



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

**TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO
ELECTRÓNICO**

TÍTULO:

**IMPLEMENTACIÓN DE UNA ESTRATEGIA DE CONTROL CINEMÁTICO EN
UN ROBOT DE TRACCIÓN DIFERENCIAL USANDO VISION ARTIFICIAL
PARA LA RED TECNOPARQUE NODO NEIVA.**

Autor:

RUBÉN DARÍO PÉREZ CUELLAR

Director:

ING. LUIS ALBERTO MUÑOZ BEDOYA

PAMPLONA-COLOMBIA

MAYO de 2018



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

**TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO
ELECTRÓNICO**

TÍTULO:

**IMPLEMENTACIÓN DE UNA ESTRATEGIA DE CONTROL CINEMÁTICO EN
UN ROBOT DE TRACCIÓN DIFERENCIAL USANDO VISION ARTIFICIAL
PARA LA RED TECNOPARQUE NODO NEIVA.**

Autor:

RUBÉN DARÍO PÉREZ CUELLAR

Director:

ING. LUIS ALBERTO MUÑOZ BEDOYA

JURADO CALIFICADOR:

LUIS ALBERTO MUÑOZ BEDOYA

MSc. LUIS ENRIQUE MENDOZA

PhD. CESAR AUGUSTO PEÑA

PAMPLONA-COLOMBIA

MAYO de 2018

II

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA,
SISTEMAS Y
TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

**TRABAJO PRESENTADO PARA OPTAR POR ÉL TÍTULO DE INGENIERO
ELECTRÓNICO**

TEMA:

**IMPLEMENTACIÓN DE UNA ESTRATEGIA DE CONTROL CINEMÁTICO EN
UN ROBOT DE TRACCIÓN DIFERENCIAL USANDO VISION ARTIFICIAL
PARA LA RED TECNOPARQUE NODO NEIVA.**

FECHA DE INCIDO DEL TRABAJO: SEPTIEMBRE 2017

FECHA DE TERMINACIÓN DEL TRABAJO: DICIEMBRE 2017

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:

**RUBÉN DARÍO PÉREZ CUELLAR
AUTOR**

**ING. LUIS ALBERTO MUÑOZ BEDOYA
DIRECTOR**

**ING. LUIS ALBERTO MUÑOZ BEDOYA
DIRECTOR DEL PROGRAMA**

JURADO CALIFICADOR:

ING. LUIS ALBERTO MUÑOZ BEDOYA

MSc. LUIS ENRIQUE MENDOZA

PhD. CESAR AUGUSTO PEÑA

**PAMPLONA N. S. COLOMBIA
MAYO de 2018**

DEDICATORIA

Este trabajo no se hubiese podido realizar sin el inmenso esfuerzo de mi madre.
Para mi abuela Marina

AGRADECIMIENTOS

A mi padre por todo el apoyo y siempre creer en mí. A mis hermanos, mis tíos y tías, amandis y memito muchísimas gracias por la paciencia y colaboración durante el pregrado.

Ingeniero Adrián Fernando Chávarro gracias por permitirme hacer parte del proyecto Agrobot Eco.

Infinitas gracias a Jah.

ÍNDICE DE CONTENIDO

| | Pág. |
|---|--------------------------------------|
| | I |
| 1. CAPITULO INTRODUCTORIO..... | 1 |
| 1.1 INTRODUCCION..... | ¡Error! Marcador no definido. |
| 1.2 PLANTEAMIENTO DEL PROBLEMA..... | 3 |
| 1.3 JUSTIFICACION..... | ¡Error! Marcador no definido. |
| 1.4 DELIMITACION..... | ¡Error! Marcador no definido. |
| 1.4.1 Objetivo general..... | 4 |
| 1.4.2 Objetivos específicos..... | 4 |
| 1.4.3 Acotaciones..... | 5 |
| 1.5 ANTECEDENTES..... | 5 |
| 1.5.1 Navegación de un robot móvil mediante visión artificial: (Universidad nacional de Colombia)..... | 5 |
| 1.5.2 Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines: (Universidad Anahuac México Sur)..... | 7 |
| 1.5.3 Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA: (Servicio Nacional de Aprendizaje SENA Regional Huila)..... | 8 |
| 2. CAPITULO MARCO TEORICO..... | 11 |
| 2.1 ROBOTS MOVILES..... | 11 |
| 2.1.1 VEHÍCULOS CON RUEDAS..... | 11 |
| 2.2 VISION ARTIFICIAL..... | 18 |
| 2.2.1 RECONOCIMIENTO DE OBJETOS..... | 19 |
| 2.2.2. PROCESAMIENTO DIGITAL DE IMÁGENES..... | 19 |

| | |
|--|----|
| 2.3 MODELO DE COLOR | 20 |
| 2.3.1 MODELO RGB | 20 |
| 2.3.2 MODELO CMYK | 21 |
| 2.3.3 MODELO HSV | 22 |
| 2.4 HISTOGRAMA | 23 |
| 2.5 PROTOTIPO | 26 |
| 2.5.1 PANEL SOLAR | 26 |
| 2.5.2 CONTROLADOR DE CARGA SOLAR..... | 26 |
| 2.5.3 L298N | 27 |
| 2.5.4 MOTORES | 28 |
| 2.5.5 BATERÍA | 31 |
| 2.5.6 ARDUINO | 32 |
| 2.5.7 BLUETOOTH | 33 |
| 2.5.8 CÁMARA..... | 34 |
| 3. CAPITULO METODOLOGIA | 35 |
| 3.1 ROBOT MÓVIL | 35 |
| 3.2 CONTROL CINEMÁTICO | 38 |
| 3.2.1 COMUNICACIÓN BLUETOOTH..... | 43 |
| 3.3 PROCESAMIENTO DE IMÁGENES..... | 44 |
| 3.3.1 POSICION Y ORIENTACION EN EL PLANO | 49 |
| 3.4 INTERFAZ GRÁFICA DE USUARIO | 51 |
| 4. CAPITULO RESULTADOS Y VALIDACIÓN | 54 |
| 4.1 ESPACIO RECREADO | 54 |
| 4.2 ROBOT MOVÍL | 57 |
| 4.2.1 VELOCIDAD LINEAL..... | 58 |
| 4.3 PROCESAMIENTO DE IMÁGENES..... | 61 |
| 4.3.1 PREPROCESAMIENTO..... | 66 |

| | |
|------------------------------|----|
| 4.4 CONTROL CINEMATICO | 74 |
| 4.5 INTERFAZ GRAFICA | 82 |
| CONCLUSIONES | 92 |
| BIBLIOGRAFÍA..... | 93 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1 SECADO DE CAFE EN PATIO..... | 2 |
| Figura 2 ROBOT SAV..... | 5 |
| Figura 3 ORIENTACION SAV | 6 |
| Figura 4 IMAGEN DE OBJETOS A DIGITALIZAR | 7 |
| Figura 5 OBJETOS SEGMENTADOS | 7 |
| Figura 6 CONJUNTO DE PUNTOS CONSIDERADOS | 8 |
| Figura 7 ROBOT MOVIL CON TRACCION DIFERENCIAL UTILIZANDO LA HERRAMIENTA LEGO..... | 9 |
| Figura 8 SIMULACION DEL DESPLAZAMIENTO XY DEL ROBOT MOVIL PARA DIFERENTES VALORES DE K1 Y K2, UTILIZANDO EL CONTROL EN LAZO CERRADO..... | 10 |
| Figura 9 SISTEMA ACKERMAN | 12 |
| Figura 10 CONFIGURACION TIPO TRICICLO..... | 13 |
| Figura 11 SISTEMA TRACCION DIFERENCIAL..... | 13 |
| Figura 12 ROBOT SKID STEER EN EL PLANO..... | 14 |
| Figura 13 ROBOT CONFIGURACION ORUGA | 15 |
| Figura 14 CONFIGURACION SÍNCRONA..... | 15 |
| Figura 15 CONFIGURACION OMNIDIRECCIONAL | 16 |
| Figura 16 ANÁLISIS DE POSICIONAMIENTO DEL ROBOT..... | 16 |
| Figura 17 ANÁLISIS DE LA ORIENTACION DEL ROBOT..... | 17 |
| Figura 18 IDENTIFICAR CON CLARIDAD OBJETOS | 19 |
| Figura 19 PROCESAMIENTO DIGITAL DE IMÁGENES EN MATLAB | 20 |
| Figura 20 MODELO RGB | 21 |
| Figura 21 MODELO CMYK..... | 22 |
| Figura 22 MODELO HSV | 23 |
| Figura 23 HISTOGRAMA DE LOS TRES CANALES RGB | 24 |
| Figura 24 HISTOGRAMA DE LOS CANALES RGB INDIVIDUALES..... | 24 |
| Figura 25 UMBRALIZACION DE UNA IMAGEN CON AYUDA DEL HISTOGRAMA | 25 |
| Figura 26 RESULTADO DE LA UMBRALIZACION | 25 |
| Figura 27 CARACTERISTICAS ELECTRICAS DEL PANEL SOLAR..... | 26 |
| Figura 28 CONTROLADOR DE CARGA SOLAR..... | 27 |
| Figura 29 CONTROLADOR DE MOTORES L298N | 27 |
| Figura 30 MOTORREDUCTOR 50:1 37Dx70L mm con 64 CPR Encoder..... | 28 |
| Figura 31 CONEXIONES MOTORREDUCTOR..... | 29 |
| Figura 32 SALIDA EN CUADRATURA DEL ENCODER EFECTO HALL | 30 |
| Figura 33 FUNCIONAMIENTO SENSOR EFECTO HALL..... | 31 |

| | |
|--|----|
| Figura 34 BATERIA 12V 7A | 31 |
| Figura 35 ARDUINO..... | 32 |
| Figura 36 MODULO BLUETOOTH..... | 33 |
| Figura 37 CÁMARA MICROSOFT HD-3000..... | 34 |
| Figura 38 PRIMER MODELO DE AGROBOT EN EXPO-RURALES SENA..... | 35 |
| Figura 39 MODELO FINAL AGROBOT ECO | 36 |
| Figura 40 AGROBOT ECO | 37 |
| Figura 41 AGROBOT ECO, CIRCUITOS | 37 |
| Figura 42 AGROBOT ECO, ESTRUCTURA..... | 38 |
| Figura 43 AGROBOT ECO, ESTRUCTURA..... | 38 |
| Figura 44 POSICIONAMIENTO DE UN ROBOT SKID STEER | 39 |
| Figura 45 VELOCIDADES DERECHA E IZQUIERDA DEL ROBOT | 39 |
| Figura 46 ANGULOS PARA CORREGIR LA ORIENTACION DEL VEHICULO | 42 |
| Figura 47 ESQUEMA DE CONTROL..... | 43 |
| Figura 48 ENVIO DE INFORMACION VIA BLUETOOTH | 43 |
| Figura 49 RECEPCION DE INFORMACION ENVIADA VIA BLUETOOTH | 44 |
| Figura 50 CIRCULO DE COLOR ROJO | 44 |
| Figura 51 HISTOGRAMA DE LOS CANALES DEL MODELO HSV | 45 |
| Figura 52 CIRCULO DE COLOR AZUL | 45 |
| Figura 53 HISTOGRAMA DE LOS CANALES DEL MODELO HSV | 45 |
| Figura 54 SELECCION DEL UMBRAL DEL COLOR EN CADA UNO DE LOS CANALES, ROJO | 46 |
| Figura 55 SELECCION DEL UMBRAL DEL COLOR EN CADA UNO DE LOS CANALES, AZUL..... | 46 |
| Figura 56 REGION DEL COLOR ESPECIFICO, ROJO | 47 |
| Figura 57 REGION DEL COLOR ESPECIFICO, AZUL..... | 47 |
| Figura 58 MASCARA RGB PARA RECUPERAR EL COLOR SOLO DE LA REGION ESPECIFICADA, ROJO..... | 47 |
| Figura 59 MASCARA RGB PARA RECUPERAR EL COLOR SOLO DE LA REGION ESPECIFICADA, AZUL..... | 48 |
| Figura 60 SEGMENTACION DEL COLOR ROJO, EJEMPLO 1 | 48 |
| Figura 61 SEGMENTACION DEL COLOR AZUL, EJEMPLO 2 | 48 |
| Figura 62 CIRCULOS NARANJAS PARA ORIENTAR Y UBICAR EL ROBOT EN EL PLANO | 49 |
| Figura 63 DETERMINAR EL ANGULO DE UN VECTOR EN EL PLANO..... | 50 |
| Figura 64 OBJETOS SEGMENTADOS Y VECTOR DE ORIENTACION EN EL PLANO | 50 |
| Figura 65 PLANOS DE DIBUJO Y VISUALIZACION DE PUNTOS DE LA TRAYECTORIA..... | 51 |

| | |
|---|----|
| Figura 66 AJUSTE EN LA INTENSIDAD DE LOS VALORES DE LA IMAGEN | 52 |
| Figura 67 VISUALIZACION DE LA POSICION | 52 |
| Figura 68 VISUALIZACION DE LA VELOCIDAD | 52 |
| Figura 69 PLANO DE VISUALIZACION DE LAS IMAGENES | 53 |
| Figura 70 VALOR DE LOS ANGULOS MEDIDOS | 53 |
| Figura 71 FOTOGRAFIA #1, ESPACIO RECREADO | 54 |
| Figura 72 FOTOGRAFIA #2, ESPACIO RECREADO | 55 |
| Figura 73 POSICION DE LA CÁMARA | 56 |
| Figura 74 MEDIDA DEL ESPACIO | 57 |
| Figura 75 MOVIMIENTO HACIA ADELANTE | 57 |
| Figura 76 GIRO HACIA LA IZQUIERDA | 58 |
| Figura 77 GIRO HACIA LA DERECHA | 58 |
| Figura 78 ENCODERS PARA MEDIR LA VELOCIDAD | 59 |
| Figura 79 COLOR EN LOS CIRCULOS DEL ROBOT | 61 |
| Figura 80 HISTOGRAMA DEL COLOR SELECCIONADO | 62 |
| Figura 81 IMAGEN SIN MODIFICAR INTENSIDAD | 62 |
| Figura 82 IMAGEN INTENSIDAD AJUSTADA | 63 |
| Figura 83 HISTOGRAMA, FIGURA 81 | 63 |
| Figura 84 HISTOGRAMA, FIGURA 82 | 64 |
| Figura 85 CIRCULOS SOBRE EL ROBOT | 64 |
| Figura 86 IMAGEN CAPTURADA POR LA INTERFAZ E INTESIDAD AJUSTADA | 65 |
| Figura 87 OBJETOS SEGMENTADOS DE LA FIGURA 86 | 65 |
| Figura 88 COLORES SEGMENTADOS IMAGEN BINARIZADA | 66 |
| Figura 89 IMAGEN FILTRADA POR AREA | 66 |
| Figura 90 IMAGEN DILATADA | 67 |
| Figura 91 CIRCULOS ETIQUETADOS EN LA IMAGEN | 67 |
| Figura 92 PUNTO MEDIO DE LA BASE | 68 |
| Figura 93 REFERENCIA PARA EL VECTOR DE DESPLAZAMIENTO | 68 |
| Figura 94 VECTOR DE DESPLAZAMIENTO | 69 |
| Figura 95 PLANO DE REFERENCIA DEL ROBOT | 69 |
| Figura 96 ANGULO DEL ROBOT | 70 |
| Figura 97 DIBUJO DEL TRAYECTO | 71 |
| Figura 98 PUNTOS QUE REPRESENTAN EL TRAYECTO | 72 |
| Figura 99 TRAYECTO ALEATORIO PARA VISUALIZAR EL VECTOR GUIA | 73 |
| Figura 100 VECTOR GUIA DEL TRAYECTO | 73 |
| Figura 101 ANGULOS DEL MODELO | 74 |
| Figura 102 TRAYECTORIA DE PRUEBA | 74 |
| Figura 103 PUNTOS DE LA TRAYECTORIA | 75 |
| Figura 104 DELTA DE PHI | 77 |

| | |
|--|----|
| Figura 105 VELOCIDADES DE LAS RUEDAS | 78 |
| Figura 106 ERROR DE POSICION | 80 |
| Figura 107 COMPARACION DE TRAYECTORIAS | 81 |
| Figura 108 TRAYECTORIA DE PRUEBA | 81 |
| Figura 109 TRAYECTORIA DE PRUEBA | 82 |
| Figura 110 INTERFAZ GRAFICA DE USUARIO | 82 |

ÍNDICE DE ECUACIONES

| | |
|------------|----|
| (1) | 17 |
| (2) | 18 |
| (3) | 18 |
| (4) | 18 |
| (5) | 40 |
| (6) | 40 |
| (7) | 40 |
| (8) | 41 |
| (9) | 41 |
| (10) | 41 |
| (11) | 42 |
| (12) | 42 |

RESUMEN

El trabajo consiste en recrear un espacio que pueda ser identificado por una cámara, de tal modo que pueda ser representado como un plano bidimensional en una interfaz gráfica que permita dibujar cualquier trayectoria sobre este. El trayecto dibujado por el usuario en la interfaz será el recorrido que deba realizar un robot por el espacio recreado. La comunicación entre el robot y la interfaz se realiza mediante un módulo bluetooth, de modo que se puedan ejercer las acciones de control en el robot con ayuda de una tarjeta de desarrollo libre (Arduino). La estrategia de control se desarrolla en base al tipo del robot escogido y su modelo cinemático. El objetivo es implementar un algoritmo de visión artificial que en conjunto con el modelo cinemático identifiquen: el robot, su posición, su orientación en el plano y así poder corregir su trayectoria con ayuda del modelo cinemático que relaciona las variables. Para identificar el robot se ubican de manera estratégica círculos, que permitan estimar la orientación y posición del robot en el plano, estos círculos son segmentados en la imagen por su color. El histograma de la imagen, en el modelo de color HSV, determina el umbral para la selección del color de los círculos y así segmentar los mismos para las medidas de las variables que otorga la cámara al modelo.

ABSTRACT

The work consists of recreating a space that can be identified by a camera, in such a way that it can be represented as a two-dimensional plane in a graphical interface that allows drawing any trajectory on it. The path drawn by the user in the interface will be the path that a robot must make through the recreated space. The communication between the robot and the interface is carried out by means of a bluetooth module, so that the control actions can be exercised on the robot with the help of a free development card (Arduino). The control strategy is developed based on the type of robot chosen and its kinematic model. The objective is to implement an artificial vision algorithm that together with the kinematic model identify: the robot, its position, its orientation in the plane and thus be able to correct its trajectory with the help of the kinematic model that relates the variables. To identify the robot, circles are strategically located, which allow estimating the orientation and position of the robot in the plane, these circles are segmented in the image by their color. The histogram of the image, in the HSV color model, determines the threshold for the selection of the color of the circles and thus segment them for the measurements of the variables that the camera grants to the model.

1. CAPITULO

INTRODUCTORIO

En este capítulo encontrará la introducción, planteamiento del problema, justificación, las delimitaciones a tener en cuenta y finalmente antecedentes de trabajos relacionados.

1.1 INTRODUCCIÓN

En la actualidad la robótica y el control se han convertido en elementos fundamentales para la industria, gracias a estas tecnologías los procesos de producción cumplen a cabalidad las necesidades de las empresas, como en las líneas de manufactura. El éxito de estas nuevas herramientas en grandes industrias ha abierto las puertas para su aplicación en el sector agrícola, de modo que permitan a los agricultores mejorar su calidad de vida y su productividad.

Actualmente, organizaciones como FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) destacan el proceso de transformación que vive la agricultura, en el que el acceso a la información y a las modernas tecnologías de la comunicación, como la informática, la conectividad y la electrónica, resultan una necesidad para los agricultores de todo el mundo, ya que, gracias a las nuevas tecnologías, es posible desarrollar una agricultura de elevada precisión y calidad [27].

En Colombia una de las principales industrias agrícolas es la del café, destacándose por su calidad de producción se ha convertido en uno de los mayores productores de café a nivel mundial, ubicándolo en tercer lugar por detrás de Brasil y Vietnam, con la tasa representativa de aproximadamente el 10% de la producción a nivel internacional [46]. A nivel nacional, el departamento del Huila se ha convertido en los últimos años en uno de los departamentos líderes en la producción de café, llegando a encontrarse por encima del promedio nacional de productividad de este sector [28].

Aunque cada fábrica tiene sus diferencias en líneas de producción de café, el proceso es bastante similar en cada una de ellas, pues se hacen indispensables labores como el lavado, despulpado, secado y empaque.

El secado es una de las tareas claves en la producción del café, determinante dentro de la cadena de valor debido a su alta influencia en la calidad de los productos resultantes de los granos [29]. Un mal manejo de esta etapa puede causar defectos en los estándares de calidad aplicados al café en Colombia, estos parámetros exigen un contenido de humedad entre 10 – 12% en los granos.

Al comercializar un café que no se encuentra dentro del rango de 10% – 12%, el caficultor puede tener pérdidas económicas al tener un nivel de humedad inferior a 10% (por la pérdida de peso al momento de vender el café) o presentar un café con problemas de taza si la humedad es superior al 12% [29].

Existen diferentes sistemas para el secado de café, algunos métodos aprovechan los rayos solares, como son: secado en patio, secado en carros de madera, secador solar parabólico, entre otras [30]. Sistemas mecánicos también son empleados para el secado del café, estos se caracterizan en función a su unidad de calentamiento, ventilación y combustible utilizado para calentar el aire [31].

El secado al sol es un método económico y ambientalmente amigable, para este cumplir o superar los estándares mínimos de calidad del café de Colombia. Es el más usado por los pequeños productores en Colombia, quienes producen cerca del 70% de la cosecha nacional [29].

Figura 1 SECADO DE CAFÉ EN PATIO



Fuente: Secado de café al sol y mecánico, COFFEEIQ [30]

Tecnoparque nodo Neiva trabaja en la elaboración de proyectos aplicados al sector agrícola con el objetivo de aportar al crecimiento de la agricultura en la región. Enfocados en la tarea del secado del café nace el proyecto ‘Agrobot Eco’, que se espera sea un robot móvil capaz

de recorrer un patio de secado describiendo una trayectoria determinada por el caficultor, optimizando el proceso de secado y volteo uniforme del café.

Este trabajo consiste en la construcción de un prototipo que permita implementar y a su vez evaluar el modelo cinemático de un robot móvil en un espacio controlado para seguir un trayecto supervisado por una cámara. La estrategia de control a implementar está basada en el estudio cinemático y cinemática inversa del robot Skid steer (Agrobot Eco) escogido para el prototipo y que se asemeja al modelo de un robot de tracción diferencial.

El modelo de color HSV y el análisis del histograma de sus componentes permiten determinar un umbral para el color de los círculos en cada una de ellas, con el fin de segmentar de manera precisa el color determinado en la imagen. Los cuales son colocados sobre el robot en el panel solar en forma de triángulo, estos serán los puntos de partida para establecer el vector de desplazamiento del robot que permite medir su orientación y posición en el plano.

En función a estas variables entregadas del procesamiento de las imágenes capturadas por la cámara, lectura y envío de las velocidades lineales del robot hacia el algoritmo por un módulo bluetooth son las entradas del modelo de control para luego corregir su posición y ser guiado en el trayecto.

1.2 PLANTEAMIENTO DEL PROBLEMA

La red Tecnoparque nodo Neiva siempre en búsqueda de herramientas tecnológicas que fortalezcan el sector agrícola de la región. Y en función del notorio crecimiento de la industria cafetera en el departamento del Huila, busca desarrollar una herramienta novedosa y asequible que permita optimizar la tarea del secado del café en patio.

Para el secado del café en patio existen recomendaciones que ayudan al caficultor a obtener un mejor resultado en su producto. Por ejemplo, en función de la cantidad de café para secar se establece el área necesaria para su distribución en una relación de una arroba por metro cuadrado, en capas aproximadamente de 3,5 cm [30], con este espesor y revolviendo el café por lo menos cuatro veces al día, se logra un secado parejo al grano [31].

Los rayos solares a los que se expone el caficultor al voltear el café para el secado uniforme en el patio pueden causar que se desarrollen enfermedades debido a la alta exposición solar, como son: lesiones oculares, quemaduras solares, carcinoma baso celular, entre otras [32].

1.3 JUSTIFICACIÓN

Los robots son muy comunes en las grandes industrias de manufactura o en líneas de producción alimentaria, sin embargo, en la agricultura es poco común ver el uso de robots

que contribuyan en el ciclo productivo. La robótica aplicada al sector agrícola está en desarrollo avanzado, y aunque muy lejos de alcanzar la “robótica industrial” que buscan priorizar el aumento de la producción, está más cerca de la “robótica de servicios” cuyo objetivo se enfoca en ayudar o sustituir al humano en el desarrollo de actividades donde existen riesgos o limitaciones físicas [33].

Agrobot Eco es una alternativa tecnológica de servicio que busca evitar la exposición solar del caficultor en el patio de secado al momento de voltear el café y aprovechar esta radiación como fuente de energía para el movimiento del robot, lo que mantiene la buena relación con el medio ambiente, una de las principales características que tiene este tipo de secado.

La denominada Agricultura de Precisión contribuye específicamente a optimizar el uso de los recursos e insumos [33]. Agrobot Eco optimiza los recursos humanos puesto que el tiempo que se dedicaba para el volteo del café se puede dedicar a otra labor por parte del caficultor, teniendo la plena seguridad de que el volteo del café se realiza de manera uniforme.

1.4 DELIMITACIÓN

1.4.1 Objetivo general

- Implementar una estrategia de control cinemático en un robot de tracción diferencial usando visión artificial para la red Tecnoparque nodo Neiva.

1.4.2 Objetivos específicos

1. Establecer un estado del arte de tracción diferencial, visión artificial y control cinemático para plataformas robóticas.
2. Definir los dispositivos electrónicos y estrategias de comunicación basados en el entorno de trabajo.
3. Elaborar el software para la adquisición de imágenes, estudio de las ecuaciones de cinemática y comunicación con el prototipo robótico.
4. Desarrollar una estrategia de control para la ubicación y desplazamiento de múltiples trayectorias en el entorno de trabajo.
5. Validar la funcionalidad del controlador usando diversas trayectorias desconocidas en el prototipo robótico.

1.4.3 Acotaciones

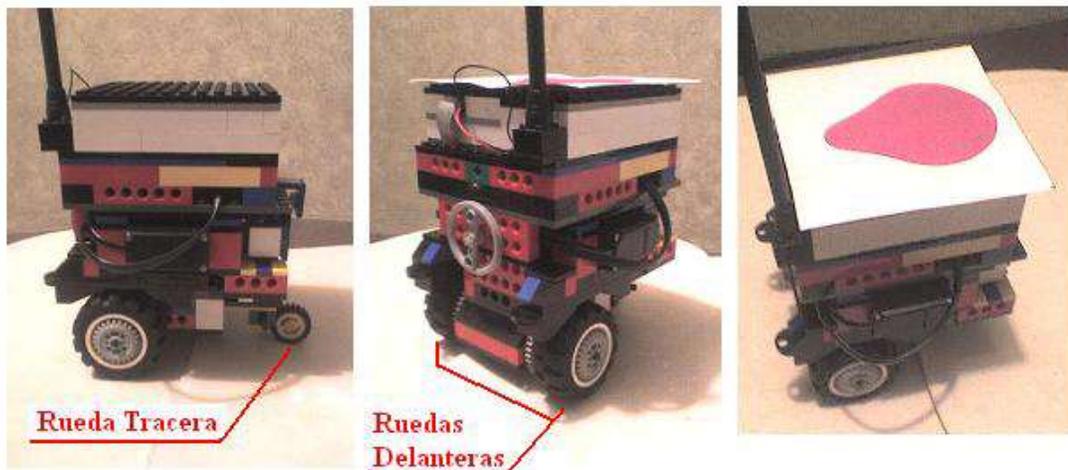
- El trabajo será realizado en conjunto con la red Tecnoparque nodo Neiva, y el desarrollo de este se llevará a cabo en la misma ciudad.
- El entorno de trabajo será un ambiente controlado.
- El robot de tracción diferencial fue construido por el Tecnoparque como una herramienta didáctica y será usado para la implementación del control cinemático.
- Los dispositivos electrónicos serán seleccionados dentro del grupo de recursos con los que cuenta el Tecnoparque.

1.5 ANTECEDENTES

1.5.1 Navegación de un robot móvil mediante visión artificial: (Universidad nacional de Colombia).

Uno de los métodos para ubicar y guiar un robot móvil a través de una trayectoria es usando visión artificial. En este trabajo construyen el prototipo del robot de tracción diferencial utilizando piezas técnicas lego, el robot se desplaza en un área delimitada de color blanco, donde se encuentran figuras geométricas planas de diferentes colores que son identificadas por una cámara web. Con ayuda de MATLAB el algoritmo reconoce la posición y orientación del robot en el plano de modo que este pueda ser guiado a la ubicación de una figura escogida por el usuario sin colisionar con ningún obstáculo que se le pueda presentar.

Figura 2 ROBOT SAV



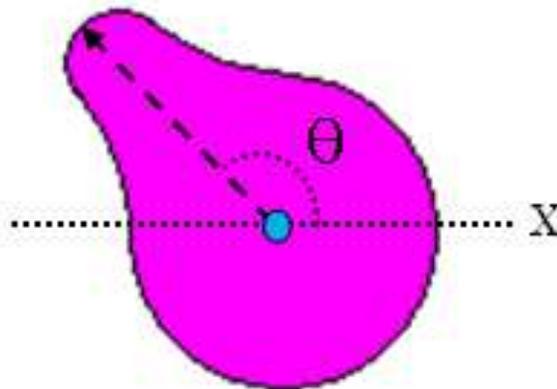
Fuente: Navegación de un robot móvil mediante visión artificial, Universidad Nacional [1]

Uno de los mayores desafíos en la toma de imágenes para su posterior procesamiento es la adaptación a los cambios de iluminación, donde realizan un preprocesamiento para ajustar los colores de la imagen.

Para identificar cada uno de los colores hacen uso de una red neuronal probabilística, Para su entrenamiento se utilizó como dato de entrada un vector constituido por una paleta de 256 niveles de color y la salida es un vector de igual tamaño, con valores de cero y uno, donde un uno representa la posición correspondiente al nivel de color que se quiere reconocer y un cero lo que se desea descartar [1].

La metodología para determinar la orientación la realizaron segmentando el color fucsia de la figura en forma de lágrima que se encuentra en la parte superior del robot, utilizando la red PNN, luego se calcula el centroide y se toma una línea recta entre este y el píxel más lejano del borde [1]. La orientación es el ángulo que existe entre la línea y el eje x del sistema de coordenadas general, de esta manera es posible ubicar y orientar el robot desde su lugar posición inicial hasta el lugar donde se encuentra la figura que determina la posición final.

Figura 3 ORIENTACIÓN SAV



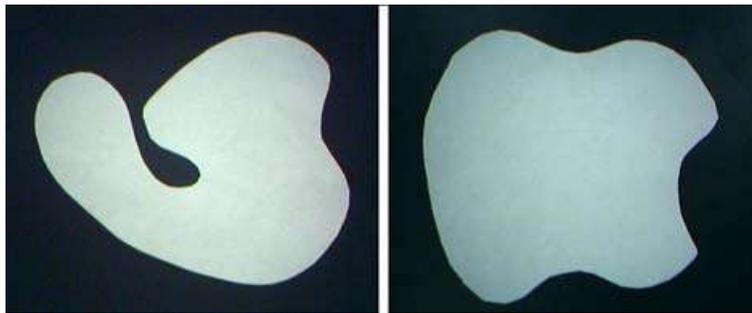
Fuente: Navegación de un robot móvil mediante visión artificial, Universidad Nacional [1]

De igual modo haciendo uso de una red RBF es posible determinar si el robot es próximo algún obstáculo. Los datos de entrada corresponden a un vector de 10 posiciones, donde los primeros nueve elementos indican si existe un obstáculo alrededor de la posición central y el décimo elemento el ángulo discretizado [1].

1.5.2 Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines: (Universidad Anahuac México Sur).

En este trabajo se presenta el uso del procesamiento de imágenes para determinar la trayectoria que debe seguir un robot manipulador a partir de una imagen específica para obtener el contorno del objeto.

Figura 4 IMAGEN DE OBJETOS A DIGITALIZAR



Fuente: Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines [2]

El contorno del objeto nos determina la trayectoria que debe seguir el robot, de ahí la importancia de la toma de una buena imagen para el procesamiento. Se toma como punto de partida figuras de color blanco sobre un espacio negro para facilitar el proceso de binarización y así determinar el contorno de la figura.

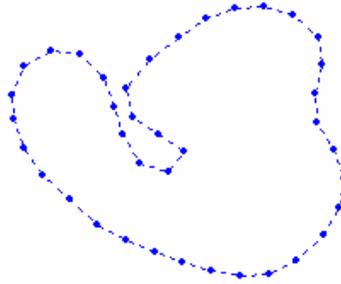
Figura 5 OBJETOS SEGMENTADOS



Fuente: Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines [2]

Con el objetivo de solucionar las interpolaciones para el robot y ayudar a suavizar el contorno del objeto hacen uso de interpolación por splines.

Figura 6 CONJUNTO DE PUNTOS CONSIDERADOS



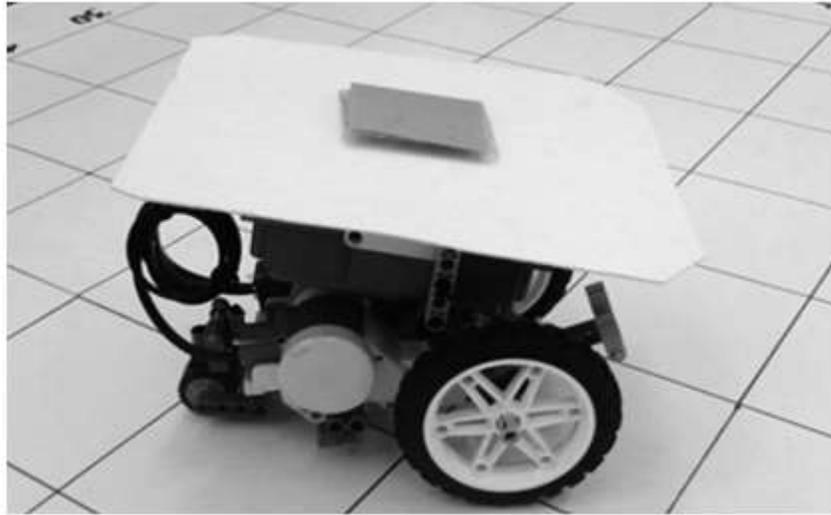
Fuente: Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines [2]

1.5.3 Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA: (Servicio Nacional de Aprendizaje SENA Regional Huila).

El principal antecedente fue el trabajo desarrollado por el ingeniero Adrián Fernando Chávarro que nació como una estrategia de transferencia de conocimientos para apoyar el programa de Formación Tecnológica en Integración de Automatismos Mecatrónicos del Centro de la Industria, la Empresa y los Servicios del SENA Regional Huila.

El proyecto consiste en llevar un robot de tracción diferencial de un punto inicial hasta un punto final. El robot es de tracción diferencial y fue construido con ayuda de la herramienta lego. El objetivo principal es realizar el modelamiento del robot con el fin de poder representarlo en una herramienta de software como es Matlab-Simulink.

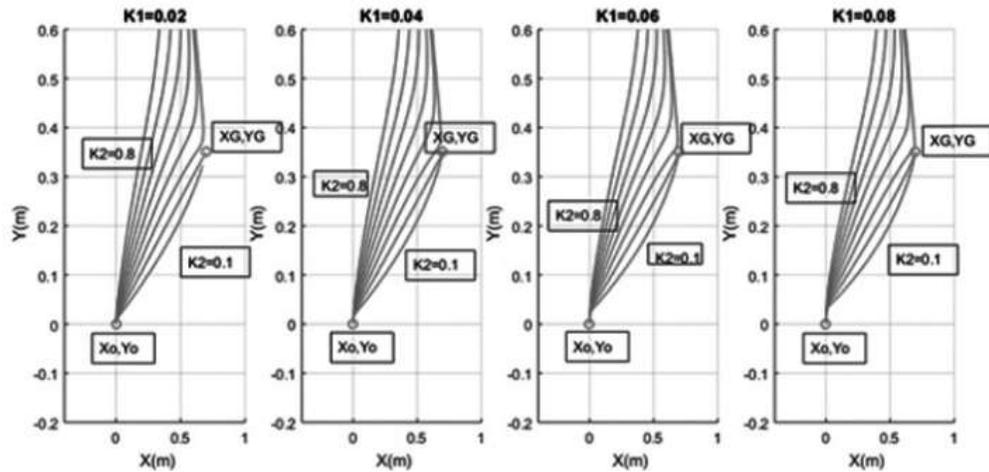
Figura 7 ROBOT MOVIL CON TRACCIÓN DIFERENCIAL UTILIZANDO LA HERRAMIENTA LEGO



Fuente: Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA [3]

El estudio del modelo arroja las variables que intervienen en el desplazamiento del robot, como son las velocidades de las ruedas derecha e izquierda. Las ecuaciones del desplazamiento cinemático son representadas como funciones dentro de los bloques en lazo cerrado del sistema, junto con el modelo matemático del robot móvil. Sin embargo este sistema de control a lazo cerrado debe encontrar el valor de los parámetros necesarios para lograr la estabilidad del sistema en lazo cerrado, K_1 y K_2 , estos están relacionados con las ecuaciones que describen la cinemática del robot y pueden ser obtenidos mediante simulación, es decir, al escoger diferentes parámetros la trayectoria del robot móvil puede ser simulada y evaluada determinando los valores más apropiados para los parámetros de entrada.

Figura 8 SIMULACIÓN DEL DESPLAZAMIENTO X,Y DEL ROBOT MOVIL PARA DIFERENTES VALORES DE K1 Y K2, UTILIZANDO EL CONTROL EN LAZO CERRADO



Fuente: Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA [3]

2. CAPITULO MARCO

TEÓRICO

El presente capítulo aborda algunos conceptos básicos sobre robots móviles y sus diferentes características, además de algunos conceptos sobre visión artificial y procesamiento de imágenes. Finalmente precisaremos algunos de los componentes con los que fue elaborado el prototipo y su función para el cumplimiento de los objetivos propuestos.

2.1 ROBOTS MÓVILES

En este apartado caracterizamos diferentes tipos de robots móviles, permitiendo de esta forma entender el porqué del uso de un sistema de tracción u otro.

2.1.1 VEHÍCULOS CON RUEDAS

Los vehículos con ruedas son una buena alternativa al momento de realizar tareas en lugares donde los terrenos son duros y sin obstáculos [4].

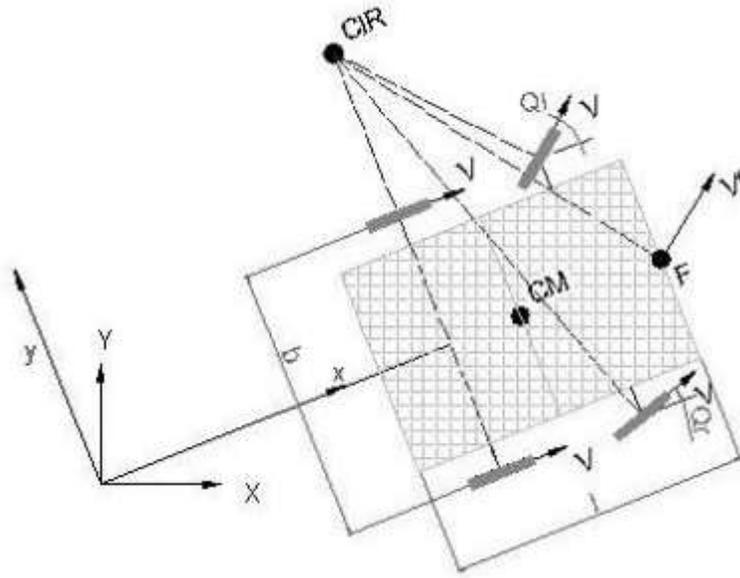
La limitación más significativa es que en terrenos blandos puede existir deslizamiento en el impulso, dependiendo de las características del terreno pueden presentarse también vibraciones. La tracción de las ruedas es ineficiente en este tipo de terrenos [5].

Los robots móviles emplean diferentes tipos de locomoción en función a la configuración de sus ruedas, brindándole características asociadas a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad [6]. La mayor maniobrabilidad la tienen los vehículos omnidireccionales, pues están en la capacidad de trasladarse simultánea e independientemente en cada eje del sistema de coordenadas y rotar según el eje perpendicular [6].

2.1.1.1 ACKERMAN O TIPO COCHE

El sistema se basa en dos ruedas traseras que se encargan de la tracción del robot, que se montan de forma paralela, las ruedas delanteras son del tipo direccionamiento, y se utilizan para seguir la trayectoria del robot. Gracias a su similitud con los vehículos convencionales recibe el nombre de tipo coche.

Figura 9 SISTEMA ACKERMAN



Fuente: Análisis y diseño del control de posición de un robot móvil con tracción diferencial [6]

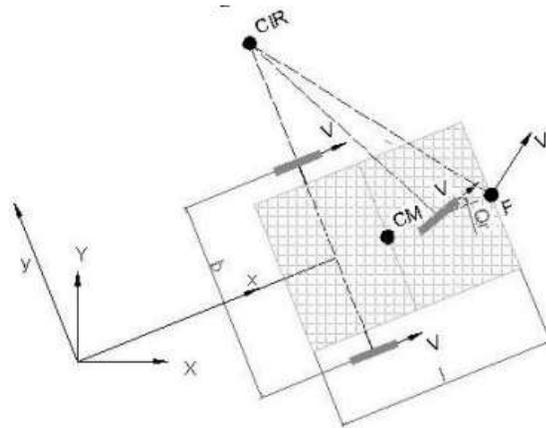
La rueda delantera interior gira un ángulo ligeramente superior a la rueda exterior, buscando que los ejes de prolongación de las ruedas delanteras se corten con el centro instantáneo de rotación (CIR), que se encuentra en el mismo punto del eje de prolongación de las ruedas traseras. Todo esto para eliminar el deslizamiento que provoca los sobre virajes del robot.

En los robots móviles tipo Ackerman se tienen dos ángulos de giros, uno para cada rueda delantera, lo que genera cierta dificultad al momento de realizar el control, por lo que en la mayoría de las ocasiones se unifican los ángulos de direccionamiento en uno solo.

2.1.1.2 TRICICLO CLÁSICO

Este sistema cuenta con una rueda delantera que cumple dos funciones, direccionamiento y tracción. La maniobrabilidad de este tipo es mayor a la del tipo Ackerman debido a que solo tiene una rueda de direccionamiento, pero esto a su vez causa que su desplazamiento no sea estable en terrenos difíciles. El centro de gravedad cumple una función clave, ya que este tiende a desplazarse cuando el vehículo se mueve en una pendiente, provocando una pérdida de tracción o incluso volcamiento.

Figura 10 CONFIGURACIÓN TIPO TRICICLO

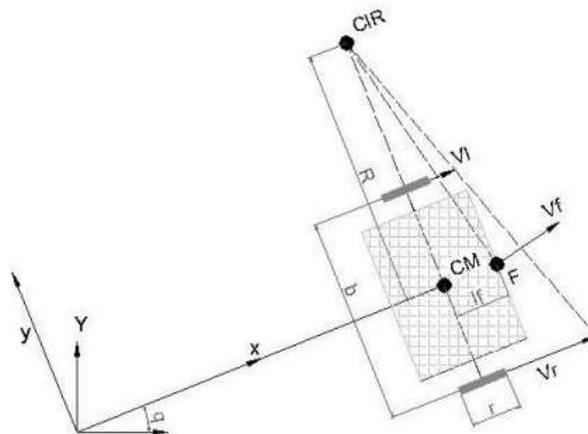


Fuente: Análisis y diseño del control de posición de un robot móvil con tracción diferencial [6]

2.1.1.3 TRACCIÓN DIFERENCIAL

Este tipo de sistema toma su nombre por la diferencia de velocidades de las ruedas laterales que le dan a su vez tracción y direccionamiento. La combinación del movimiento de las ruedas provoca el movimiento alrededor del CIR. Una de las ventajas que tiene este modelo de robot móvil es que puede cambiar su orientación sin realizar movimientos de traslación, a lo que se llama cambio de spin [6]. Las variables de control para este sistema son las velocidades angulares de las ruedas izquierda y derecha.

Figura 11 SISTEMA TRACCIÓN DIFERENCIAL

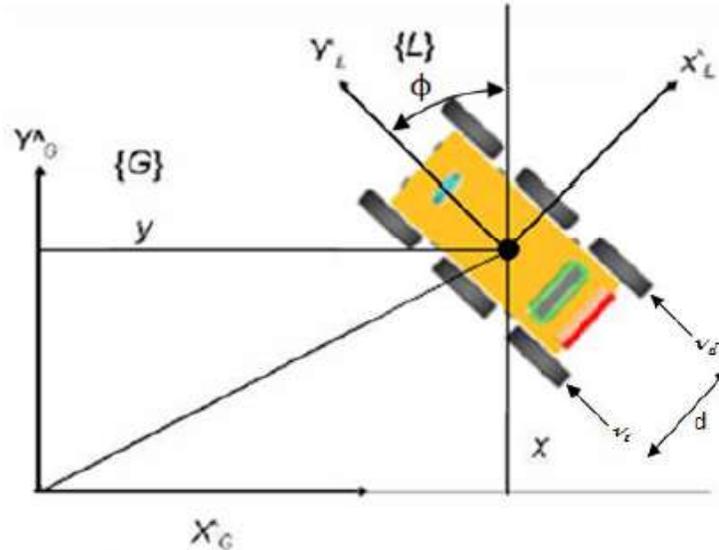


Fuente: Análisis y diseño del control de posición de un robot móvil con tracción diferencial [6]

2.1.1.4 SKID STEER

En cada lado del vehículo se posicionan igual número de ruedas que funcionan simultáneamente, el movimiento es el resultado de combinar las velocidades de las ruedas de la izquierda con las de la derecha. Este tipo de robot ha tenido gran aceptación y ha sido empleado en aplicaciones mineras e incluso en misiones de exploraciones espaciales.

Figura 12 ROBOT SKID STEER EN EL PLANO

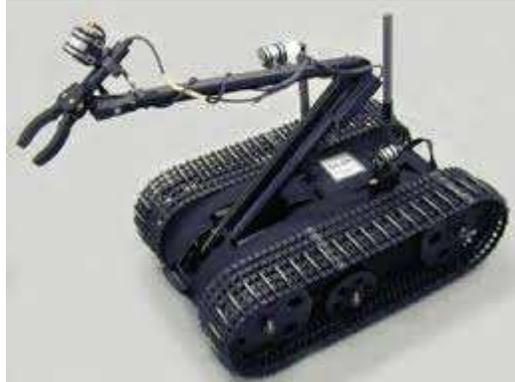


Fuente: DISEÑO, MODELAMIENTO Y SIMULACIÓN 3D DE UN ROBOT MÓVIL PARA EXPLORACIÓN DE TERRENOS [7]

2.1.1.5 PISTAS DE DESLIZAMIENTO

Estos son los modelos conocidos como tipo oruga en los que tanto la tracción y el direccionamiento se consiguen gracias a las pistas de deslizamiento. Puede considerarse un comportamiento similar al Skid steer. Este tipo de sistema con pistas deslizantes es muy conveniente usarlo en terrenos irregulares en los que presenta un buen rendimiento debido a que la resistencia al desgaste es mayor.

Figura 13 ROBOT CONFIGURACIÓN ORUGA

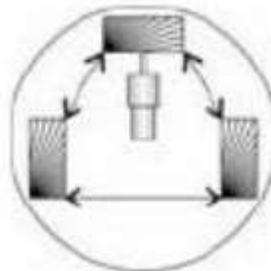


Fuente: DESARROLLO DE UNA APLICACIÓN PARA EL GUIADO AUTOMÁTICO DE UN VEHÍCULO ELÉCTRICO [9]

2.1.1.6 SÍNCRONAS

Consiste en la actuación simultánea de todas las ruedas, que giran de forma síncrona [6], en este modelo cada rueda es capaz de ser conducida y dirigida. Algunas configuraciones son:

Figura 14 CONFIGURACIÓN SÍNCRONA



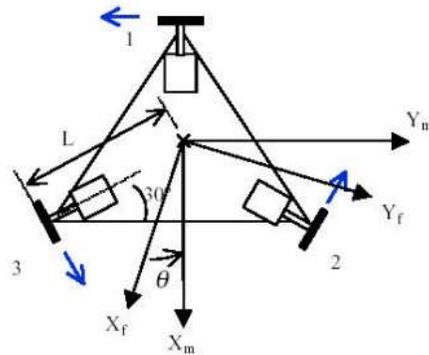
Fuente: Desarrollo de un sistema de control de un robot móvil [8]

- Tres ruedas direccionales se montan en los vértices de un triángulo equilátero muchas veces debajo de una plataforma cilíndrica.
- Todas las ruedas impulsan y giran al unísono.

2.1.1.7 TRACCIÓN OMNIDIRECCIONAL

Este sistema de tracción se basa en la utilización de tres ruedas directrices y motrices. Esta configuración tiene tres grados de libertad, por lo que puede realizar cualquier movimiento y posicionarse en cualquier posición y cualquier orientación [6].

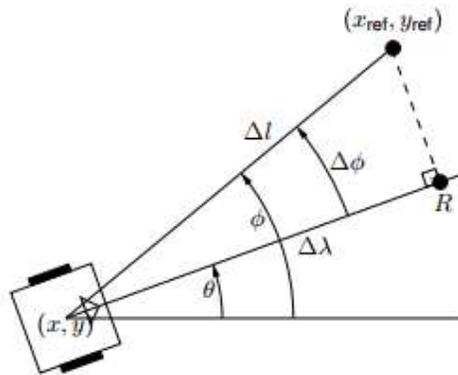
Figura 15 CONFIGURACIÓN OMNIDIRECCIONAL



Fuente: Análisis y diseño del control de posición de un robot móvil con tracción diferencial [6]

2.1.1.8 POSICIÓN Y ORIENTACIÓN DE UN ROBOT MÓVIL DE TRACCIÓN DIFERENCIAL

Figura 16 ANÁLISIS DE POSICIONAMIENTO DEL ROBOT



Fuente: POSITION AND ORIENTATION CONTROL OF A TWO-WHEELED DIFFERENTIALLY DRIVEN NONHOLONOMIC MOBILE ROBOT [11]

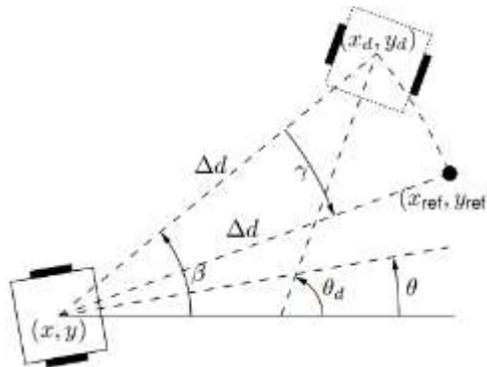
Como podemos apreciar en la figura 16 Δl es la distancia que hay entre el robot y el punto deseado en el plano cartesiano. El control del posicionamiento del robot se resolverá si aseguramos que Δl sea cero. Sin embargo l no se encuentra en el modelo, lo que hace necesario realizar el estudio en base de $\Delta \lambda$ y ϕ . $\Delta \lambda$ es la distancia hasta el punto R, que es el punto más cercano de la referencia deseada; ϕ es el ángulo que forma el vector que enlaza la posición del robot con la referencia deseada. De esa manera podemos definir $\Delta \phi$ como la diferencia entre el ángulo y la orientación del robot $\Delta \phi = \phi - \theta$ [11].

Con estas variables definidas se obtiene la ecuación 1, donde si $\Delta \lambda$ y $\Delta \phi$ son cero de igual modo Δl será cero. Entonces lograremos nuestra meta x_{ref} y y_{ref} .

$$\Delta l = \frac{\Delta \lambda}{\cos(\Delta \phi)}$$

(1)

Figura 17 ANÁLISIS DE LA ORIENTACION DEL ROBOT



Fuente: POSITION AND ORIENTATION CONTROL OF A TWO-WHEELED DIFFERENTIALLY DRIVEN NONHOLONOMIC MOBILE ROBOT [11]

La estrategia para calcular la referencia interna se presenta en la figura 17. La referencia (x_{ref}, y_{ref}) se calcula girando el vector de longitud Δd que apunta desde la posición del robot a la posición deseada en un ángulo de γ . γ es esencialmente igual a la diferencia entre la orientación final deseada (θ_d) y el ángulo para moverse a la posición deseada (β).

$$x_{ref} = x + \Delta d \cdot \cos(\beta - \gamma)$$

$$y_{ref} = y + \Delta d \cdot \sin(\beta - \gamma)$$

(2)

Si β y θ_d coinciden, $\gamma = 0$ y el robot va directamente a la posición final $(x_{ref}, y_{ref}) = (x_d, y_d)$. De lo contrario, siempre que el robot intente moverse a la referencia interior (x_{ref}, y_{ref}) la diferencia entre θ_d y β aumenta, γ disminuye y (x_{ref}, y_{ref}) tiende a (x_d, y_d) [11], ecuación 2.

Δd La distancia a la referencia desaseada y β , se calculan en referencia a las ecuaciones 3 y 4 respectivamente.

$$\begin{aligned} \Delta d &= \sqrt{(x_d - x)^2 + (y_d - y)^2} \\ &= \sqrt{\Delta x_d^2 + \Delta y_d^2} \end{aligned}$$

(3)

$$\beta = \tan^{-1}\left(\frac{y_d - y}{x_d - x}\right) = \tan^{-1}\left(\frac{\Delta y_d}{\Delta x_d}\right)$$

(4)

2.2 VISION ARTIFICIAL

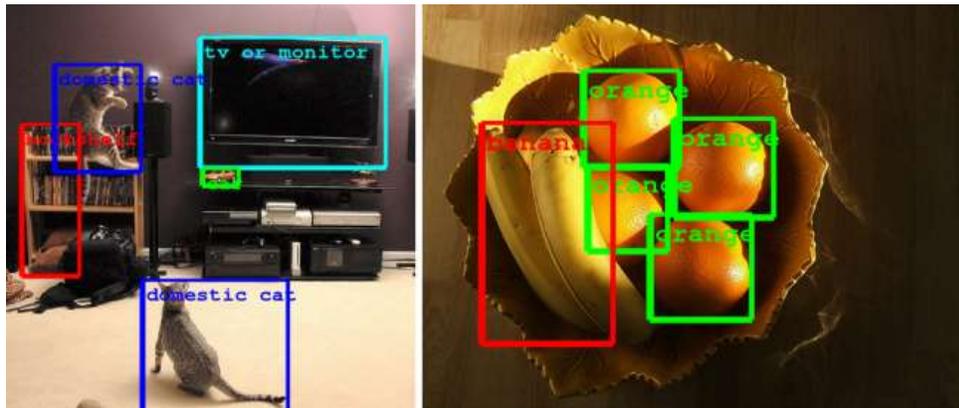
La visión artificial abarca cualquier proceso óptico mediante el cual un sistema inteligente es capaz de extraer información de un entorno para su interpretación mediante el uso de la computadora [10]. Los sistemas de visión artificial se enfocaban principalmente en la visión humana, sin embargo, el sistema visual humano es muy complejo de estudiar debido al desconocimiento de las áreas precisas donde el cerebro que se encarga de procesar toda la información visual.

El comienzo de cualquier proceso de reconocimiento es la cuantificación digital del entorno. El sistema ha de ser capaz de procesar la información obtenida y aplicar los algoritmos adecuados para su procesamiento y correcta interpretación. Ésta resultaría imposible sin un especificado pre-procesado de la imagen, debido a que los patrones a reconocer pasan a ser infinitos. Muchas veces no toda la información necesaria para completar una tarea está contenida en una imagen [10].

2.2.1 RECONOCIMIENTO DE OBJETOS

Un procesamiento de la imagen captada por un sistema de visión artificial es fundamental para un reconocimiento correcto y eficaz. Al procesar la imagen se elimina la mayor parte de información irrelevante para reconocer el objeto, y se extraen únicamente aquellos parámetros que permiten una identificación e interpretación inconfundible del objeto [11]. La selección de estos parámetros es una de las tareas más críticas en el reconocimiento. El tipo de procesamiento empleado también depende del propósito del sistema y del tipo de objetos que se deseen extraer.

Figura 18 IDENTIFICAR CON CLARIDAD OBJETOS



Fuente: Análisis digital de imágenes y reconocimiento de objetos [34]

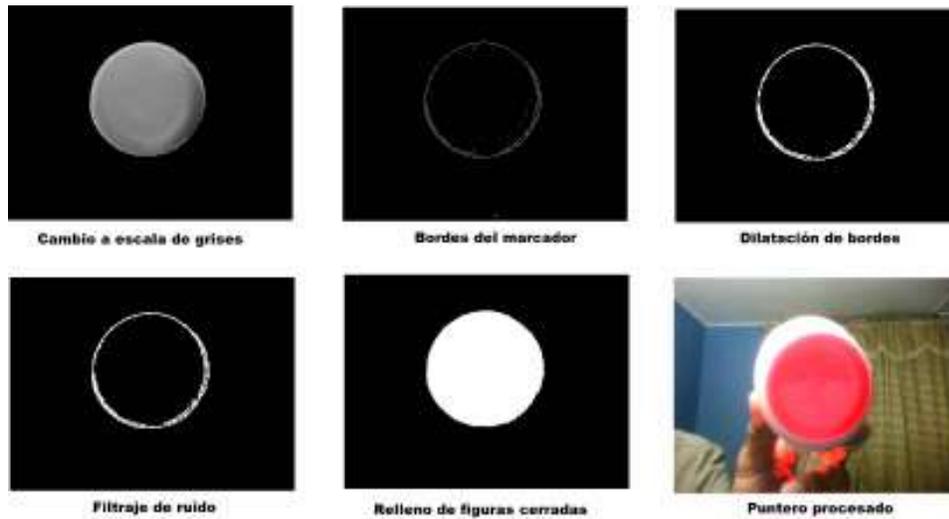
2.2.1.1 IMAGEN DIGITAL

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica. Dependiendo si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen de mapa de bits, la cual está formada por puntos (píxeles), que son unidad de información de color y luminosidad, o imágenes vectoriales, la cual está formada por objetos gráficos que se definen matemáticamente, Las formas están construidas por vectores, que son representaciones de ecuaciones matemáticas [14].

2.2.2. PROCESAMIENTO DIGITAL DE IMÁGENES

El procesamiento digital de una imagen es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. Puede ser subdividido en seis áreas principales y están agrupados de acuerdo con la complicación y delicadeza que lleva su implementación. Estas son: captura y adquisición, preprocesamiento, segmentación, descripción, reconocimiento e interpretación. No todas las aplicaciones requieren de todos los procesos y depende de la complejidad del problema que se va a resolver [12].

Figura 19 PROCESAMIENTO DIGITAL DE IMÁGENES EN MATLAB



Fuente: Towards a Smart Classroom: Development of an Augmented Reality Application for Education and Tele-education [36]

La utilidad del procesamiento de imágenes es muy amplia y abarca muchos campos. Un ejemplo son las imágenes obtenidas con fines de diagnóstico médico. Otro ejemplo son las imágenes aéreas obtenidas para realizar exámenes del terreno. Mediante este método se pueden analizar los recursos naturales, las fallas geológicas del terreno, etcétera [35].

2.3 MODELO DE COLOR

Un modelo de color establece un conjunto de colores primarios a partir de los que, mediante mezclas, se pueden obtener otros colores hasta cubrir todo el espectro visible, además del propio blanco, negro y grises, y aún más. Por ejemplo, hay colores, como el marrón o el magenta, que no están presentes en el espectro visible, y es nuestro cerebro el que lo interpreta a partir de la combinación de ondas con diferentes longitudes.

Los modelos de color más comunes son RGB (utilizado en monitores) y CMYK (utilizado para impresión) [42].

2.3.1 MODELO RGB

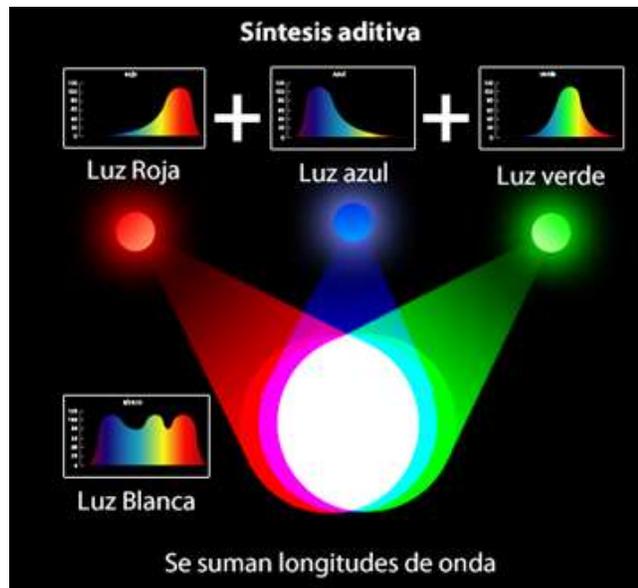
Los conos son los sensores del ojo humano responsables de la visión en color. Por medio de evidencias experimentales se sabe que entre 6 o 7 millones de conos pueden ser divididos en tres categorías de sensores que corresponden de manera burda al rojo, Verde y azul. Aproximadamente 65% de todos los conos son sensibles a la luz roja, 33% a la luz verde y sólo un 2% a la luz azul (pero los conos azules son los más sensibles de todos). Debido a estas características de absorción, los colores son vistos como combinaciones variables de

los llamados colores primarios de luz o colores secundarios de pigmento rojo (R), verde (G) y azul (B).

Cada pixel de las imágenes tomadas por la herramienta de software matemático MATLAB está formada por un número de bits, en relación su tamaño. El valor de estos bits entre 0-255 de los tres canales, determina la profundidad de los pixeles y dan un valor representativo para distinguir los colores presentes en la imagen [15]. RGB es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios.

Es un sistema de color muy usado para televisión y páginas web [37].

Figura 20 MODELO RGB



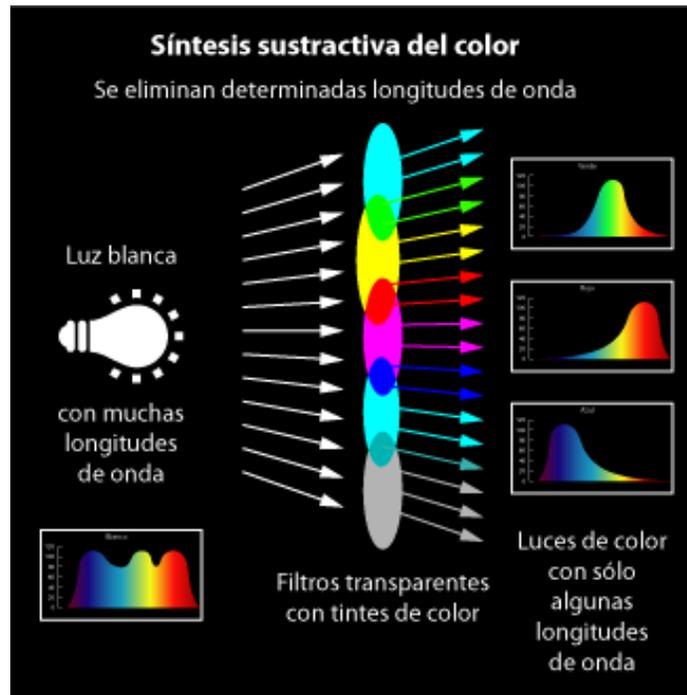
Fuente: Síntesis aditiva [38]

2.3.2 MODELO CMYK

El modelo CMYK (acrónimo de Cyan, Magenta, Yellow y Key) es un modelo de Colores sustractivo que se utiliza en la Impresión en colores. La mezcla de colores CMY ideales es sustractiva (pues imprimir cian, magenta y amarillo en fondo blanco resulta en el color negro). El modelo CMYK se basa en la absorción de la luz. El color que presenta un objeto corresponde a la parte de la Luz que incide sobre éste y que no es absorbida por el objeto.

El cian es el opuesto al rojo, lo que significa que actúa como un filtro que absorbe dicho color (-R +G +B). Magenta es el opuesto al verde (+R -G +B) y amarillo el opuesto al azul (+R +G -B).

Figura 21 MODELO CMYK



Fuente: Síntesis sustractiva [39]

2.3.3 MODELO HSV

Espacio de color HSV. Representación tridimensional del color basado en los componentes de tinte, matiz o tonalidad (Hue, en inglés), saturación (Saturation) y brillo o valor (value).

Fue definido en 1978 por Alvy Ray Smith. A diferencia del modelo RGB ampliamente usado en los monitores, televisores, etc., si bien las coordenadas de aquel son euclidianas; el color HSV sigue una representación más parecida a las coordenadas cilíndricas [40]. Además, es una representación más cercana a la forma en que los humanos perciben los colores y sus propiedades, pues se agrupan las tonalidades de color, lo cual es distinto al caso RGB donde los colores no están necesariamente tan agrupados [40].

El modelo de color HSV es una transformación no lineal del modelo RGB en coordenadas cilíndricas de manera que cada color viene definido por las siguientes dimensiones:

- Tinte o matiz: Ángulo que representa el matiz, normalmente definido entre 0° y 360°.
- Saturación: Nivel saturación del color, dado entre 0 y 1, 0 representa sin saturación alguna (blanco), hasta 1 que sería el matiz en toda su intensidad. Es común también darlo en percentiles 0%-100%.

- Brillo: Nivel del brillo entre 0 y 1. 0 es negro; 1, blanco. Al igual que la saturación puede darse en porcentajes entre 0% y 100%. De esta forma el 50% indica el nivel medio o normal del brillo del color.

Figura 22 MODELO HSV



Fuente: La teoría del color ¿Qué es el color? ¿Cuáles son los colores primarios y secundarios? [41]

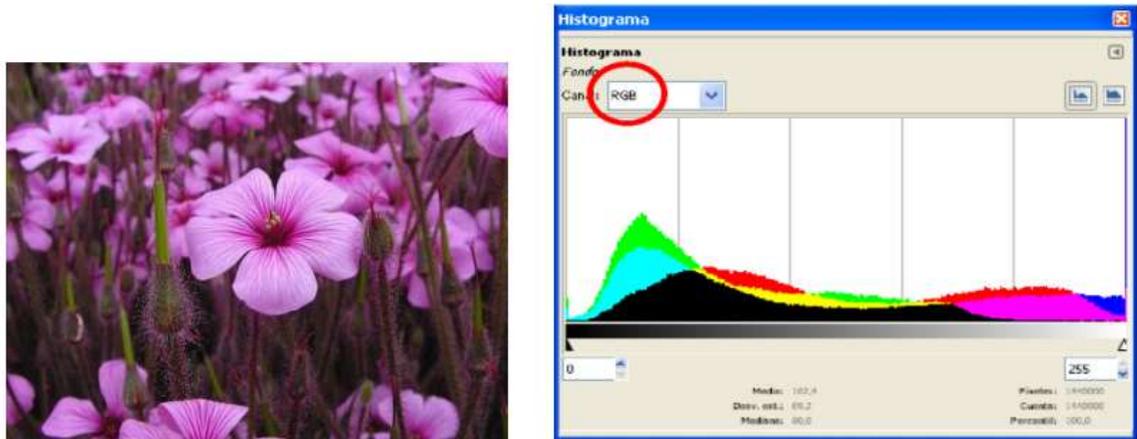
2.4 HISTOGRAMA

El histograma es la representación gráfica de las frecuencias relativas con las que aparecen los distintos colores en una determinada imagen. En general se representa como un gráfico de barras en el que las abscisas son los distintos colores de la imagen y las ordenadas la frecuencia relativa con la que cada color aparece en la imagen. El histograma proporciona información sobre el brillo y el contraste de la imagen, y puede ser utilizado para ajustar estos parámetros, eliminar ciertas tonalidades molestas, etc. [43]

Si el conjunto de posibles valores es suficientemente pequeño, cada uno de esos colores pueden ser puestos en un rango por si mismos; entonces el histograma no es más que el conteo de los píxeles que tiene cada posible color. Más a menudo, el espacio es dividido en un número apropiado de rangos, con frecuencia arreglados como una rejilla regular, cada una conteniendo muchos valores de colores similares. El histograma de color puede también ser representado y mostrado como una función suave definida sobre el espacio de color que aproxima el conteo de píxeles. [44]

Para las imágenes en color es posible obtener el histograma de canales de color, que cabe entenderlo como la superposición de 3 histogramas distintos, los que corresponden a cada uno de los canales que forman la imagen u obtener el histograma individual de cada uno de los canales. [45]

Figura 23 HISTOGRAMA DE LOS TRES CANALES RGB



Fuente: El Histograma de una imagen digital [45]

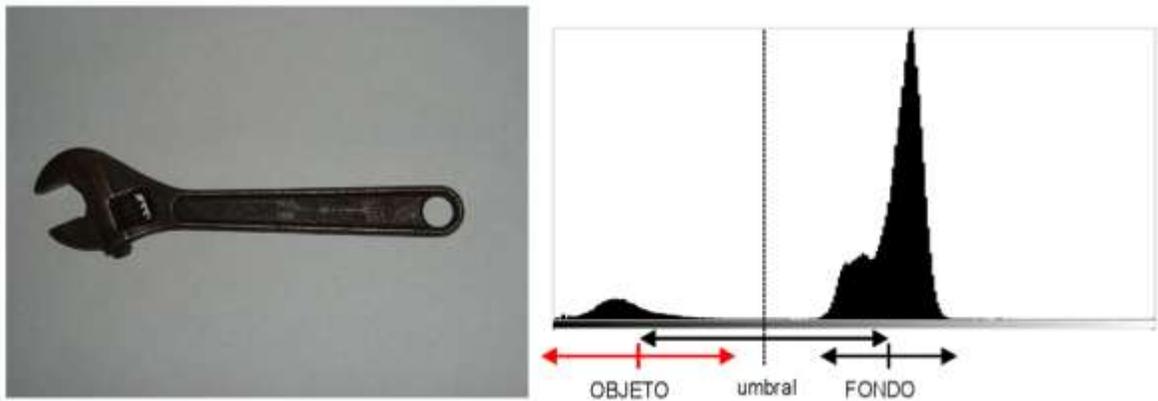
Figura 24 HISTOGRAMA DE LOS CANALES RGB INDIVIDUALES



Fuente: El Histograma de una imagen digital [45]

El histograma resulta una herramienta de gran interés también en tareas de análisis automático de imágenes, como las que se llevan a cabo en el área de la visión por computador. La Figura 25 muestra un ejemplo típico. Para una aplicación de inspección automática, se trata de determinar los píxeles de la imagen que corresponden a la herramienta, que se sabe depositada sobre un fondo de color más claro, bastante uniforme.

Figura 25 UMBRALIZACION DE UNA IMAGEN CON AYUDA DEL HISTOGRAMA



Fuente: El Histograma de una imagen digital [45]

El histograma muestra dos grandes agrupaciones de niveles que aparecen con cierta frecuencia en la imagen, una correspondiente al objeto oscuro y otra debida al fondo claro. Una vez determinadas estas dos agrupaciones, el nivel de gris equidistante entre ambas, resultaría un nivel de umbral adecuado. Obteniendo como resultado, en color rojo, todos los píxeles que se consideran pertenecientes a la herramienta por ser más oscuros que el nivel de umbral. [45]

Figura 26 RESULTADO DE LA UMBRALIZACION



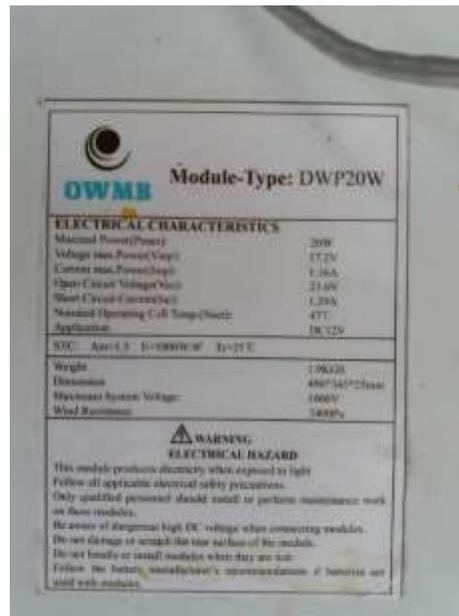
Fuente: El Histograma de una imagen digital [45]

2.5 PROTOTIPO

2.5.1 PANEL SOLAR

La idea principal del equipo de trabajo que desarrollo la idea del robot para el volteo de café en planchón es que los rayos solares que afectan a la persona encargada de realizar la tarea sean aprovechados por un panel solar para cargar la batería que mueve el prototipo, evitando así que el trabajador desarrolle cualquier enfermedad asociada a la piel.

Figura 27 CARACTERÍSTICAS ELÉCTRICAS DEL PANEL SOLAR



The image shows a label for a solar panel with the following text:

OWMB Module-Type: DWP20W

| ELECTRICAL CHARACTERISTICS | |
|------------------------------------|--------|
| Maximal Power(Pmax): | 20W |
| Voltage max.Powert(Vmp): | 17.2V |
| Current max.Powert(Isc): | 1.16A |
| Open Circuit Voltage(Voc): | 21.6V |
| Short Circuit Current(Isc): | 1.23A |
| Maximal Operating Cell Temp.(Max): | 47°C |
| Application: | 18-32V |

STC: Am:1.5, I:1000W/m², Tc=25°C

| | |
|-------------------------|--------------|
| Weight: | 1.06KG |
| Dimension: | 480*340*25mm |
| Maximum System Voltage: | 1000V |
| Wind Resistance: | 1400Pa |

WARNING
ELECTRICAL HAZARD
This module produces electricity when exposed to light.
Follow all applicable electrical safety precautions.
Only qualified personnel should install or perform maintenance work on these modules.
Be aware of dangerous high DC voltage when connecting modules.
Do not attempt to connect the two surfaces of the module.
Do not handle or install modules when they are wet.
Follow the factory manufacturer's recommendations if batteries are used with modules.

Fuente: Fotografía Autor

2.5.2 CONTROLADOR DE CARGA SOLAR

Para recargar la batería y alimentar los circuitos que hacen parte del robot móvil hacemos uso de un controlador de carga solar marca Paco 12V/24V 8A. Es un controlador serie de bajas pérdidas, se ajusta automáticamente al nivel de tensión 12V o 24V, la corriente de cortocircuito de entrada es 8A y el voltaje máximo de colector solar es de 47V.

Figura 28 CONTROLADOR DE CARGA SOLAR

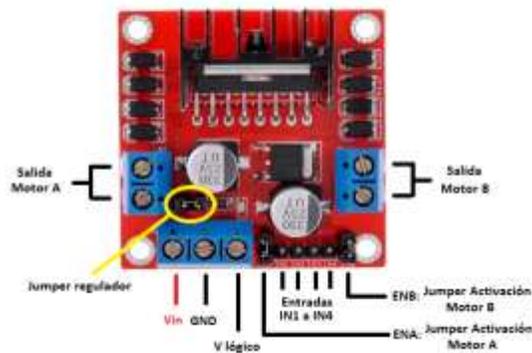


Fuente: Fotografía Autor

2.5.3 L298N

El módulo controlador de motores L298N puente H permite controlar la velocidad y la dirección de dos motores de corriente continua o un motor paso a paso, gracias a que cuenta con 2 puente H. El rango de tensiones en el que trabaja este módulo va desde 3V hasta 35V, y una intensidad de hasta 2A [16].

Figura 29 CONTROLADOR DE MOTORES L298N



Fuente: EL MÓDULO CONTROLADOR DE MOTORES L298N [16]

Si el jumper del regulador está cerrado se activará el regulador de tensión del L298N, y en V lógico se tendrá una salida de 5V [16]. Si quitamos el jumper desactivaremos el regulador y será necesario alimentar la parte lógica del módulo, así que tendremos que meter una tensión de 5V por la conexión V lógico para que el módulo funcione.

El regulador sólo funciona con tensiones hasta 12V en V_{in} , por encima de este valor tendremos que quitar el jumper y alimentar la parte lógica del módulo desde otra fuente [16].

Las salidas para los motores A y B nos darán la energía para mover los motores. Se debe tener en cuenta la polaridad al conectarlos, para que giren en el sentido deseado.

Los pines IN1 e IN2 nos sirven para controlar el sentido de giro del motor A, y los pines IN3 e IN4 el del motor B. Funcionan de forma que, si IN1 está a HIGH e IN2 a LOW, el motor A gira en un sentido, y si está IN1 a LOW e IN2 a HIGH lo hace en el otro. Y lo mismo con los pines IN3 e IN4 del motor B [16].

Para controlar la velocidad de giro de los motores tenemos que quitar los jumpers y usar los pines ENA y ENB. Los conectaremos a dos salidas PWM de Arduino de forma que le enviemos un valor entre 0 y 255 que controle la velocidad de giro. Si tenemos los jumpers colocados, los motores girarán a la siempre a la misma velocidad [16].

2.5.4 MOTORES

Los motores que se escogieron para desarrollar el prototipo son motorreductores de 12V de corriente continua, el cual cuenta con una caja de reducción de relación 50:1 y un Encoder de cuadratura. La salida de la caja reductora es de 200 RPM, La corriente de trabajo es de 300mA sin carga y de 5A a plena carga. Posee un torque de 12Kg-cm y la resolución del Encoder es de 64 cuentas por revolución (CPR), que corresponde a 3200 CPR del eje de salida de la caja de engranajes.

Figura 30 MOTORREDUCTOR 50:1 37Dx70L mm con 64 CPR Encoder



Fuente: Motorreductor 50:1 37Dx70L mm con 64 CPR Encoder [17]

El eje de salida es en D, con 6mm de diámetro y 1.52cm de longitud y tiene un conector de 6 cables, dos para la alimentación del motor y los otros 4 para la alimentación y las salidas

de señal del Encoder. Las medidas del motorreductor son 37mm de diámetro, 66mm de longitud, y pesa 220g.

Figura 31 CONEXIONES MOTORREDUCTOR

| Color | Función |
|----------|--|
| rojo | potencia del motor (se conecta a un terminal del motor) |
| Negro | potencia del motor (se conecta a la otra terminal del motor) |
| Verde | codificador GND |
| Azul | codificador Vcc (3.5 - 20 V) |
| Amarillo | salida del codificador A |
| Blanco | salida codificador B |

Fuente: Motorreductor 50:1 37Dx70L mm con 64 CPR Encoder [17]

El Encoder del motor es de efecto hall y este sensor requiere una tensión de entrada, Vcc, entre 3.5 y 20 V y consume un máximo de 10 mA. Las salidas A y B son ondas cuadradas de 0 V a Vcc aproximadamente 90 ° desfasadas [17]. La frecuencia de las transiciones le indica la velocidad del motor y el orden de las transiciones le indica la dirección.

La siguiente figura de osciloscopio que brinda el fabricante muestra las salidas A y B del Encoder (amarillo y blanco) utilizando una tensión del motor de 12 V y el sensor Hall Vcc de 5 V [17]:

Figura 32 SALIDA EN CUADRATURA DEL ENCODER EFECTO HALL

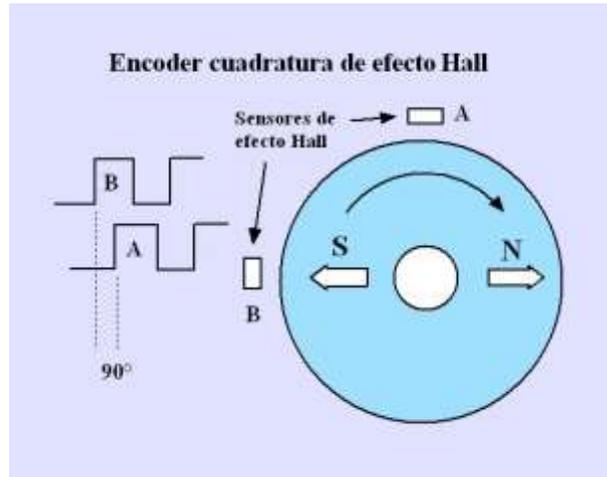


Fuente: Motorreductor 50:1 37Dx70L mm con 64 CPR Encoder [17]

Los sensores de efecto Hall son transductores que generan un voltaje de salida en respuesta a la presencia de un campo magnético variable. Las aplicaciones típicas son en sensores de proximidad, medición de corriente y medición de velocidad en motores de corriente directa [47].

Para aplicaciones de cálculo de velocidad y posición, los sensores son colocados con una separación de 90° , con respecto al círculo completo de giro del eje del motor. Un imán de ferrita induce en los sensores las señales A y B mostradas en la figura 33. Ambas señales están desfasadas 90° , debido a la posición de los sensores con respecto al círculo de giro.

Figura 33 FUNCIONAMIENTO SENSOR EFECTO HALL



Fuente: Principios de funcionamiento del Encoder cuadratura de efecto Hall para medición de velocidad en motores de corriente directa [47]

Principios de funcionamiento: en el ejemplo de la imagen, al girar el imán de ferrita con 2 polos, induce voltajes en los sensores de efecto Hall.

2.5.5 BATERÍA

La batería con la que cuenta el vehículo es de 12V a 7A de larga duración, tiene un peso de 2,15Kg y es usada en UPS (Sistema de alimentación ininterrumpida), EPS (Emergency Power System), iluminación de emergencia, equipos de comunicaciones y marinos.

Figura 34 BATERÍA 12V 7A

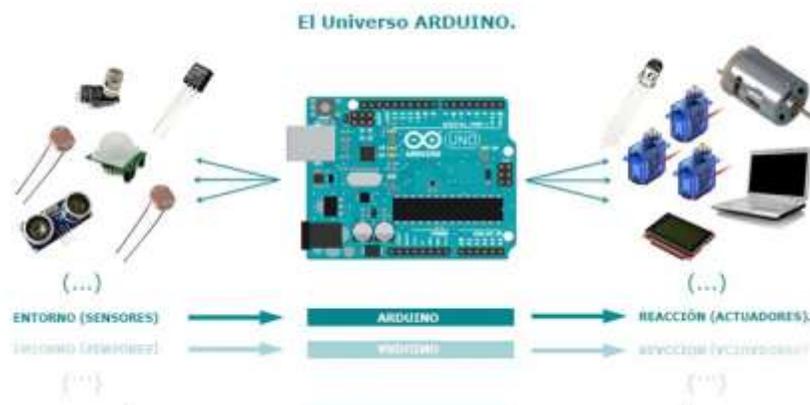


Fuente: KELCOM INTERNATIONAL [20]

2.5.6 ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

Figura 35 ARDUINO



Fuente: Aprendiendo arduino [19]

Arduino puede medir el entorno mediante la recepción de entradas desde una variedad de sensores y puede controlar su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo, con Flash, Processing, MaxMSP, etc.) [18].

Algunas ventajas de arduino son:

- **Economía:** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras [18].
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows [18].
- **Programación simple y clara:** El entorno de programación de Arduino es fácil de usar, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también [18].

- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados [18].
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo [18].

2.5.7 BLUETOOTH

Para comunicar el prototipo del robot con la computadora y usando los recursos con los que cuenta el Tecnoparque se decidió usar un módulo bluetooth HC-06, que es un dispositivo que es realmente económico y de fácil uso [22].

Figura 36 MODULO BLUETOOTH



Fuente: Modulo HC-06 [21]

Este módulo puede ser configurado usando algunos comandos AT, no da mucho acceso al firmware como el HC-05 pero también puede configurarse para satisfacer las necesidades de la aplicación [22].

- Prueba de funcionamiento:
Enviar: AT
Recibe: OK
- Configurar el Baudrate:
Enviar: AT+BAUD<Numero>
El parámetro número es un carácter hexadecimal de '1' a 'c' que corresponden a los siguientes Baud Rates: 1=1200, 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200, 9=230400, A=460800, B=921600, C=1382400
Recibe: OK<Baudrate>
- Configurar el Nombre de dispositivo Bluetooth:

Enviar: AT+NAME<Nombre>

Recibe: OKsetname

- Configurar el código PIN de emparejamiento:

Enviar: AT+PIN<pin de 4 dígitos>

Recibe: OK<pin de 4 dígitos>

- Obtener la versión del firmware:

Enviar: AT+VERSION

Recibe: Linvor1.8

2.5.8 CÁMARA

La cámara con la que se van a tomar las imágenes es la Microsoft LifeCam HD-3000, que cuenta con las siguientes características:

- Vídeo en auténtica alta definición de 720p, Graba vídeo en auténtica alta definición hasta 30 fps.
- Panorámica 16:9, Ofrece grabación de video cinematámico.
- Tecnología TrueColor, La tecnología TrueColor proporciona automáticamente imágenes de vídeo de colores brillantes y vívidos en casi todas las situaciones de iluminación.
- Micrófono con eliminación de ruidos, Sonido de máxima nitidez y micrófono unidireccional integrado con eliminación de ruidos acústicos.
- Base de acoplamiento universal, Funciona con el equipo portátil o el de escritorio.

Figura 37 CÁMARA MICROSOFT HD-3000



Fuente: Microsoft, LifeCam HD-3000 [48]

3. CAPITULO METODOLOGÍA

El presente capítulo explicara la construcción del prototipo del robot, así como el estudio de su modelo cinemático. El procesamiento de las imágenes capturadas y el funcionamiento de la interfaz gráfica de usuario.

3.1 ROBOT MÓVIL

Para empezar, el prototipo del robot era tipo Skid steer y ya estaba construido por los aprendices Sena, sin embargo, este robot móvil presentaba problemas en su locomoción debido al peso de la estructura que sostenía los paneles solares afectando el agarre de las ruedas con el terreno. Además, los motores con lo que se desplazaba este prototipo no contaban con circuitos que permitieran medir su velocidad lineal.

Figura 38 PRIMER MODELO DE AGROBOT EN EXPO-RURALES SENA



Fuente: Ministerio del trabajo [23].

Con los problemas presentados en el primer prototipo del vehículo se desarrolló un nuevo prototipo en el cual pudiéramos usar motores con Encoder con el objetivo de medir las velocidades lineales de la rueda derecha e izquierda, además este nuevo vehículo y su estructura permite usar un panel solar de mayor eficiencia y no un conjunto de paneles solares pequeños como se planteaba en el primer prototipo.

En búsqueda de una estructura para el vehículo económica y eficiente, se construye el robot móvil con tubos de PVC. El tamaño del robot fue adecuado para que tuviera el tamaño del panel solar, obteniendo el espacio suficiente para ubicar la batería y el cargador solar, problemas también presentes en el primer diseño.

Esta nueva estructura permite además adecuar los motores con Encoder y para optimizar el uso de los recursos del Tecnoparque se usaron las mismas llantas usadas del primer modelo del robot.

Figura 39 MODELO FINAL AGROBOT ECO



Fuente: Autor

Aunque la implementación y evaluación de la estrategia trol son en un espacio controlado de fondo uniforme, el objetivo es que Agrobot Eco se pueda desplazar por un patio de secado de café. De ahí que los aprendices Sena adaptaran al robot para cada de las ruedas una protección con escobillas y evitar que el café tenga contacto con las ruedas y esto pueda afectar en su calidad.

El modelo final del prototipo de Agrobot Eco fue el siguiente y consta de:

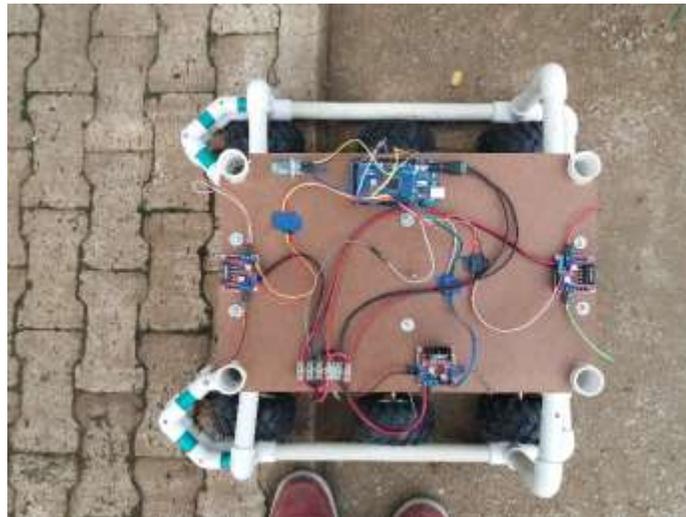
- 6 motores con Encoder de efecto hall.
- 3 L298N puente H.
- 1 modulo bluetooth HC-06.
- 1 Arduino Mega.
- 1 batería 12V 7.2Ah
- 1 cargador solar 12V/24V.
- 1 panel Solar 12V.

Figura 40 AGROBOT ECO



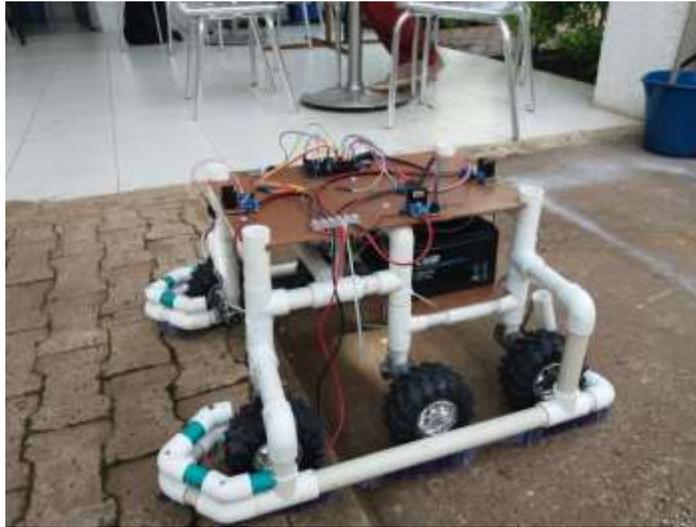
Fuente: Autor

Figura 41 AGROBOT ECO, CIRCUITOS



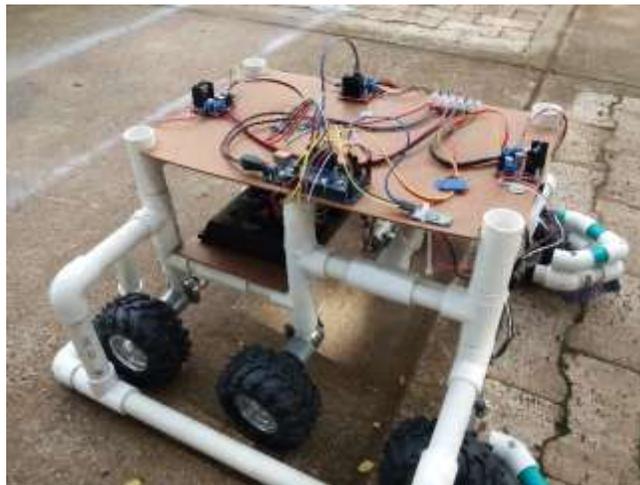
Fuente: Autor

Figura 42 AGROBOT ECO, ESTRUCTURA



Fuente: Auto

Figura 43 AGROBOT ECO, ESTRUCTURA



Fuente: Autor

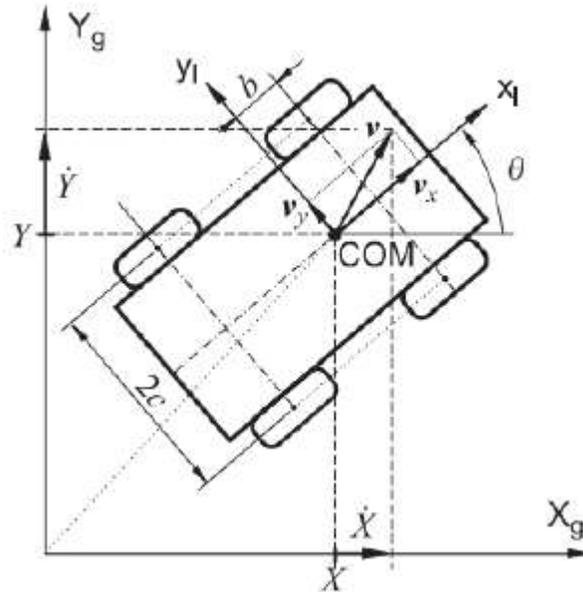
3.2 CONTROL CINEMÁTICO

La estrategia de control a implementar es basada en el estudio cinemático de un robot de tracción de diferencial, pero aplicada a un robot Skid steer al cual corresponde Agrobot Eco.

Para empezar el estudio del modelado del robot se deben conocer algunas variables relacionadas a las dimensiones del robot. Las medidas que interesan en el proceso son: la

distancia entre las ruedas, $2C$ en la gráfica o L , y el radio de estas, en algunos casos lo llaman b en la gráfica o R .

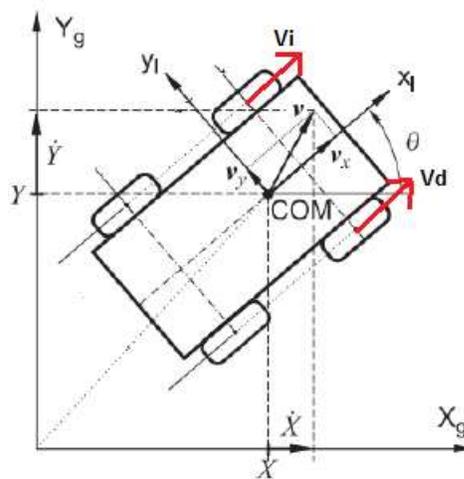
Figura 44 POSICIONAMIENTO DE UN ROBOT SKID STEER



Fuente: Ecuaciones cinemáticas de un robot móvil de 4 ruedas tipo “Skid-steer” [24]

Para lograr un movimiento controlado se debe tener dominio sobre las velocidades de las ruedas derecha e izquierda. El objetivo del estudio del modelo del robot es relacionar estas velocidades con los estados del sistema (X, Y, θ) [25].

Figura 45 VELOCIDADES DERECHA E IZQUIERDA DEL ROBOT



Fuente: Ecuaciones cinemáticas de un robot móvil de 4 ruedas tipo “Skid-steer” [24]

Para que el robot sea capaz de desplazarse en línea recta las velocidades de sus ruedas deben ser iguales, por lo que se define la velocidad lineal del robot como:

$$\mathbf{v} = R \frac{V_d + V_i}{2}$$

(5)

Ahora bien, para que el robot pueda rotar sobre su centro de masa, las velocidades de las ruedas deben tener la misma magnitud, pero signo diferente. De este modo se puede definir la velocidad angular del robot como:

$$\mathbf{w} = R \frac{V_d - V_i}{L}$$

(6)

Una vez ubicado el robot en el plano y encontradas las variables que hacen parte del modelo, se hallan las ecuaciones que definen la dinámica del movimiento del robot en cada eje.

$$\begin{aligned} \dot{x} &= v * \cos\theta \\ \dot{y} &= v * \text{sen}\theta \\ \dot{\phi} &= w \end{aligned}$$

(7)

Debido a que no en todo momento el robot se encuentra alineado con el eje global, se plantea una matriz de rotación. Donde se reemplazan las velocidades lineares y angulares obtenidas [25].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R \frac{V_d + V_i}{2} \\ R \frac{V_d - V_i}{L} \end{bmatrix}$$

(8)

Con el objetivo de obtener el modelo cinemático directo del robot se multiplican las matrices de rotación y la matriz de velocidades, obteniendo una ecuación matricial que representa las velocidades en X, Y y angular ϕ del prototipo en relación con a la variación de las velocidades angulares de las ruedas derecha e izquierda [25].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\cos\theta}{2} \\ R \frac{\sen\theta}{2} & R \frac{\sen\theta}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} W_d \\ W_i \end{bmatrix}$$

(9)

Para lograr el desplazamiento deseado del robot, se deben alterar las señales de entrada del sistema que son las velocidades angulares de las ruedas. Por lo tanto, se busca con la ayuda de la cinemática inversa una relación matemática que pueda determinar estas velocidades en función de una posición y orientación de entrada.

Encontrar esta relación matemática significa determinar la pseudoinversa de la matriz inicial del modelo, ya que esta es una matriz no cuadrada [25].

$$A^+ = (A^T * A)^{-1} * A^T$$

$$A = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\cos\theta}{2} \\ R \frac{\sen\theta}{2} & R \frac{\sen\theta}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix}$$

(10)

La matriz de los coeficientes es llamada "A" a partir de la cual se determina su pseudoinversa "A+".

$$A^+ = \begin{bmatrix} \frac{\cos\theta}{R} & \frac{\sen\theta}{R} & \frac{L}{R} \\ \frac{\cos\theta}{R} & \frac{\sen\theta}{R} & -\frac{L}{R} \end{bmatrix}$$

41

(11)

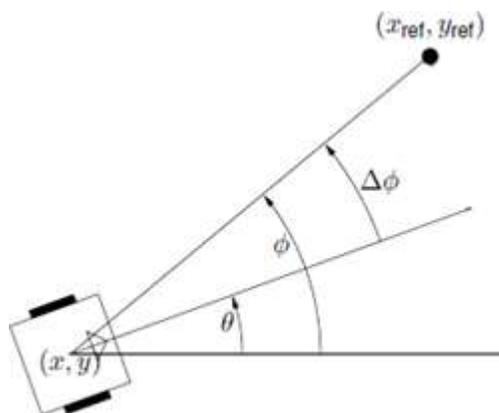
Finalmente se puede determinar la cinemática inversa del robot como:

$$\begin{bmatrix} \mathbf{W}_d \\ \mathbf{W}_i \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & \frac{L}{R} \\ \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & -\frac{L}{R} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}$$

(12)

Para poder completar el análisis de las variables de entrada para el controlador debemos determinar $\Delta\Phi = \Phi - \theta$, donde Φ es el ángulo que forma el punto objetivo respecto al plano y θ es el ángulo que forma el vector orientación del vehículo.

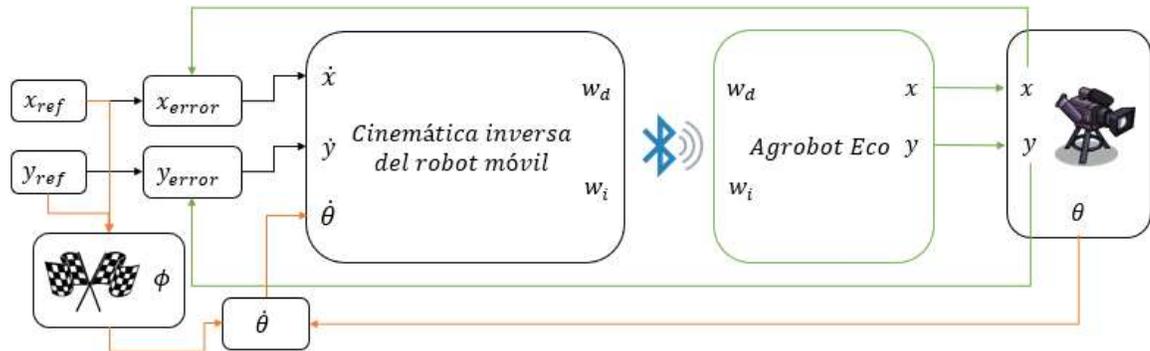
Figura 46 ÁNGULOS PARA CORREGIR LA ORIENTACIÓN DEL VEHÍCULO



Fuente: POSITION AND ORIENTATION CONTROL OF A TWO-WHEELED DIFFERENTIALLY DRIVEN NONHOLONOMIC MOBILE ROBOT [11]

Finalmente, en relación con el modelo descrito se realiza el diagrama de bloques que resume la estructura de la estrategia de control.

Figura 47 ESQUEMA DE CONTROL



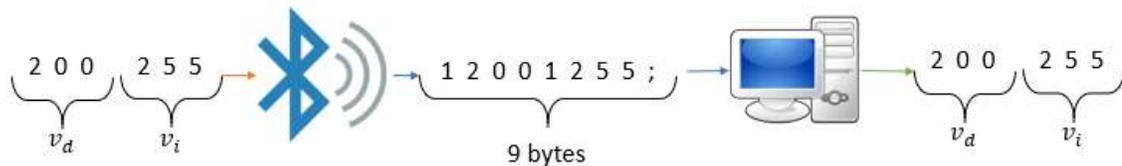
Fuente: Autor

3.2.1 COMUNICACIÓN BLUETOOTH

La comunicación entre el robot y el algoritmo es a partir de un módulo bluetooth, permitiendo el envío y recepción de datos, principalmente, de la velocidad lineal de las ruedas del robot, que son las variables que permiten corregir la posición y orientación del robot durante el trayecto.

La tarjeta de desarrollo Arduino Mega, con la que cuenta Agrobot Eco, basada en un microcontrolador ATmega2560. Realiza la lectura de la señal cuadrada de los Encoder de efecto hall y determina la velocidad lineal de las ruedas, en rpm, para ser enviada al modelo vía bluetooth. Este dato enviado desde el robot es procesado antes de su envío para completar el tamaño mínimo especificado del buffer para no tener pérdidas de información. El tamaño del buffer especificado para el envío y recepción de datos es de 9 bytes, 8 bytes que corresponden al valor de los datos y un carácter (';') que permita conocer donde termina la trama de datos. Estos caracteres una vez recibidos en el algoritmo son nuevamente procesados separar los primeros 4 bytes que representan la velocidad lineal de la rueda derecha de los últimos 4 bytes que representan la velocidad lineal de la rueda izquierda.

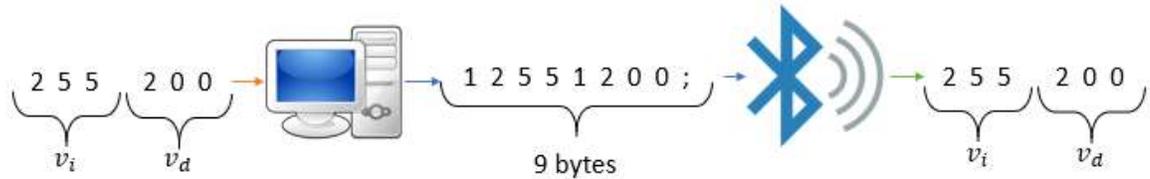
Figura 48 ENVIÓ DE INFORMACIÓN VÍA BLUETOOTH



Fuente: Autor

Evaluated the variables in the model, the linear velocities obtained must be sent to the robot with the aim of correcting its position and orientation during the trajectory. These data are also processed to fit the same size of the reading buffer, 9 bytes, with the difference that the robot receives in its first 4 bytes the linear velocity of the left wheel and in the second part of the buffer the linear velocity of the right wheel.

Figura 49 RECEPCIÓN DE INFORMACIÓN ENVIADA VÍA BLUETOOTH



Fuente: Autor

3.3 PROCESAMIENTO DE IMÁGENES

The robot recognition by the camera is performed starting from the strategic location of circles of the same color on the solar panel of Agrobot Eco.

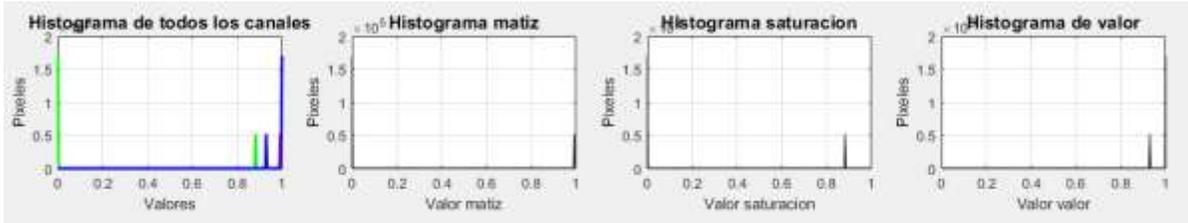
To segment the circles of the image from their color, the histogram of each one of the channels of the color model HSV, normalizing its value in each channel between 0 and 1.

Figura 50 CÍRCULO DE COLOR ROJO



Fuente: Autor

Figura 51 HISTOGRAMA DE LOS CANALES DEL MODELO HSV



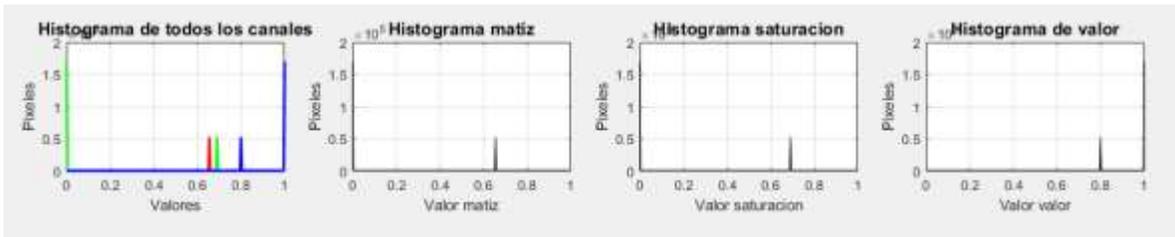
Fuente: Autor

Figura 52 CÍRCULO DE COLOR AZUL



Fuente: Autor

Figura 53 HISTOGRAMA DE LOS CANALES DEL MODELO HSV



Fuente: Autor

Como se ve en la graficas anteriores de los histogramas a cada color le corresponde un umbral en cada canal del modelo HSV. En base al histograma del círculo del color rojo se puede determinar que los umbrales para este color son:

```
umbralMatizbajo = 0.80;  
umbralMatizbajo = 1;  
umbralSaturacionbajo = 0.8;  
umbralSaturacionalto = 1;  
umbralValorbajo = 0.8;  
umbralValoralto = 1.0;
```

Para el color azul los umbrales serian:

```
umbralMatizbajo = 0.6;  
umbralMatizbajo = 0.7;  
umbralSaturacionbajo = 0.6;  
umbralSaturacionalto = 0.7;  
umbralValorbajo = 0.7;  
umbralValoralto = 0.8;
```

En cada uno de los canales del modelo HSV se aplican los umbrales seleccionados para segmentar el color de los círculos. La aplicación de los umbrales encontrados de los histogramas en las mismas componentes del modelo HSV permite resaltar únicamente el color del umbral aplicado en la imagen en cada uno de los canales.

Para el círculo de color rojo, el umbral encontrado operado a sus componentes del modelo HSV, arroja como resultado las siguientes imágenes, que llamaremos máscaras, para cada canal.

Figura 54 SELECCIÓN DEL UMBRAL DEL COLOR EN CADA UNO DE LOS CANALES, ROJO



Fuente: Autor

A su vez para el círculo de color azul, el umbral encontrado aplicado a sus componentes del modelo HSV, arroja como resultado las siguientes máscaras para cada uno de sus canales.

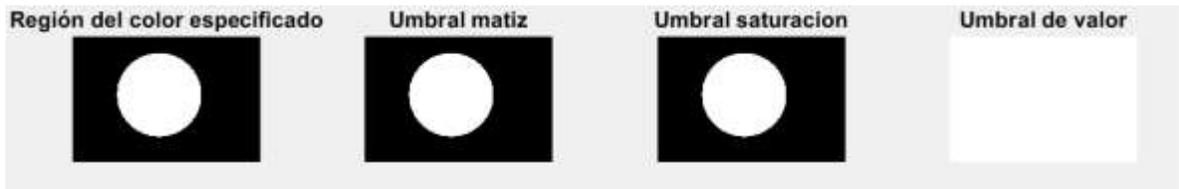
Figura 55 SELECCIÓN DEL UMBRAL DEL COLOR EN CADA UNO DE LOS CANALES, AZUL



Fuente: Autor

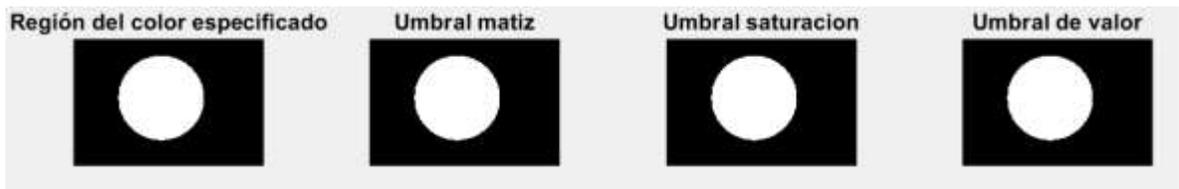
La unión de estas máscaras entrega la imagen donde se resaltan solo los objetos cuyo color este dentro del umbral establecido en relación del histograma de color.

Figura 56 REGIÓN DEL COLOR ESPECÍFICO, ROJO



Fuente: Autor

Figura 57 REGIÓN DEL COLOR ESPECÍFICO, AZUL

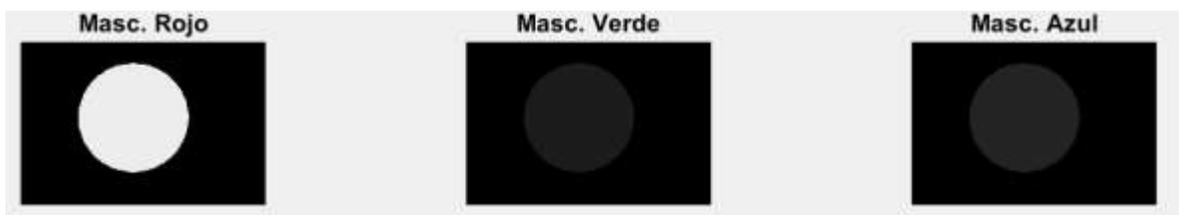


Fuente: Autor

Hasta aquí la imagen resalta los objetos o zonas del color dentro del umbral, sin embargo, no se observa con color.

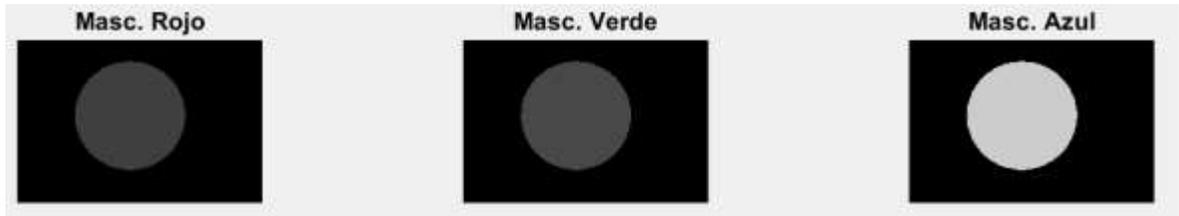
La imagen de la región del color especificado es usada para enmascarar solo las partes del color determinado a partir del modelo RGB, esta imagen se opera con cada uno de los canales del modelo RGB de la imagen inicial.

Figura 58 MASCARA RGB PARA RECUPERAR EL COLOR SOLO DE LA REGIÓN ESPECIFICADA, ROJO



Fuente: Autor

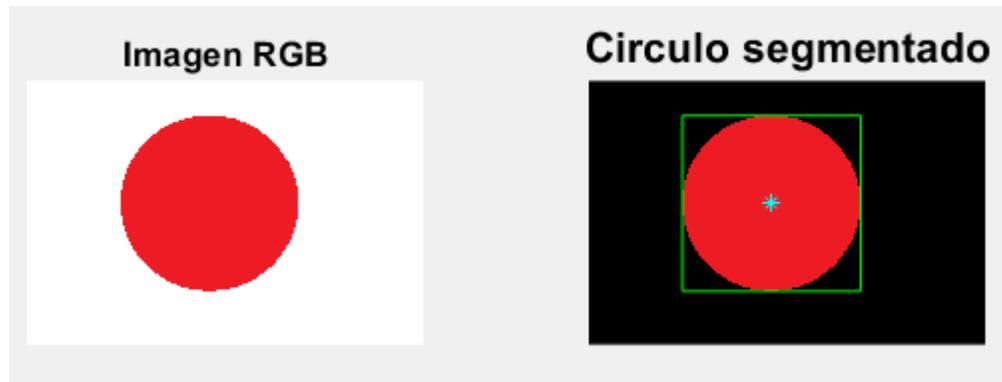
Figura 59 MASCARA RGB PARA RECUPERAR EL COLOR SOLO DE LA REGIÓN ESPECIFICADA, AZUL



Fuente: Autor

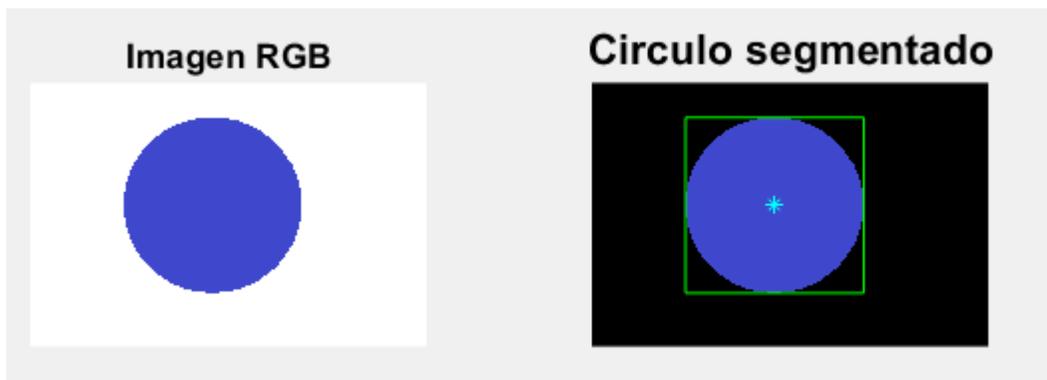
Estas 3 nuevas mascararas para cada canal concatenadas forman nuevamente la imagen RGB, con el color del circulo segmentado.

Figura 60 SEGMENTACIÓN DEL COLOR ROJO, EJEMPLO 1



Fuente: Autor

Figura 61 SEGMENTACIÓN DEL COLOR AZUL, EJEMPLO 2

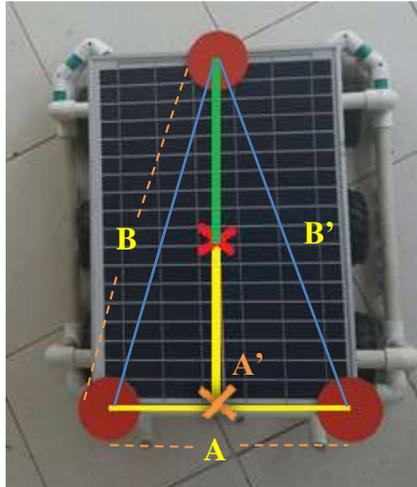


Fuente: Autor

3.3.1 POSICIÓN Y ORIENTACIÓN EN EL PLANO

Tres círculos son ubicados sobre el panel solar del robot, formando un triángulo. Gracias a la forma rectangular del panel solar este triángulo tiene características muy importantes que permiten ubicar el vector de desplazamiento, calcular su posición y orientación en el plano.

Figura 62 CÍRCULOS NARANJAS PARA ORIENTAR Y UBICAR EL ROBOT EN EL PLANO



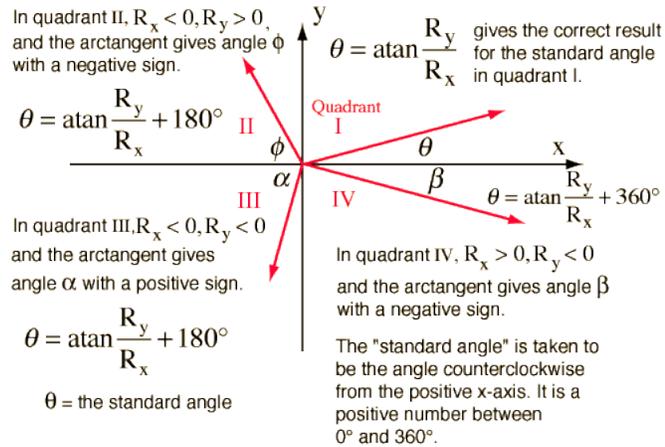
Fuente: Autor.

La base de este triángulo será la distancia más pequeña, A, que exista entre los puntos, donde $B \cong B'$. Entre el punto medio de la base (A') y el círculo más lejano se proyecta una recta. En el punto medio de esta recta (X en rojo) se estima que se encuentra el centro de masa del robot y la posición del robot.

Una vez tenemos la ubicación del centro de masa del robot se proyecta un vector entre el centro de masa y el centroide del círculo más lejano, línea de color verde. Este vector es la referencia para medir el ángulo de orientación del robot en el plano.

Para medir el ángulo de un vector es necesario conocer su componente en el eje X y su componente en el eje Y.

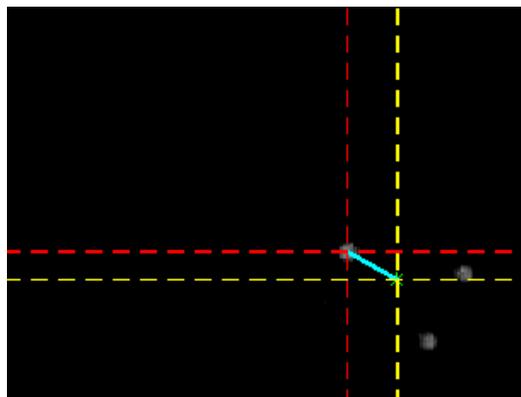
Figura 63 DETERMINAR EL ANGULO DE UN VECTOR EN EL PLANO



Fuente: Obtener el ángulo dado un vector [26].

Para obtener las medidas de (Rx, Ry) se establece un eje de referencia en el punto de ubicación del robot, línea a rayas amarilla Figura 64, y otro en el centroide del círculo más lejano, línea a rayas roja Figura 64. El eje de referencia rojo ubicado en el centroide del círculo más lejano se mueve en los cuatro cuadrantes de la referencia amarilla conforme el robot cambie su orientación, de esta manera midiendo la diferencia entre los dos ejes referenciados se obtiene Rx y Ry . Para medir correctamente el ángulo se debe conocer el cuadrante de la referencia amarilla se encuentra el vector color cyan en la figura 64, y en función a su ubicación sumar, o no, grados de más a la medida obtenida.

Figura 64 OBJETOS SEGMENTADOS Y VECTOR DE ORIENTACIÓN EN EL PLANO



Fuente: Autor.

3.4 INTERFAZ GRÁFICA DE USUARIO

Para facilitar el dibujo de la trayectoria para el vehículo, se realiza una interfaz gráfica de usuario en Matlab donde se pueda determinar el recorrido en un plano determinado, el tamaño de este plano es el mismo de la resolución escogida para la cámara y así que la trayectoria se ajuste al tamaño de las imágenes capturadas. En un segundo plano se pueden ver los puntos que representan los objetivos que debe ir alcanzando el robot para describir el trayecto satisfactoriamente.

Figura 65 PLANOS DE DIBUJO Y VISUALIZACIÓN DE PUNTOS DE LA TRAYECTORIA



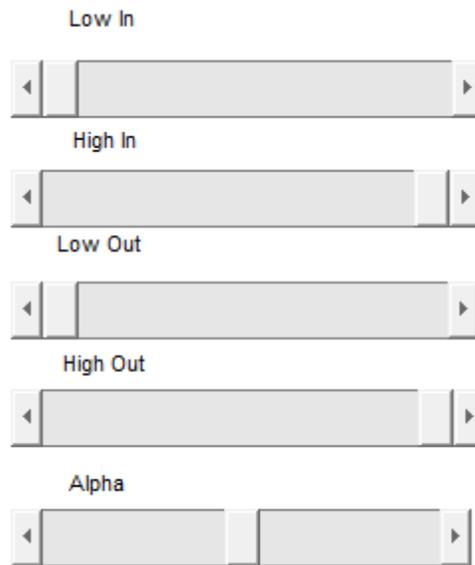
Fuente: Autor

Las imágenes captadas por la cámara se exponen a cambios de luz producidos naturalmente, esto puede cambiar el histograma debido al cambio en la tonalidad del color en los círculos y el color del mismo fondo de la imagen, ocasionando que no se segmente en la imagen el umbral establecido para el color de los círculos.

Debido a estos cambios en la luz, la interfaz permite modificar la intensidad de los valores presentes en la imagen. Los sliders varían en un rango ($[LOW_IN; HIGH_IN]$, $[LOW_OUT; HIGH_OUT]$), entre 0 y 1, en estos rangos se mapean los valores de intensidad en la imagen capturada y son modificados en la nueva imagen obtenida.

Gamma, en la interfaz llamada Alpha, es la forma de la curva que relaciona los valores de entrada y salida. Si gamma es menor que 1, se pesa el mapeo hacia valores de salida más altos (más brillantes). Si gamma es mayor que 1, se pondera la asignación hacia valores de salida más bajos (más oscuros).

Figura 66 AJUSTE EN LA INTENSIDAD DE LOS VALORES DE LA IMAGEN



Fuente: Autor

La interfaz también permite observar el comportamiento en la medida de las variables que hacen parte del modelo cinemático. Aquí se puede apreciar cómo cambian las coordenadas conforme cambia de posición el robot y se puede ver la velocidad del desplazamiento.

Figura 67 VISUALIZACIÓN DE LA POSICIÓN

POSICION (metros)



Fuente: Autor

Figura 68 VISUALIZACIÓN DE LA VELOCIDAD

Vizq (RPM) Vder



Fuente: Autor

En un tercer plano se pueden ver las imágenes capturadas por la cámara e indicar cuando finalizar la captura y el algoritmo de control. Este plano deja ver el vector de desplazamiento del robot móvil y el vector de guía durante el recorrido. La medida del ángulo de estos vectores individuales y su delta se pueden ver, como las otras variables, de manera clara y con sus unidades para su interpretación.

Figura 69 PLANO DE VISUALIZACIÓN DE LAS IMÁGENES



FINALIZAR VIDEO ANTES DE DIBUJAR Y CARGAR UN NUEVO TRAYECTO

Fuente: Autor

Una vez dibujada la trayectoria, el botón de “video” activa la comunicación vía bluetooth con el robot y comienza su recorrido. Para cargar una nueva trayectoria es necesario cerrar el canal de comunicaciones y detener la señal de video con “finalizar”.

Figura 70 VALOR DE LOS ÁNGULOS MEDIDOS

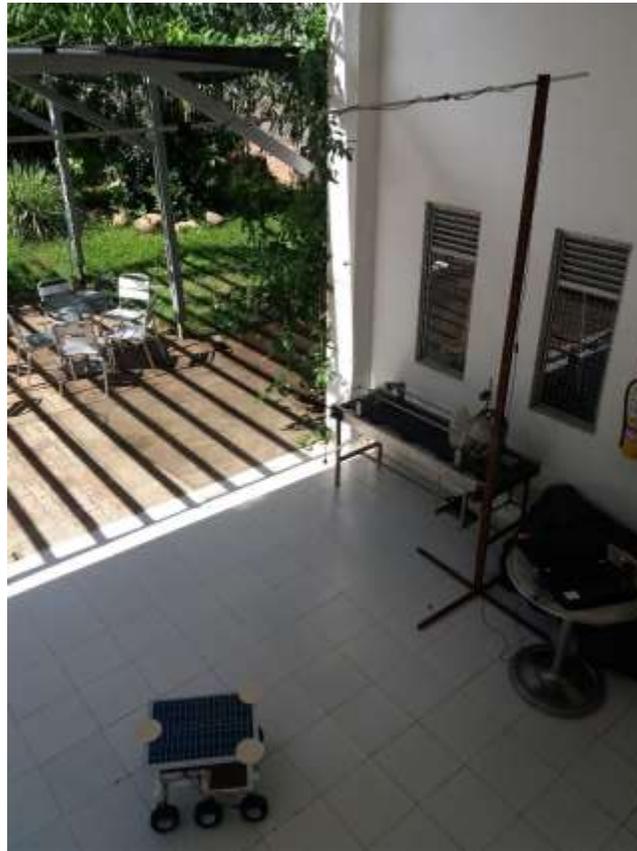


Fuente: Autor

4. CAPITULO RESULTADOS Y VALIDACIÓN

4.1 ESPACIO RECREADO

Figura 71 FOTOGRAFÍA #1, ESPACIO RECREADO



Fuente: Autor

El espacio recreado es realmente la cafetería de las instalaciones del Tecnoparque, que es amplia principalmente, y además poder aprovechar el color blanco de las baldosas buscando tener un fondo uniforme para identificar el color de los círculos que ubican al robot.

La altura a la que se encuentra la cámara es de 3m y es la altura máxima a la sé que pudieron construir los dos soportes metálicos en los que descansa una varilla de aluminio de 4m que sostiene la cámara.

Figura 72 FOTOGRAFÍA #2, ESPACIO RECREADO



Fuente: Autor

La cámara es colocada en la mitad de la varilla de aluminio buscando capturar una mayor área de trabajo, alejando también del área de captura de imagen las bases de los soportes metálicos.

La posición de la cámara debe ser en un ángulo cenital, para obtener las imágenes desde la vista superior y segmentar desde encima el color de los círculos en el robot.

Figura 73 POSICIÓN DE LA CÁMARA



Fuente: Autor.

El tamaño de la imagen es de 160x120 píxeles, los ejes del plano donde se realiza el dibujo tienen el eje y va de 0 a 120, mientras que, por las características de la captura de video, la imagen de video va en su eje y de 120 a 0. Lo que hace necesario que el algoritmo se adapte a este cambio en los ejes.

Es importante resaltar que de la forma que se coloca la cámara también cambia el punto de origen representado en el espacio real, lo que cambia la forma de ver el movimiento.

Usando círculos como referencia podemos conocer el espacio real capturado por la imagen, estos círculos están colocados extremos y origen del plano del espacio recreado, ver Figura 74, en el eje x la medida es 0 es de 2,45 mientras en el eje y es de 0 a 1,95m aproximadamente.

Figura 74 MEDIDA DEL ESPACIO



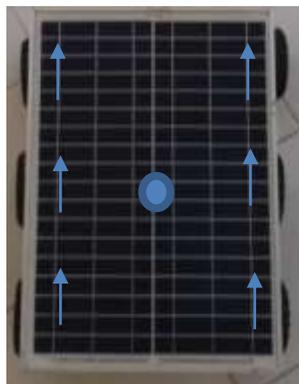
Fuente: Autor

4.2 ROBOT MOVÍL

Guiar el robot a través de la trayectoria supone dos movimientos, uno de orientación y uno de desplazamiento, en relación con el punto objetivo. Sin embargo, dependiendo del tipo del robot la forma de desplazarse y orientarse hacia el objetivo es distinta.

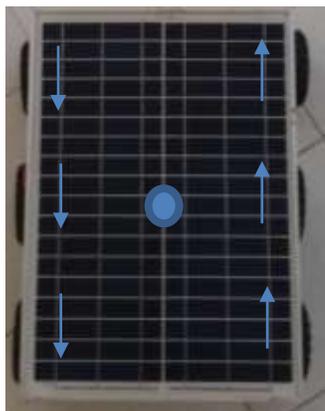
Agrobot Eco es un robot tipo Skid steer, que se desplaza fácilmente en cualquier sentido, principalmente gracias a que puede cambiar su orientación sobre su centro de masa. Esta característica permite corregir la orientación del vector de desplazamiento del robot sin la necesidad de desplazar su centro de masa.

Figura 75 MOVIMIENTO HACIA ADELANTE



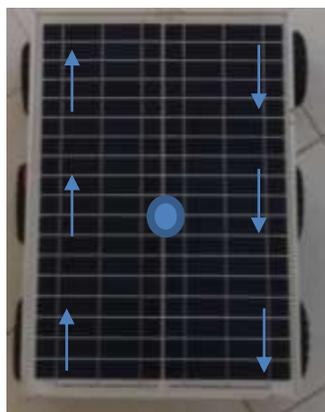
Fuente: Autor

Figura 76 GIRO HACIA LA IZQUIERDA



Fuente: Autor

Figura 77 GIRO HACIA LA DERECHA



Fuente: Autor

4.2.1 VELOCIDAD LINEAL

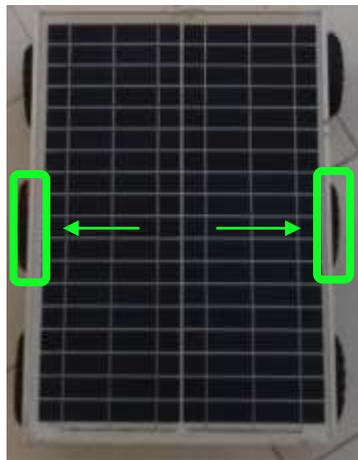
Para corregir la posición y orientación del robot en el trayecto es necesario modificar la velocidad lineal de sus ruedas derecha e izquierda.

El tipo del modelo escogido para Agrobot Eco beneficia altamente el desarrollo del prototipo y evaluación. En cualquier movimiento que realice el robot la velocidad inyectada a sus ruedas será la misma. Incluso en el movimiento de orientación, donde lo único que cambia es el sentido de giro de las llantas de un lado del robot, pero la magnitud de su velocidad es idéntica.

La tarjeta de desarrollo usada para el prototipo es un Arduino mega 2560, con un microprocesador ATmega 2560. Medir la velocidad lineal de los motores de todas las ruedas del costado derecho y del costado izquierdo del robot aumentaría la demanda energética de la tarjeta. Esto porque para cada motor con Encoder es necesario alimentar el circuito del sensor de efecto hall (5V) de manera independiente de la fuente de voltaje del motor (12V).

Con el objetivo de no aumentar la demanda energética de la tarjeta de desarrollo o utilizar más baterías, y gracias a que la magnitud de la velocidad aplicada en las ruedas en este modelo es la misma a cada lado; se miden las velocidades de las ruedas intermedias del lado derecho e izquierdo, permitiendo, alimentar los dos Encoder desde la tarjeta de desarrollo y a partir de la medición de estos dos canales conocer la velocidad de las tres ruedas de la derecha y de las tres ruedas de la izquierda.

Figura 78 ENCODERS PARA MEDIR LA VELOCIDAD



Fuente: Autor

Medir la velocidad en el Encoder de efecto hall, representa medir los pulsos generados por el Encoder en un tiempo de muestreo. Los motores usados para el prototipo tienen la misma relación, 800 pulsos/vuelta. Esto es indispensable ya que de otro modo la velocidad medida puede ser diferente para cada motor, provocando que el robot se mueva sin ningún sentido.

```

//funcion interrupcion velocidad DERECHA
void interrupcion_isr()
{
    pulsos=pulsos+1; // contador de pulsos
    // if(pulsos==800){
    //     vuelta=vuelta+1;
    //     pulsos=0;
    // }
}
//funcion interrupcion velocidad IZQUIERDA
void interrupcion_isr1()
{
    pulsos1=pulsos1+1;
}

```

El tiempo de muestreo de la velocidad de los motores es de 120mS, en este tiempo se cuentan la cantidad de pulsos que ha generado el Encoder y se calcula la velocidad en RPM.

```

if(millis() - time > sampleTime)
{
    time = millis(); // Tiempo de la próxima actualización
    rpm=(pulsos/resolucion)*500; // formula para calcular rpm
    rpml=(pulsos1/resolucion)*500; //60000mS para el minuto(120mS)
}

```

Las velocidades se procesan y adecuan al tamaño establecido del buffer de envío y recepción de datos, 9 bytes, y ser enviados vía bluetooth hacia el algoritmo. Serial2 es el puerto de comunicación al que está conectado el módulo Bluetooth en la tarjeta de desarrollo.

```

vf = 1000+rpm;
vf1= 1000+rpml;
Serial2.print(vf);
Serial2.print(vf1);
Serial2.print(";");
}

```

Estas variables son recibidas y procesadas por el algoritmo en MATLAB, para luego ser ingresadas al modelo cinemático en conjunto con las demás variables de entrada.

Una vez procesados todos los datos por la estrategia de control, esta envía de vuelta por bluetooth la nueva magnitud de las velocidades de las ruedas, bien sea para corregir la orientación o realizar un desplazamiento. Las nuevas velocidades obtenidas también son recibidas en un paquete de 9 bytes.

```

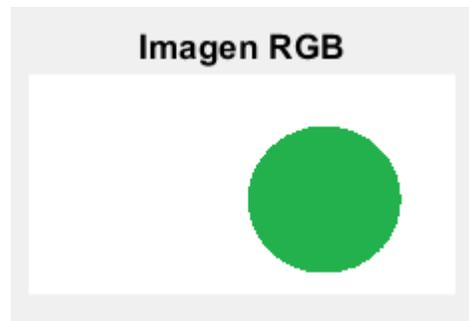
void serialEvent2() {
  while (Serial2.available()) {
    // get the new byte:
    char inChar = (char)Serial2.read();
    if (inChar==58 )
    { contadorx=0;
    }
    else
    { contadorx=contadorx+1;
    if (contadorx==1 ) { tem1=0; }
    if (contadorx==2 ) { tem1= (inChar)*100 ; }
    if (contadorx==3 ) { tem1=tem1+ (inChar)*10 ; }
    if (contadorx==4 ) { tem1=tem1+ (inChar) ; }
    if (contadorx==6 ) { tem2=0; }
    if (contadorx==7 ) { tem2= (inChar)*100 ; }
    if (contadorx==8 ) { tem2=tem2 + (inChar)*10 ; }
    if (contadorx==9 ) { tem2=tem2 + (inChar) ; }
    if (contadorx==9 )
    {
      vi=tem1;
      vd=tem2;
    }
  }
}
}
}

```

4.3 PROCESAMIENTO DE IMÁGENES

El color que van a llevar los círculos para ser detectados se muestra en la figura 79, este es el color que debe ser segmentado para identificar la ubicación y orientación del robot.

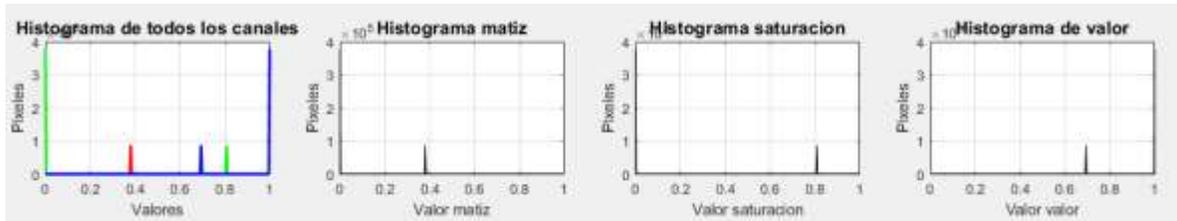
Figura 79 COLOR EN LOS CÍRCULOS DEL ROBOT



Fuente: Autor

El histograma para este color en particular es el siguiente para cada uno de los canales del modelo HSV.

Figura 80 HISTOGRAMA DEL COLOR SELECCIONADO



Fuente: Autor

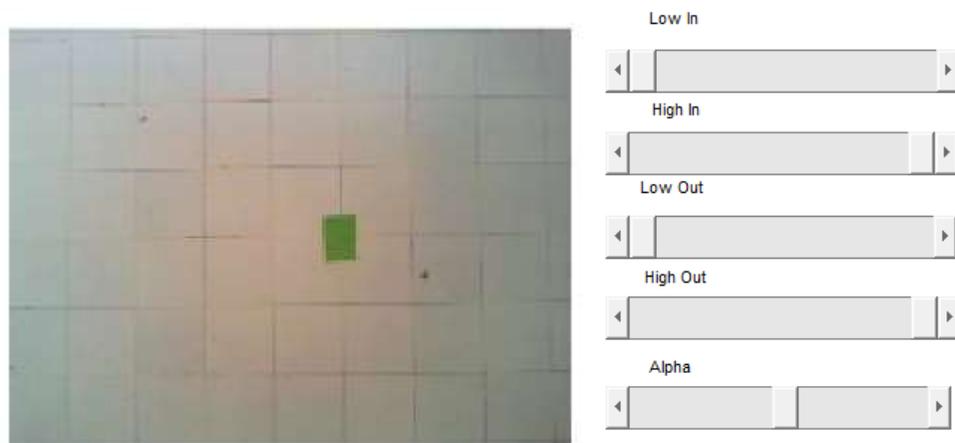
Los valores de cada histograma arrojan el umbral en el que se encuentra el color en cada canal.

```

umbralMatizbajo = 0.15;
umbralMatizalto = 0.60;
umbralSaturacionbajo = 0.36;
umbralSaturacionalto = 1;
umbralValorbajo = 0;
umbralValoralto = 0.8;
    
```

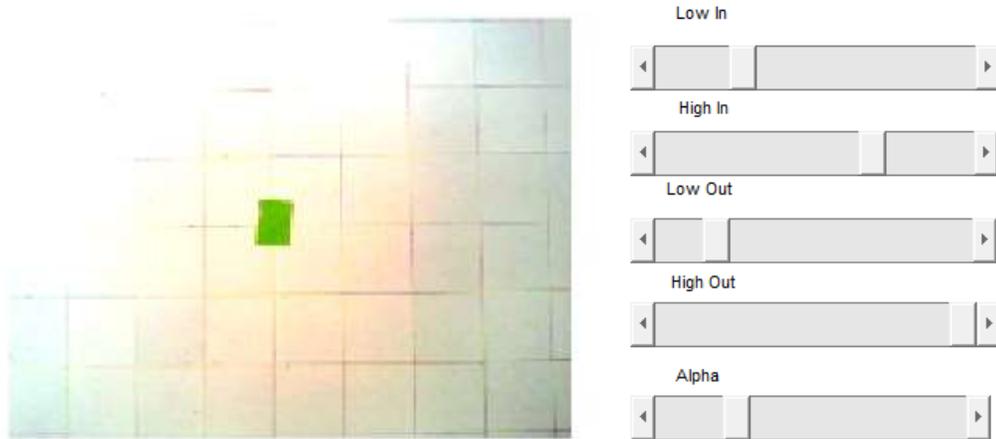
Debido a que el espacio de trabajo se encuentra expuesto a cambios luminosidad, ocasionando que el histograma cambie con la intensidad de la luz y no se encuentre el color dentro del umbral establecido. El usuario puede modificar la intensidad de los valores en la imagen en RGB, destacando el color sobre el blanco de la baldosa y que el color se encuentre en el umbral.

Figura 81 IMAGEN SIN MODIFICAR INTENSIDAD



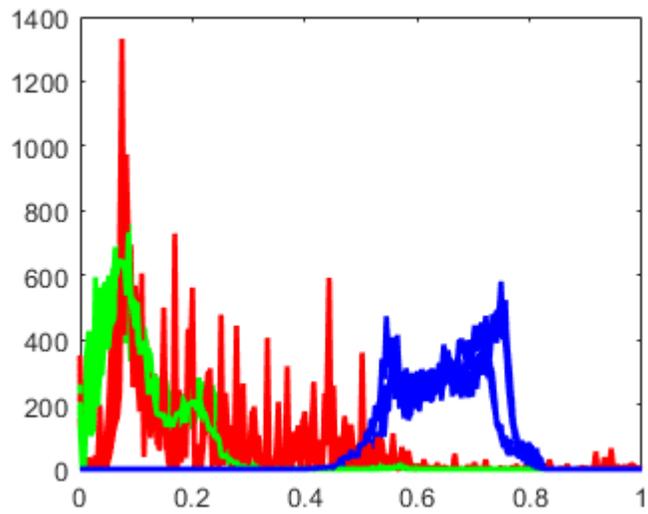
Fuente: Autor

Figura 82 IMAGEN INTENSIDAD AJUSTADA



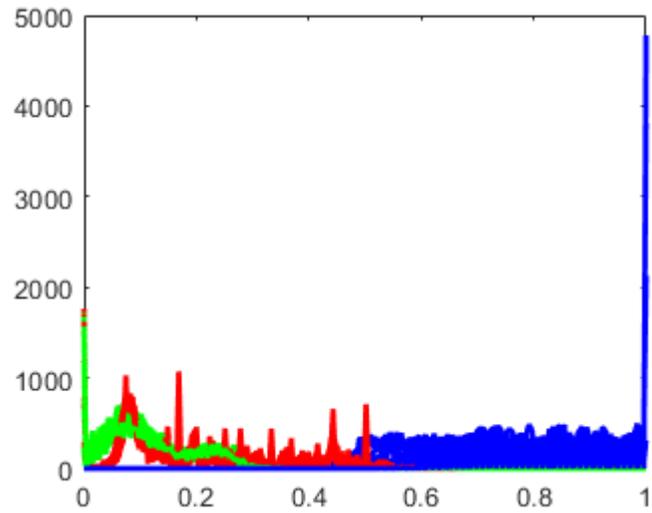
Fuente: Autor

Figura 83 HISTOGRAMA, FIGURA 81



Fuente: Autor

Figura 84 HISTOGRAMA, FIGURA 82



Fuente: Autor

Superado los inconvenientes que se puedan presentar con los cambios en la luminosidad del lugar, finalmente se pueden segmentar de manera correcta los tres círculos sobre el robot.

Figura 85 CÍRCULOS SOBRE EL ROBOT



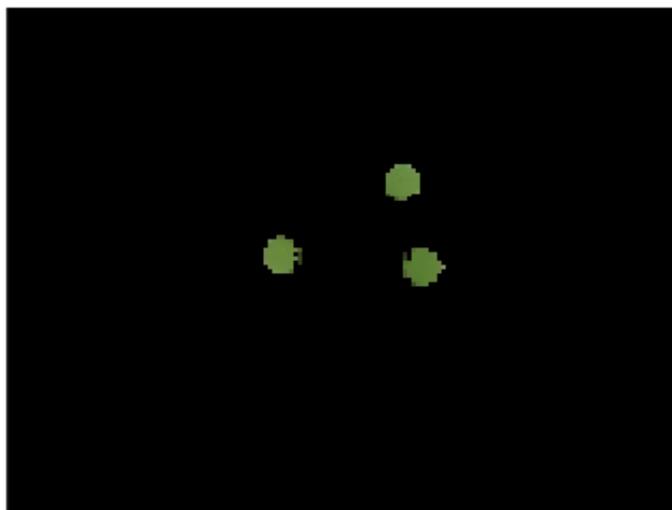
Fuente: Autor

Figura 86 IMAGEN CAPTURADA POR LA INTERFAZ E INTENSIDAD AJUSTADA



Fuente: Autor

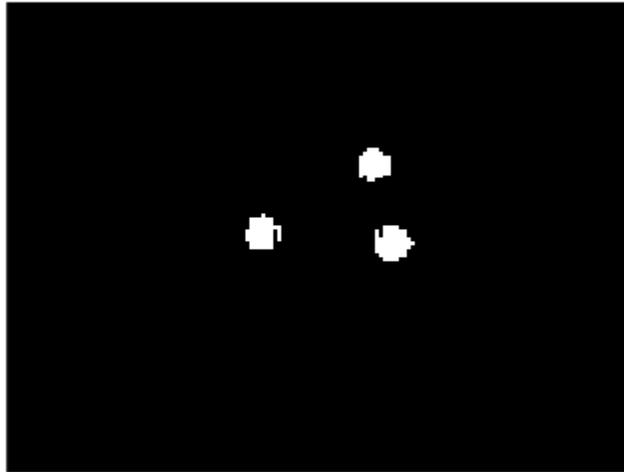
Figura 87 OBJETOS SEGMENTADOS DE LA FIGURA 86



Fuente: Autor

4.3.1 PREPROCESAMIENTO

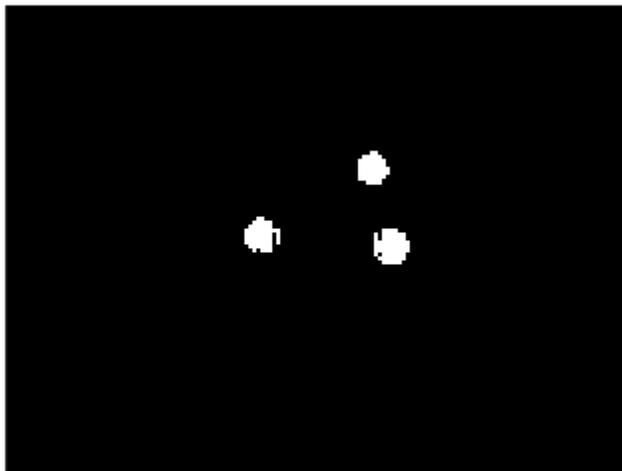
Figura 88 COLORES SEGMENTADOS IMAGEN BINARIZADA



Fuente: Autor

Segmentado el color y antes de etiquetar los objetos, se realiza un filtrado por área para eliminar los pixeles del contorno que pueden quedar separados por pequeños espacios del objeto, pero que de igual manera inciden en el algoritmo ya que pueden ser etiquetados como un objeto de mas en la imagen.

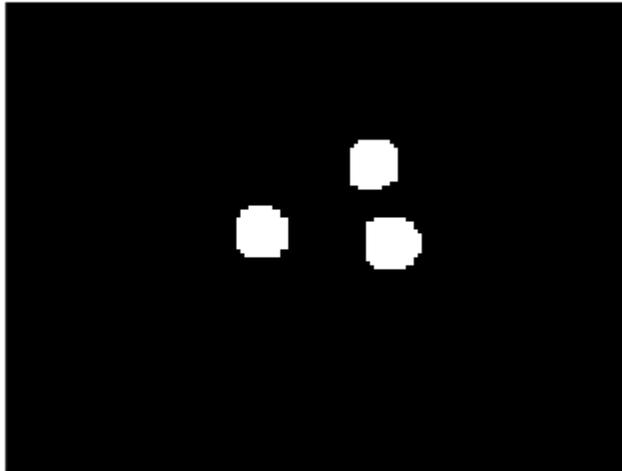
Figura 89 IMAGEN FILTRADA POR ÁREA



Fuente: Autor

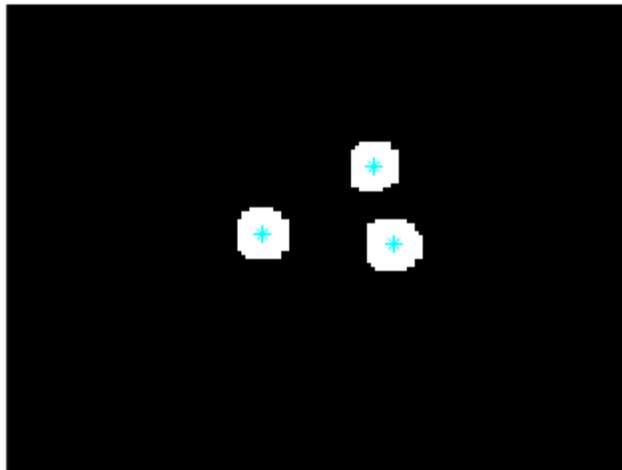
Para mejorar nuevamente la forma de los círculos se dilata la imagen obtenida después del filtro. El resultado de la dilatación permite etiquetar correctamente los círculos.

Figura 90 IMAGEN DILATADA



Fuente: Autor

Figura 91 CÍRCULOS ETIQUETADOS EN LA IMAGEN



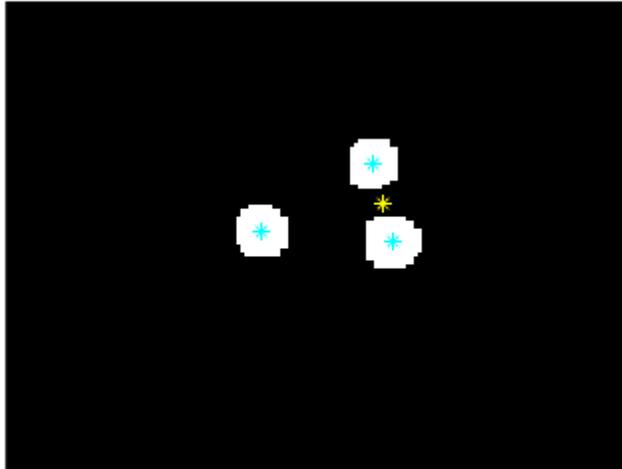
Fuente: Autor

Segmentados y etiquetados los objetos, a partir de estos tres centroides se miden las variables necesarias para el modelo cinemático.

4.3.1.1 PUNTO MEDIO DE LA BASE

Encontramos los centroides mas cercanos, la base del triangulo, y medimos el punto medio entre estos dos puntos.

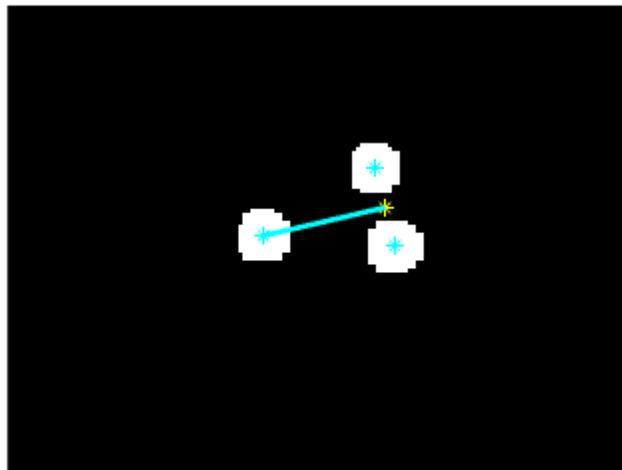
Figura 92 PUNTO MEDIO DE LA BASE



Fuente: Autor

El vector de desplazamiento se ubica desde el centro de masa del vehiculo y en la direccion de su desplazamiento. El punto que referencia la orientacion en el triangulo es el centroide mas lejano al punto medio de la base, entre estos dos puntos se proyecta una recta pasa por el centro de masa del robot.

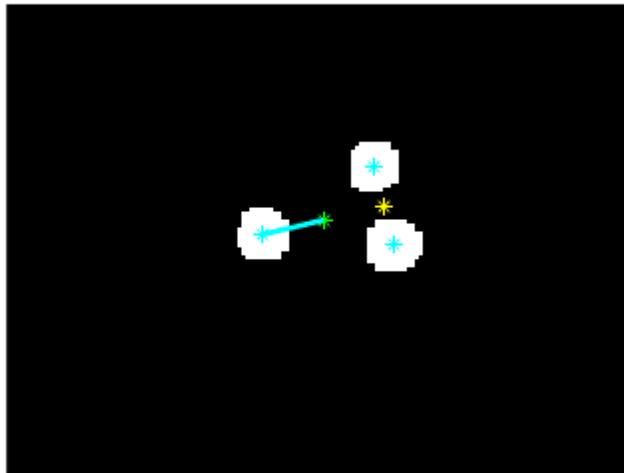
Figura 93 REFERENCIA PARA EL VECTOR DE DESPLAZAMIENTO



Fuente: Autor

El centro de masa del robot se encuentra en la mitad de esta recta, y es desde este punto hasta el centroide mas lejano que se proyecta el vector de desplazamiento final del robot.

Figura 94 VECTOR DE DESPLAZAMIENTO

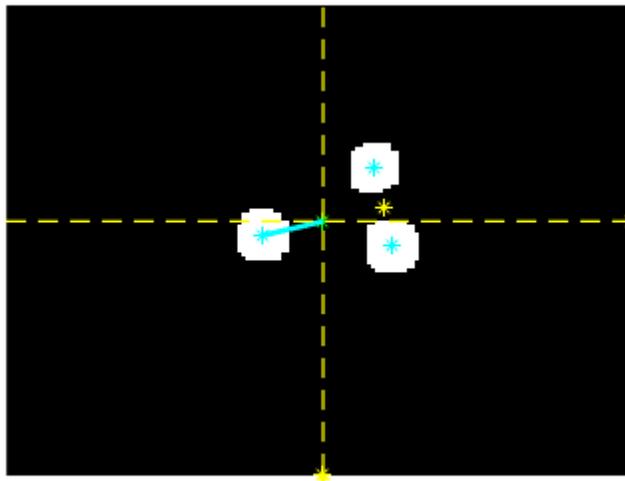


Fuente: Autor

4.3.1.2 MEDIR ANGULO DE ORIENTACION DEL ROBOT

El vector de desplazamiento se mueve en el plano capturado por la cámara, es necesario conocer el ángulo que forma este vector en el plano de robot referenciado desde la cámara para poder corregir su orientación en el trayecto.

Figura 95 PLANO DE REFERENCIA DEL ROBOT



Fuente: Autor

La medida del ángulo se hace en relación a los deltas del desplazamiento en cada uno de los ejes. Luego es necesario conocer el cuadrante en el que se encuentra el vector para así obtener la medida del ángulo del vector de desplazamiento en el plano.

Figura 96 ANGULO DEL ROBOT



Fuente: Autor

4.3.1.3 ANGULO DEL VECTOR GUIA

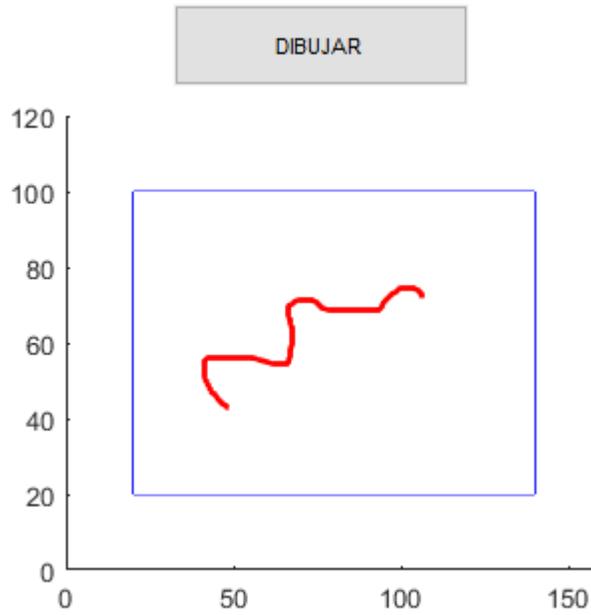
El dibujo de la trayectoria se representa en una serie de puntos, la cantidad de puntos depende de la forma de la trayectoria. La función desarrollada `freehanddraw` permite dibujar a mano alzada sobre un plano de un tamaño determinado y arroja como resultado una serie de puntos que representan el dibujo realizado.

Con `freehanddraw` dibuje un objeto de línea a mano alzada suave en el eje predeterminado. Haga clic con el botón izquierdo una vez para comenzar a dibujar, haga clic con el botón derecho para finalizar o haga doble clic para cerrar el contorno y terminar. Esta función se encuentra en el File Exchange de Mathworks para descargar y ha tenido una gran puntuación entre la comunidad gracias a que es muy útil [49].

El tamaño del plano de dibujo para la interfaz es el mismo del tamaño de la captura de las imágenes, 160x120, guardando así mismo una sola porción.

Dentro de ese plano de dibujo existe un rectángulo que limita el área de dibujo, con el fin de no tener puntos al límite, donde por alcanzar el objetivo puede que se salga el robot de la imagen.

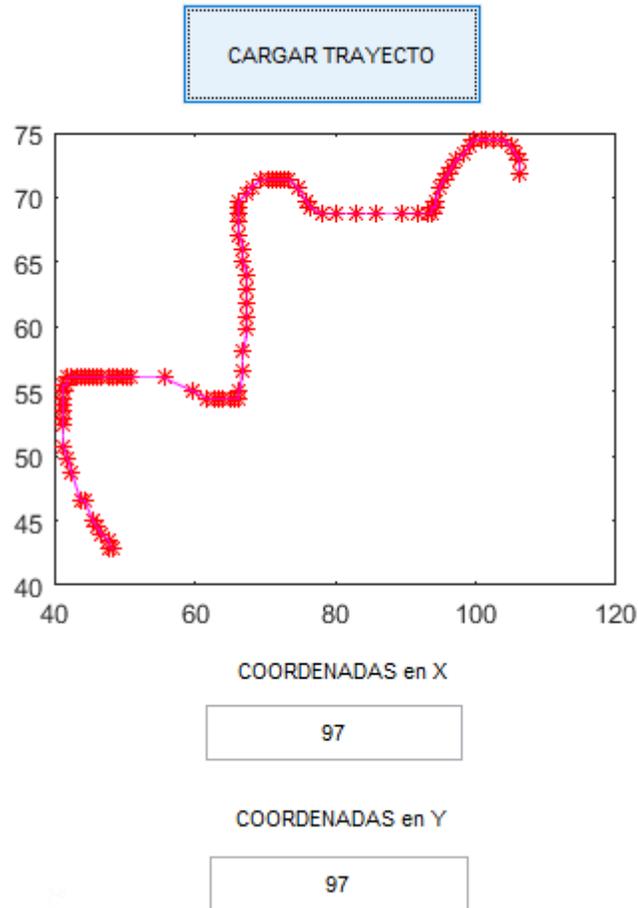
Figura 97 DIBUJO DEL TRAYECTO



Fuente: Autor

Cualquier trayecto dibujado, como el de la figura anterior, será representado por las coordenadas de los puntos que lo conforman. La interfaz permite también ver cuántos puntos en total representa el dibujo.

Figura 98 PUNTOS QUE REPRESENTAN EL TRAYECTO

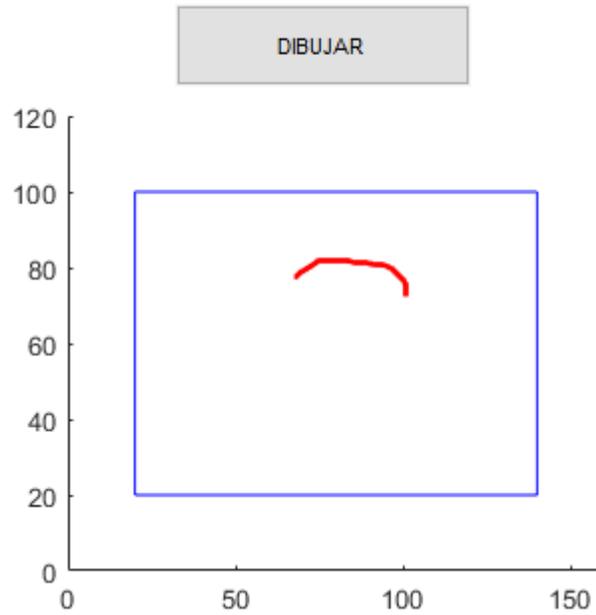


Fuente: Autor.

En la posición del punto objetivo se debe orientar y desplazar el robot. Por eso es necesario también medir el ángulo que forma la posición de la meta en el plano del robot.

Para medir el ángulo de esta coordenada se proyecta un vector guía en su dirección. El ángulo de este vector guía es calculado en función del plano del robot y de este modo se puede determinar el delta entre los dos ángulos y la magnitud del ángulo de corrección de orientación.

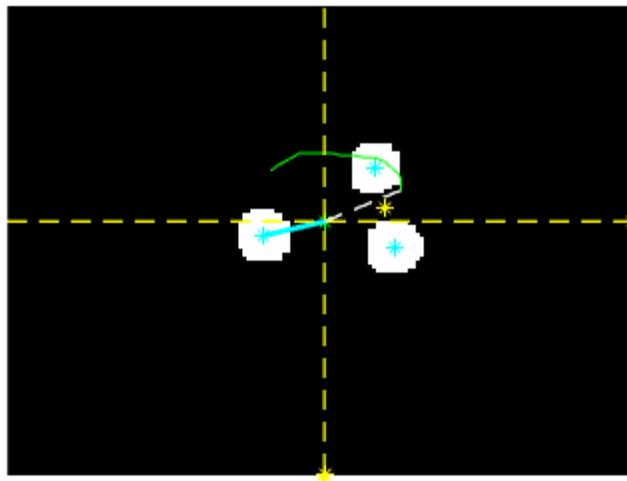
Figura 99 TRAYECTO ALEATORIO PARA VISUALIZAR EL VECTOR GUÍA



Fuente: Autor

Donde se empieza a dibujar será la primera coordenada del trayecto, en la siguiente figura se pueden apreciar el vector guía hacia el primer objetivo y el vector de desplazamiento.

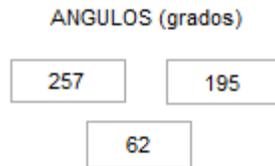
Figura 100 VECTOR GUÍA DEL TRAYECTO



Fuente: Autor

La interfaz me permite ver la magnitud de estos ángulos, siendo el primer recuadro la medida del ángulo del robot, el segundo recuadro la medida del ángulo del vector guía y el inferior el delta entre los dos ángulos.

Figura 101 ÁNGULOS DEL MODELO



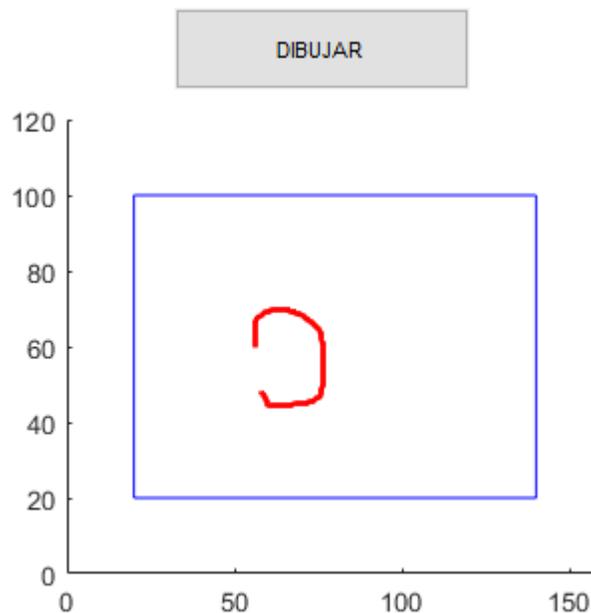
Fuente: Autor

4.4 CONTROL CINEMÁTICO

Para evaluar el modelo del control propuesto se dibujan diferentes trayectorias en la interfaz.

En el siguiente trayecto se muestra el comportamiento de las variables y funcionamiento del prototipo.

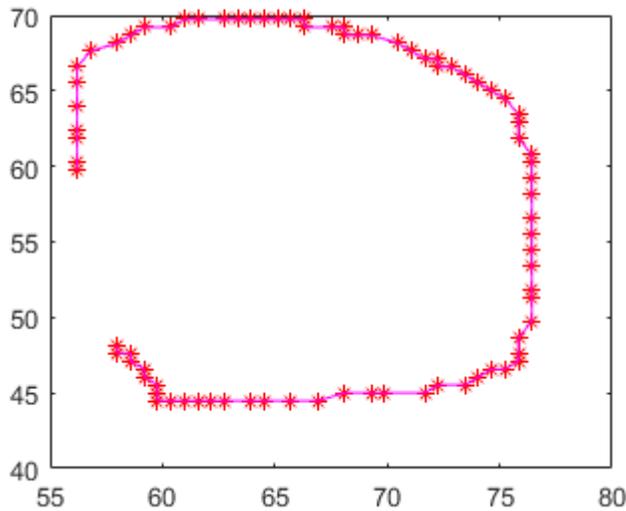
Figura 102 TRAYECTORIA DE PRUEBA



Fuente: Autor

No se cierra el trayecto para que cuando termine el trayecto sea al mismo algoritmo el que lleve nuevamente al robot a la posición inicial y realice la ruta nuevamente.

Figura 103 PUNTOS DE LA TRAYECTORIA



Fuente: Autor

Para poder ejercer control sobre las ruedas del prototipo, se debe conocer primero su velocidad. Una vez establecida la conexión bluetooth con el robot se puede leer los 9 bytes con la información que este envía de los Encoder de efecto hall.

```
% LEER LA VELOCIDAD LINEAL DE LAS RUEDAS ENVIADAS POR ARDUINO
recbyte = fread(bt,9);
d = char(recbyte(1:9));
d1 = d';
token = strtok(d1, ';');
val = str2double(token);
d1 = floor(val/10000);
d = mod(val,10000);
vder = d1-1000;
vizq = d-1000;
chrder = int2str(vder);
chrizq = int2str(vizq);
set(handles.edit19, 'String', chrizq);
set(handles.edit21, 'String', chrder);
```

La variación del ángulo de corrección es muy importante en el modelo para que el carro este siempre en dirección a su objetivo.

Sin embargo, la velocidad de ejecución del algoritmo de control es lenta y el tiempo de muestro de las imágenes también, lo que ocasiona que el carro se tenga que mover mucho más despacio y no se adelante a la captura de las imágenes. Este adelanto hace que el carro se pase del ángulo de orientación al ir más rápido que la cámara o se pase de la posición de la meta.

En la tarjeta de desarrollo se usa un algoritmo de “pasos”, que aplica la velocidad recibida 100mS y se detiene 100mS.

Para optimizar el algoritmo y mejorar el funcionamiento, se da un margen de error en el ángulo de corrección y al alcance de la meta. Consiguiendo que el robot siga la trayectoria adaptándose a la velocidad de ejecución del algoritmo en la interfaz de usuario, con un margen de error aceptable.

Delta de phi

Al ser la velocidad del robot mayor a la velocidad de captura, la medida de la orientación del robot en el algoritmo siempre estaba “atrasada” haciendo que el robot nunca encuentre la orientación adecuada.

El margen de error para el delta de phi es de 20 grados, va desde -10 a 10 grados, cuando el delta esta en este rango, el robot supone estar orientado en la dirección correcta y se dirige al objetivo.

```
% %% MARGEN DE ERROR PERMITIDO EN LA ORIENTACION DEL ROBOT
% %% PARA DIRIGIRSE DIRECTAMENTE HACIA LA META.
if dphi<=10 && dphi>=-10
    vi = 1;
    vd = 1;
end
```

Las velocidades de la rueda izquierda y derecha en ‘1’ es un valor que indica a la tarjeta de desarrollo que el sentido de giro de la rueda es el mismo y está en una orientación adecuada para alcanzar el objetivo.

En la gráfica 104 se puede observar cómo cambia la variable durante el recorrido de la forma de la figura 102. En la primera parte se puede ver que es negativo y se acerca buscando el cero. Esto indica que el vector de dirección tiene un ángulo mayor que el de desplazamiento del carro, pero esta diferencia no es menor a -180.

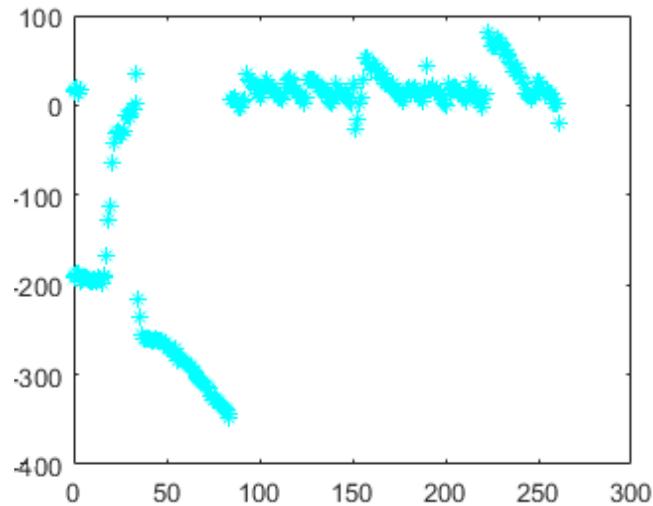
El modelo cinemático siempre orienta al robot haciéndolo ir en contra de las manecillas del reloj, no tiene en cuenta que si entre el vector de desplazamiento y el de dirección hay más de 180 y menos de -180 él puede acercarse por el ángulo menor, haciendo más rápido el movimiento de orientación.

```
if dphi>180
    dphi = -180;
    %display('IZQUIERDA'); %se vuelve negativo
end
if dphi<-180
    dphi = 180;
    %display('DERECHA'); %se vuelve positivo
end
```

Esto explica el comportamiento del delta en la figura 104, entre las muestras 35 y 80, donde el ángulo del delta de va aumentando y en las muestras llegando al 90 ya es cero. Esto es debido a que el vector de desplazamiento se acerca al de dirección por el ángulo más corto.

El delta de phi es grande cuando el robot llega al punto inicial en el sentido contrario al del trayecto. Una vez va sobre el trayecto varía entre el rango de error como se puede ver después de las 100 muestras en la figura 104.

Figura 104 DELTA DE PHI



Fuente: Autor

Con el cálculo de los ángulos de los dos vectores obtenidos de las imágenes capturadas y las velocidades de las ruedas del robot, se mide el radio de las ruedas del motor y la distancia entre ellas.

Todas estas variables hacen parte del modelo cinemático, que arroja como resultado la velocidad de las ruedas para dirigir al robot a través de la trayectoria

```
% ENTRADAS AL CONTROLADOR
rd = 0.06;%radio de la rueda (Metros)
ln = 0.47;%distancia entre las ruedas (Metros)
%VELOCIDAD LINEAR Y ANGULAR DEL ROBOT
v = ((vder+vizq)*rd)/2;%velocidad lineal
w = ((vder-vizq)*rd)/ln;%velocidad angular
% %xpunto y ypunto
drx = v*cosd(dgc);%xpunto
dry = v*sind(dgc);%ypunto
fx = (drx*cosd(dgc))/rd;
fy = (dry*sind(dgc))/rd;
dpln = (dphi*ln)/rd;
wd = (fx+fy+dpln);
```

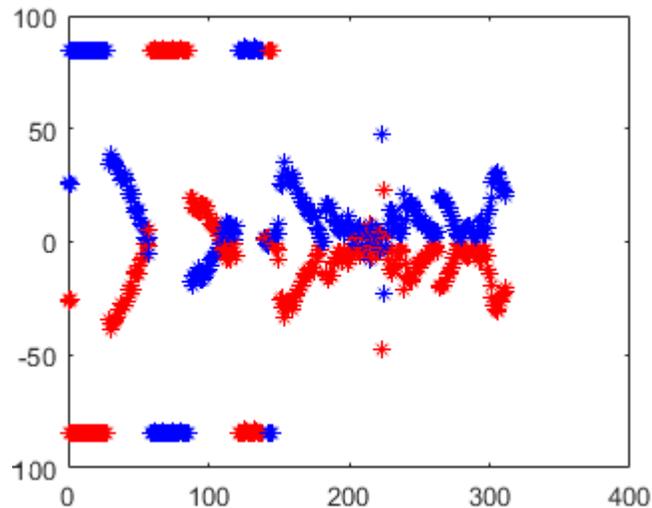
```

wi = (fx+fy-dpln);
% % VELOCIDAD DE RUEDA IZQ Y DER OBTENIDAS
vi = wi*rd;
vd = wd*rd;

```

En la figura 105 y con relación al trayecto de la figura 103, podemos visualizar la velocidad en cada iteración del algoritmo de control. En esta imagen podemos ver que, como el modelo lo indica, la magnitud de la velocidad de las ruedas es la misma, solo cambia su sentido de giro al momento de orientarse. En rojo se puede ver la velocidad de la rueda izquierda y el azul la de la rueda derecha, cuando los puntos azules son mayores a los puntos rojos el robot se está orientando hacia su izquierda y viceversa. Como se puede ver en las primeras 35 muestras de la figura 104 y las primeras 35 de la figura 105, el vector de dirección estaba en un cuadrante mayor al de desplazamiento por eso debía girar hacia la izquierda. La magnitud de estas velocidades es mayor cuando el ángulo del delta es muy grande. Estas velocidades tienden a cero al acercarse a la orientación correcta, la velocidad también determina de forma indirecta la correcta orientación el robot y cuando desplazarse directamente hacia la coordenada objetivo.

Figura 105 VELOCIDADES DE LAS RUEDAS



Fuente: Autor

En la figura 106 se puede apreciar el error en el *eje x* en magenta y el error del *eje y* en negro. Aquí podemos ver como el error tiende a estar dentro del margen.

Para enviar los datos hacia el robot se procesan los datos al tamaño del buffer de envió de 9 bytes.

```

%% ENVIAMOS LA DATA
srli=int2str(uint8(vi));srld=int2str(uint8(vd));
srli1=str2double(srli);srld1=str2double(srld);

```

```

vli=1000+srli1;vld=1000+srld1;
%disp(vli);disp(vld);
ss1=num2str(vli);ss2=num2str(vld);
li=length(ss1);ld=length(ss2);
for in=1:li
    I(in)=str2double(ss1(in));
end
I(li+1)=': ';
for in=1:ld
    D(in)=str2double(ss2(in));
end
D(ld+1)=': ';
dato = horzcat(I,D);
lt=length(dato);
disp(dato);
for in=1:lt
    fwrite(bt, char(dato(in)));
end

```

Medir el error en la posición ayuda a determinar si el robot alcanzó la meta. Debido a la captura demorada de las imágenes y del procesamiento, existe también un margen de error en la posición de -3 a 3 píxeles en cada eje de coordenadas.

En cada coordenada objetivo se mide el error, si está dentro del margen, da por alcanzado el objetivo, mide la distancia que ha recorrido en el espacio recreado y pasa a la siguiente coordenada de la trayectoria.

Una vez termina la ruta, muestra la distancia recorrida y vuelve al punto inicial de la misma trayectoria.

```

%% MEDIR EL ERROR EN LA POSICIÓN DEL ROBOT
erx = xt(hc)-xml;
ery = yta(hc)-ym1;
ltr = length(yta);

if hc <= ltr
    %MARGEN DE ERROR PERMITIDO EN LA POSICIÓN PARA RECONOCER CUANDO
    % %HA SIDO ALCANZADA LA META Y ASI PASAR AL SIGUIENTE OBJETIVO
    if (erx>=-3 && erx<=3)
        if (ery>=-3 && ery<=3)
            %GRAFICAR PUNTO META CUMPLIDA EN TRAYECTO
            vi = 0;
            vd = 0;
            %DISTANCIA ENTRE LOS PUNTOS QUE RECORRE
            pfx2 = (xt(hc)*mtx)/160; pfy2 = (yta(hc)*mty)/120;
            terx = (pfx2-pfx1)^2; tery = (pfy2-pfy1)^2;
            dist = sqrt(terx+tery);
            distancia = distancia+dist;
            hc = hc+1;
        end
    end
end

```

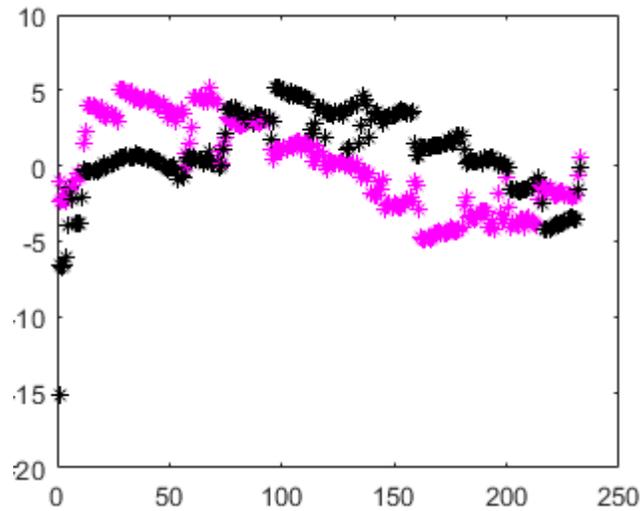
```

end
end

%% RECORRIDAS TODAS LAS POSICIONES, VUELVE A LA 1ERA POSICIÓN
% Y MUESTRA LA DISTANCIA RECORRIDA EN METROS
if hc>ltr
    disp(distancia);
    hc=1;
end

```

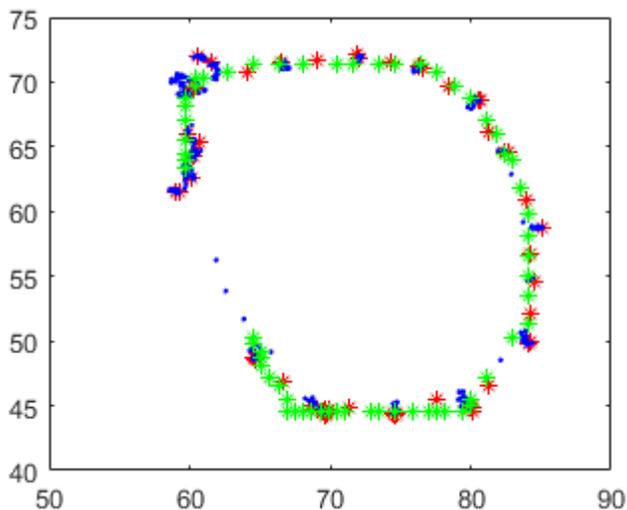
Figura 106 ERROR DE POSICIÓN



Fuente: Autor

Para evaluar el trayecto realizado frente a la trayectoria dibujada, figura 102, se visualizan los puntos de la trayectoria (verde), la posición del robot (azul) y cuando el robot alcanza una coordenada dentro del margen de error de la posición se marca un punto (rojo).

Figura 107 COMPARACIÓN DE TRAYECTORIAS



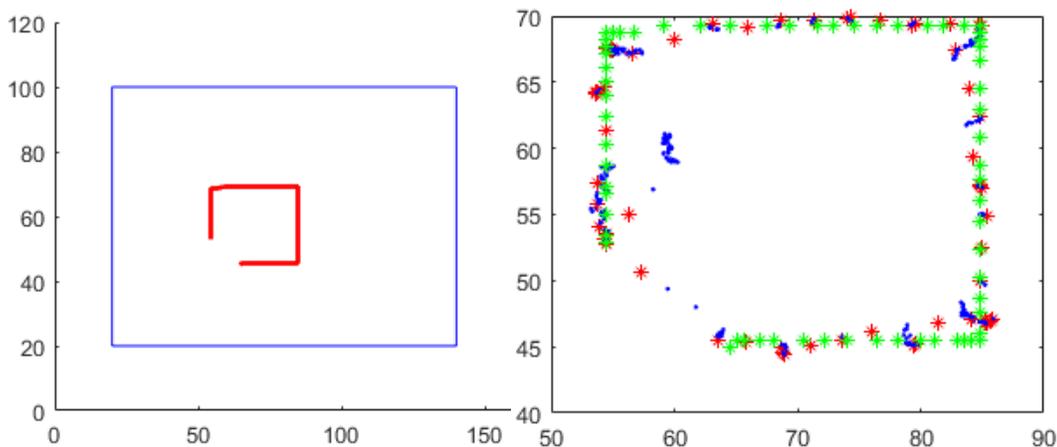
Fuente: Autor

Se puede observar la trayectoria que describe el robot por los puntos azules y las coordenadas alcanzadas en rojo. Las zonas azules de más puntos cercanos son movimientos prolongados cuando la diferencia entre el ángulo de los vectores es grande.

Como se puede ver también en la figura 107, al terminar la trayectoria se va desplazando nuevamente al punto inicial.

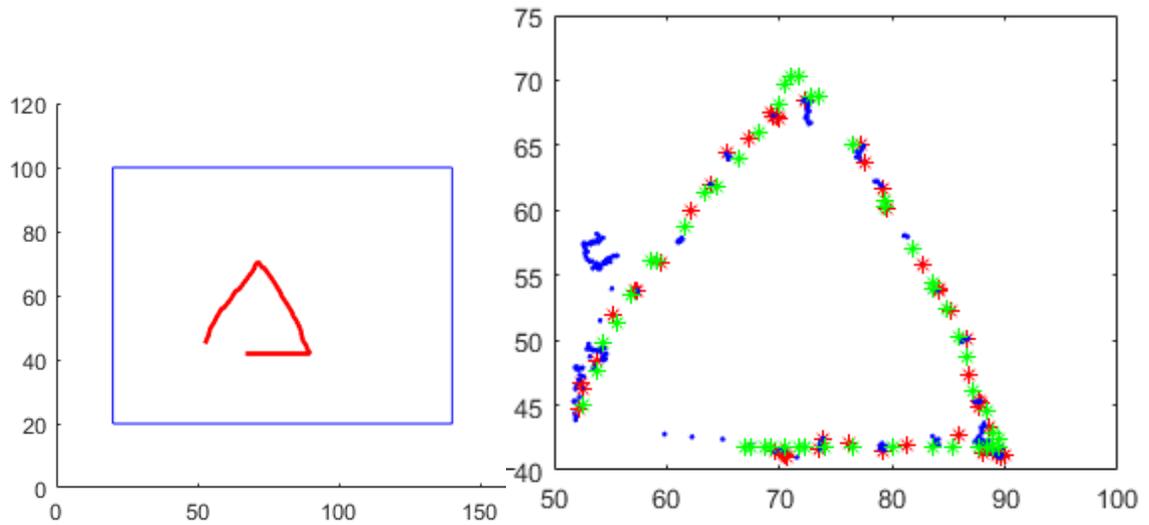
Otros trayectos dibujados aleatoriamente para evaluar el modelo cinemático.

Figura 108 TRAYECTORIA DE PRUEBA



Fuente: Autor

Figura 109 TRAYECTORIA DE PRUEBA



Fuente: Autor

4.5 INTERFAZ GRAFICA

En la figura 110 se observa la interfaz, y a continuación el código de ejecución en cada uno de los botones de la GUI.

Figura 110 INTERFAZ GRAFICA DE USUARIO



Fuente: Autor

En la ejecución del botón de dibujar se encuentra la función `freehanddraw`, que permite realizar el dibujo a mano alzada sobre el plano.

```
BOTON DE DIBUJAR
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(~, ~, handles)
% hObject     handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
axes(handles.axes1);
rectangle('Position',[20 20 120 80],'EdgeColor','b');
set(gca,'xlim',[0 160]);
set(gca,'ylim',[0 120]);
[~, xs, ys] = freehanddraw (gca,'color','r','linewidth',2);
set(handles.axes1,'UserData',xs);
set(handles.text3,'UserData',ys);
return
```

Para cargar el trayecto se toman las coordenadas de la ruta del plano bidimensional y se muestran en un nuevo plano para comparar los recorridos del robot y el dibujado.

```
CARGAR TRAYECTO
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(~, ~, handles)
% hObject     handle to pushbutton2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
xs=get(handles.axes1,'userdata');
ys=get(handles.text3,'userdata');
lx=length(xs);
ly=length(ys);
chr1 = int2str(lx);
chr2 = int2str(ly);
set(handles.edit1,'string',chr1);
set(handles.edit2,'string',chr2);
axes(handles.axes2);
plot(xs, ys, '*r'), hold on
plot(xs, ys, 'm'), hold off
```

Establecido el trayecto y conociendo los puntos se puede empezar la captura de las imágenes para el procesamiento, extraer la información, establecer la comunicación y ejecutar las ordenes enviadas por el controlador.

```
BOTON DE VIDEO
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton3 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```

%BOTON PARA FINALIZAR VIDEO
set(handles.pushbutton4,'userdata',0);
%VALORES INICIALES PARA EL AJUSTE DE LA IMAGEN
handles.low_in = 0;
handles.high_in = 1;
handles.low_out = 0;
handles.high_out = 1;
handles.alpha = 1;
%% CONECTAR BLUETOOTH
bt = instrhwinfo ('Bluetooth');
IDbt = bt.RemoteIDs;
%IDnm = bt.RemoteNames;
%ID del HC06 btspp://98D33380B57B
L = length(IDbt);
for im = 1:L
    s1 = 'btspp://98D33380B57B';
    s2 = IDbt(im,1);
    tf = strcmp(s1,s2);
    if tf == 1
        %set(handles.edit12,'string','ENCONTRADO!');
        bt = Bluetooth (s1, 1);
        set(handles.text13,'userdata',bt);
        fopen(bt);
    end
end

set(handles.edit12,'string','CONECTADO!');
webcamlist;
camera = webcam(1);
camera.Resolution = '160x120';
preview(camera);
pause(3);

i=0; %Para graficar las velocidades
hc=1; %variable para recorrer cada uno de los puntos del trayecto
distancia=0; %Acumulador de la distancia que se va recorriendo

%CARGAR LA TRAYECTORIA
xt = get(handles.axes1,'userdata');
yt = get(handles.text3,'userdata');
[fc, ~] = size(yt);
yta = yt;
for o = 1:fc
    yta(o,1) = 120 - yt(o,1); %Ajuste a los ejes
end

while get(handles.pushbutton4,'userdata') == 0
%% VISION ARTIFICIAL
    rgbIm = snapshot(camera); % Take a picture

    handles.low_in = get(handles.slider3,'Value');
    handles.high_in = get(handles.slider4,'Value');
    handles.low_out = get(handles.slider5,'Value');

```

```

handles.high_out = get(handles.slider6, 'Value');
handles.alpha = get(handles.slider7, 'Value');
imad = imadjust(rgbIm, [handles.low_in
handles.high_in], [handles.low_out handles.high_out], [handles.alpha]);
hsvIm = rgb2hsv(imad);
% canales HSV
hIm = hsvIm(:, :, 1);
sIm = hsvIm(:, :, 2);
vIm = hsvIm(:, :, 3);

    umbralMatizbajo = 0.15;
    umbralMatizalto = 0.60;
    umbralSaturacionbajo = 0.36;
    umbralSaturacionalto = 1;
    umbralValorbajo = 0;
    umbralValoralto = 0.8;

%
matiz = (hIm >= umbralMatizbajo) & (hIm <= umbralMatizalto);
sat = (sIm >= umbralSaturacionbajo) & (sIm <= umbralSaturacionalto);
valor = (vIm >= umbralValorbajo) & (vIm <= umbralValoralto);

colorObj = uint8(matiz & sat & valor);

ImaR = colorObj .* rgbIm(:, :, 1);
ImaG = colorObj .* rgbIm(:, :, 2);
ImaB = colorObj .* rgbIm(:, :, 3);
nRGBIm = cat(3, ImaR, ImaG, ImaB);

BW = imbinarize(colorObj, 'adaptive');
%filtrado por area
BW2 = bwareaopen(BW, 50);
se = strel('disk', 3);
dilatedIm = imdilate(BW2, se);

axes(handles.axes3);
imshow(dilatedIm); hold on; % Show the picture

[q, n] = bwlabel(dilatedIm);
q1 = regionprops(q);
cn=cat(1, q1.Centroid);
display(n);
%% TRAYECTO A REALIZAR
plot(xt, yta, 'g');hold on;

drawnow;
%%
if n==3
% plot(cn(1,1),cn(1,2), 'c*');hold on;
% plot(cn(2,1),cn(2,2), 'c*');hold on;

```

```

% plot(cn(3,1),cn(3,2),'c*');hold on;
%% DISTANCIAS ENTRE LOS CENTROIDES, IDENTIFICACION DE LA BASE DEL
TRIANGULO
d1 = sqrt((cn(2,1)-cn(1,1))^2+(cn(2,2)-cn(1,2))^2);%Distancia P1->P2
d2 = sqrt((cn(3,1)-cn(2,1))^2+(cn(3,2)-cn(2,2))^2);%Distancia P2->P3
d3 = sqrt((cn(3,1)-cn(1,1))^2+(cn(3,2)-cn(1,2))^2);%Distancia P3->P1
    if d1<d2 && d1<d3
        px2 = cn(2,1);
        px1 = cn(1,1);
        py2 = cn(2,2);
        py1 = cn(1,2);
        xm = (px1+px2)/2;
        ym = (py1+py2)/2;
%         plot(xm,ym,'y*'), hold on
    end
    if d2<d1 && d2<d3
        px2 = cn(2,1);
        px3 = cn(3,1);
        py2 = cn(2,2);
        py3 = cn(3,2);
        xm = (px3+px2)/2;
        ym = (py3+py2)/2;
%         plot(xm,ym,'y*'), hold on
    end
    if d3<d2 && d3<d1
        px3 = cn(3,1);
        px1 = cn(1,1);
        py3 = cn(3,2);
        py1 = cn(1,2);
        xm = (px1+px3)/2;
        ym = (py1+py3)/2;
%         plot(xm,ym,'y*'), hold on
    end
end
%% ENTRE EL PUNTO MEDIO DE LA BASE Y EL PIXEL MAS LEJANO A ESTE PUNTO
% FORMAN EL VECTOR DE DESPLAZAMIENTO DEL ROBOT Y PERMITEN CONOCER SU
%POSICIÓN
dp1 = sqrt((xm-cn(1,1))^2+(ym-cn(1,2))^2);%Distancia P1->PC
dp2 = sqrt((xm-cn(2,1))^2+(ym-cn(2,2))^2);%Distancia P2->PC
dp3 = sqrt((xm-cn(3,1))^2+(ym-cn(3,2))^2);%Distancia P3->PC

    if dp1>dp2 && dp1>dp3
        plx = cn(1,1); %pixel lejano
        ply = cn(1,2); %pixel lejano
        xm1 = (xm+plx)/2; %centro del carro
        ym1 = (ym+ply)/2; %centro del carro
        plot(xm1,ym1,'g*'), hold on;
        %UBICACION DEL CENTRO DEL TRIANGULO(POSICIÓN DEL ROBOT)
        pcx = [xm1 plx];
        pcy = [ym1 ply];
        %SE DIBUJA EL VECTOR DE DESPLAZAMIENTO DEL CARRO
        line(pcx,pcy,'color','c','linewidth',2), hold on;
    end
end
    if dp2>dp1 && dp2>dp3
        plx = cn(2,1);

```

```

        ply = cn(2,2);
        xml = (xm+plx)/2;
        yml = (ym+ply)/2;
        plot(xml,yml,'g*'), hold on;
        %UBICACION DEL CENTRO DEL TRIANGULO(POSICIÓN DEL ROBOT)
        pcx = [xml plx];
        pcy = [yml ply];
        %SE DIBUJA EL VECTOR DE DESPLAZAMIENTO DEL CARRO
        line(pcx,pcy,'color','c','linewidth',2), hold on;
    end
    if dp3>dp2 && dp3>dp1
        plx = cn(3,1);
        ply = cn(3,2);
        xml = (xm+plx)/2;
        yml = (ym+ply)/2;
        plot(xml,yml,'g*'), hold on;
        %UBICACION DEL CENTRO DEL TRIANGULO(POSICIÓN DEL ROBOT)
        pcx = [xml plx];
        pcy = [yml ply];
        %SE DIBUJA EL VECTOR DE DESPLAZAMIENTO DEL CARRO
        line(pcx,pcy,'color','c','linewidth',2), hold on;
    end

%% VISUALIZACION DE LOS EJES DE REFERENCIA PARA LA MEDICION DEL ANGULO
% DE ORIENTACION DEL CARRO
% EJE DE REFERENCIA UBICADO EN EL CENTRO DEL ROBOT, COLOR AMARILLO
%eje x
% prx1=0;prx2=160;pry1=yml;pry2=yml;
% plot(prx1,pry1,'y*'), hold on
% plot(prx2,pry2,'y*'), hold on
% ecr1=[prx1 prx2];
% ecr2=[pry1 pry2];
% line(ecr1,ecr2,'color','y','linewidth',1,'linestyle','--'), hold on
% %eje y
% prx_1=xml;prx_2=xml;pry_1=120;pry_2=0;
% plot(prx_1,pry_1,'y*'), hold on
% plot(prx_2,pry_2,'y*'), hold on
% eyr1=[prx_1 prx_2];
% eyr2=[pry_1 pry_2];
% line(eyr1,eyr2,'color','y','linewidth',1,'linestyle','--'), hold on
%% MEDICION DEL ANGULO DEL VECTOR DE DESPLAZAMIENTO DEL ROBOT
dx1 = xml-plx;
dy = yml-ply;
if xml>plx
    dx = -dx1;
end
if xml<plx
    dx = -(dx1);
end
ag = atan(dy/dx); %radianes
agt = (agt*180)/pi;
agt1 = round(agt,0);
%DETERMINAR EL CUADRANTE DEL VECTOR
if dx>0 && dy>0

```

```

    chrang = int2str(agt1);
    dgc = agt1;
    %Angulo teta, angulo del carro
    set(handles.edit7, 'string', chrang);
end
if dx<0 && dy>0
    agt2 = agt1+180;%2 cuadrante
    dgc = agt2;
    %Angulo teta, angulo del carro
    chrang = int2str(agt2);
    set(handles.edit7, 'string', chrang);
end
if dx<0 && dy<0
    agt3 = agt1+180;%3 cuadrante
    dgc = agt3;
    %Angulo teta, angulo del carro
    chrang = int2str(agt3);
    set(handles.edit7, 'string', chrang);
end
if dx>0 && dy<0
    agt4 = agt1+360;%4 cuadrante
    dgc = agt4;
    %Angulo teta, angulo del carro
    chrang = int2str(agt4);
    set(handles.edit7, 'string', chrang);
end
%% VISUALIZACION DEL VECTOR GUÍA DEL ROBOT
ro1=[xm1 xt(hc)];
ro2=[ym1 yta(hc)];
line(ro1,ro2,'color','w','linewidth',1,'linestyle','--'), hold on;
%% MEDICION DEL ANGULO DEL VECTOR GUÍA EN FUNCION DEL PLANO
dxu1 = xm1-xt(hc);
dyu = ym1-yta(hc);
if xm1>xt(hc)
    dxu = -dxu1;
end
if xm1<xt(hc)
    dxu = -(dxu1);
end
agu = atan(dyu/dxu); %radianes
agt1 = (agu*180)/pi;
agt1 = round(agt1,0);
if dxu>0 && dyu>0
    chrangu = int2str(agt1);
    dgc1 = agt1;
    %ANGULO DE VECTOR GUÍA DEL ROBOT, phi
    set(handles.edit17, 'string', chrangu);
end
if dxu<0 && dyu>0
    agt2 = agt1+180;%2 cuadrante
    dgc1 = agt2;
    %ANGULO DE VECTOR GUÍA DEL ROBOT, phi
    chrangu = int2str(agt2);
    set(handles.edit17, 'string', chrangu);

```

```

end
if dxu<0 && dyu<0
    agtu3 = agtu1+180;%3 cuadrante
    dgcu = agtu3;
    %ANGULO DE VECTOR GUÍA DEL ROBOT, phi
    chrangu = int2str(agt3);
    set(handles.edit17, 'string', chrangu);
end
if dxu>0 && dyu<0
    agtu4 = agtu1+360;%4 cuadrante
    dgcu = agtu4;
    %ANGULO DE VECTOR GUÍA DEL ROBOT, phi
    chrangu = int2str(agt4);
    set(handles.edit17, 'string', chrangu);
end
%% DIFERENCIA ENTRE LOS DOS ÁNGULOS DELTA DE PHI
% si dphi es mayor a 180 (negativo para que gire izquierda)
% si dphi es menor de -180 (positivo para que gire derecha)
dphi = dgc-dgcu;
%GRAFICA DEL DELTA DE LOS ÁNGULOS
% axes(handles.axes2);
% plot(i,dphi,'c*');hold on;
i=i+1;
if dphi>180
    dphi = -180;
    %display('IZQUIERDA'); %se vuelve negativo
end
if dphi<-180
    dphi = 180;
    %display('DERECHA'); %se vuelve positivo
end
crang1 = num2str(dphi);
set(handles.edit22, 'String', crang1);
%% CONTROLADOR
% LEER LA VELOCIDAD LINEAL DE LAS RUEDAS ENVIADAS POR ARDUINO
recbyte = fread(bt,9);
d = char(recbyte(1:9));
d1 = d';
token = strtok(d1, ';');
val = str2double(token);
d1 = floor(val/10000);
d = mod(val,10000);
vder = d1-1000;
vizq = d-1000;
chrder = int2str(vder);
chrizq = int2str(vizq);
set(handles.edit19, 'String', chrizq);
set(handles.edit21, 'String', chrder);
% ENTRADAS AL CONTROLADOR
rd = 0.06;%radio de la rueda (Metros)
ln = 0.47;%distancia entre las ruedas (Metros)
%VELOCIDAD LINEAR Y ANGULAR DEL ROBOT
v = ((vder+vizq)*rd)/2;%velocidad lineal
w = ((vder-vizq)*rd)/ln;%velocidad angular

```

```

% %xpunto y ypunto
drx = v*cosd(dgc);%xpunto
dry = v*sind(dgc);%ypunto
fx = (drx*cosd(dgc))/rd;
fy = (dry*sind(dgc))/rd;
dpln = (dphi*ln)/rd;
wd = (fx+fy+dpln);
wi = (fx+fy-dpln);
% % VELOCIDAD DE RUEDA IZQ Y DER OBTENIDAS
vi = wi*rd;
vd = wd*rd;
% % POSICIÓN (EN METROS) DEL CARRO
mtx=2.45;mty=1.95;%MEDIDAS REALES DEL ESPACIO RECREADO
pfx1 = (xm1*mtx)/160;pfy1=(ym1*mty)/120;
pfy1 = mty-pfy1;
chr3 = num2str(pfx1);
chr4 = num2str(pfy1);
set(handles.edit5,'string',chr3);
set(handles.edit6,'string',chr4);
% %% MARGEN DE ERROR PERMITIDO EN LA ORIENTACION DEL ROBOT
% % PARA DIRIGIRSE DIRECTAMENTE HACIA LA META.
if dphi<=10 && dphi>=-10
    vi = 1;
    vd = 1;
end
%% MEDIR EL ERROR EN LA POSICIÓN DEL ROBOT
erx = xt(hc)-xm1;
ery = yta(hc)-ym1;
ltr = length(yta);
%GRAFICA TRAYECTO
axes(handles.axes2);
plot(xt, yt,'g*');hold on;
plot(xm1, 120-ym1, '.b'), hold on
if hc <= ltr
    %MARGEN DE ERROR PERMITIDO EN LA POSICIÓN PARA RECONOCER CUANDO
% %HA SIDO ALCANZADA LA META Y ASI PASAR AL SIGUIENTE OBJETIVO
    if (erx>=-3 && erx<=3)
        if (ery>=-3 && ery<=3)
            %GRAFICAR PUNTO META CUMPLIDA EN TRAYECTO
            plot(xm1, 120-ym1, 'r*'), hold on
            vi = 0;
            vd = 0;
            %DISTANCIA ENTRE LOS PUNTOS QUE RECORRE
            pfx2 = (xt(hc)*mtx)/160; pfy2 = (yta(hc)*mty)/120;
            terx = (pfx2-pfx1)^2; tery = (pfy2-pfy1)^2;
            dist = sqrt(terx+tery);
            distancia = distancia+dist;
            hc = hc+1;
        end
    end
end
%% RECORRIDAS TODAS LAS POSICIONES, VUELVE A LA 1ERA POSICIÓN
% Y MUESTRA LA DISTANCIA RECORRIDA EN METROS
if hc>ltr

```

```

        disp(distancia);
        hc=1;
    end
    %% ENVIAMOS LA DATA
    srli=int2str(uint8(vi));srld=int2str(uint8(vd));
    srlil=str2double(srli);srldl=str2double(srld);
    vli=1000+srli1;vld=1000+srld1;
    %disp(vli);disp(vld);
    ss1=num2str(vli);ss2=num2str(vld);
    li=length(ss1);ld=length(ss2);
    for in=1:li
        I(in)=str2double(ss1(in));
    end
    I(li+1)=': ';
    for in=1:ld
        D(in)=str2double(ss2(in));
    end
    D(ld+1)=': ';
    dato = horzcat(I,D);
    lt=length(dato);
    disp(dato);
    for in=1:lt
        fwrite(bt, char(dato(in)));
    end

end

end
end

```

Botón finalizar, termina el proceso de captura de imágenes y Cierra el canal de comunicación mediante bluetooth, todo esto para que se pueda cargar un nuevo trayecto.

```

BOTON FINALIZAR
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
bt=get(handles.text13,'userdata');
fclose(bt);
closepreview();
set(handles.edit12,'string','DESCONECTADO');
set(handles.pushbutton4,'userdata',1);

```

CONCLUSIONES

1. El robot de tracción diferencial puede funcionar para desarrollar el prototipo, pero este modelo de robot no es el más apto para realizar la tarea de voltear el café. De ahí que se desarrolla el prototipo en función de un robot tipo Skid steer, cuyo modelo se aproxima al modelo de un robot de tracción diferencial y es una plataforma robótica que puede corregir su orientación sin necesidad de trasladarse.
2. La comunicación bluetooth entre el robot y algoritmo es buena, sin embargo, es necesario establecer un tamaño para el buffer de datos de envío y recibo en el algoritmo. Esto permite codificar la información del tamaño del buffer y llegar a destino y ser decodificada sin perder ningún dato. Ampliar o disminuir el tamaño del buffer puede ocasionar errores en la lectura de las velocidades del modelo.
3. La combinación de los modelos de color RGB y HSV permiten segmentar los colores de una manera mucho más precisa. Modificar la intensidad de los valores en una imagen en RGB cambian el histograma de los canales de HSV, logrando identificar un umbral para los colores dentro de un fondo uniforme.
4. El umbral del color cambia con la incidencia de la luz solar en el espacio recreado, cambiando el histograma, lo que hace que se deba ajustar manualmente la intensidad de los valores de la imagen RGB perdiendo autonomía el robot. Lo que descarta el uso de imágenes para identificar el robot en el patio de café.
5. La velocidad de procesamiento de imágenes obtenida no es la mejor, el tiempo de ejecución del algoritmo en la interfaz obliga a que el robot deba ir más lento para obtener las medidas precisas de los ángulos y su posición.
6. La velocidad de la captura de imágenes obliga a que existan estos márgenes de error, de no tener estos márgenes, el robot nunca alcanzaría la posición correcta porque se adelantaría y hasta podría salirse de la zona de captura de la cámara. Tampoco podría orientarse, cuando se hace la captura de la imagen el carro siempre va más rápido.
7. El modelo de control cinemático, aunque con todas las restricciones que causa el procesamiento de imágenes, es bueno. pues podemos ver en las figuras 107, 108 y 109 que sigue el trayecto dentro del margen de error aceptado por el usuario.

BIBLIOGRAFÍA

- [1] Acevedo, H. G., Suárez, M. B., & Cantillo, J. C. C. NAVEGACIÓN DE UN ROBOT MÓVIL MEDIANTE VISIÓN ARTIFICIAL.
- [2] Cajiga, J. S., Soto, J. V., & Ortega, J. P. (2006). Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines. In Segundo Congreso Internacional de Ingeniería, UAQ, marzo.
- [3] Chávarro, A. F. C., Antolínez, L. K. R., & Cruz, K. J. Z. (2017). Modelamiento, simulación y control de posicionamiento automático de un robot móvil con tracción diferencial como herramienta para apoyar la formación en robótica en ambientes de aprendizaje SENA. *Revista del Sistema de Ciencia Tecnología e Innovación (SENNOVA)*, 2(2), 50-69.
- [4] Ortigoza, R. S., Marcelino-Aranda, M., Ortigoza, G. S., Guzman, V. M. H., Molina-Vilchis, M. A., Saldana-Gonzalez, G., ... & Olguin-Carbajal, M. (2012). Wheeled mobile robots: a review. *IEEE Latin America Transactions*, 10(6), 2209-2217.
- [5] Quintero, B.B., Zaráish, L., Cruz, B.G., & Rocimar, E. GENERACION DE TRAYECTORIAS PARA EL DESPLAZAMIENTO DE UN ROBOT MOVIL EN ENTORNOS NO ESTRUCTURADOS UTILIZANDO VISION ARTIFICIAL.
- [6] Azcón, A. B. (2003). Análisis y diseño del control de posición de un robot móvil con tracción diferencial. *Escola Tecnica Superior Enginyeria, Universitat Povira, Departament D'enginyeria Electronica Electrica I Automtica. I Virgili*.
- [7] Pérez Pérez, N. C., Avellaneda, S., & Arturo, D. (2009). Diseño, modelamiento y simulación 3D de un robot móvil para exploración de terrenos.
- [8] Jiménez Jiménez, C. (2017). Desarrollo de un sistema de control de un robot móvil.
- [9] MORENO, C. A. G. (2016). *Desarrollo de una aplicación para el guiado automático de un vehículo eléctrico* (Doctoral dissertation).
- [10] Rossius, S. (2013). *Reconocimiento de objetos mediante WebCam en tiempo real* (Doctoral dissertation).
- [11] Vieira, F. C., Medeiros, A. A., Alsina, P. J., & Araújo Jr, A. P. (2004). Position and Orientation Control of a Two-Wheeled Differentially Driven Nonholonomic Mobile Robot. In *ICINCO (2)*(pp. 256-262).
- [12] García, I. (2008). Visión Artificial y Procesamiento digital de imágenes usando Matlab. *Ibarra (Ecuador): Escuela de Ingeniería de la PUCE-SI*.
- [13] Zhou, H., Wu, J., & Zhang, J. (2010). *Digital Image Processing: Part II*. Bookboon.
- [14] La imagen digital. [Online]. <http://www.slideshare.net/martapotoc/la-imagen-digital-28010025>

- [15] Martínez, E. Curso de procesamiento digital de imágenes. *Obtenido de http://turing.iimas.unam.mx/~elena/PDI-Lic/Tema_3_AB.pdf*.
- [16] EL MÓDULO CONTROLADOR DE MOTORES L298N. [Online]. <https://www.promotec.net/l298n/#>
- [17] 50:1 Metal Gearmotor 37Dx70L mm with 64 CPR Encoder. [Online]. <https://www.pololu.com/product/2824>
- [18] ¿QUÉ ES ARDUINO? [Online]. <http://arduino.cl/que-es-arduino/>
- [19] Qué es Arduino. [Online]. <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- [20] CB1270. [Online]. <https://www.kelcom.cz/cgb-battery-cb1270-1095.html>
- [21] HC-06. [Online]. <http://www.electronicoscaldas.com/modulos-rf/482-modulo-bluetooth-hc-06.html>
- [22] Bluetooth HC-05 y HC-06 Tutorial de Configuración. [Online]. <https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>
- [23] Formación para el trabajo con ciencia y tecnología mejoraría productividad en el sector rural. [Online]. <http://www.mintrabajo.gov.co/web/guest/prensa/comunicados/2017/septiembre/formacion-para-el-trabajo-con-ciencia-y-tecnologia-mejoraria-productividad-en-el-sector-rural>
- [24] Kozłowski, K., D. Pazderski, I. Rudas, and J. Tar. "Modeling and Control of a 4-wheel Skid-steering Mobile Robot: From Theory to Practice." (n.d.): n. pag. Budapest Tech Jubilee Conference. Annie, 4 Sept. 2004. Web. 27 Apr. 2015.
- [25] Guzmán, L. E. S., Villa, M. A. M., & Vásquez, E. L. R. (2014). Seguimiento de trayectorias con un robot móvil de configuración diferencial. *Ingenierías USBMed*, 5(1), 26-34.
- [26] Obtener el ángulo dado un vector. [Online]. <https://geus.wordpress.com/2006/12/07/obtener-el-angulo-dado-un-vector/>
- [27] La nueva era tecnológica de la agricultura. [Online]. <http://www.vanguardia.com/mundo/tecnologia/389529-la-nueva-era-tecnologica-de-la-agricultura>
- [28] "El Huila tiene productividad por encima del promedio nacional". [Online]. <http://www.lanacion.com.co/2018/04/24/huila-productividad-del-promedio-nacional/>
- [29] Secado del café, otra de las claves para que el Café de Colombia sea de alta calidad. [Online]. <http://www.cafedecolombia.com/ci-fnc-es/index.php/comments/el-secado-del-cafe-otra-de-las-claves-para-que-el-cafe-de-colombia-adquiera/>
- [30] Secado de café al sol y mecánico. [Online]. <http://www.coffeeiq.co/secado-de-cafe-al-sol-y-mecanico/>

- [31] Beneficio del café II: secado del café pergamino. [Online]. https://www.cenicafe.org/es/publications/cartilla_21_Secado_del_cafe.pdf
- [32] Enfermedades causadas por la radiación solar. [Online]. <http://blog.cecofar.es/enfermedades-radiacion-solar/>
- [33] El uso de robots en tareas agrícolas. [Online]. <http://www.interempresas.net/Horticola/Articulos/151745-El-uso-de-robots-en-tareas-agricolas.html>
- [34] Análisis digital de imágenes y reconocimiento de objetos. [Online]. <http://visualismo.blogspot.com.co/2014/09/analisis-digital-de-imagenes-y.html>
- [35] Procesamiento de imágenes. [Online]. http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm
- [36] Onate, P. B., Lara-Cueva, R., & Rivadeneira, J. (2015, October). Towards a smart classroom: Development of an augmented reality application for education and tele-education. In *Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2015 CHILEAN Conference on* (pp. 395-400). IEEE.
- [37] Introducción al Diseño Gráfico. [Online]. <https://piktochart.com/es/blog/introduccion-al-diseno-grafico/>
- [38] Síntesis aditiva. [Online]. http://www.glosariografico.com/sintesis_aditiva
- [39] Síntesis sustractiva. [Online]. http://www.glosariografico.com/sintesis_sustractiva
- [40] Modelo HSV. [Online]. https://www.ecured.cu/Modelo_HSV
- [41] La teoría del color ¿Qué es el color? ¿Cuáles son los colores primarios y secundarios? [Online]. <https://beatrizxe.com/es/blog/teoria-del-color-que-es-el-color-colores-primarios-secundarios.html>
- [42] MODELOS DE COLOR (RGB, CMYK, HSV/HSL). [Online]. <https://ahenav.com/2014/04/09/modelos-de-color/>
- [43] El histograma. [Online]. https://www.uv.es/gpoei/eng/Pfc_web/realizado/histograma/histo.htm
- [44] Histograma de color. [Online]. https://es.wikipedia.org/wiki/Histograma_de_color
- [45] Vanacloig, V. L. A. (2011). El histograma una imagen digital.
- [46] Principales productores de café del mundo. [Online]. <http://www.infocafe.es/cafe/principales-productores-cafe.php>
- [47] Principios de funcionamiento del encoder cuadratura de efecto Hall para medición de velocidad en motores de corriente directa. [Online]. <https://www.puntofotante.net/FUNCIONAMIENTO-ENCODER-CUADRATURA-EFECTO-HALL.htm>
- [48] LifeCam HD-3000. [Online]. <https://www.microsoft.com/accessories/es-es/products/webcams/lifecam-hd-3000/t3h-00002>

[49] freehanddraw, by Brett Shoelson. [Online].
<https://www.mathworks.com/matlabcentral/fileexchange/7347-freehanddraw>