de corrección.

Fuente: (Barahona & Ríos, 2021)

En el Grafico 9-3 podemos observar cómo es el comportamiento del servomotor al momento que va corrigiendo el giro después de 5 metros de trayecto, verificamos que a pesar que es una línea recta la trayectoria según la imagen del interior de la ESPOCH, este tiende a girar hacia la izquierda debido que tenemos grados de corrección positivos, por lo cual este nos menciona que tenemos una ligera desviación de la vía detectada por la visión artificial del prototipo, además puede ser una mañana alineación de las llantas del prototipo.

3.2.3.2 Escenario B

En trayectoria se tomó como referencia el exterior de la Espoch, para ser más exactos la Av. Canónigo Ramos, debido que la señalización de la ciclovía es mucho más eficiente que dentro de la ESPOCH, en donde se observa el comportamiento del giro de cada una de las posiciones del servomotor.

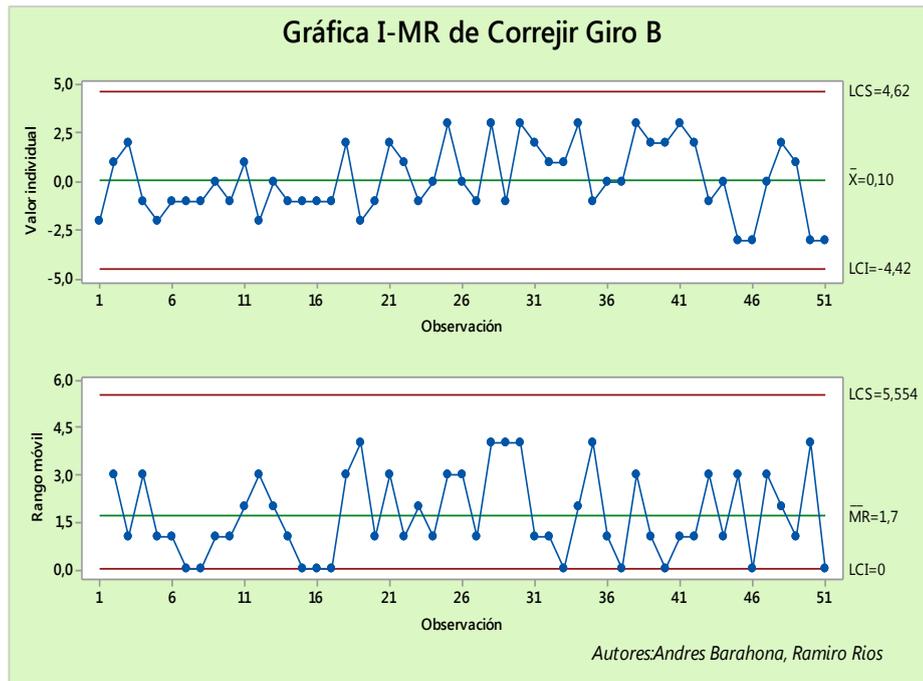


Gráfico 10-3: Comportamiento del giro de corrección B.

Fuente: (Barahona & Ríos, 2021)

En el Grafico 10-3, se puede verificar que el giro del prototipo tiende hacia la izquierda debido a que se toma una mayor concentración de ángulos positivos para poder corregir su giro y poder estar en el centro de la ciclovía.

3.2.3.3 Escenario C

En esta trayectoria es una de las complicadas debido que se toma como referencia una curva muy pronunciada para poder detectar si realiza un correcto centrado del prototipo en el carril por lo cual tenemos que implementar el graficas de control MR debido que para controlar el correcto funcionamiento del prototipo.

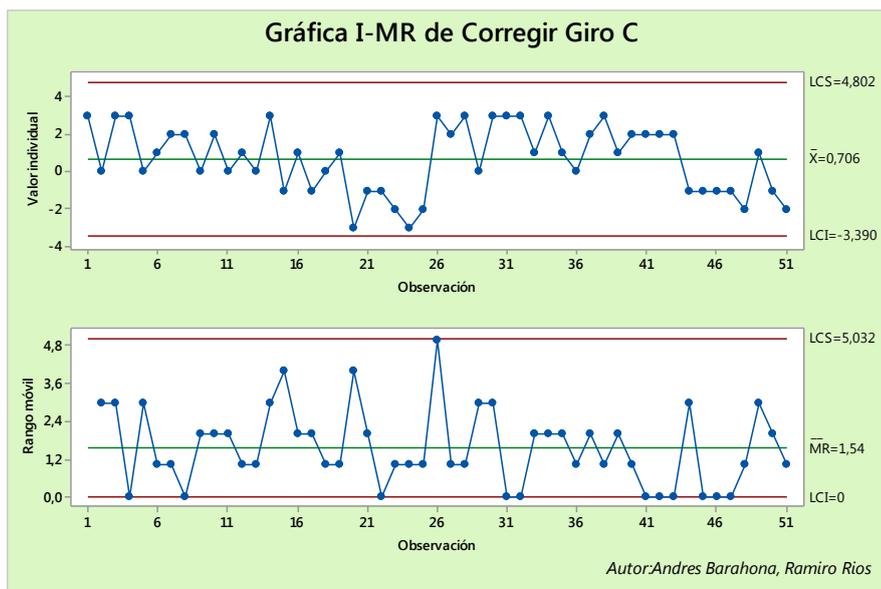


Gráfico 11-3: Comportamiento del giro de corrección C.

Fuente: (Barahona & Ríos, 2021)

En el Grafico 11-3, como se observa la curva es muy pronunciada nos dio como resultado un desvió de la carretera en los 2 metro y medio por lo cual se tuvo que volverle a introducir hacia la pista, dando como resultado un funcionamiento adecuado del prototipo.

3.3 Discusión

Una de las principales propuestas para la ejecución del tema se titulación, se basó en la idea de implementar un del prototipo tipo Ackermann para la carrera de ingeniería automotriz, debido que en todo el país no se desarrolla plataformas para la ejecución de algoritmos de navegación autónoma y principalmente en la ciudad de Riobamba.

Según una de las propuestas de temas de titulación en otros países mencionan que:

“Como bien se puede observar, este vehículo no deja de ser un mero prototipo, que en un futuro y siguiendo las modificaciones y mejoras que se describirán a continuación, puede llegar a ser un prototipo competitivo en el mercado de la automatización en los vehículos, y demostrando que la visión artificial puede ser nuestro futuro aliado.” (ÁVILA, E.2019)

Tomando como base los resultados obtenidos en pruebas realizadas se trato de superar estas dificultades mencionadas por el autor y llegar a la propuesta establecida para crear un prototipo final para desarrollo de conocimientos en el área de visión artificial y navegación autónoma en nuestro país.



Figura 8-3: Prototipo Ackerman, cámara LIDAR.

Fuente: (Merino, T. 2018)

El prototipo Ackermann usado en las pruebas de detección de señalética en el campus ESPOCH presenta evidencias visuales de acuerdo con la luminosidad del ambiente que posee la ciudad de Riobamba, con datos realizados en un periodo de tiempo en condiciones ideales de (9-13) horas del día obteniendo un porcentaje con un resultado favorable mayor al 80% de efectividad en todas las pruebas .De acuerdo a los resultados vamos a tener varias respuestas según factores como el clima y tipo de luz que tenga la ciudad para el procesamiento de imagen es decir la velocidad de respuesta para reconocimiento de algoritmos por medio de visión artificial.

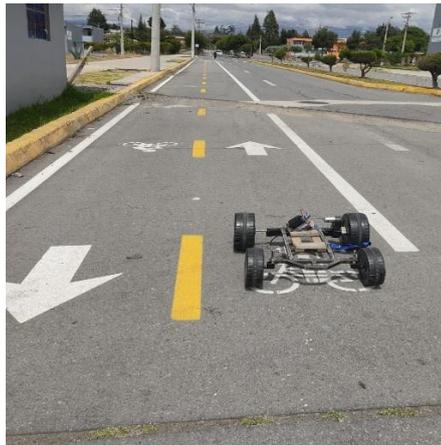


Figura 9-3: Prototipo tipo Ackermann en el campus ESPOCH

Fuente: (Barahona & Ríos, 2021)



Figura 10-3: Pruebas en el Campus ESPOCH

Fuente: (Barahona & Ríos, 2021)

En la figura 9-3 y en la figura 10-3, se observa varias inconsistencias al momento de ejecutar los algoritmos como son: la mala señalización de las vías, lactancia en el procesamiento de datos de acuerdo con el horario del día, y una mala disposición de los elementos del prototipo para la detección de imágenes, después de un tiempo total de ensayos se logró solucionar varios factores deficientes.

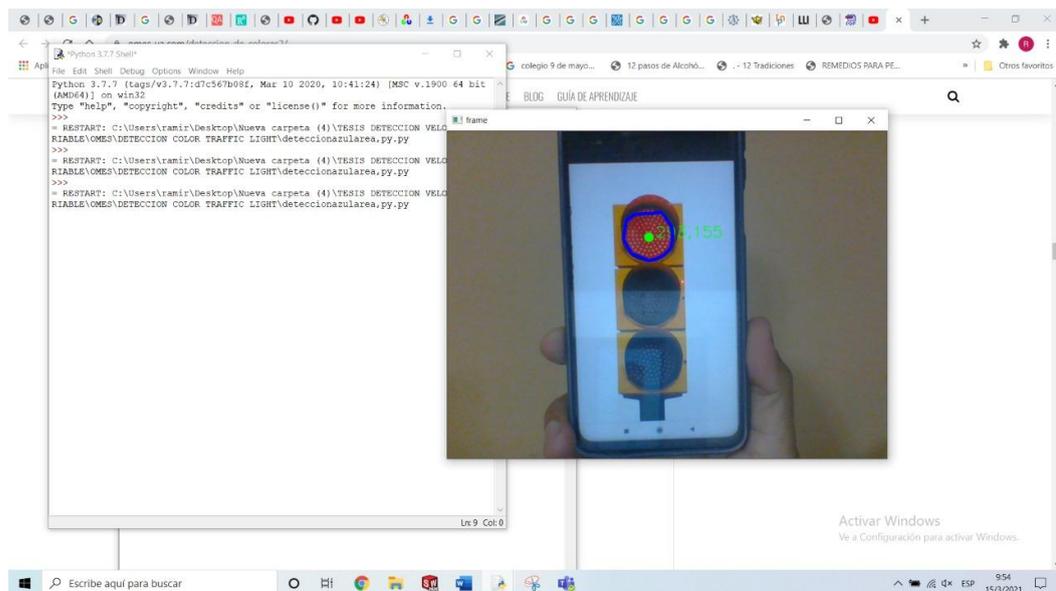


Figura 11-3: Detección del semáforo

Fuente: (Barahona & Ríos, 2021)

Con respecto a la detección del semáforo por medio de tonalidad de Colores para el prototipo Ackermann se realizó una correcta detección con un margen de error máximo del 3% el cual está dentro de los estándares de publicaciones como, Ibarra. M, Mendieta. V, Escalera H, Olmeda A. Para lo cual en sus publicaciones tiene un margen aproximado promedio del 1% al 6% en donde se identifica el semáforo mas no por su posición o foco encendido, sino por su color en ese instante procediendo al envío de datos desde la cámara web a la Raspberry procesando los datos y enviando señales para el arranque o stop del motor, los márgenes de colores en HSV que se uso fue para el verde mínimo $[36,0,0]$ y verde máximo $[86,255,255]$, con respecto al color Rojo mínimo $[0,100,20]$ y rojo máximo $[5,255,255]$. Valores los cuales están dentro del rango HSV y usados en aproximaciones cercanas en estudios de segmentación de colores y detección de estos publicados por P. Gil, F. Torres. G. Ortiz. Perez C. Valores referenciales los cuales sirvió para posterior modificación de alcances con respecto a la electrónica a usar.

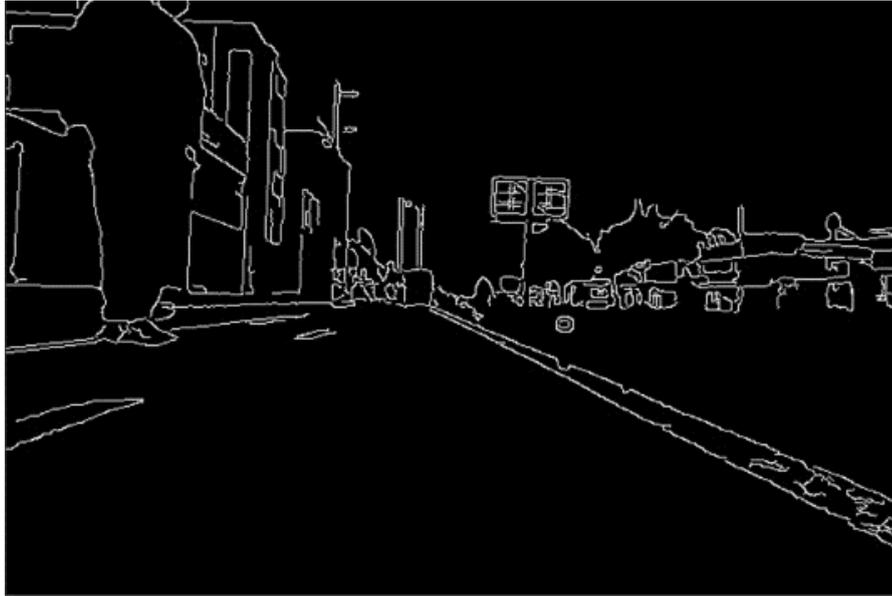


Figura 12-3: Filtros de visión artificial

Fuente: (Barahona & Ríos, 2021)

En lo que respecta a la detección del carril se utilizó un modelamiento Binario para el desarrollo de este, procediendo al uso de los filtros más comunes y porcentualmente con mayor efectividad usados en publicaciones por (CORREA & DIAZ, 2019), (MEDINA & LEAL 2010). En donde se desarrolla la codificación del algoritmo de navegación autónoma con bases en filtros tales como Canny, y Transformada de Hough, eliminando ruidos y pronunciando líneas para el seguimiento correcto obteniendo un resultado mayor 95% en el área propuesta, dándonos una fiabilidad al usar los mismo ya que en artículos similares rondaba entre los 88%-99% según estudios por (RUEDA, 2017), (MEDINA & LEAL, 2010) donde uno de los factores importantes era la meteorología de medio ambiente.

CONCLUSIONES

- Se presentó una plataforma móvil tipo Ackermann la cual esta acoplada con funciones para la validación de algoritmos de navegación autónoma por medio de visión artificial, sistema con orientación a aplicaciones de reconocimiento de carretera, detección óptima de la señal de tránsito “PARE”, detección de señalética de velocidad (30) y detección del semáforo por medio de reconocimiento de colores a través del reconociendo en cascada HAAR CASCADE.
- Se realizó una plataforma móvil óptima para su funcionamiento dentro de un carril de la ciclovía con relación a un vehículo promedio en la ciudad de Riobamba. Después de varios análisis para seleccionar: el material y según el análisis estructural (deformación unitaria y factor de seguridad), llegamos a la conclusión que el material óptimo para la construcción es el acero ASTM A36 por poseer un factor de seguridad alto de 1.4 comparado con el estándar que debe ser superior a 1 y una deformación unitaria de 0.000307mm con respecto al peso de un niño de 8 años resistiendo su deformación unitaria en la simulación.
- El análisis para la selección de cada uno de los algoritmos tanto de: STOP, VELOCIDAD, DETECCION DE CARRIELES y SEMAFORO. Evidencia un procesamiento de datos óptimo y fiable con un porcentaje mayor al 90% de funcionamiento dentro de un análisis por visión computacional.
- La implementación para seleccionar los algoritmos de navegación autónoma en el prototipo se hizo por medio de pruebas estadísticas tales como el T-Student y recolección de datos en tiempo real validando así cada uno de los algoritmos específicos de navegación autónoma con un margen de error mínimo dentro de los límites propuestos en relación con estudios acerca del control de Raspberry pi en visión artificial.
- La obtención de resultados para la detección de la señalética varía de acuerdo: al color, tipo de señalética y tipo de clima que se posea en el momento del procesamiento de esta, ya que el enfoque de la cámara y el tono de los colores varían. La distancia del enfoque focal a la que cada señal es detectada varía por el tipo de procesamiento y la cámara posee, en el caso de la señal de pare en comparación con la señal giro a la derecha poseen un rango de detección diferente en 10 cm ya que la señal giro a la derecha posee menos características a ser detectadas en la imagen.
- Las funciones desarrolladas en este sistema pretenden servir como base en cursos de autotrónica para permitir a los estudiantes transferir los conocimientos teóricos a experimentos prácticos, así como contar con una plataforma para la incorporación de diversas tareas en aplicaciones de navegación autónoma por medio de visión artificial, tales como la detección de líneas de carretera, señales de tránsito y semáforos.

RECOMENDACIONES

- Una de las principales limitaciones del prototipo es que el área de trabajo es limitada debido a la ubicación fija de la cámara web, a futuro se debería implementar una LIDAR que otorga una visión más precisa de un rango de 360°. Se puede implementar esta herramienta visual para cubrir un rango de visión y no perder el procesamiento de información por una mala ubicación o limitaciones del prototipo.
- Mediante la prueba de medición y precisión del prototipo, se puede determinar que el sistema de visión puede fallar por la tarde. Se recomienda utilizar el prototipo por la mañana o al mediodía que la visión de la computadora es óptima.
- Se recomienda antes de las pruebas de detección de carriles realizar un análisis de entorno donde o de la vía donde se va a movilizar el prototipo, debido que en Mayra de los carriles tiene un rango de error excesivo en vías o se mueve la cámara para obtener un rango de visión mayor.
- Para un mejor rendimiento a futuro se deberíamos implementar un GPS en el prototipo para geolocalización del mismo. Detectar el entorno de donde se encuentra el prototipo y poder monitorear a distancia hacia el destino que podamos asignar.
- Para la activación del servomotor debe tener una alimentación independiente, debido que este elemento realiza un torque elevado para el giro de las ruedas y es importante no conectar las salidas de alimentaciones de la Raspberry debido que dificultaría el rendimiento del sistema.

GLOSARIO

Autónomo: Es un vehículo capaz de imitar las capacidades humanas de manejo y control.

Raspberry: Es un ordenador de bajo coste y formato compacto destinado al desarrollo para hacer accesible la informática a todos los usuarios.

Python: Es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, nos sirve especialmente para la ejecución de programas de visión artificial.

OpenCV: Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision.

Puente H: Es un dispositivo electrónico capaz de invertir la polaridad de acuerdo a las necesidades del algoritmo.

BIBLIOGRAFÍA

- ACOSTA, Rebeca Junieth, et al.** *Reconocimiento de placas vehiculares aplicando procesamiento de imágenes digitales en Python-OpenCV. Tesis Doctoral. s.l. : (TESIS DE GRADO), 2020.*
- ALEMÁN, Enrique.** *Detección de carril utilizando barrido de ventanas en imágenes.* 2020. págs. 66-77.
- BARAHONA, Andrés Fabricio y RIOS, Kleber Ramiro.** *Desarrollo de una plataforma tipo Ackerman para la validación de algoritmos en el campus.* 2021.
- BARRETO, Andrés Mauricio.** *Implementación de un Sistema Radar Móvil Usando un Sensor Doppler para la Detección de Obstáculos.* 2020.
- BERMEJO, Jesús María Leal.** *Características de la infraestructura que pueden favorecer la conducción asistida y automatizada.* 2020. págs. 106-117.
- BORJA, F.** *Navegación automática en un vehículo con distribución Ackermann.* 2016.
- BOSE, Siddharth.** *Traffic Detection Using OpenCV.* 2020.
- BRICEÑO, Rodrigo Fernando y PAZMIÑO, Raúl Santiago.** *Detección del estado de somnolencia para conductores mediante el análisis de ondas cerebrales. . Tesis de Licenciatura.* 2020.
- BYELOBORODOV, Yevgeniy y RASHAD, Sherif.** *Design of Machine Learning Algorithms for Behavioral Prediction of Objects for Self-Driving Cars.* 2020. págs. p.0101-0105.
- CHEN, Qiang.** *Traffic light and moving object detection for a guide-dog robot.* 2020. págs. p. 675-678.

- COVA ROCAMORA, Roberto.** *Navegación y Conducción Autónoma de vehículos con geometría Ackerman.* 2019.
- CRUZ, Vianey Guadalupe.** *Evaluación del desempeño de sensores infrarrojo, ultrasónico y visión para el conteo de vehículos.* Instituto de Ingeniería y Tecnología. 2020.
- DHAROORI, , Rakesh Acharya y NARAYANAN, Sathiya.** *Vision-Based Lane Detection for Advanced Driver Assistance Systems.* 2020. págs. p. 537-546.
- DÍAZ, Jairo Eduardo.** *Visión artificial profunda aplicada a la identificación temprana de cáncer no melanoma y queratosis actínica.* 2020. págs. p. 751-766.
- DIONIZ, Ayoze.** *Transmisión digital con cámara óptica para comunicaciones VLC basada en Raspberry Pi.* 2020.
- FARAG, Wael.** *Real-time detection of road lane-lines for autonomous driving.* 2020. págs. p. 265-274.
- GANDHI, S. Girish.** *Traffic Sign Recognition Robot using Raspberry-Pi.* 2016.
- GUILCAPI, Jaime Rodrigo y LLAMBO, Fredy Bladimir.** *Sistema electrónico de corrección del gesto deportivo en base a la biomecánica en deportistas de marcha atlética.* 2020.
- JIMENEZ, Magaly Gabriela.** *Desarrollo de un sistema de visión artificial para la detección de aglomeración de personas en un semáforo.* s.l. : (Tesis de Grado), 2015.
- LORENTE MONZÓ, Jorge.** *Desarrollo de una aplicación de reconocimiento en de señales de tráfico en tiempo real mediante Python.* 2020.
- MAINA, Hernán Javier.** *Visión artificial para el reconocimiento automático, en tiempo real, de líneas urbanas de autobuses.* 2020.

- MONTERREY, Ángel Mauricio y SOSA, Camilo Andrés.** *Diseño de un sistema de semaforización inteligente para controlar flujo vehicular a partir de procesamiento de imágenes.* 2020.
- MONTESDEOCA, Erik Daniel.** *Implementación de un sistema de reconocimiento del uso de mascarillas como medida de precaución contra el covid19 usando deep learning.* 2020.
- MONTOYA, Santiago Andres y NIÑO, Jefersson.** *Diseño e implementación de un prototipo ADAS con comunicación IoT con radar lidar para detección de pseudoaccidentes.* 2020.
- OLEA, Julio Alfonso Rodriguez.** *Vehículo autónomo de configuración Ackermann para seguimiento de carril mediante procesamiento digital de imagen.* 2020. págs. p. 205-210.
- OLIVEIRA, Ricardo Enrique.** *Control Adaptativo para el Seguimiento de Trayectorias de un Robot Móvil en Carsim.* 2020.
- PALACIOS, Miguel Angel, SUCAR, L. Enrique y ESCALANTE, Hugo Jair.** *Deteccion de novedades visuales basada en aprendizaje profundo para manejo autonomo.* 2020.
- PUNAGIN, Akash y PUNAGIN, Sahana.** *Analysis of Lane Detection Techniques on Structured Roads using OpenCV.* 2020.
- QUINTERO, Lissette del Carmen y ACOSTA, Rebeca Junieth.** *Reconocimiento de placas vehiculares aplicando procesamiento de imágenes digitales en Python-OpenCV.* 2020.
- ROSSI, Alfa.** *Real-time Lane detection and Motion Planning in Raspberry Pi and Arduino for an Autonomous Vehicle Prototype.* 2020.

SARMIENTOS, Esther Diaz, AZUELA, Humberto Sossa y BARCELÓ, Alberto Petrilli.

Reconstrucción autónoma de un mapa métrico de una pista de competencias usando visión artificial. 2020. págs. p. 223-233.

VELÁSQUEZ, Jeison Alexander y LAZO, Anderson José. *Propuesta de un sistema de seguridad con cámaras y sensores usando tecnología RASPBERRY-PI, en el colegio bautista de la concepción, Masaya en el año 2020. 2020.*

VICENTE, Javier Jesús. *Análisis de algoritmos de detección de características de OpenCV en Raspberry Pi. 2020.*

ANEXOS

ANEXO A: Código de Stop.

 *DetectLaneLines.py - C:\Users\ramir\Desktop\Self-Driving_Car_Project-master\DetectLaneLines.py (3.7.7)*

```
File Edit Format Run Options Window Help
#####
### LEER SEÑAL DE TRANSITO###
###          STOP          ###
### OBJETIVO            ###
### > conducir hasta    ###
###  detectar una señal ###
###  de STOP            ###
#####
from picamera.array import PiRGBArray
from picamera import PiCamera
import RPi.GPIO as GPIO
from time import sleep
import numpy as np
import cv2

# Inializations: #
#Camera fps/size
camera=PiCamera()
camera.resolution=(640,480)
camera.framerate=20
rawCapture=PiRGBArray(camera,size=(640,480))
sleep(0.1)
#Car PIN setups
enA=14
in1=15
in2=18
templ=1
servo=22
angle=45
GPIO.setmode(GPIO.BCM)
GPIO.setup(enA,GPIO.OUT)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(servo,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
pwm_servo=GPIO.PWM(servo,100)
pwm_motor=GPIO.PWM(enA,1000)
pwm_servo.start(12.5)
pwm_motor.start(25)

def make_coordinate(img,line_parameters):
    slope,intercept=line_parameters
    y1=img.shape[0]
    y2=int(y1*(3/5))
    x1=int((y1-intercept)/slope)
    x2=int((y2-intercept)/slope)
    return np.array([x1,y1,x2,y2])
```

ANEXO B: Código Detección de línea

```
#####
### LEER DETECCION DE VIA ###
###                               ###
### OBJETIVO                       ###
### > conducir dentro             ###
###   de las lineas hasta         ###
###   llegar a un punto B|       ###
#####
from picamera.array import PiRGBArray
from picamera import PiCamera
import RPi.GPIO as GPIO
from time import sleep
import numpy as np
import cv2

# Inializations: #
#Camera fps/size
camera=PiCamera()
camera.resolution=(640,480)
camera.framerate=20
rawCapture=PiRGBArray(camera,size=(640,480))
sleep(0.1)
#Car PIN setups
enA=14
in1=15
in2=18
temp1=1
servo=22
angle=45
GPIO.setmode(GPIO.BCM)
GPIO.setup(enA,GPIO.OUT)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(servo,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
pwm_servo=GPIO.PWM(servo,100)
pwm_motor=GPIO.PWM(enA,1000)
pwm_servo.start(12.5)
pwm_motor.start(25)

def make_coordinate(img,line_parameters):
    slope,intercept=line_parameters
    y1=img.shape[0]
    y2=int(y1*(3/5))
    x1=int((y1-intercept)/slope)
    x2=int((y2-intercept)/slope)
    return np.array([y1,x1,y2,x2])
```

ANEXO C: Código de detección del semáforo.

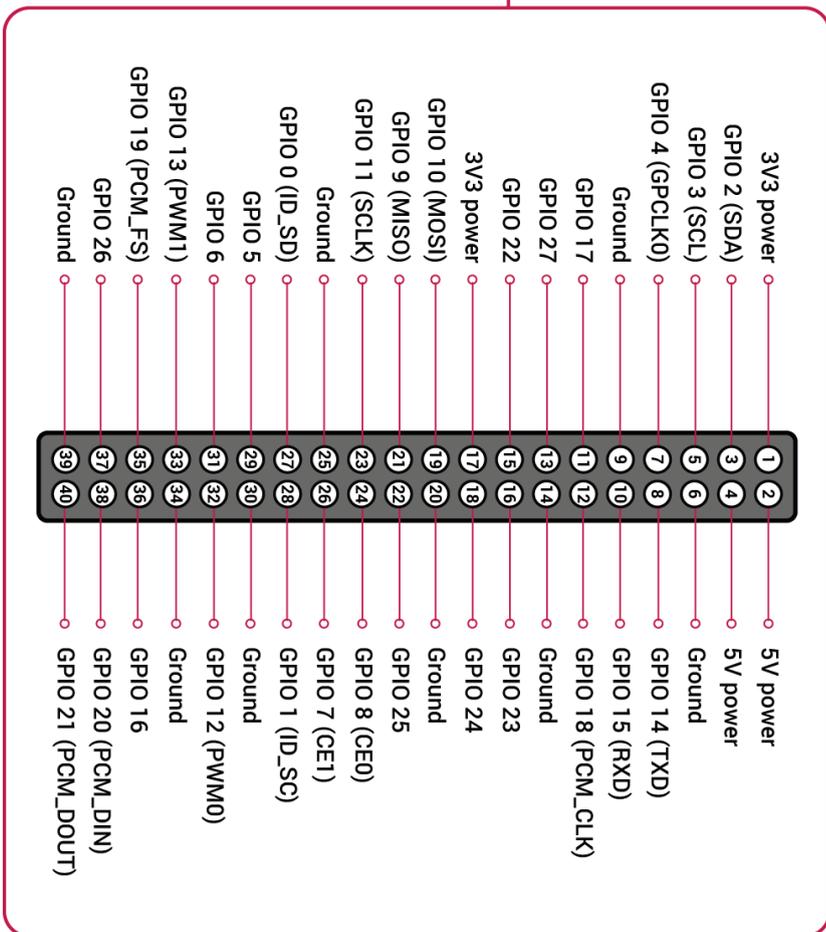
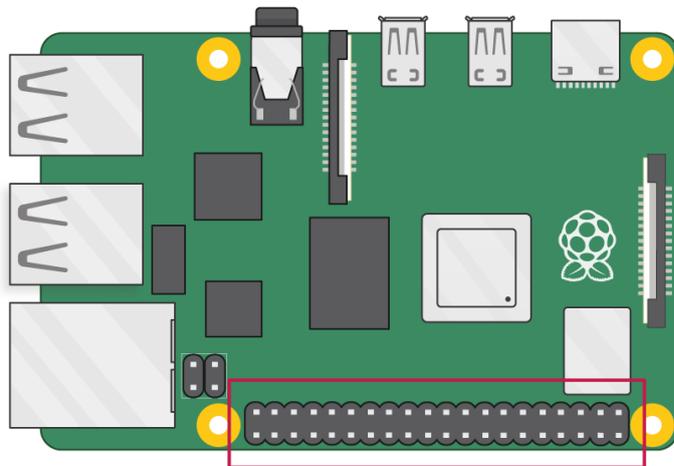
```
import numpy as np
import matplotlib.pyplot as plt
# change this to your caffe root dir
caffe_root = '/home/jfg/Documents/work/caffe'
import os
import sys
sys.path.insert(0, caffe_root + '/python')
import caffe
from google.protobuf import text_format
from caffe.proto import caffe_pb2
import cv2

def get_labelname(labelmap, labels):
    num_labels = len(labelmap.item)
    labelnames = []
    if type(labels) is not list:
        labels = [labels]
    for label in labels:
        found = False
        for i in range(0, num_labels):
            if label == labelmap.item[i].label:
                found = True
                labelnames.append(labelmap.item[i].display_name)
                break
        assert found
    return labelnames

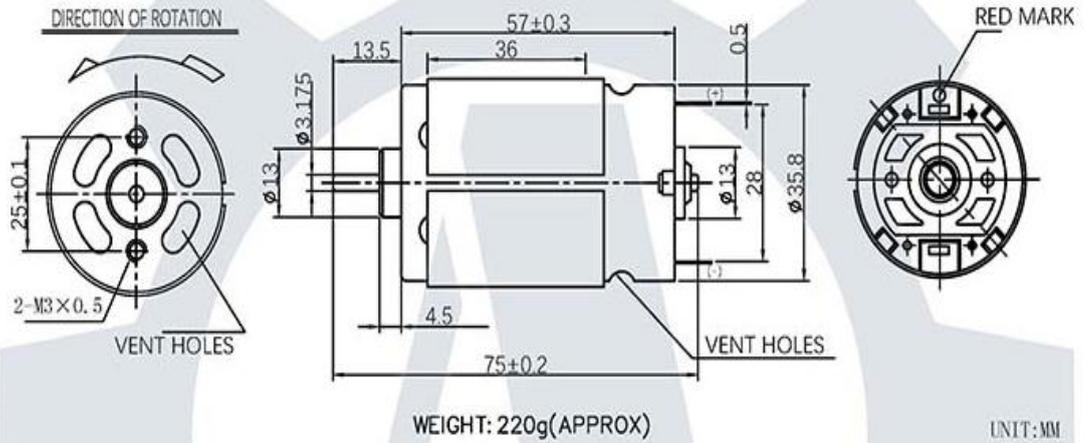
def ssd_detection(detect_image, map_label, is_show=True):
    model_def = 'models/VGGNet/LISA/SSD_414x125/deploy.prototxt'
    model_weights = 'models/VGGNet/LISA/SSD_414x125/VGG_LISA_SSD_414x125_iter_120000.caffemodel'
    if not os.path.exists('./predict_result'):
        os.mkdir('./predict_result')
    if len(detect_image.split('/')) != 0:
        image_save_path = './predict_result/' + detect_image.split('/')[-1]
    else:
        image_save_path = './predict_result/' + detect_image

    net = caffe.Net(model_def,
                    model_weights,
                    caffe.TEST)
    transformer = caffe.io.Transformer({'data': net.blobs['data'].data.shape})
    transformer.set_transpose('data', (2, 0, 1))
    transformer.set_mean('data', np.array([104, 117, 123]))
    transformer.set_raw_scale('data', 255)
    transformer.set_channel_swap('data', (2, 1, 0))
```

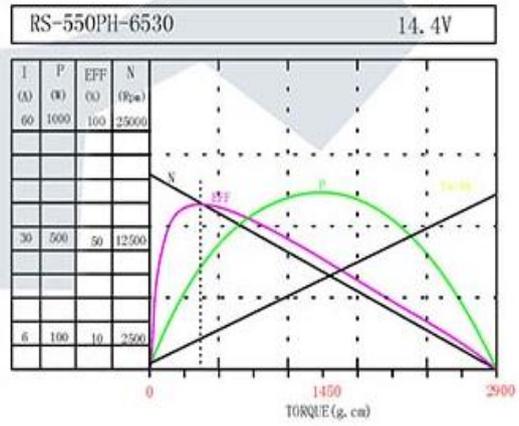
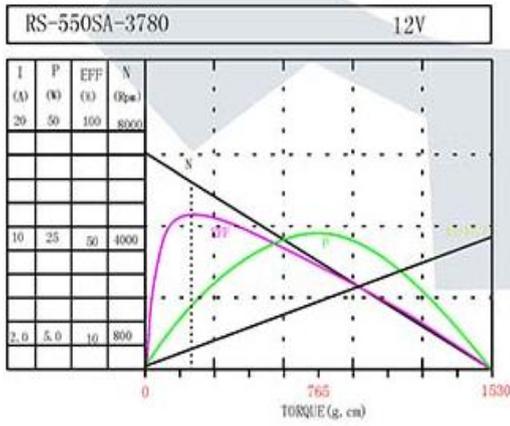

ANEXO F: Datasheet Raspberry Pi II



ANEXO G: Datasheet motor DC

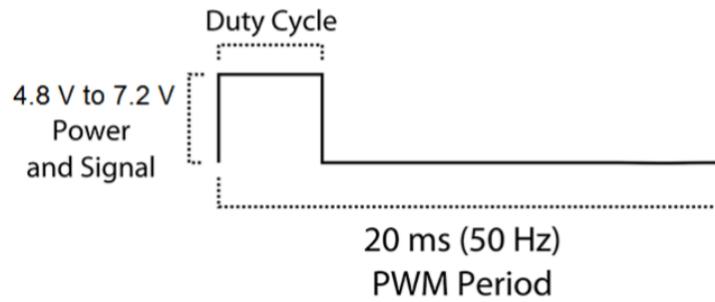
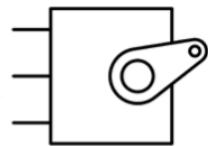


型号 MODEL	电压 VOLTAGE		空载 NO LOAD		最大效率 AT MAXIMUM EFFICIENCY					堵转 STALL	
	使用范围 OPERATING RANGE	额定值 NOMINAL	转速 SPEED	电流 CURRENT	转速 SPEED	电流 CURRENT	扭矩 TORQUE	输出功率 OUT PUT	效率 EFF	扭矩 TORQUE	电流 CURRENT
	V	V	r/min	A	r/min	A	g. cm	W	%	g. cm	A
RS-550SA-3780	6.0~18	12.0	7280	0.25	6300	1.70	206	13.3	65.2	1530	11.2
RS-550PH-6530	6~18	14.4	20500	1.4	17500	7.64	424	76.2	69.3	2900	44.0

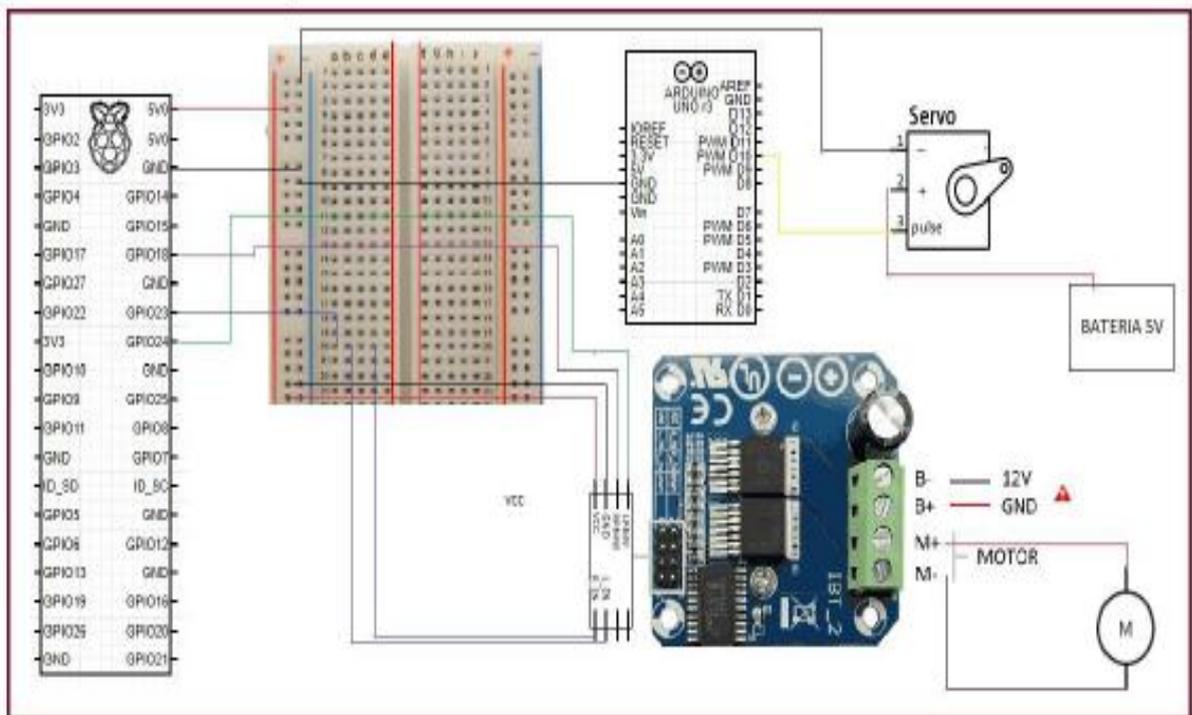


ANEXO H: Datasheet ServoMotor

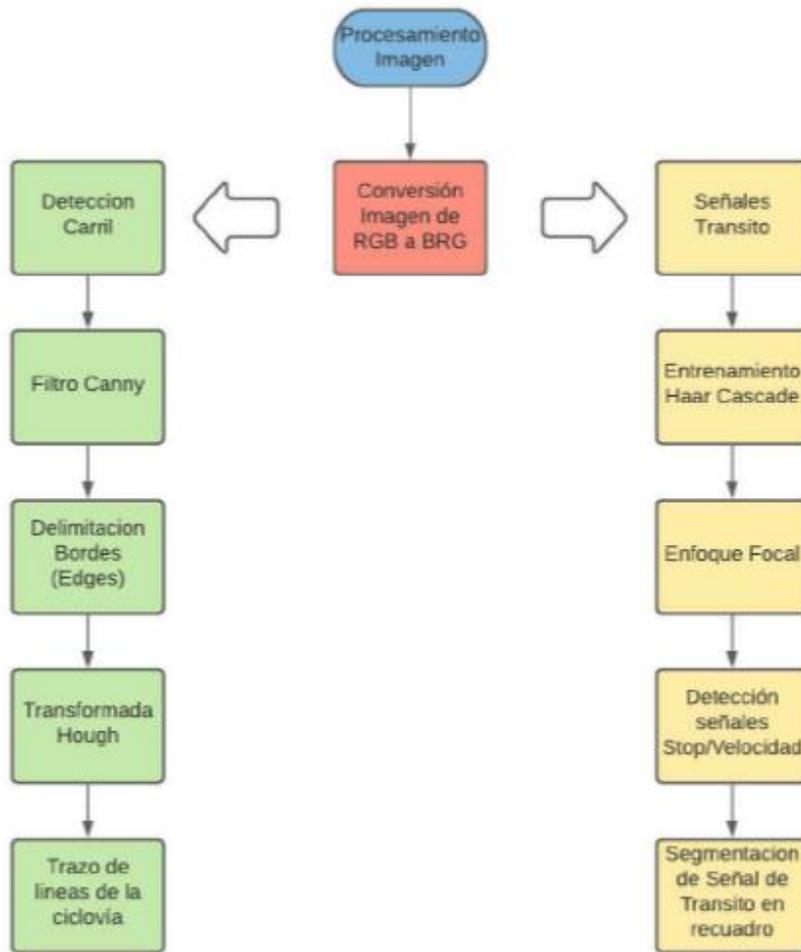
PWM=Orange (⏏)
Vcc = Red (+)
Ground=Brown (-)



ANEXO I: Diagrama de conexión prototipo ESPOCH01



ANEXO J: Diagrama de flujo Procesamiento de la imagen del prototipo ESPOCH01



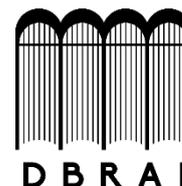
ANEXO K: Prototipo Vehículo ESPOCH01





**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO**

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN**



UNIDAD DE PROCESOS TÉCNICOS

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega:

INFORMACIÓN DEL AUTOR/A (S)
Andrés Fabricio Barahona Andino Kleber Ramiro Ríos Sigcho
INFORMACIÓN INSTITUCIONAL
Facultad: Mecánica
Carrera: Ingeniería Automotriz
Título a optar: Ingeniero Automotriz
f. Analista de Biblioteca responsable: