

**SISTEMA DE RECONOCIMIENTO AUTÓNOMO DE MADURACIÓN DE
FRUTOS DE CACAO MEDIANTE LA TÉCNICA DE VISIÓN ARTIFICIAL**

ANDRES YESID MARTÍNEZ ARDILA
WILBER FRANCISCO SANTAMARIA ALZA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA (UNAB)
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
PROGRAMA DE INGENIERÍA MECATRÓNICA
BUCARAMANGA

2019

**SISTEMA DE RECONOCIMIENTO AUTÓNOMO DE MADURACIÓN DE
FRUTOS DE CACAO MEDIANTE LA TÉCNICA DE VISIÓN ARTIFICIAL**

ANDRES YESID MARTÍNEZ ARDILA

WILBER FRANCISCO SANTAMARIA ALZA

Proyecto de grado para optar al título de Ingeniero Mecatrónico

Director:

Carlos Julio Arizmendi Pereira

Co-Director:

Hernando González Acevedo

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA (UNAB)

FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

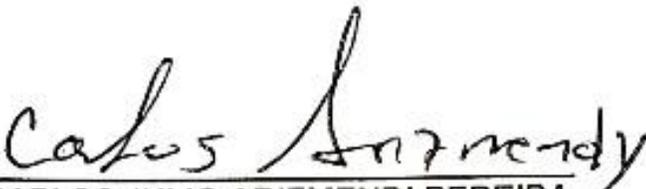
PROGRAMA DE INGENIERÍA MECATRÓNICA

BUCARAMANGA

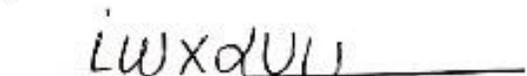
2019

Nota de aceptación

Firmas de aceptación proyecto de grado


CARLOS JULIO ARIZMENDI PEREIRA
Director de Proyecto de Grado
UNAB


DARÍO JOSE HERNÁNDEZ BOLÍVAR
Jurado de Proyecto de Grado
UNAB


JOHANN BARRAGÁN GÓMEZ
Jurado Proyecto de Grado
UNAB

Bucaramanga, 17 de junio de 2019

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	1
2. OBJETIVOS.....	3
2.1 Objetivo general	3
2.2 Objetivos específicos.....	3
3. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	4
4. MARCO TEÓRICO	6
4.1 Cultivo del cacao	6
4.2 Vehículo autónomo terrestre no tripulado (VATNT).....	8
4.3 Robótica móvil	9
4.3.1 Visión Artificial:.....	11
4.4 Redes Neuronales	14
5. METODOLOGÍA	17
5.1 Fases del proyecto	17
5.2 Metodología del proyecto	18
6. CRITERIOS PARA LA CLASIFICACIÓN DEL CACAO	20
7. DISEÑO DEL PROTOTIPO VATNT	22
7.1 Parte mecánica.....	22
7.1.1 Plataforma.....	22
7.2 Actuadores	23
7.2.1 Micromotor con eje extendido	24
7.2.2 Servomotor SG90.....	25

7.3	Sensores	26
7.3.1	Encoder	27
7.3.2	Ultrasónico	30
7.3.3	Magnetómetro	32
7.3.4	Cámara FPV (First Person View)	37
7.4	Dispositivos de comunicación.....	39
7.4.1	Xbee.....	40
7.4.2	Kit emisor receptor FPV (First Person View).....	41
7.5	Tarjetas.....	42
7.5.1	Arduino mega ADK.....	42
7.5.2	Puente H L298N.....	43
7.6	Suministro de energía.....	45
8.	TRAYECTORIA	48
8.1	Magnetómetro	48
8.2	Ultrasónicos.....	51
8.3	Modelo Matemático	54
8.3.1	Modelo eléctrico	54
8.3.2	Modelo mecánico	55
8.3.3	Modelo cinemático	55
8.3.4	Modelo dinámico	56
8.4	Desarrollo del VATNT.....	66
9.	COMUNICACIÓN	69
9.1	Comunicación digital.....	69

9.2	Comunicación análoga	73
10.	VISIÓN ARTIFICIAL.....	76
10.1	Clasificación por color.....	78
10.1.1	YCbCr	79
10.1.2	CIELAB	79
10.2	Primeras pruebas	80
11.	INTERFAZ GRÁFICA.....	81
12.	RESULTADOS OBTENIDOS.....	84
13.	CONCLUSIONES	86
14.	TRABAJO FUTURO.....	88
15.	BIBLIOGRAFÍA	89
16.	ANEXOS	93

LISTA DE FIGURAS

Figura 1 A Cacao Criollo B Cacao Forastero	7
Figura 2 A Cacao Trinitario B Cacao Sanvicentano	7
Figura 3 Modelo de red en cascada de tres capas	15
Figura 4 Diagrama de la metodología	19
Figura 5 Plataforma tanque con absorción de choque DIY SN800	23
Figura 6 Micromotor con caja de engranes 100:1 con eje extendido	24
Figura 7 Dimensiones de micromotor con eje extendido	25
Figura 8 Soporte para cámara con servomotores 9g de dos grados de libertad ...	26
Figura 9 Encoder (sensor hall)	27
Figura 10 Encoder instalado en motor	27
Figura 11 Encoder y disco magnético	28
Figura 12 Estados del encoder	29
Figura 13 Sensor ultrasónico HC_SR04	30
Figura 14 Funcionamiento sensor ultrasónico	31
Figura 15 Magnetómetro sin distorsión	34
Figura 16 Magnetómetro con distorsión de hierro duro	35
Figura 17 Magnetómetro con distorsión de metal duro y suave	36

Figura 18 Sensor magnetómetro HMC5883L	37
Figura 19 Cámara FPV CMOS 700 TVL.....	39
Figura 20 Modulo Xbee PRO	41
Figura 21 Conexión transmisor TS832	41
Figura 22 Arduino mega ADK	43
Figura 23 Conexión puente H	44
Figura 24 Batería de lipo 11.1 [v].....	46
Figura 25 Cargador LinkMan	46
Figura 26 Adaptador externo 12 [v].....	47
Figura 27 Diagrama de conexiones	47
Figura 28 Calibración magnetómetro.....	48
Figura 29 Datos magnetómetro	49
Figura 30 Software MagMaster.....	51
Figura 31 Calibración sensor ultrasónico.....	52
Figura 32 Datos de calibración sensor ultrasónico	53
Figura 33 Modelado de un motor DC.....	54
Figura 34 Identificación A) Entrada en %PWM para motor izquierdo. B) Salida en RPM motor izquierdo	57

Figura 35 Validación A) Entrada en %PWM para motor izquierdo. B) Salida en RPM motor izquierdo	57
Figura 36 Identificación A) Entrada en %PWM para motor derecho. B) Salida en RPM motor derecho.....	58
Figura 37 Validación A) Entrada en %PWM para motor derecho. B) Salida en RPM motor derecho.....	58
Figura 38 BestFit motor derecho A) Identificación. B) Validación	59
Figura 39 Respuesta motor derecho a una entrada escalón	59
Figura 40 BestFit motor izquierdo A) Identificación. B) Validación.....	60
Figura 41 Respuesta motor izquierdo a una entrada escalón.....	60
Figura 42 Arquitectura de controlador.....	61
Figura 43 Respuesta al controlador PI en lazo cerrado en el motor derecho	61
Figura 44 Acción de control PI en motor derecho	62
Figura 45 Respuesta al controlador PI en lazo cerrado en el motor izquierdo.....	62
Figura 46 Acción de control PI en motor izquierdo	63
Figura 47 Respuesta controlador PID motor derecho ante una entrada escalón ..	63
Figura 48 Respuesta controlador PID motor izquierdo ante una entrada escalón .	64
Figura 49 Bloque Simulink®	64

Figura 50 Arriba: acción de control motor izquierdo, Abajo: referencia y variable de proceso	65
Figura 51 Arriba: acción de control motor izquierdo, Abajo: referencia y variable de proceso	65
Figura 52 División del espacio de trabajo	66
Figura 53 Ubicaciones posibles desde el cual podrá tomar fotos	67
Figura 54 Ubicación óptima para hacer captura de imagen.....	67
Figura 55 Metodología para la trayectoria	68
Figura 56 Lectura de Xbee	70
Figura 57 Respuesta Xbee	71
Figura 58 Configuración Xbee	72
Figura 59 Metodología de comunicación entre Xbee.....	73
Figura 60 Cámara FPV y transmisor análogo.....	74
Figura 61 Capturadora de video	75
Figura 62 Diagrama de la conexión interna del terminal inalámbrico.....	75
Figura 63 Ubicación y orientación de la cámara	76
Figura 64 Mazorcas de Cacao	78
Figura 65 A) Fruto maduro YCbCr. B) Fruto no maduro YCbCr	79
Figura 66 A) Fruto maduro CIELAB. B) Fruto no maduro CIELAB	79

Figura 67 Pruebas de clasificación de mazorcas de cacao80

Figura 68 Aplicación HMI81

Figura 69 Función botón inicio82

Figura 70 Función botón Registros83

LISTA DE TABLAS

Tabla 1 Pines de interrupción de las diferentes tarjetas de control.....	29
Tabla 2 Comparación longitud focal con FOV	38
Tabla 3 Funcionamiento de cada canal	45
Tabla 4 Datos del magnetómetro.....	49
Tabla 5 Datos de calibración sensor ultrasónico.....	53
Tabla 6 Frecuencias de operación para la comunicación análoga	74
Tabla 7 Resultados trayectoria uno (1)	84
Tabla 8 Porcentajes de error	85

LISTA DE ANEXOS

1. INTRODUCCIÓN

El ser humano como ser biológico, ha venido evolucionando conforme pasan los años; en áreas como comunicación, educación, agricultura, vida familiar, empresa, la invención de la electricidad, la escritura, la computadora, entre otros; lo capacitó para guardar más información y ha permitido un avance de forma significativa en aras de una evolución más cultural, y así adaptarse en el medio y/o entorno en el que habita.

Esta evolución cultural se logró en gran medida, gracias a la evolución tecnológica, que nace a raíz de la necesidad de suplir las carencias, restricciones a ciertos lugares, la necesidad de potenciar y ampliar conocimientos con otras culturas, recorrer largas distancias; aparecieron las máquinas (aparatos más complejos), de esta forma el ser humano fue desarrollando mejor sus capacidades, especialmente en el área militar, especialidad donde se da inicio al uso de estas en : reconocimiento en lugares de vigilancia, combate y defensa de misiles, detención de minas antipersonal; en áreas civiles podemos encontrar su uso en: inspección y vigilancia de obras civiles, pesca, prevención de catástrofes , entre otras.

En la actualidad estas máquinas, robots o equipos de cómputo se han venido desarrollando con mejores características: mayor capacidad, menor espacio, más accesibles, tamaño nano, mejores funciones, variedad de sistemas de comunicación y control, lo anterior ha permitido que el ser humano pueda controlar, desarrollar y controlar de mejor manera su entorno sin necesidad de estar físicamente en él.

El presente trabajo está diseñado para que a partir de la aplicación de herramientas tecnológicas como lo es la computadora (la comunicación entre esta y el robot fue inalámbrica), tarjetas de adquisición de datos para la comunicación bidireccional de

ellos, actuadores, sistema de comunicación, sensores, microcontroladores, Permita desarrollar un robot (VATNT) con la capacidad de planificar trayectorias a diferentes objetivos y percibir un entorno. El diseño de la estructura del robot se realizó en SolidWorks, usando una red punto a punto que fue construida con módulos Xbee.

2. OBJETIVOS

2.1 Objetivo general

- Diseñar un sistema de reconocimiento de maduración en un fruto de cacao, utilizando técnicas de visión artificial, soportado en un vehículo autónomo terrestre no tripulado, en ambientes controlados.

2.2 Objetivos específicos

- Implementar un vehículo autónomo terrestre no tripulado que realice una trayectoria alrededor de máximo cuatro (4) arboles de cacao.
- Generar algoritmos que permitan a un robot autónomo trabajar en un entorno estructurado, siguiendo una trayectoria específica y bordeando obstáculos.
- Implementar un sistema electrónico de adquisición y transmisión inalámbrica de imágenes al computador.
- Diseñar un software clasificador de frutos maduros y no maduros del árbol de cacao mediante el uso de técnicas de procesamiento de imágenes e inteligencia artificial, en un entorno estructurado.

3. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

El “niño y la niña”, son fenómenos climáticos que han afectado severamente la vida económica, social y política de Colombia, debido a que producen nefastas alteraciones en el ciclo productivo y por tanto graves pérdidas económicas en áreas destinadas para la producción y comercialización de un determinado producto. El “niño” presenta una fase cálida y la “niña” su fase opuesta, es de aguas superficiales más frías; son nombres de dos fenómenos atmosféricos-oceánicos, que forman parte de un ciclo natural global del clima, siendo (ENOS) El Niño-Oscilación del Sur el modo dominante de variabilidad en el océano pacífico [1], [2].

Debido a la variación de estos fenómenos se afecta el ciclo regular en la maduración de los frutos de cacao, este hecho hace que algunos árboles o plantaciones maduren de manera no uniforme. Siendo el más afectado el trabajador o jornalero, quien es contratado por (destajo) modelo de contratación laboral en la que se reconoce una suma de dinero determinada a una persona por un trabajo realizado y no por el tiempo empleado, por lo anterior su pago será proporcional a la cantidad de fruto maduro recolectado; razón que ha generado negativa en los recolectores, pues no justifican invertir tiempo y dinero para recolectar una cantidad no representativa de frutos no maduros que no generan retribución monetaria, por tanto el propietario del cultivo y/o agricultor principal se ve obligado a incrementar el pago (hasta el doble del valor normal), con el fin de conseguir recolectores para su cosecha y personal que calcule un valor aproximado de la cantidad de frutos a recolectar; para determinar el valor aproximado a recolectar el propietario y/o agricultor principal debe realizar rondas periódicas de forma detallada a los extensos cultivos, convirtiéndose este proceso en poco óptimo debido a la inversión monetaria (contratar más trabajadores que realicen la labor) y tiempo empleado

supervisando cultivos que no permiten tener un cálculo acertado entre frutos maduros y no maduros y de esta forma reducir el margen de pérdida.

En razón a esta problemática se observa la necesidad de diseñar un sistema que permita al agricultor calcular el tiempo óptimo en la recolección de cultivos.

4. MARCO TEÓRICO

4.1 Cultivo del cacao

El árbol de cacao es originario de la Amazonía, Brasil, América del Sur, donde se extendió a América Central. La historia del cacao se desarrolla en Centro América hace más de 3,500 años. La palabra cacao proviene del maya “CAC” que significa rojo por el color de su mazorca y “CAU” que significa fuerza y fuego.

El árbol de cacao es una planta perenne, que una vez sembrada su vida útil es aproximadamente 30 años y puede llegar alcanzar una altura de más de 4 metros; su nombre científico es *Theobroma cocoa* que significa: Alimento de los dioses. Pertenece a la familia Malvaceae [3].

Originalmente existieron dos grupos de cacaos:

- **El Cacao Criollo:** Originario de Centroamérica, Colombia y Venezuela, se observa en la figura 1.A. Produce cacao dulce y de aroma intenso.
- **El Cacao Forastero o Calabacillo:** Es originario del Amazona, Brasil, se observa en la figura 1.B. Produce cacao amargo.

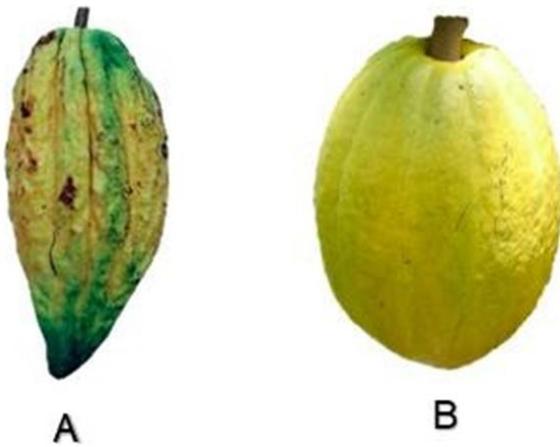


Figura 1 A Cacao Criollo B Cacao Forastero [3]

El cruce de estos dos grupos dio origen al Cacao Trinitario, se observa en la figura 2.A, llamado así, por tener su origen en la Isla de Trinidad ubicado frente a las costas de Venezuela. Del cruce repetido entre ellos, se originaron los diferentes tipos de cacao que se conocen y utilizan en la actualidad [3]. El Cacao San vicentano.se observa en la figura 2.B, es uno de los cacaos originados por estos cruces, proveniente de San Vicente de Chucuri principal productor de cacao en Colombia [4].

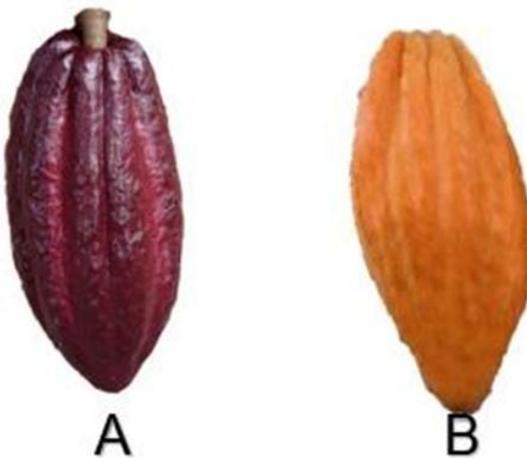


Figura 2 A Cacao Trinitario B Cacao Sanvicentano [4]

4.2 Vehículo autónomo terrestre no tripulado (VATNT)

Un Vehículo Autónomo Terrestre No Tripulado (VATNT), es un vehículo diseñado para imitar algunas de las capacidades humanas de control y manejo que permite realizar contacto con la superficie sin necesidad de un conductor a bordo o ser tripulado.

Su origen se da como una respuesta a ciertas necesidades militares, especialmente lo relacionado con labores de inteligencia y reconocimiento de áreas de difícil acceso, a partir de ello hoy en día han tenido mayor incidencia en actividades no militares, debido a que han venido diseñándose diferentes prototipos que permiten realizar labores específicas como: Rastreo, monitoreo en zonas de catástrofes, localización de víctimas, búsqueda de objetos perdidos, para realizar actividades que hacen imposible la presencia de un operador humano o que al realizarla represente un peligro inminente para quien la ejecute.

El vehículo cuenta con un conjunto de sensores que permiten observar y tener una comprensión limitada del medio ambiente, mediante la aplicación de algoritmos de control capaces de tomar decisiones autónomas sobre la siguiente acción a realizar, a través de los datos proporcionados por un humano.

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la robótica, su importancia radica principalmente en que poseen un espacio de trabajo determinado por sus dimensiones físicas con la finalidad de aumentar la movilidad del robot y de esta manera su capacidad de trabajo, mediante el uso de un sistema locomotor, para que el robot pueda desplazarse libremente en su espacio de trabajo.

El concepto de autonomía en robótica, hace referencia a la capacidad de tomar decisiones propias sin tener que ser controlado por algo o alguien y de esta forma

pueda responder ante situaciones cambiantes sin la intervención humana; estos robots se caracterizan por la capacidad para desplazarse de forma autónoma en entornos conocidos o desconocidos.

Un robot completamente autónomo tiene la capacidad de:

- Recopilar información sobre el medio ambiente.
- Detectar objetos de interés.
- Viajar entre puntos de ruta, sin asistencia de navegación humana.
- Trabajar por periodos prolongados sin intervención humana.
- Evitar situaciones que son perjudiciales para las personas y bienes.

Los diferentes comandos dados por el humano a un robot con estas características permiten realizar estrategias de cooperación entre ellos; esto permite obtener un mayor cubrimiento de áreas en un menor tiempo de ejecución.

4.3 Robótica móvil

Los robots móviles se pueden clasificar de acuerdo con el tipo de locomoción utilizado, los sistemas más conocidos son: Ruedas, patas y orugas.

Un robot móvil con ruedas (RMR), es un vehículo capaz de moverse de manera autónoma sobre una superficie, mediante la acción de las ruedas montadas en el robot. Considerando que un RMR es capaz de moverse sobre una superficie mediante la acción de ruedas montadas en él, se asumen las siguientes hipótesis:

- El robot móvil se mueve sobre una superficie plana horizontal, es decir, la energía potencial es constante.
- Los ejes guiados son perpendiculares al suelo.
- El contacto entre cada rueda y el suelo es tangencial.
- No puede existir deslizamiento.

A diferencia de los robots fijos, los móviles se encuentran en constante cambio de posición y la orientación de acuerdo a su eje de coordenadas base (sistema global) [5], de ahí que deban estar en constante reconocimiento del entorno que lo rodea para adaptarse y realizar un movimiento natural que le permita evadir los obstáculos que se presentan. Para que un sistema de robótica móvil pueda trasladarse en un entorno debe contar con sensores que le permitan monitorear el ambiente constantemente [6]. La navegación más básica de los robots móviles se basa en el modelo cinemático del sistema de propulsión, es decir, determinar el modelo de la configuración de actuadores que permiten al robot moverse dentro del entorno [7], [6]. De las configuraciones de robots móviles, la más utilizada, es la de tracción diferencial, debido a que es un sistema simple y adecuado para la navegación en entornos cotidianos no muy exigentes. Este tipo de direccionamiento viene dado por la diferencia de velocidades de las ruedas laterales. Dos ruedas montadas en un único eje son propulsadas y controladas independientemente, proporcionando tracción y direccionamiento, además, permite cambiar la orientación del robot sin movimientos de traslación [7], [8]. Un problema importante es cómo resolver el equilibrio del robot, debido a que es necesario buscar un apoyo adicional a las dos ruedas ya existentes, esto se consigue mediante una o dos ruedas de apoyo añadidas en un diseño triangular o romboidal; otro problema que presenta la configuración de tracción diferencial, es desplazarse en línea recta, ya que para que

el robot pueda lograr esto, sus dos ruedas deben girar a la misma velocidad [9], pero si cada rueda experimenta una fricción diferente sus velocidades van a variar, este problema debe ser solucionado con un sistema de control dinámico que vaya variando las velocidades a medida que el robot lo necesite [7], [9].

Para dar solución al problema, se realiza el modelo del robot diferencial, con el objetivo de tener una representación matemática que describa la cinemática del sistema [7], [10]. Posteriormente se realiza una aproximación de las variables de estado, con esto se implementa un sistema de control que permite disminuir la diferencia entre los estados deseados y los del sistema [5], [7].

4.3.1 Visión Artificial:

Cualquier robot diseñado para moverse en un entorno desconocido necesita de múltiples sensores, adaptando estrategias de control, con capacidad de procesar información procedente del exterior, que le permitan cumplir con la tarea propuesta sin colisionar. Una alternativa utilizada actualmente para la navegación de robots móviles son los sistemas de procesamiento digital de imágenes.

La visión artificial es un proceso que consta de etapas: esta inicia con la captura de una imagen que permite hacer un pre-procesado con el fin eliminar ruidos, corregir imperfecciones y eliminar a su vez los objetos que no son de interés, a través de un proceso llamado segmentación, para luego empezar con el análisis. Al separar los objetos de importancia se extraen características de este como: color, tamaño o forma, se guardan en una base de datos para seguir con el reconocimiento el cual consta de un clasificador [11].

Para el análisis de la imagen se debe tener en cuenta el fondo y el ambiente, tener claro que no se deben alterar sus condiciones, por esta razón es importante que el

área de trabajo cuenta con una iluminación adecuada; por otro lado se considera el espacio de color utilizado, siendo RGB (Red Green Blue) el más común en las cámaras digitales, en comparación con otros espacios de color como: YCbCr (Luminancia y crominancia), HSI (tono, saturación e intensidad), CMYK (cian, magenta, amarillo y negro) y CIEL*a*b*.

YCbCr

Es un espacio de color utilizado en sistemas de fotografía y video digital. Define el color en términos de un componente de luminancia y dos de crominancia. Y representa la luminancia (luma) y se encuentra en el rango de 0 a 255. Cb y Cr representan la crominancia (chroma), los colores azul y rojo respectivamente. Se encuentran en el rango 0 a 255.

Las señales YCbCr y Y'CbCr son una aproximación práctica para el procesamiento del color y para la percepción uniforme, en las cuales los colores primarios que corresponden aproximadamente al rojo, verde y azul son procesados en información perceptualmente significativa. De esta manera, el procesamiento, la transmisión y almacenamiento posteriores de imagen o vídeo pueden introducir errores en formas perceptualmente significativas. A partir de la señal del modelo Y'CbCr se separa una señal de luminancia que se puede almacenar con alta resolución o transmitida con un alto ancho de banda, y dos componentes de crominancia que pueden ser de ancho de banda reducido, sub-muestreados, comprimidos, o tratados de otra manera por separado para mejorar la eficiencia del sistema.

$$Y' = 0 + (0.299 * R'_D) + (0.587 * G'_D) + (0.114 * B'_D) \quad (1)$$

$$C_B = 128 - (0.168736 * R'_D) - (0.331264 * G'_D) + (0.5 * B'_D) \quad (2)$$

$$C_R = 128 + (0.5 * R'_D) - (0.41688 * G'_D) - (0.081312 * B'_D) \quad (3)$$

CIELAB

El espacio CIELAB es el modelo cromático usado normalmente para describir todos los colores que puede percibir el ojo humano. Fue desarrollado específicamente con este propósito por la Commission Internationale d'Eclairage (Comisión Internacional de la Iluminación), razón por la cual se abrevia CIE. Los asteriscos (*) que siguen a cada letra forman parte del nombre, ya que representan L^* , a^* y b^* , de L , a y b .

Los tres parámetros en el modelo representan la luminosidad de color (L^* , $L^*=0$ rendimientos negro y $L^*=100$ indica blanca), su posición entre rojo y verde (a^* , valores negativos indican verde mientras valores positivos indican rojo) y su posición entre amarillo y azul (b^* , valores negativos indican azul y valores positivos indican amarillo).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.35758 & 0.180423 \\ 0.212671 & 0.71516 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

$$L^* = 116 * h\left(\frac{Y}{Y_w}\right) - 16 \quad (5)$$

$$a^* = 500 \left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right) \right] \quad (6)$$

$$b^* = 200 \left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right) \right] \quad (7)$$

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.00885 \\ 7.787q + \frac{16}{116} & q \leq 0.00885 \end{cases}$$

Donde X_w , Y_w , Z_w son los valores tri-estimulo CIE 'XYZ' del punto blanco de referencia [11].

HSI

Este espacio de color está basado en como el ojo humano percibe por la forma de describir los colores, su tono y saturación en contraste con el modelo RGB que representa un color en base a sus colores primarios (rojo, verde y azul) [11].

$$\theta = \cos^{-1} \left(\frac{0.5[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^2} \right) \quad (8)$$

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases}$$

$$S = 1 - \frac{3}{R + G + B} (\min(R, G, B)) \quad (9)$$

$$I = \frac{(R + G + B)}{3} \quad (10)$$

4.4 Redes Neuronales

Las redes neuronales artificiales (RNA) son un sistema de computación artificial compuesto por elementos simples que operan en paralelo, las cuales están inspiradas en el cerebro humano [12], [13]. Cada neurona (también llamada nodo) en una red neuronal recibe un número de entradas; A dichos valores se les aplica una función denominada función de activación, dando como resultado el nivel de activación de la neurona, que corresponde a la salida de la neurona [14]. Las conexiones entre neuronas tienen pesos asociados con ellas y cada neurona pasa su salida a la entrada de las siguientes neuronas que están conectadas.

El número de neuronas a la entrada de la RNA es ajustado en función de la cantidad y características de la entrada, el ajuste a los pesos de las neuronas se realiza a través de las capas ocultas y de salida [12]. Por lo general, la red tiene tantas neuronas en su capa de salida como clases tenga el problema. En la etapa de entrenamiento se ajustan los parámetros de la red para aproximar el mapeo de la superficie de clasificación de los datos de entrada en la forma más acertada posible. En la Figura 3 se observa la representación de una red neuronal en cascada de tres capas.

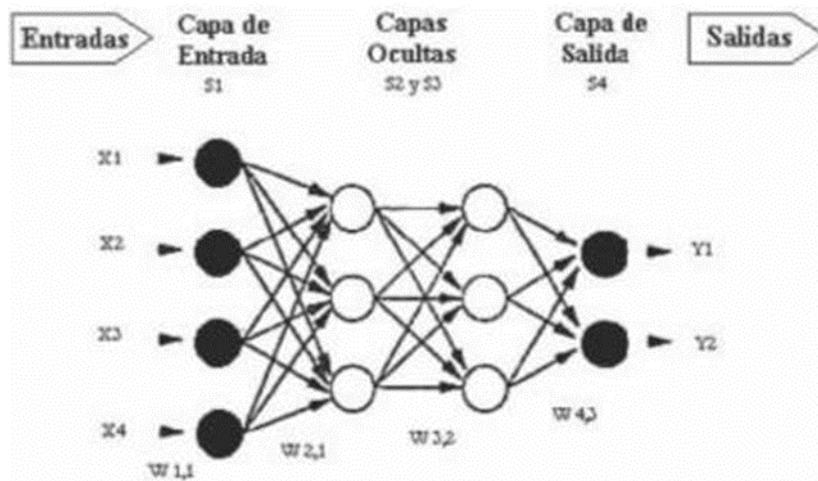


Figura 3 Modelo de red en cascada de tres capas [15]

Los siguientes métodos se pueden utilizar para minimizar el error de generalización [12]:

- Cross-validation: se dividen los datos en k subconjuntos de tamaño aproximadamente igual, la red es entrenada k veces, quedando afuera todos los posibles subconjuntos.
- K fold cross-validation: se dividen los datos en k subconjuntos de tamaño aproximadamente igual, la red es entrenada k veces, quedando afuera cada

vez un subconjunto diferente, el cual se utiliza para computar el diseño que se desea aplicar. Cuando k es igual al tamaño de la muestra, el proceso se denomina leave-one-out.

- Leave-v-out cross-validation: Es una versión más elaborada de cross-validation, pero que acarrea mayor costo computacional.
- Split-sample o hold-out: Se utiliza únicamente un subconjunto (el subconjunto de validación) para estimar el error de generalización. A diferencia del cross-validation, el split-sample se suele utilizar para realizar early stopping en redes neuronales.

5. METODOLOGÍA

5.1 Fases del proyecto

- Fase de estructuración: Buscar y clasificar información obtenida por medio de libros, tesis y demás.
- Fase de investigación: Investigar cada uno de los factores que influyen en la delimitación de los aspectos de diseño.
- Fase de requerimientos: Especificar los requerimientos del proyecto, tanto software como hardware para el mecanismo del robot móvil, el sistema de comunicación inalámbrica y procesamiento de imágenes.
- Fase de diseño: Evaluar cada una de las propuestas para escoger la que mejor se adapta a las especificaciones del proyecto.
- Fase de realización: implementar un sistema de trayectorias para el robot móvil reconociendo y clasificando frutos de cacao por medio de un sistema HMI (Human Machine Interface).
- Fase de valoración: valorar el funcionamiento y resultados obtenidos de los análisis realizados a las mazorcas de cacao garantizando el cumplimiento de los objetivos propuestos.
- Fase de conclusión: validar los resultados obtenidos teniendo en cuenta los objetivos planteados para la finalización del proyecto.

5.2 Metodología del proyecto

- El VATNT tendrá definido un entorno estructurado, el cual le permitirá moverse por las trayectorias generadas por el software en busca de los puntos óptimos para la toma de imágenes en cada árbol.
- La trayectoria es generada una vez sea definida la posición de los árboles en el software, creando dos puntos por árbol para que el vehículo realice un seguimiento con ayuda de sensores ultrasónicos para la ubicación y un magnetómetro para la orientación en un entorno estructurado, capturando a 180° dos imágenes del árbol donde se observen las mazorcas de cacao.
- Al terminar la trayectoria el vehículo compartirá los datos tomados en su recorrido al servidor de forma inalámbrica para que el software realice el procesamiento de las imágenes, utilizando técnica de visión artificial, separando los frutos maduros y no maduros.
- El software indicará al usuario, el total de frutos en el cultivo, la cantidad por árbol y el porcentaje de frutos maduros y no maduros en cada uno, teniendo registros de datos estadísticos y fotográficos para que el usuario lleve un control y seguimiento de ellos.

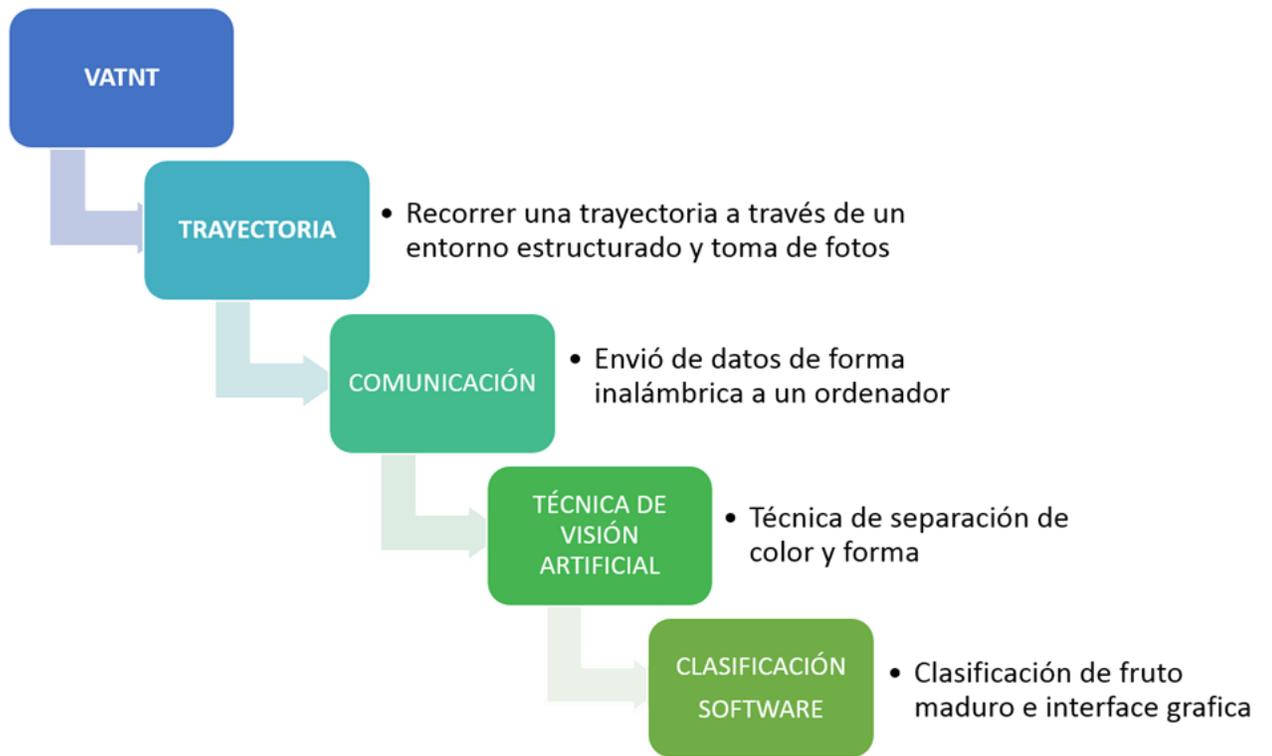


Figura 4 Diagrama de la metodología [Autores]

6. CRITERIOS PARA LA CLASIFICACIÓN DEL CACAO

Para obtener una buena cosecha y post cosecha de cacao se debe tener en cuenta varios elementos importantes, debido a que el cacao como producto agrícola es el resultado de múltiples variaciones e interacciones desde factores climáticos hasta herramientas que utiliza el agricultor para su cultivo, y así lograr resultados positivos, pues a medida del buen manejo que se dé al inicio de la cosecha dependerá la calidad de cacao que se obtenga.

El proceso consiste en los siguientes pasos:

- La cosecha o recolección: inicia cuando el fruto de mazorca está maduro (se identifica por el color de su cascara: naranja), es importante que se recojan a tiempo para evitar riesgo de germinación de la almendra; Al madurar cambian del color, variando según los grupos y mezclas de cacao de cada zona. En los cacaos híbridos la variación de color es mayor que en los cacaos criollos.
- No deben recolectarse frutos verdes o amarillos porque afectan de manera desfavorable el proceso de fermentación.
- Quiebra: consiste en la acción que se realiza al partir el fruto o mazorca y extraer las almendras, para seguido someterlas a fermentarlas.
- Fermentación: es un proceso que consiste en remover el mucilago azucarado que cubre el grano, elevar la temperatura, destruir las células pigmentadas, transformar alcoholes por levadura.

- Secado: se realiza con el fin de eliminar la humedad y el ácido acético producido en el proceso de fermentación.
- Limpieza y selección: este paso nos permite limpiar residuos del cacao como: pajillas, granos picados y placentas.

La recolecta puede hacerse cada quince días o cada mes. Si la plantación es pequeña se puede hacer cortes todas las semanas. En el año, hay dos cosechas grandes, y durante los picos de las cosechas, la corta debe realizarse cada ocho días.

Este proceso termina con la fase de almacenamiento del cacao el cual debe empacarse en sacos de yute, este paso debe realizarse en perfectas condiciones de: clima, higiene, el lugar debe estar libre de malos olores, contaminación por humo, libre de humedad, entre otros; en caso contrario, todo el esfuerzo realizado por obtener un cacao de calidad fracasaría.

7. DISEÑO DEL PROTOTIPO VATNT

Para el diseño del vehículo autónomo terrestre no tripulado se tuvo en cuenta tres aspectos importantes: la parte mecánica, electrónica y el software; todo eso escalado teniendo en cuenta el espacio estructurado en el que se trabajó.

7.1 Parte mecánica

Por la parte mecánica tenemos la plataforma móvil, los micromotores y los servos para ubicar la cámara.

7.1.1 Plataforma

El vehículo autónomo cuenta con una plataforma tanque DIY SN800 de las siguientes características: dimensión 9,7 x 18 x 4,5 [cm], dos (2) orugas, ocho (8) llantas lisas y cuatro (4) llantas dentadas, material flexible (polietileno de alta densidad) como se muestra en la figura 5, peso aproximadamente de 166 [g], lo anterior la convierte en una plataforma compacta, manejable y ligera. [17]

Las llantas lisas cuentan con sistema de absorción de resorte que permiten realizar la función de amortiguador y puede llegar a comprimir hasta 0,7 [cm], lo cual las hace perfectas en superficies rizadas.

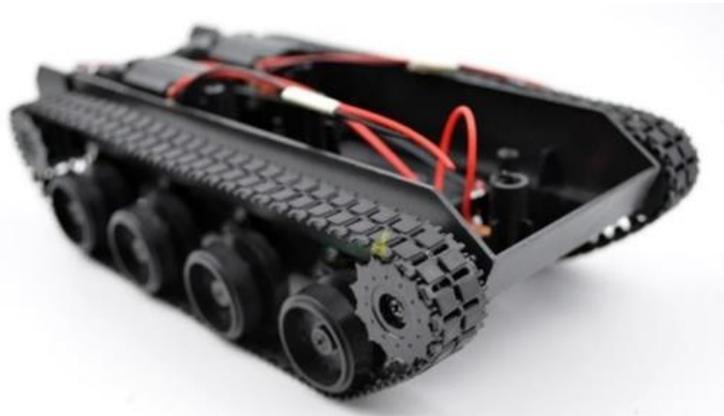


Figura 5 Plataforma tanque con absorción de choque DIY SN800 [17]

7.2 Actuadores

Son dispositivos capaces de convertir energía eléctrica, neumática o hidráulica en una actividad de un proceso (generar movimiento), la cual es generada y enviada por un controlador.

Según la fuerza que ejercen, existen varios tipos de actuadores:

- Hidráulicos
- Neumáticos
- Eléctricos

Cada uno de los sistemas se utilizan en labores específicas, debido a que cada uno presenta diferentes características; para el VATNT los actuadores (micromotores – servomotores) que se emplearon son eléctricos, en razón en su facilidad de control, sencilla instalación y silenciosos.

7.2.1 Micromotor con eje extendido

Los actuadores encargados de darle movimiento a esta plataforma móvil son dos (2) micromotores con eje extendido que cuentan con caja reductora de 100:1 de la marca Pololu. Se observa en la figura 6.



Figura 6 Micromotor con caja de engranes 100:1 con eje extendido [18]

Estos micromotores de corriente continua están disponibles en una amplia gama de relaciones de engranes, desde 5:1 hasta 1000:1, para el VATNT se utilizó motor de alta potencia de 6 a 12 [v] y relación de engranes de 100:1 con escobillas de carbón de alta duración, este tiene una sección transversal de 10 x 12 [mm] con una longitud de 9 [mm] y un diámetro de 3 [mm] para el eje de salida en la caja de engranes; esta versión cuenta con un eje extendido de 4,5 x 1 [mm], se observa en la figura 7. Que funciona como soporte para instalar un encoder. Este eje extendido gira a la misma velocidad que la entrada de la caja de engranes y ofrece una forma de conocer la velocidad y dirección de giro para los motores [18].

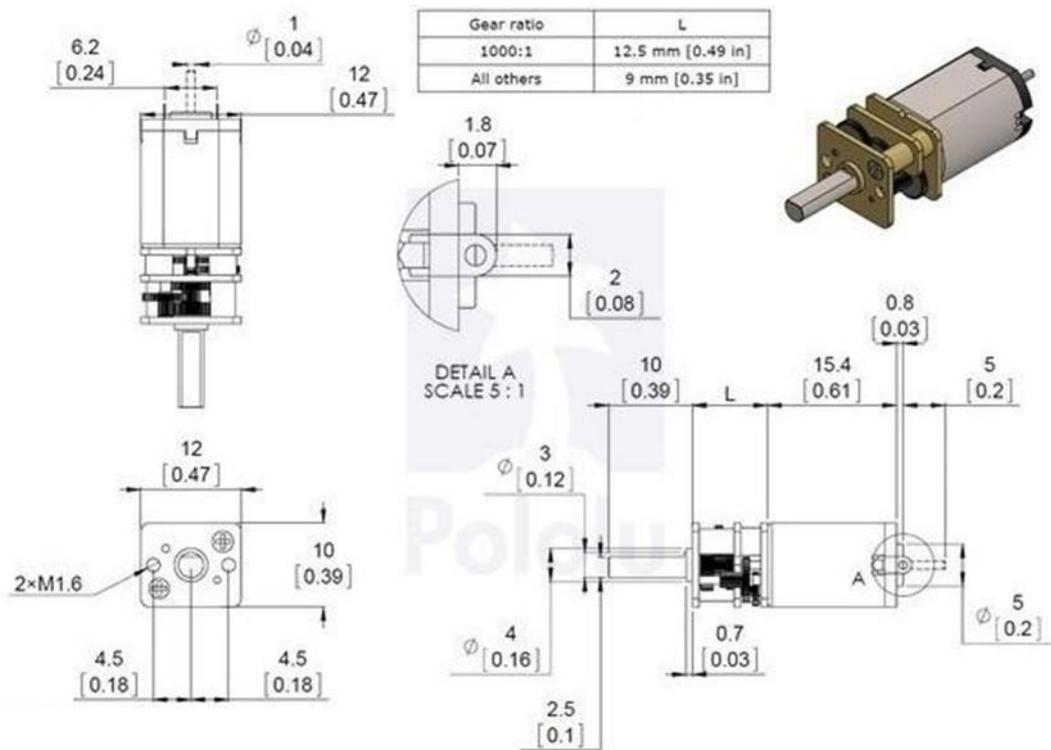


Figura 7 Dimensiones de micromotor con eje extendido [18]

7.2.2 Servomotor SG90

Un servomotor es un dispositivo electromecánico que está compuesto por un motor eléctrico, un juego de engranes y una tarjeta la cual nos permite controlar la posición del eje de salida. Está diseñado para ser controlado en posición, moverse a una posición específica en grados y mantenerse fija en esa posición.

Funcionan por medio de modulación de ancho de pulso (PWM). Estos servos disponen de tres cables, dos para alimentación Vcc y Gnd 4.8 a 6 [V] y un tercero para aplicar el tren de pulsos de control, que hace que el circuito de control diferencial interno ponga el servo en la posición indicada.

Estos servos son de gran calidad y pequeñas dimensiones 22 x 12 x 31 [mm], con un peso de 9 [g] la cual los hace muy adaptables al prototipo, la función que cumplen es darle la dirección a la cámara para la captura de imágenes, además cambiar el ángulo de visión con respecto al plano 'XY', se observa en la figura 8. Estos servomotores tienen en su interior un (1) motor que trabaja de 3 a 7.2 [v] con una caja de engranes de nylon la cual le da un torque de salida de 1.8 [kgf.cm] dándole la capacidad de rotar 180° aproximadamente.



Figura 8 Soporte para cámara con servomotores 9g de dos grados de libertad [19]

7.3 Sensores

Los sensores que se emplearon para el diseño del vehículo autónomo terrestre no tripulado, permitieron conocer su información en un entorno estructurado como lo son: velocidad, posición y orientación, con el fin de garantizar una ubicación óptima en el plano 'XY' en la captura de imágenes.

7.3.1 Encoder

El sensor funciona desde 2.7 a 18 [v] y tiene salidas digitales que proporcionan la información, la cual puede ser conectada directamente a un circuito digital. Este módulo se observa en la figura 9. Es compatible con todos los micromotores de eje extendido de la marca Pololu.



Figura 9 Encoder (sensor hall) [20]

Este encoder encaja dentro de la sección transversal de 12 x 10 [mm] de los motores y va soldado directamente a M1 y M2 del encoder al motor, se observa en la figura 10.



Figura 10 Encoder instalado en motor [20]

Se adquirieron dos (2) sensores de efecto hall de doble canal y dos (2) discos magnéticos de seis (6) polos, se observa en la figura 11. Al agregarlos a los motores

ofrecen una codificación en cuadratura. La placa del codificador detecta la rotación del disco magnético y proporciona una resolución de doce (12) conteos por revolución del eje del motor al contar los bordes de ambos canales. Para calcular los conteos por revolución del eje de salida de la caja de engranajes se multiplica la relación de los engranajes por 12. En este caso es una relación 100:1. [20]

$$100 \times 12 = 120$$

Es decir que se necesitan 120 conteos para generar una (1) revolución en el eje de salida de la caja de engranes.

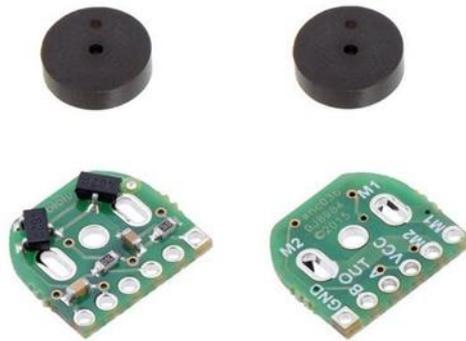


Figura 11 Encoder y disco magnético [20]

El codificador tiene dos (2) señales de salida al controlador, que deben estar conectadas a pines de interrupción para mejorar su lectura. Como lo muestra la tabla 1.

Tablero	Pines de interrupción	Pin LED (no usar)
Teensy 3.0 - 3.6	Todos los pines digitales	13
Teensy lc	2 - 12, 14, 15, 20 – 23	13
Teensy 2.0	5, 6, 7, 8	11
Teensy 1.0	0, 1, 2, 3, 4, 6, 7, 16	
Teensy ++ 2.0	0, 1, 2, 3, 18, 19, 36, 37	6
Teensy ++ 1.0	0, 1, 2, 3, 18, 19, 36, 37	
Arduino Due	Todos los pines digitales	13
Arduino Uno	2, 3	13
Arduino Leonardo	0, 1, 2, 3	13
Arduino Mega	2, 3, 18, 19, 20, 21	13
Sanguino	2, 10, 11	0

Tabla 1 Pines de interrupción de las diferentes tarjetas de control [21]

Los codificadores pueden detectar el movimiento en cualquier dirección, al detectar agujeros o marcas a medida que se mueven más allá de las 2 posiciones. Cuando el disco azul en la figura 12 gira en el sentido de las agujas del reloj, los cambios se detectan primero con el pin 1 y luego con el pin 2. Cuando gira en el sentido contrario a las agujas del reloj, el pin 2 es el primero en detectar cambios. Este esquema se denomina "codificación en cuadratura" porque las formas de onda detectadas por los dos (2) pines están desfasadas 90 grados. [21]

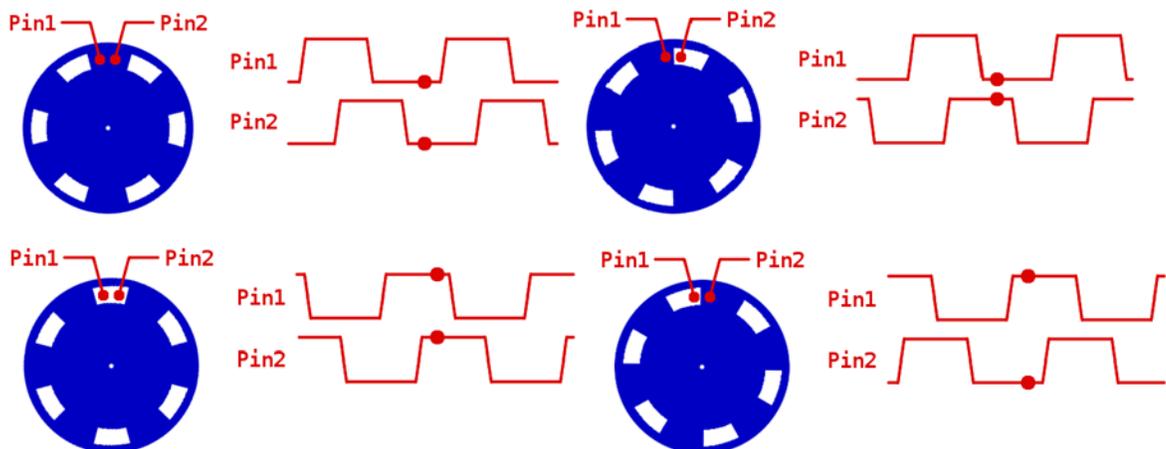


Figura 12 Estados del encoder [21]

7.3.2 Ultrasónico

Como su nombre lo indica, los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. Este proceso inicia con la emisión de una onda ultrasónica y termina con la recepción que rebota desde el objeto, por lo anterior estos sensores se encargan de medir la distancia del objeto respecto al tiempo que tarde en la emisión y recepción de la onda.

Para el diseño del vehículo autónomo terrestre no tripulado se implementaron cuatro (4) sensores ultrasónicos HC-SR04, los cuales permiten conocer la posición de vehículo en las coordenadas 'XY', el cual es de bajo consumo energético de 5 [V] y 15[mA] con un rango de operación de 2 a 450 [cm], destacado por su pequeño tamaño, buena precisión. Este sensor es el más utilizado dentro de los sensores de tipo ultrasónico en los proyectos de robótica y automatización como sistemas de nivel o distancia por la cantidad de información acerca de él. Se observa en la figura 13.



Figura 13 Sensor ultrasónico HC_SR04 [23]

Las siguientes características hacen del ultrasónico HC_SR04 un sensor eficiente y óptimo para el prototipo:

- Detecta objetos transparentes: las ondas ultrasónicas pueden reflejarse en una superficie de vidrio o líquido y retornar al cabezal, incluso los objetos transparentes pueden ser detectados.
- Resistente a niebla y suciedad: La detección no se ve afectada por la acumulación de polvo o suciedad.
- Detecta objetos de forma compleja: La detección de presencia es estable, incluso para objetos como bandejas de malla o resortes.

El sensor HC_SR04 posee dos (2) transductores piezoeléctricos, un emisor y un receptor. El emisor piezoeléctrico envía ocho (8) pulsos de ultrasonido a 40[Khz] luego de recibir la orden en el pin TRIG, las ondas de sonido viajan por el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a alto 5[v] por un tiempo igual al que tardó la onda desde que fue emitida hasta que fue detectada; el tiempo del pulso ECHO es medido por el micro controlador y así se puede calcular la distancia al objeto como se observa en la figura 14.

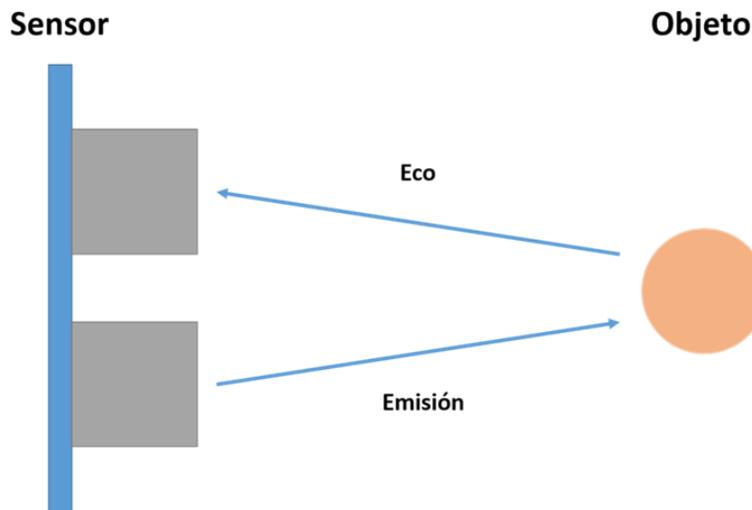


Figura 14 Funcionamiento sensor ultrasónico [Autores]

La distancia se puede calcular utilizando la siguiente formula:

$$Distancia[m] = \frac{t_{eco} \times V_{sonido}}{2} \quad (11)$$

Donde la $V_{sonido} = 343,2$ [m/s] a 20°C con un 50% de humedad y al nivel del mar y t_{eco} es el tiempo que tarda en llegar la señal del emisor al receptor en μs .

7.3.3 Magnetómetro

El magnetómetro es un sensor que mide la fuerza y dirección del campo magnético local, esta medición es el resultado de combinar el magnetismo emitido por los objetos cercanos y el norte magnético de la tierra.

El campo magnético de la tierra es un campo auto sostenido que se asemeja a un dipolo magnético, con un extremo cerca del polo norte geográfico de la tierra y el otro, cerca del polo sur geográfico de la tierra. La fuerza del campo magnético de la tierra tiene variaciones entre 0,3 [Gauss] en Sudamérica y 0,6 [Gauss] en Norteamérica, aunque es relativamente estable en el tiempo, las corrientes eléctricas en la ionosfera pueden causar alteraciones que desvíen los campos magnéticos de la superficie hasta un (1) grado. Normalmente, las variaciones diarias en la intensidad del campo son del orden de 0.25 [mGauss], lo que equivaldría a una variación de aproximadamente 0.03 grados.

El modelado del sensor se usa para corregir matemáticamente los errores de desalineación y bias (sesgo/orientación). El magnetómetro utilizado utiliza un modelo de sensor lineal.

$$H = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & M_{XY} & M_{XZ} \\ M_{YX} & 1 & M_{YZ} \\ M_{ZX} & M_{ZY} & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & \frac{1}{S_z} \end{bmatrix} \left(B_D + \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} - \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \right) \quad (12)$$

Esto escrito en ecuaciones algebraicas da lo siguiente:

$$H_x = \frac{B_D + B_x - V_x}{S_x} + \frac{M_{XY}(B_D + B_y - V_y)}{S_y} + \frac{M_{XZ}(B_D + B_z - V_z)}{S_z} \quad (13)$$

$$H_y = \frac{M_{YX}(B_D + B_x - V_x)}{S_x} + \frac{B_D + B_y - V_y}{S_y} + \frac{M_{YZ}(B_D + B_z - V_z)}{S_z} \quad (14)$$

$$H_z = \frac{M_{ZX}(B_D + B_x - V_x)}{S_x} + \frac{M_{ZY}(B_D + B_y - V_y)}{S_y} + \frac{B_D + B_z - V_z}{S_z} \quad (15)$$

El magnetómetro presenta dos (2) tipos de distorsiones una de hierro duro y otra de hierro suave; las distorsiones de hierro duro son creadas por objetos que producen un campo magnético como un altavoz o un hierro magnetizado. Si la pieza de material magnético está físicamente unida al mismo marco de referencia que el sensor, este tipo de distorsión de hierro duro causará una polarización permanente en la salida del sensor.

Las distorsiones de hierro suave se consideran desviaciones o alteraciones en el campo magnético existente. Estas distorsiones estiran o alteran el campo magnético dependiendo de en qué dirección se actúa el campo en relación al sensor. Este tipo de distorsiones son causados por metales como níquel y el hierro.

En el caso de que no existan distorsiones las mediciones deben formar (un círculo centrado en 'X=0, Y=0' y el radio del círculo es igual a la magnitud del campo magnético. se observa en la figura 15.

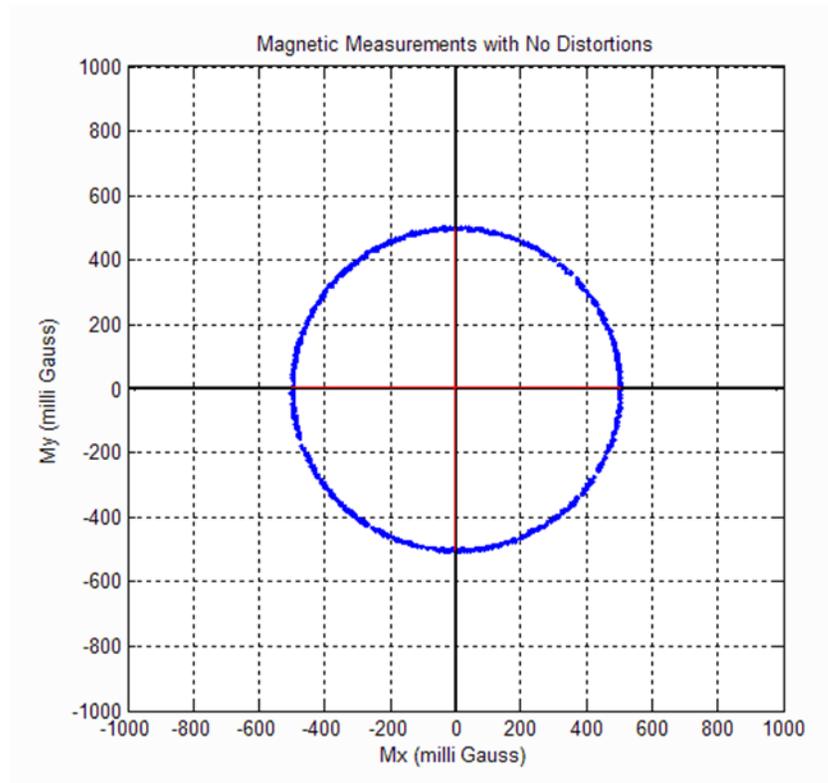


Figura 15 Magnetómetro sin distorsión [23]

En la figura16 se observa la distorsión con hierro duro en la que el centro del círculo se encuentra en 'X=200, Y=100'. De esto se puede concluir que hay una polarización de hierro duro de 200 [mGauss] en el eje 'X' y una de 100 [mGauss] en el eje 'Y' es decir que la distorsión de hierro duro solo ájela el centro del círculo del origen.

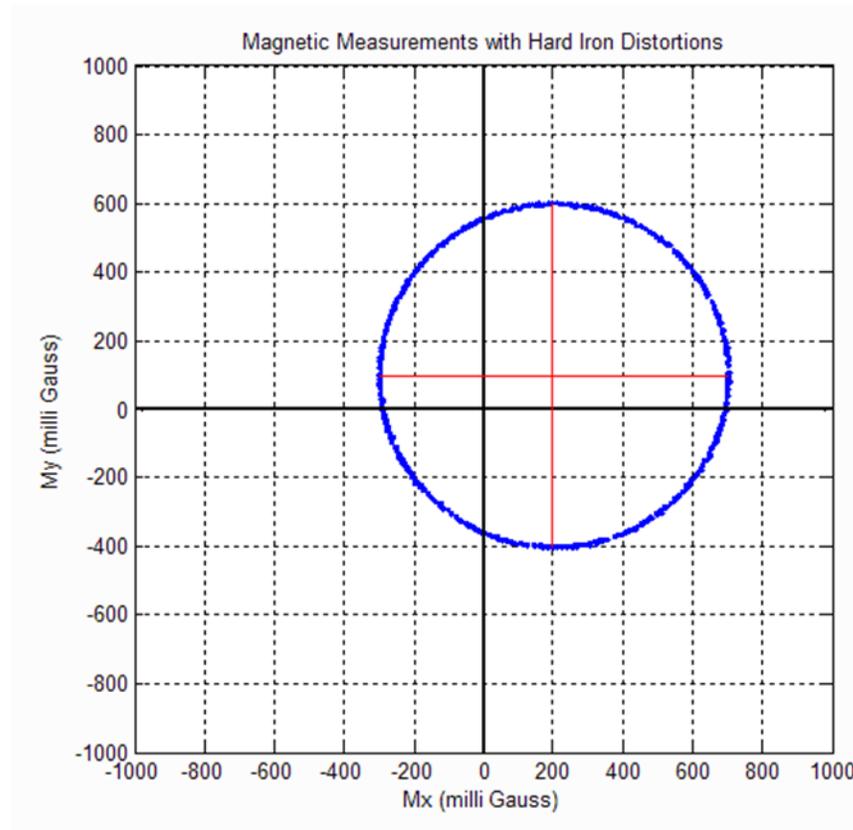


Figura 16 Magnetómetro con distorsión de hierro duro [23]

Las distorsiones de hierro suave, distorsionan y deforman los campos magnéticos existentes. Cuando se traza la salida magnética, las distorsiones de hierro suave son fáciles de reconocer, ya que distorsionarán la salida circular. Los efectos de hierro suave deforman el círculo en una forma elíptica.

En la figura 17 se observa la distorsión de hierro fuerte y suave donde el centro de la elipse se encuentra en 'X=200, Y=100' ya que las distorsiones del hierro son las mismas de antes. Cada elipse tiene un eje mayor y menor que corresponde a las dimensiones largas y cortas con respecto al campo magnético generado por los metales duros y suaves.

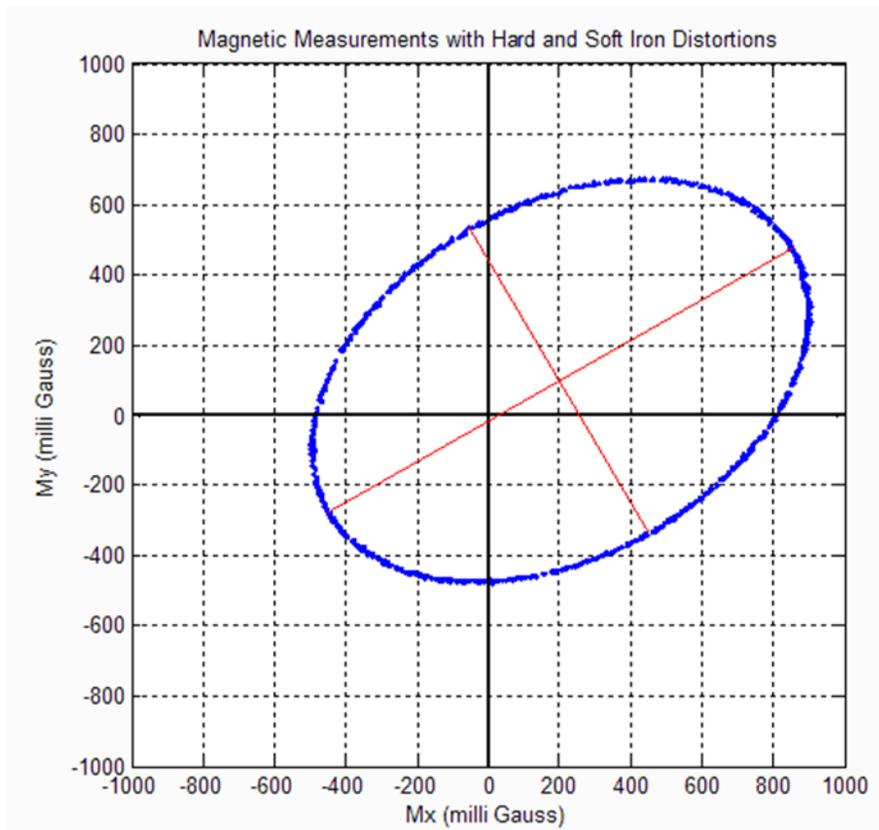


Figura 17 Magnetómetro con distorsión de metal duro y suave [23]

Es posible eliminar los efectos de las distorsiones de hierro duro y suave en las salidas del magnetómetro. Los magnetómetros utilizan el siguiente modelo de calibración para corregir las distorsiones de hierro duro y suave.

$$M = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_4 & C_6 \\ C_7 & C_8 & C_9 \end{bmatrix} \begin{bmatrix} H_X - C_{10} \\ H_Y - C_{11} \\ H_Z - C_{12} \end{bmatrix} \quad (16)$$

El modelo anterior consta de 12 parámetros de compensación de hierro duro y suave. Los primeros nueve (9) parámetros correctos para el hierro suave, mientras que los últimos tres (3) parámetros $\{C_{10}, C_{11}, C_{12}\}$ compensan el hierro duro [23].

Los sensores inerciales como el acelerómetro y el giroscopio dan información precisa a corto plazo, debido a esto, el uso de ellos en un proyecto se convierte en poco fiable, a diferencia el magnetómetro da información a largo plazo, razón por la cual se convierte en un sensor más fiable al no acumular margen de error al calcular la orientación del VATNT, Por lo anterior se implementó en el vehículo autónomo terrestre no tripulado un sensor HMC5883L. Se observa en la figura 18.

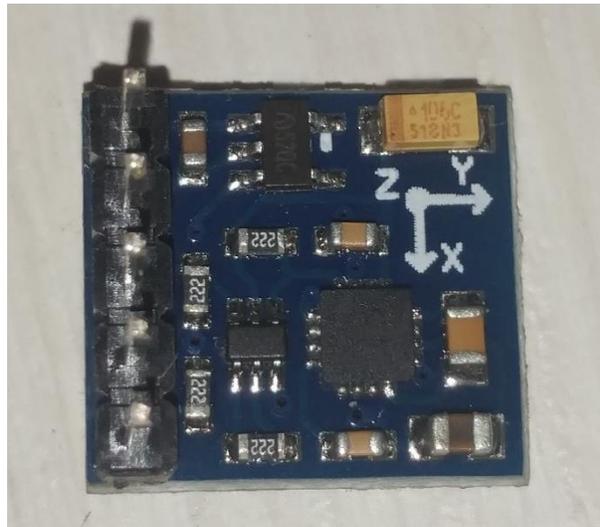


Figura 18 Sensor magnetómetro HMC5883L [Autores]

Este módulo se caracteriza por su funcionamiento basado en un campo magnético de tres ejes, lleva a cabo la comunicación a través de interfaz I2C, con un rango de operación de $\pm 1,3 - 8$ [Gauss] con un voltaje de $3 - 5$ [V].

7.3.4 Cámara FPV (First Person View)

La cámara FPV es vital para el vehículo autónomo terrestre no tripulado, debido a que esta funciona como el ojo del robot, la imagen es transmitida en tiempo real por medio de un video transmisor; por otro lado existen dos (2) tipos de sensores de

imagen usados para FPV, los cuales son CCD y CMOS cada una con sus características y ventajas [24].

Las cámaras CCD tienen menos ruido en condiciones de poca luz, menor latencia y menos efecto gelatina en las grabaciones. Las cámaras CMOS tienen mayor resolución, mejores colores y consumen menos corriente.

Antes de seleccionar la cámara FPV también debe tenerse en cuenta el campo de visión o FOV (Field Of View). En la tabla 2 se muestra los diferentes tipos de FOV dependiendo la longitud focal del lente.

Longitud Focal	FOV aproximado
2.1mm	160° – 170°
2.3mm	145°
2.5mm	130° – 140°
2.8mm	120° – 130°
3.0mm	110° – 125°

Tabla 2 Comparación longitud focal con FOV [24]

La resolución de las cámaras FPV al ser análogas se miden en TVL (TV Lines o en español líneas de televisión) esta resolución es basada en la cantidad de líneas horizontales que tiene la imagen, normalmente existen cámaras análogas con 380, 480, 600, 700, 800 o 1200 TVL, pero a mayor resolución aumenta su latencia, por lo anterior la más utilizada en proyectos de quadrotor es de 600 TVL.

El VATNT requería una cámara de bajo consumo de corriente, buena resolución de imagen y color, economía, enfoque manual, tamaño, peso, baja latencia y conexión

práctica, razón por la cual se optó por una cámara CMOS de 700 TVL. Se observa en la figura 19.



Figura 19 Cámara FPV CMOS 700 TVL [25]

La cámara empleada en el VATNT para la captura de imágenes de los frutos de cacao, cuenta con un (1) FOV de 120° , tres (3) pines: dos (2) empleados para la alimentación y uno (1) de señal.

7.4 Dispositivos de comunicación

Para la comunicación del vehículo autónomo terrestre no tripulado se implementó dos sistemas de comunicación inalámbrica independientes para enviar una señal analógica y una digital.

La comunicación analógica es enviada a través de un video trasmisor FPV, mientras la digital por medio de módulos ZigBee, esta es recibida por un terminal inalámbrico conectado al ordenador.

7.4.1 Xbee

Los Xbee's son pequeños módulos capaces de comunicarse de forma inalámbrica unos a otros, su función es remplazar un par de cables en una comunicación serial.

Los módulos Xbee permiten enlaces seriales de señales TTL en distancias de 30 [mts] en interiores, 100 [mts] en exteriores con línea de vista y hasta 1.5 [Km] con los módulos Pro.

Los módulos Xbee utilizan el protocolo IEEE 802.15.4 mejor conocido como ZigBee. Este protocolo se creó pensando en implementar redes de sensores. El objetivo es crear redes tipo mesh que tengan las propiedades de auto-recuperación y bajo consumo de energía.

Para el diseño del vehículo autónomo terrestre no tripulado se implementaron dos (2) Xbee PRO los cuales son perfectos para la comunicación con el servidor, gracias a su buen rango de trabajo con respecto a la distancia de cobertura, brinda información importante del vehículo como lo es la ubicación, orientación y recibe las coordenadas a donde tiene que dirigirse.

El modelo Serie 1 Xbee Pro funciona a 3,3[v], 215[mA] y 2,4[GHz] es fabricado por (Maxstream) y es de alta potencia de salida con una velocidad de transferencia de 250 [kbps] (Max). Los modelos Pro tienen los mismos comandos que la serie básica pero ofrecen una salida con más potencia, en éste caso de 60[mW]. Para la comunicación en parejas (punto a punto), es necesario utilizar dos modelos de la misma serie para garantizar su funcionamiento, se observa en la figura 20.



Figura 20 Modulo Xbee PRO [26]

7.4.2 Kit emisor receptor FPV (First Person View)

La comunicación de captura y envío de imágenes del vehículo autónomo terrestre no tripulado al servidor se hace mediante un sistema análogo FPV.

Cuenta con un transmisor de señal TS832 de 600 [mW], con una impedancia de salida de 50 [Ω], un voltaje de operación entre 7 y 16 [V], y una corriente de 220 [mA]. Opera a una temperatura entre -10 y 85 [°C] una banda de video entre 0 y 8 [MHz]. Su conexión es sencilla cuenta con una entrada de alimentación y de video y salida de alimentación para la cámara, se observa en la figura 21.

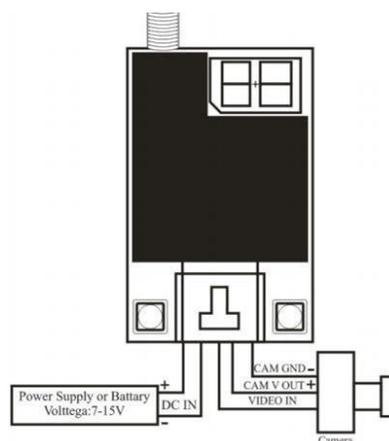


Figura 21 Conexión transmisor TS832 [27]

El receptor RC832 trabaja a una corriente de 200[mA], voltaje de entrada de 12[v] DC, al igual que el transmisor cuenta con dos (2) botones de conmutación y una pantalla de 7 segmentos para la frecuencia y el canal, también poseen una memoria de apagado la cual permite que al volverse a encender quede con el mismo registro con el cual fue apagado [27]. Su conexión es básica, se necesita solamente una fuente de 12 [v] como entrada y cuenta con salida de audio y video.

7.5 Tarjetas

Se implementaron tarjetas para el control de movimiento, adquisición de datos y control de potencia del vehículo autónomo terrestre no tripulado, estas fueron necesarias para su completo funcionamiento.

7.5.1 Arduino mega ADK

El arduino ADK R3 es una tarjeta microcontroladora que se fundamenta en el Atmega2560.

Dentro de sus componentes están 54 pines digitales de entrada y salida. Los pines de entrada hacen referencia a un estado de alta impedancia, es decir, hacen demandas extremadamente pequeñas en el circuito; por el contrario, los pines de salida se encuentran en un estado de baja impedancia, lo que significa que pueden brindar una cantidad importante de corriente a otros circuitos. Además, 15 de estos pines pueden ser utilizados como salidas PWM, lo que ayuda a que se generen salidas analógicas puras desde pines digitales. A su vez cuenta con 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP (Programación serie en circuito), y un botón de reinicio.

Teniendo en cuenta los datos de la tabla 1, el arduino mega ADK cuenta con seis (6) pines de interrupción los cuales son suficientes para implementar los sensores del vehículo autónomo terrestre no tripulado, estos permiten ser reemplazados por otra salida digital gracias a su gran cantidad de pines disponibles. La versatilidad del arduino mega brinda cuatro (4) puertos seriales de los cuales se implementaron dos (2) para la comunicación con el servidor debido a que la mayoría de tarjetas de la familia arduino no cuenta con más de un (1) puerto serial. Figura 22.



Figura 22 Arduino mega ADK [Autores]

7.5.2 Puente H L298N

La placa de potencia implementada para el movimiento y cambio de sentido de los motores implementados es el módulo L298N, el cual posee dos (2) canales de puente en H, estos se pueden utilizar para controlar los dos (2) actuadores del vehículo autónomo terrestre no tripulado. Cuenta con un diseño y conexión sencilla, se observa en la figura 23.

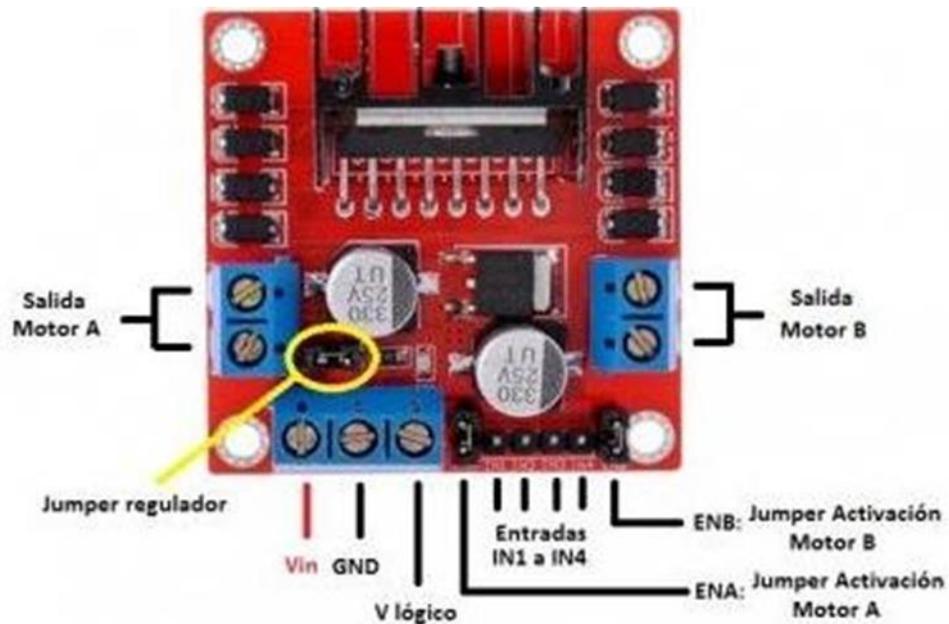


Figura 23 Conexión puente H [28]

El módulo L298 cuenta con un regulador de voltaje para alimentar el dispositivo siempre y cuando no supere los 12 [v] de entrada, además cuenta con un filtro de ruido y un diodo de protección ante corriente inversa lo cual hace que su funcionamiento sea más estable y fiable. Permite controlar el sentido de giro y velocidad mediante señales TTL [28], posee seis (6) entradas de control, de las cuales dos (2) son para habilitar el funcionamiento de cada canal de salida y las otras cuatro (4) son para el control de velocidad que además admiten entradas de señal PWM que envía la placa controladora.

Este módulo permite controlar el funcionamiento de los motores a una corriente por cada canal hasta de 2 [A]. Cuenta con un disipador de calor acorde con las características de este, puede llegar a trabajar con una entrada de hasta 46v perfecto para el diseño del VATNT debido a que su alimentación es de 12 [v] y los actuadores con carga exigen una corriente máxima de 1.6 [A]. El funcionamiento del módulo se observa en la tabla 3.

ENA	EN 1	EN 2	Descripción
0	N / A	N / A	Motor A está apagado
1	0	0	Motor A se detiene (frenado)
1	0	1	El motor A está encendido y da vuelta al revés
1	1	0	Motor A está en y hacia delante de giro
1	1	1	Motor A se detiene (frenado)

Tabla 3 Funcionamiento de cada canal [28]

7.6 Suministro de energía

La alimentación del vehículo autónomo terrestre no tripulado está dividida en dos secciones: inalámbrica y alámbrica. La parte inalámbrica corresponde al vehículo móvil el cual se desplaza con libertad en el espacio de trabajo con una alimentación de 11.1 [v] y 1100 [mAh] la cual es proporcionada por una batería de lipo de tres celdas, se observa en la figura 24. La batería de lipo se encuentra ubicada en la parte inferior de la plataforma móvil la cual puede ser extraída para ser cargada.

Al incluir la batería de lipo al vehículo autónomo terrestre no tripulado le brinda una autonomía de cuatro (4) horas de funcionamiento continuo gracias a las modificaciones de protección que se realizaron al vehículo los cuales disminuyen el consumo de energía apagando los sistemas de comunicación análoga mientras el vehículo se encuentra en movimiento.



Figura 24 Batería de lipo 11.1 [v] [Autores]

El dispositivo encargado de cargar la batería de lipo es un cargador LinkMan LK1008D que tiene un voltaje de entrada de 10 a 15 [v] y de 8,4 a 12,6 [v] de salida con una corriente de 800 [mA] y es compatible con baterías de 2 y 3 celdas, se observa de la figura 25.



Figura 25 Cargador LinkMan [Autores]

Para el funcionamiento del cargador de la batería de lipo se requiere una fuente de alimentación externa de 10 a 15 [v] y se utilizó un adaptador de 12 [v] a 2 [A]. Se observa en la figura 26.



Figura 26 Adaptador externo 12 [v] [Autores]

En la figura 27 se observa el diagrama de conexiones del vehículo autónomo terrestre no tripulado.



Figura 27 Diagrama de conexiones [Autores]

8. TRAYECTORIA

Con el fin de obtener una lectura y respuesta óptima en la trayectoria, se inició con la calibración de los sensores y actuadores, teniendo en cuenta todas las partes del VATNT, y así evitar que, al entrar en contacto con él, se perturbe el sistema.

8.1 Magnetómetro

Se inició con la calibración del magnetómetro, sensor que permite leer la orientación del VATNT con respecto al norte magnético, para lograr esto se escalaron los datos cada 10°, comparando los datos tomados con los grados que se giró el vehículo. Se observa en la figura 28.

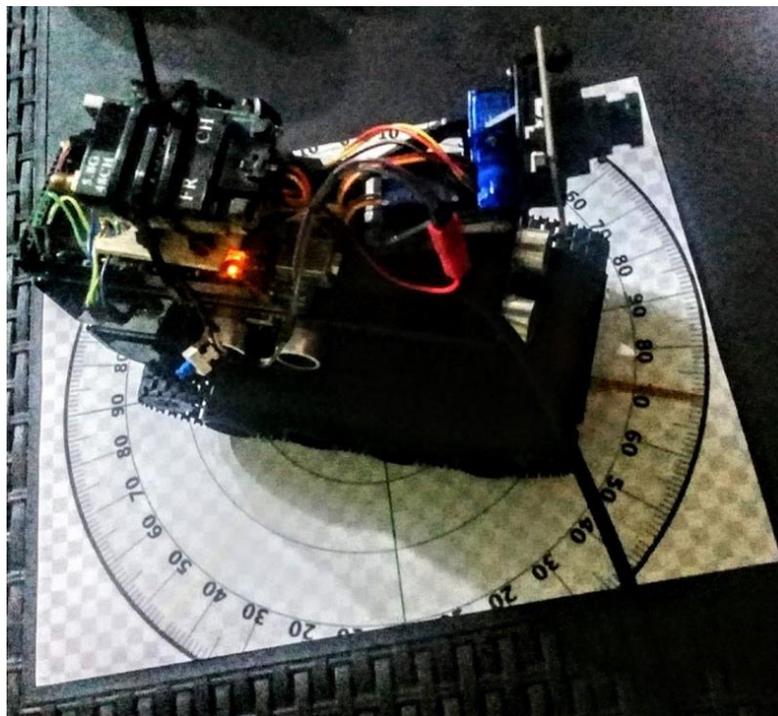


Figura 28 Calibración magnetómetro [Autores]

El resultado obtenido, tabla 4, se tabuló en una hoja Excel y se graficó, para encontrar la ecuación de la recta, y así implementarla en el código, que permite mejorar la lectura de los datos que varían al verse afectados por componentes metálicos presentes en el vehículo autónomo y de esta manera optimizar la orientación del vehículo en el espacio estructurado, figura 29.

Giro	Sensor
0	124
10	134
20	143
30	152
40	160
50	168
60	177
70	185
80	193

90	200
100	208
110	216
120	226
130	235
140	245
150	255
160	265
170	276
180	288

Tabla 4 Datos del magnetómetro [Autores]

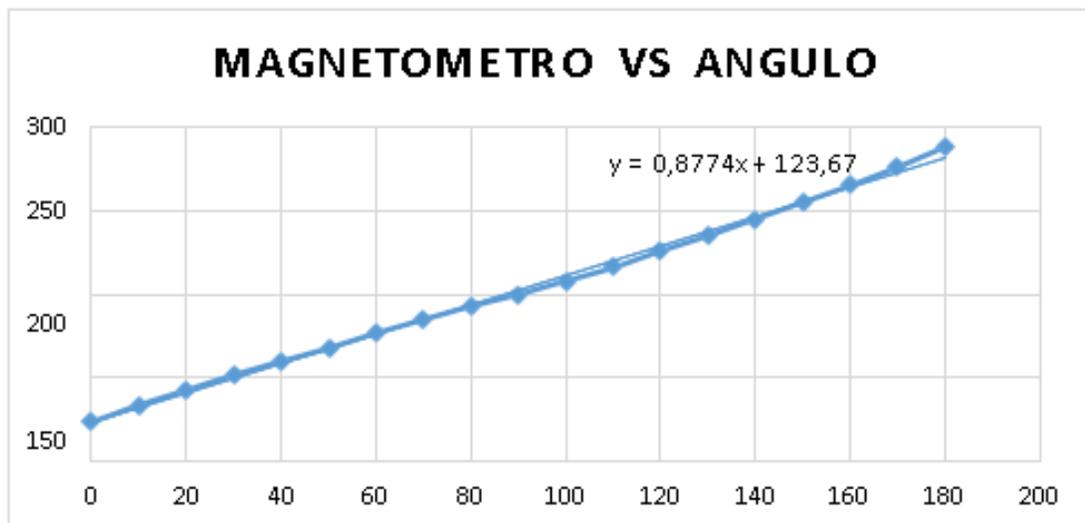


Figura 29 Datos magnetómetro [Autores]

El magnetómetro mide su lectura respecto al norte magnético y para que su lectura sea respecto al norte geográfico (sea funcional el VATNT en el lugar que se encuentre) se debe sumar la declinación magnética presente en el lugar, siendo esta de -8° 28' en la ciudad de Bucaramanga [29] pasándolo a número entero con un valor de -8.22.

Para obtener la lectura exacta de la orientación se implementó en el código la fórmula: $y=0.8774*(ang-8.22) +123.67$; siendo 'y' la salida y 'ang' el valor de la lectura del magnetómetro.

Para garantizar 0° respecto al eje 'y' en el VATNT, se implementó el siguiente método de calibración aplicando la ecuación 17.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} * \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \right) \quad (17)$$

Donde X_C , Y_C y Z_C , son los valores calibrados del sensor, los valores de 'M' son la matriz de transformada; X, Y y Z son los valores tomados del sensor y B_x , B_y y B_z son la matriz de bias [30], los cuales se obtuvieron con el uso del software MagMaster. Se observa en la figura 30.

Para calcular las matrices se debe tener presente los datos de cada uno de los ejes 'XYZ' y los ejes negativos de 'XYZ' del magnetómetro a 0° y 180° a cada uno, para ingresarlos al programa y este realice el cálculo de los respectivos valores de la matriz de transformada M y la matriz de Bias y así permita ser implementada en el código de lectura de este sensor como se ve en la ecuación 17.

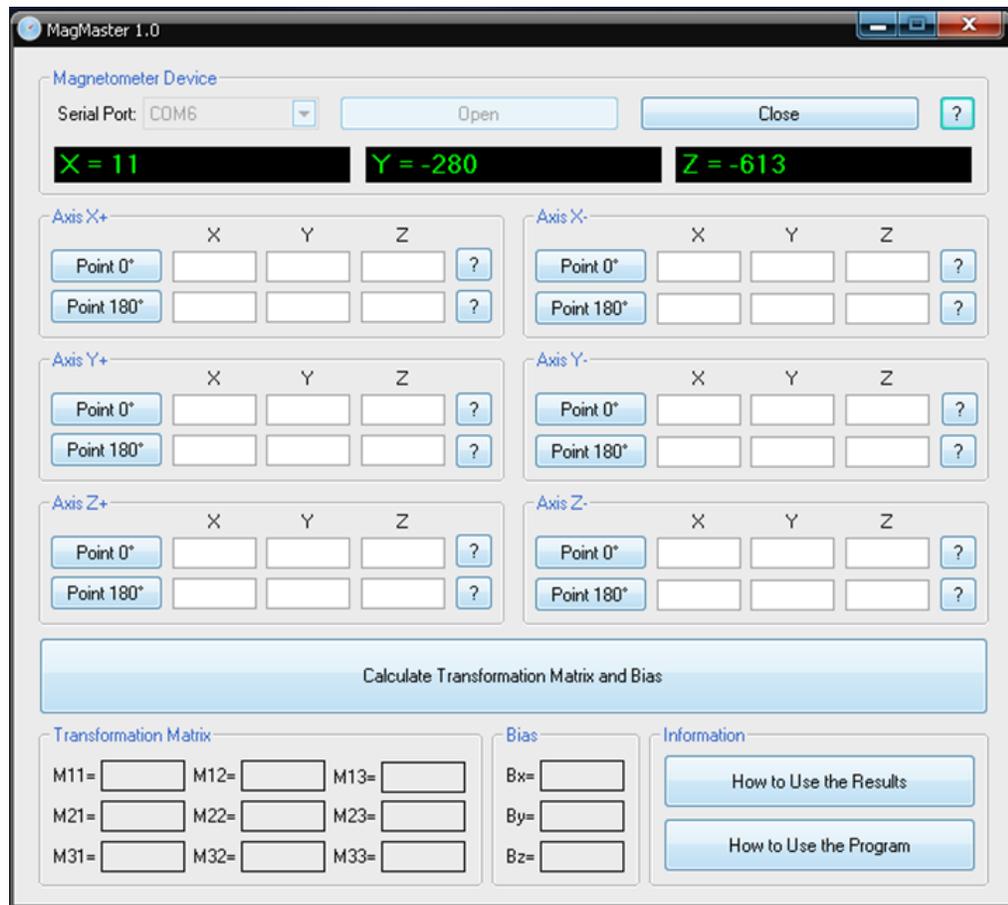


Figura 30 Software MagMaster [Autores]

8.2 Ultrasónicos

El siguiente paso fue la calibración de los sensores de distancia tomando datos cada 5 [cm]. se observa en la figura 31. Estos sensores son ultrasonidos y están ubicados de tal forma que puedan leer la coordenada 'XY' dentro de su espacio de trabajo, con el fin de tener una mejor lectura de la distancia tanto en el eje 'X' como en el eje 'Y'.

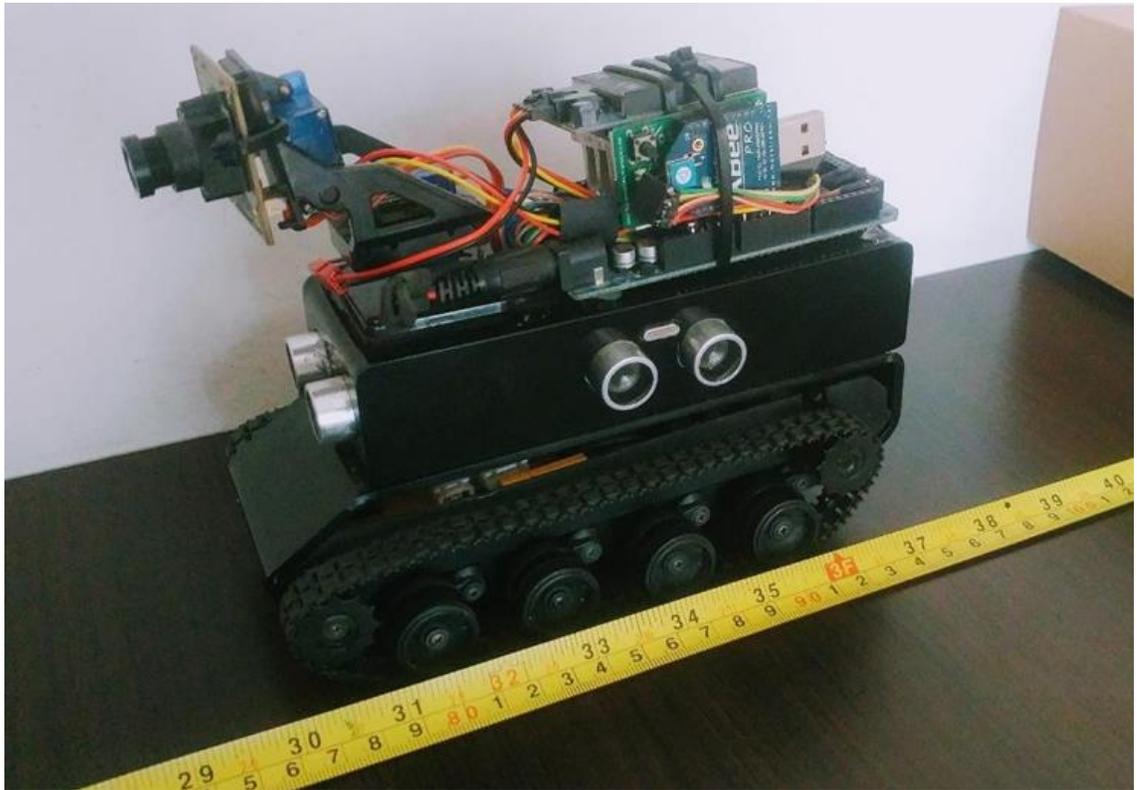


Figura 31 Calibración sensor ultrasónico [Autores]

Al momento de censar su ubicación sobre el plano 'XY' para cada eje se utilizaron dos (2) sensores puestos a 180° cada uno respecto al otro, el cual inicia realizando lectura del eje más cercano al límite del entorno estructurado, hasta llegar a la mitad de este, posterior a esto leerá el siguiente sensor que este ubicado más cerca al siguiente limite, restando la distancia de este al punto medio del VATNT, lo anterior permitirá leer sin errores la ubicación exacta.

Con los datos obtenidos, tabla 5, se realiza una gráfica del sensor comparando la distancia real que se desplaza el vehículo para obtener la pendiente y la ecuación de esta en Excel, figura 32.

Distancia	Sensor
5	5
10	11
15	16
20	22
25	27
30	32
35	37
40	44
45	50
50	53

55	58
60	62
65	67
70	72
75	78
80	83
85	88
90	94
95	99
100	104
105	109

Tabla 5 Datos de calibración sensor ultrasónico [Autores]

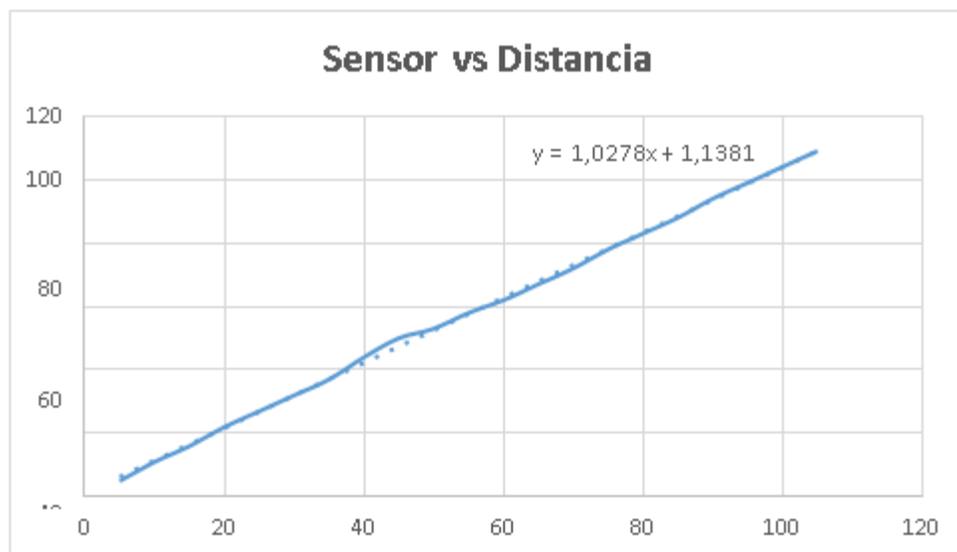


Figura 32 Datos de calibración sensor ultrasónico [Autores]

Obtenida la ecuación se procede a implementarla en el código para la mejora en la adquisición de datos de la siguiente forma: 'y=1.0278*x+1.1381' siendo 'y' la salida y 'x = (sensor)+9' en el caso de ser el sensor más cercano a cero (0) y 'x = (sensor)-9' en el caso donde el sensor de lectura es el más lejano de cero (0) en el eje.

8.3 Modelo Matemático

El modelo matemático de un motor DC se obtiene mediante dos (2) ecuaciones, una mecánica: modela principalmente el movimiento del rotor, y otra eléctrica: modela el circuito eléctrico interno del motor.

Al aplicar una tensión V_{in} al motor, circula una corriente i_m la cual inducirá una fuerza contra electromotriz.

8.3.1 Modelo eléctrico

El modelo eléctrico del motor se obtiene teniendo en cuenta la tensión aplicada, la corriente, la resistencia eléctrica, inductancia del rotor y la fuerza contra-electromotriz. Se observa en la figura 33.

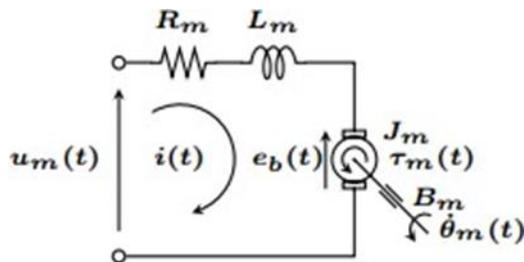


Figura 33 Modelado de un motor DC [31]

$$V_{in} = R_m * i_m + L_m \left(\frac{di}{dt} \right) + e_b \quad (18)$$

Donde:

V_{in} : Tensión aplicada al motor.

R_m : Resistencia del motor.

i_m : Corriente eléctrica del motor.

L_m : Inductancia del motor.

e_b : Fuerza contra electromotriz.

8.3.2 Modelo mecánico

El modelo mecánico del motor DC se obtiene teniendo en cuenta la aceleración del rotor, la inercia, la velocidad angular y el par de fricción, de la siguiente forma:

$$T_m = J_m \ddot{\theta}_m + J_m \dot{\theta}_m + \tau_c \quad (19)$$

Donde:

T_m : Par del motor.

J_m : Inercia del rotor.

$\ddot{\theta}_m$: Aceleración angular del motor.

$\dot{\theta}_m$: Velocidad del motor.

τ_c : Par de fricción.

8.3.3 Modelo cinemático

La cinemática es el área dentro de la mecánica relacionada con el movimiento de los cuerpos. La cinemática de un robot móvil describe la evolución de la posición/orientación del mismo, con respecto a un sistema de referencia fijo, en función de las variables que actúan (posición/velocidad de las ruedas).

Con la cinemática directa, se obtiene la posición y orientación del robot, teniendo los parámetros de control.

$$x(k + 1) = x(k) - \left(\frac{T}{2} * r * \sin \beta(k) * (\omega_d + \omega_i) \right) \quad (20)$$

$$y(k + 1) = y(k) + \left(\frac{T}{2} * r * \cos \beta(k) * (\omega_d + \omega_i) \right) \quad (21)$$

$$\beta(k + 1) = \beta(k) - \left(\frac{T}{b} * T * (\omega_d - \omega_i) \right) \quad (22)$$

Con la cinemática inversa, se obtienen los parámetros de control, teniendo en cuenta la orientación y posición del robot.

$$\omega_i = \frac{x'}{r * \sin \beta} - \frac{b * \beta'}{2 * r} \quad (23)$$

$$\omega_d = \frac{1}{r} * \left(-x' * \sin \beta + y' * \cos \beta + \frac{b * \beta'}{2} \right) = \frac{x'}{r * \sin \beta} + \frac{b * \beta'}{2 * r} \quad (24)$$

$$\beta = \arctan \left(\frac{x'}{y'} \right) \quad (25)$$

8.3.4 Modelo dinámico

Para el control de velocidad de cada motor, se realizó un análisis con caja negra para obtener una aproximación real del modelo matemático de cada motor, utilizando los datos almacenados para la identificación, graficados como se observa en las figuras 34 y 35, y unos datos de validación para verificar la planta obtenida.

Se observa en las figuras 36 y 37. Utilizando la entrada (porcentaje de PWM), salida (velocidad angular en RPM), y tiempo (segundos).

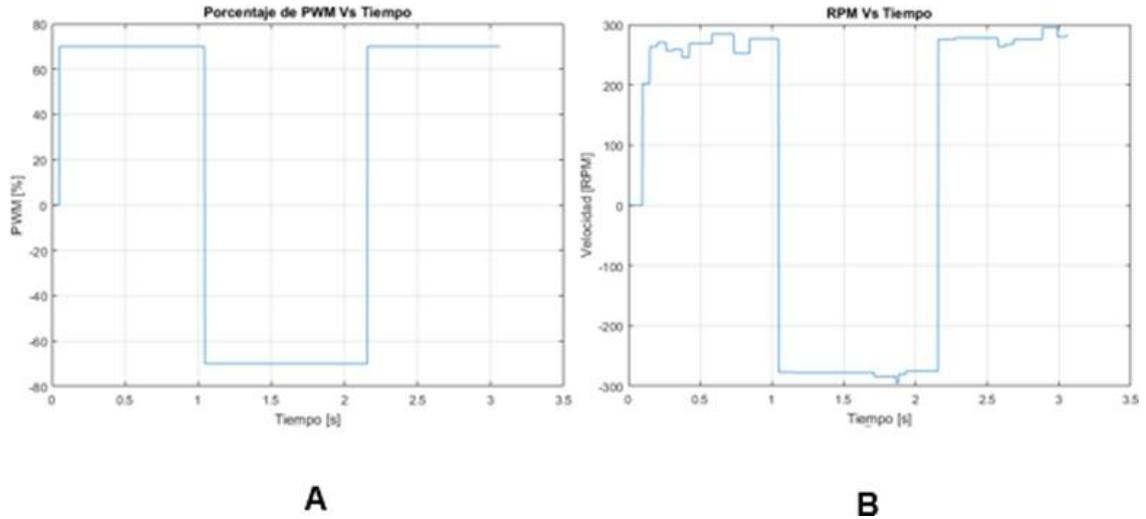


Figura 34 Identificación A) Entrada en %PWM para motor izquierdo. B) Salida en RPM motor izquierdo [Autores]

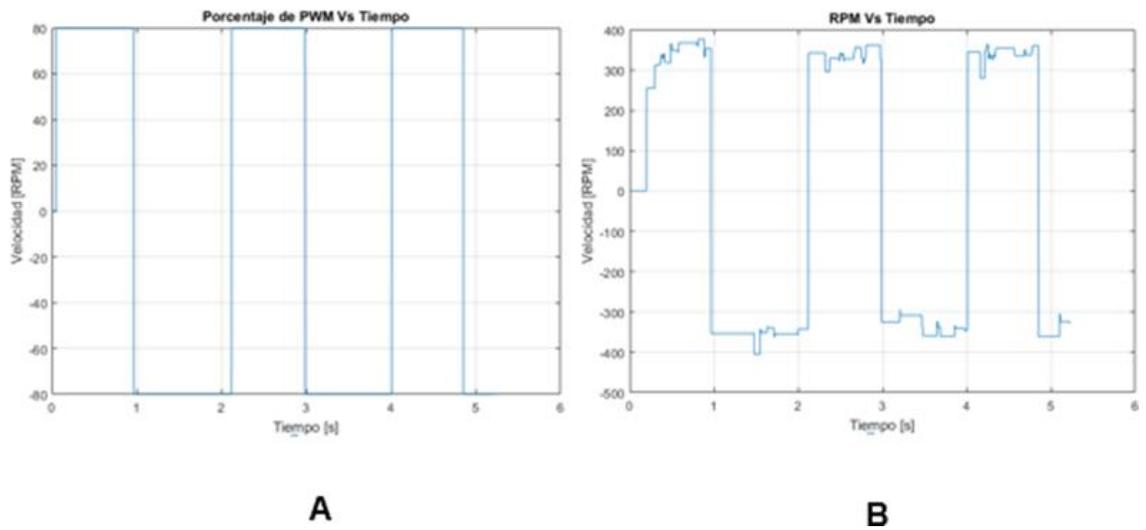


Figura 35 Validación A) Entrada en %PWM para motor izquierdo. B) Salida en RPM motor izquierdo [Autores]

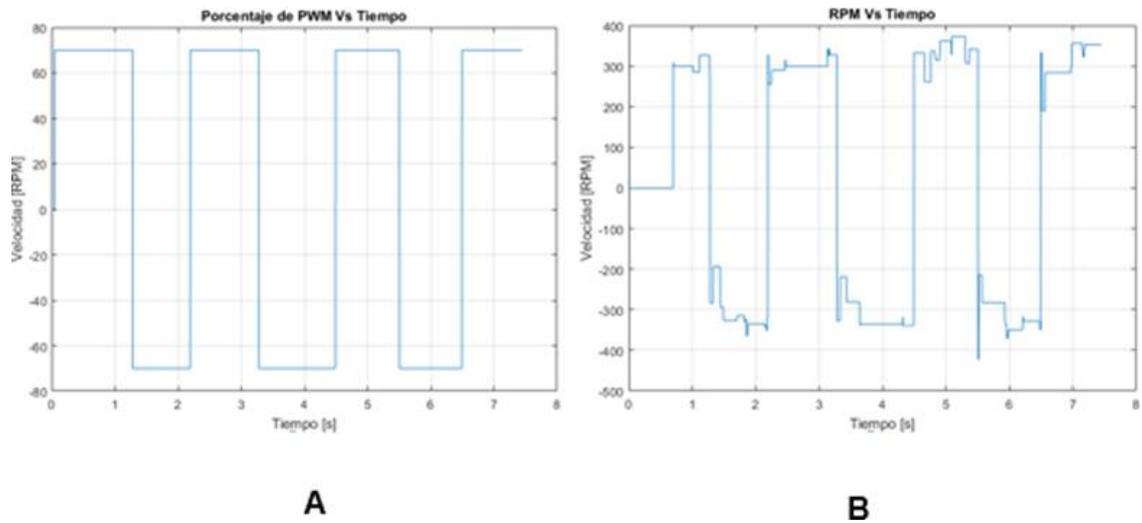


Figura 36 Identificación A) Entrada en %PWM para motor derecho. B) Salida en RPM motor derecho [Autores]

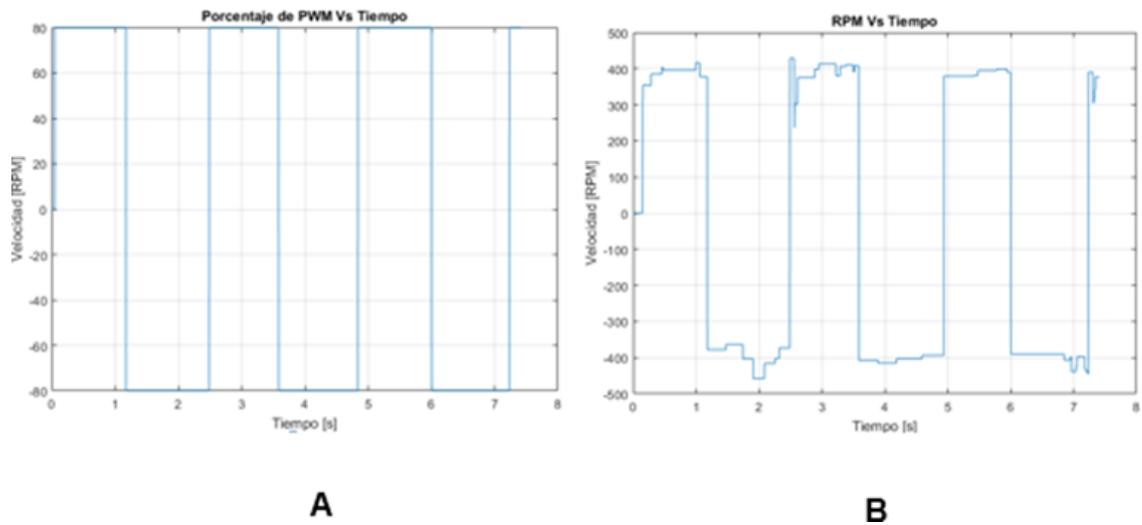


Figura 37 Validación A) Entrada en %PWM para motor derecho. B) Salida en RPM motor derecho [Autores]

Empleando la toolbox Ident de MATLAB®, se ingresaron los datos con un periodo de muestreo de 0.002 s, y se estableció la función de transferencia de cada motor, validando para obtener el mejor BestFit teniendo como criterio el tiempo de establecimiento, valor en estado estable y sobrepaso sean similares a los experimentales ante una entrada escalo unitario. Para el motor izquierdo se

encontró el mejor BestFit de 82.16 como se ve en la figura 38.A con la opción transfer functions y tuvo un BestFit de 92.98 de validación como se observa en la figura 38.B, se obtuvo la función de transferencia:

$$\frac{1.498e07 s^2 + 1.919e07 s + 4.737e08}{s^4 + 2476 s^3 + 3.013e06 s^2 + 4.66e06 s + 9.51e07} \quad (26)$$

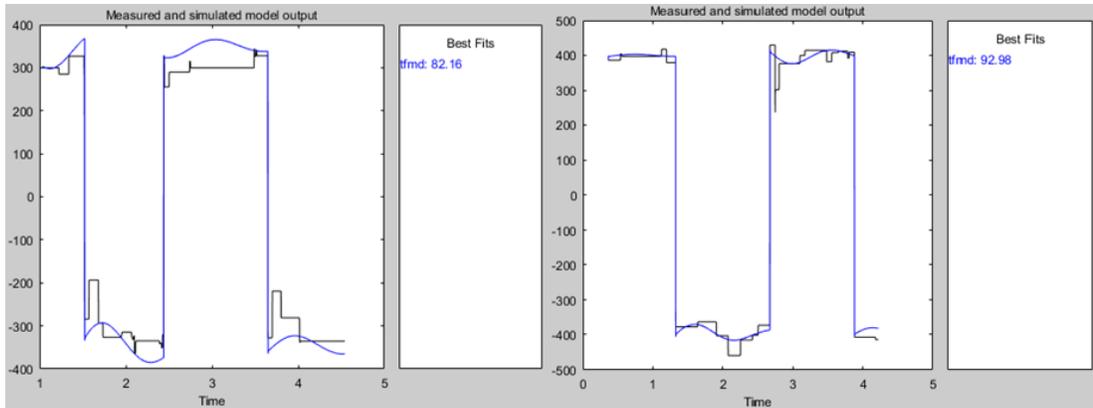


Figura 38 BestFit motor derecho A) Identificación. B) Validación [Autores]

En la figura 39 se observa la respuesta de la función de transferencia obtenida ante una entrada escalón y se observa con un tiempo de establecimiento de 0.00338 segundos.

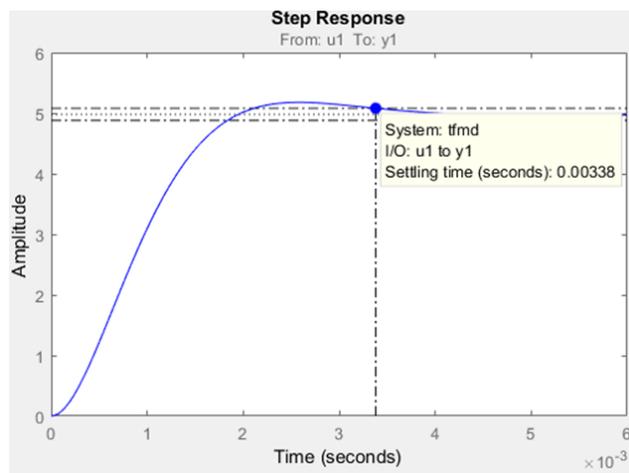


Figura 39 Respuesta motor derecho a una entrada escalón [Autores]

Para el motor izquierdo, su mejor BestFit de identificación fue de 93.53 como se ve en la figura 40.A con la opción transfer-functions y tuvo un BestFit de 87.39 de validación como se observa en la figura 40.B, se obtuvo la función de transferencia:

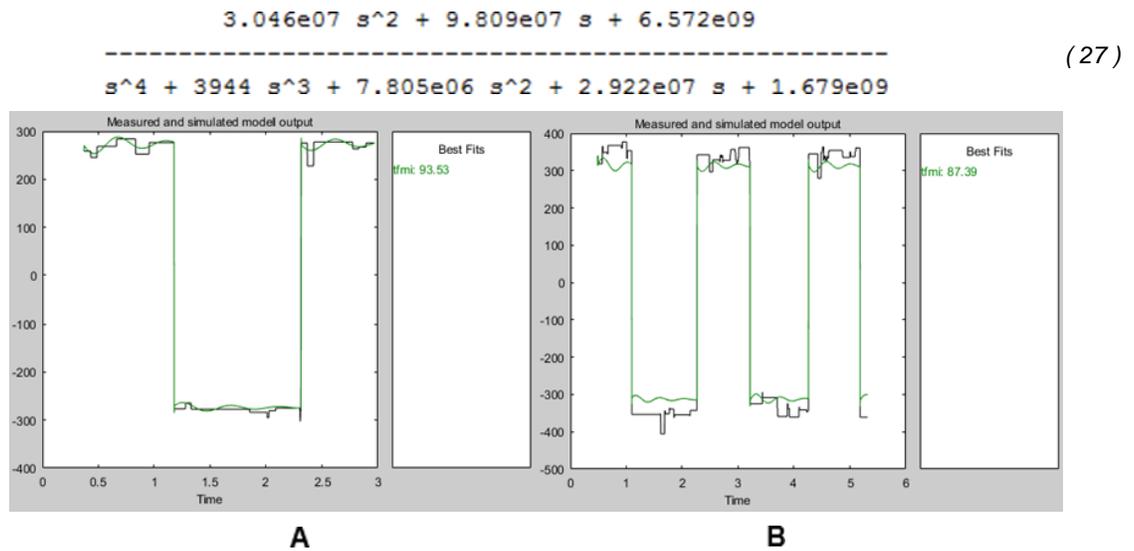


Figura 40 BestFit motor izquierdo A) Identificación. B) Validación [Autores]

En la figura 41 se observa la respuesta de la función de transferencia del motor izquierdo obtenido ante una entrada escalón y se observa con un tiempo de establecimiento de 0.0021 segundos.

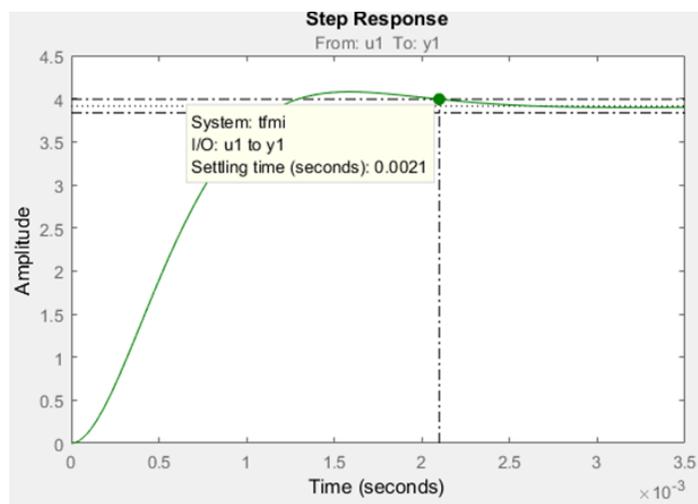


Figura 41 Respuesta motor izquierdo a una entrada escalón [Autores]

Ya teniendo el modelo matemático para la velocidad cada motor, se procede a diseñar cada controlador a lazo cerrado, con la toolbox rltool de MATLAB®, utilizando la arquitectura que se muestra en la figura 42, donde C es el compensador o la función de transferencia del controlador, y G es la planta o función de transferencia del motor, H y F para que no afecten el sistema se les asigna un valor de 1.

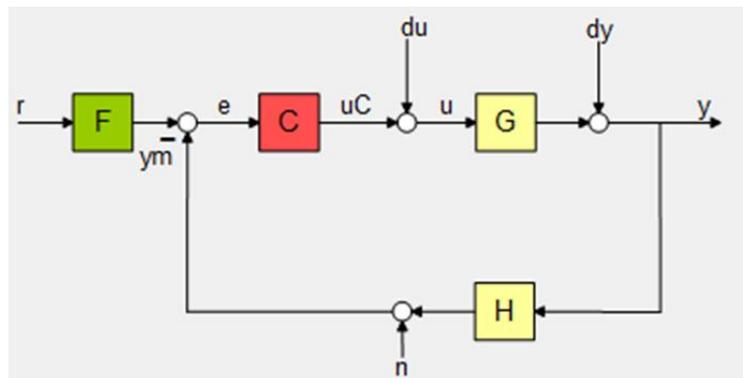


Figura 42 Arquitectura de controlador [Autores]

El mejor compensador diseñado fue un PI el cual redujo el sobre paso que tenía la planta, llegando al punto de consigna y dando una mejor acción de control figura 44, comparado con el PID que también se diseñó y con el cual se comparó, y su tiempo de establecimiento fue de 0.00257 segundos como se muestra en la figura 43 ante la entrada escalón.

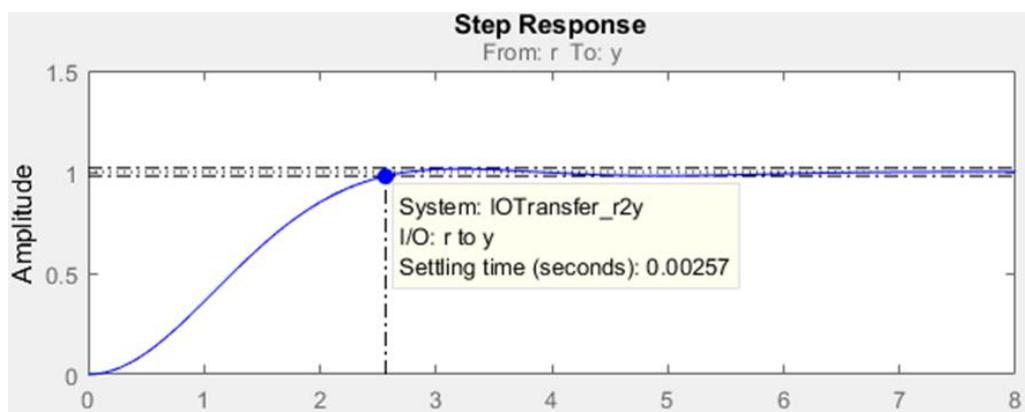


Figura 43 Respuesta al controlador PI en lazo cerrado en el motor derecho [Autores]

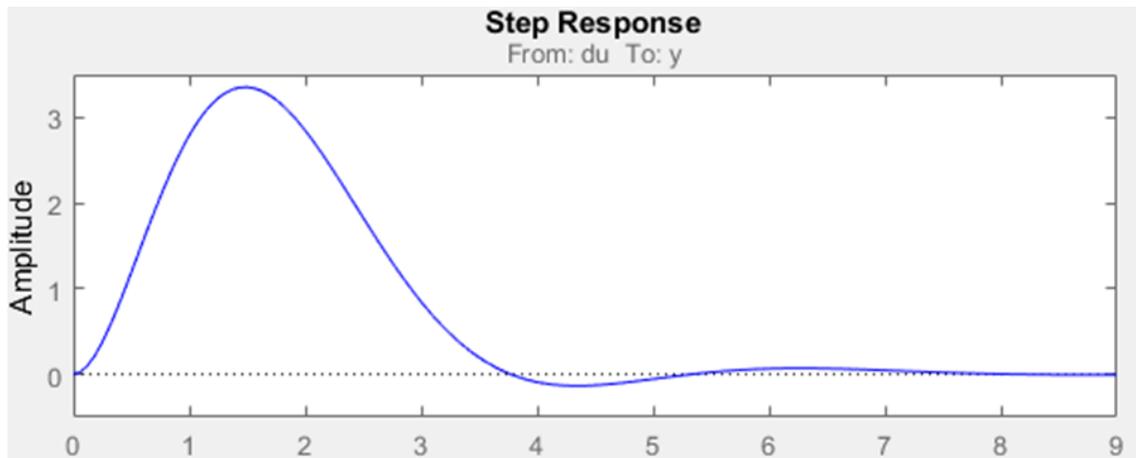


Figura 44 Acción de control PI en motor derecho [Autores]

El mejor compensador para el motor izquierdo fue un PI al igual que el derecho, redujo el sobre paso que tenía la planta, llegando al punto de consigna y dando una mejor acción de control figura 46, y su tiempo de establecimiento fue de 0.00233 segundos como se muestra en la figura 45 ante la entrada escalón.

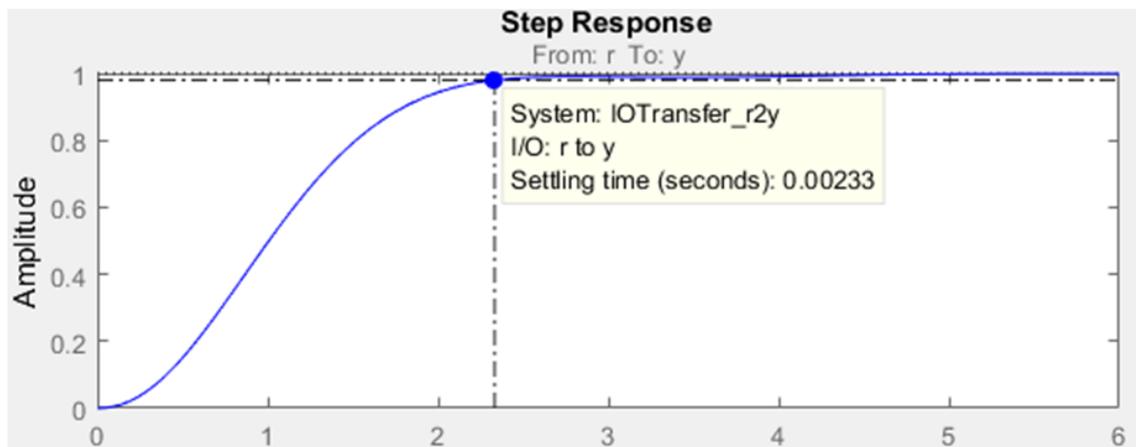


Figura 45 Respuesta al controlador PI en lazo cerrado en el motor izquierdo [Autores]

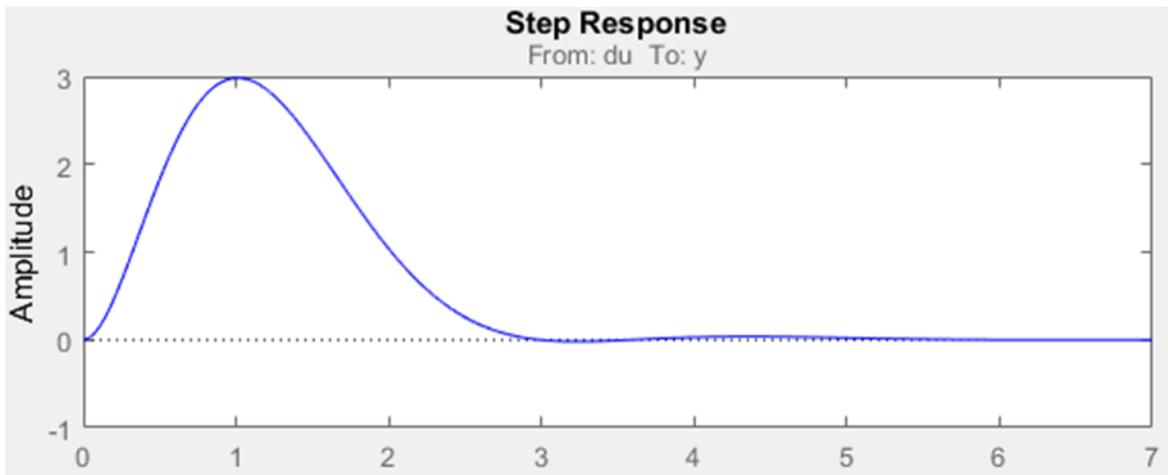


Figura 46 Acción de control PI en motor izquierdo [Autores]

El controlador PID diseñado para el motor derecho cuenta con un tiempo de establecimiento de 0.00302s y un sobrepaso de 5.63%. Se observa en la figura 47, mientras el compensador PID para el motor izquierdo tiene un tiempo de establecimiento de 0.00308s y un sobrepaso de 9.75%, se observa en la figura 48. Los cuales fueron descartados porque no cumplían con los requerimientos de diseño, para el diseño de los compensadores era requerido un tiempo de establecimiento bajo en su respuesta y además contara con una respuesta sin sobre paso.

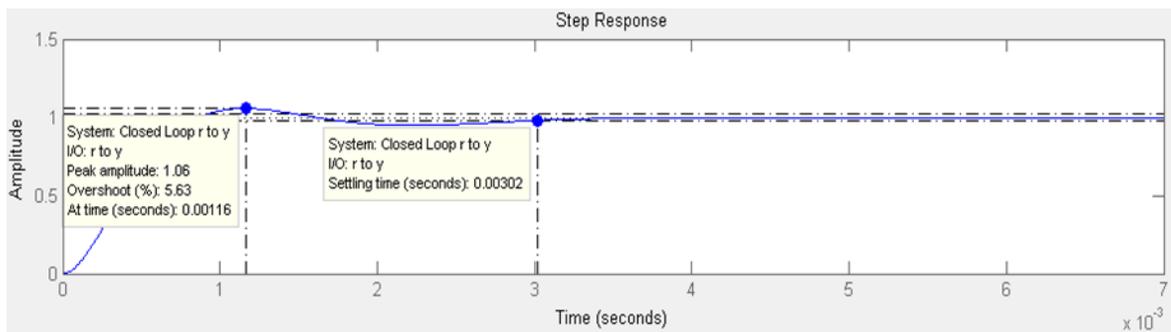


Figura 47 Respuesta controlador PID motor derecho ante una entrada escalón [Autores]

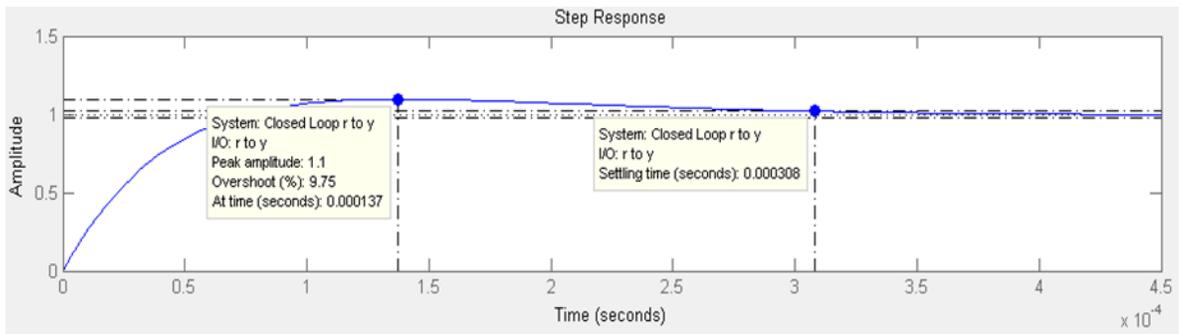


Figura 48 Respuesta controlador PID motor izquierdo ante una entrada escalón [Autores]

Se comprobó el correcto funcionamiento de cada compensador en Simulink®, siguiendo la arquitectura mostrada anteriormente en la figura 49, donde la entrada fue randomica entre -400 y 400 [rpm].

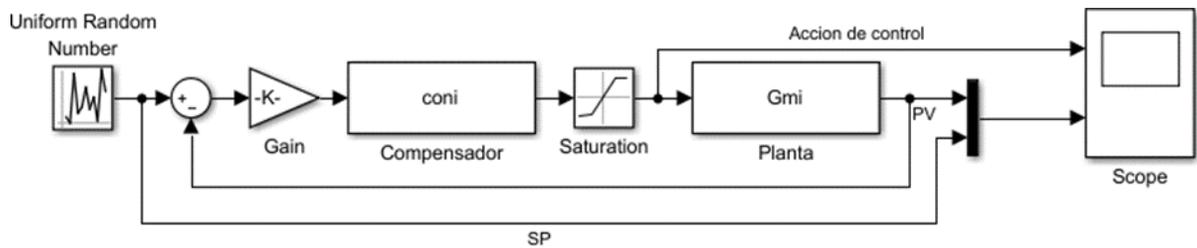


Figura 49 Bloque Simulink® [Autores]

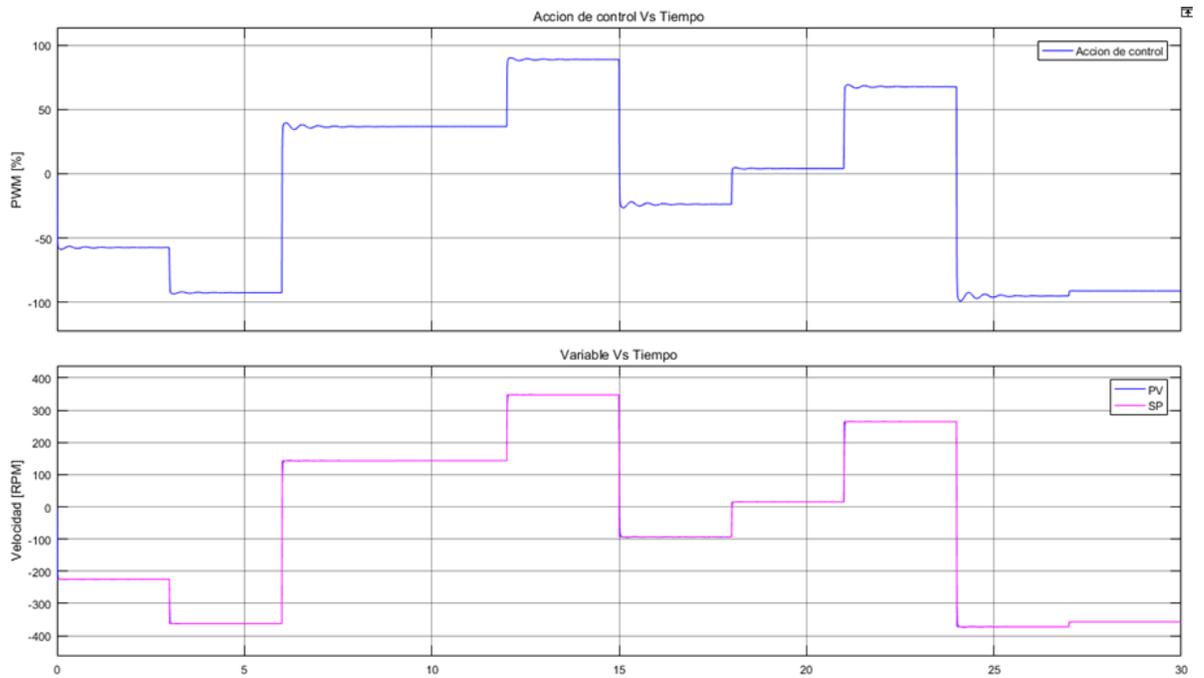


Figura 50 Arriba: acción de control motor izquierdo, Abajo: referencia y variable de proceso [Autores]

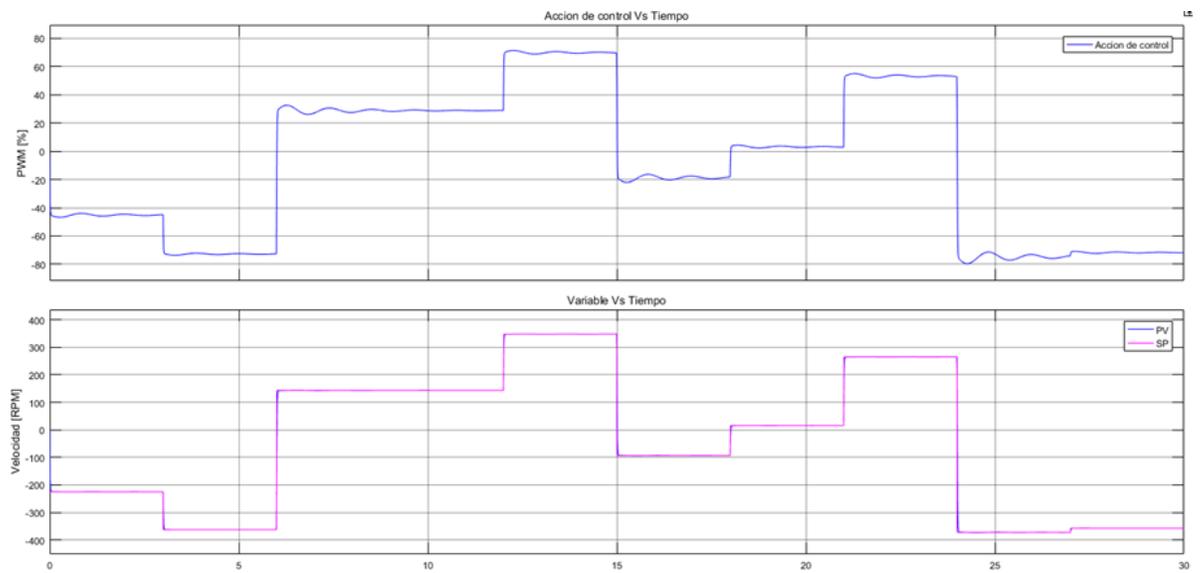


Figura 51 Arriba: acción de control motor izquierdo, Abajo: referencia y variable de proceso [Autores]

8.4 Desarrollo del VATNT

Al tener definida el área de trabajo y la ubicación de cada árbol se dividió en puntos a los cuales el vehículo puede llegar sin ningún problema eliminando los puntos dentro de un radio de 60cm alrededor de cada árbol esto con el fin de buscar un lugar óptimo para la toma de imágenes. Se observa en la figura 52. Siendo el punto verde: la ubicación de cada árbol, la circunferencia verde: el radio de separación donde el VATNT tomará la foto y los puntos rojos: ubicación a la cual el VATNT podrá llegar.

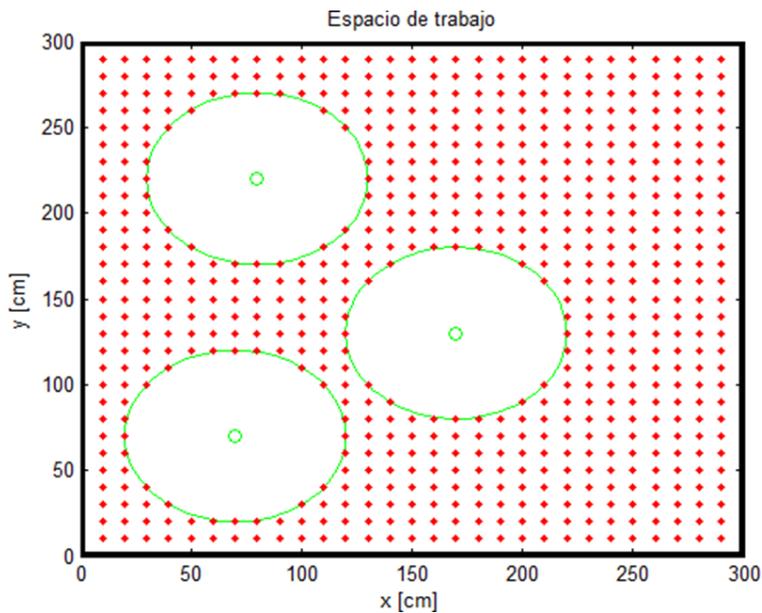


Figura 52 División del espacio de trabajo [Autores]

Al tener todos los puntos posibles en los cuales el vehículo puede llegar dentro del área de trabajo, se procede a buscar los dos (2) lugares desde el cual hará las capturas de cada foto a un radio de 60 cm, eliminando los puntos que se encuentren sobre el mismo eje 'X' o 'Y' de todos los árboles evitar que el vehículo se mueva sobre estos ejes. Se observa en la figura 53.

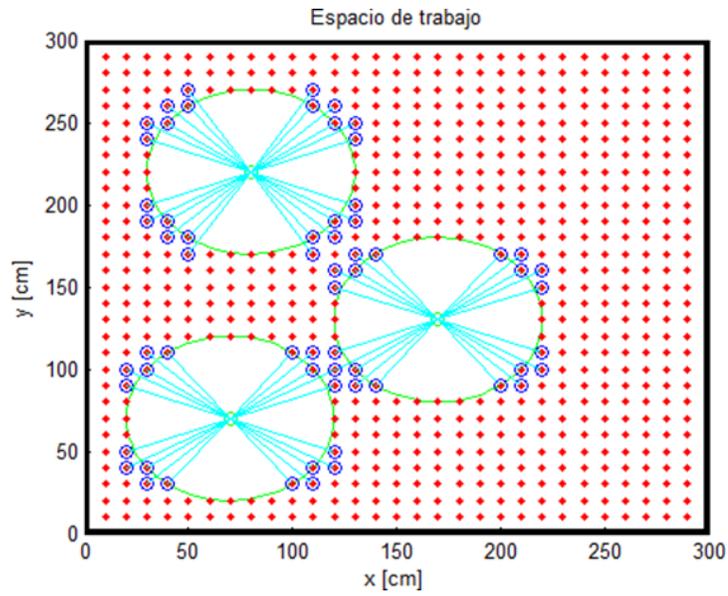


Figura 53 Ubicaciones posibles desde el cual podrá tomar fotos [Autores]

Seguido se observa si hay punto en común entre los árboles, si existe, se envían las coordenadas al vehículo con el fin de reducir la trayectoria, eliminando los demás puntos. Se observa en la figura 54.

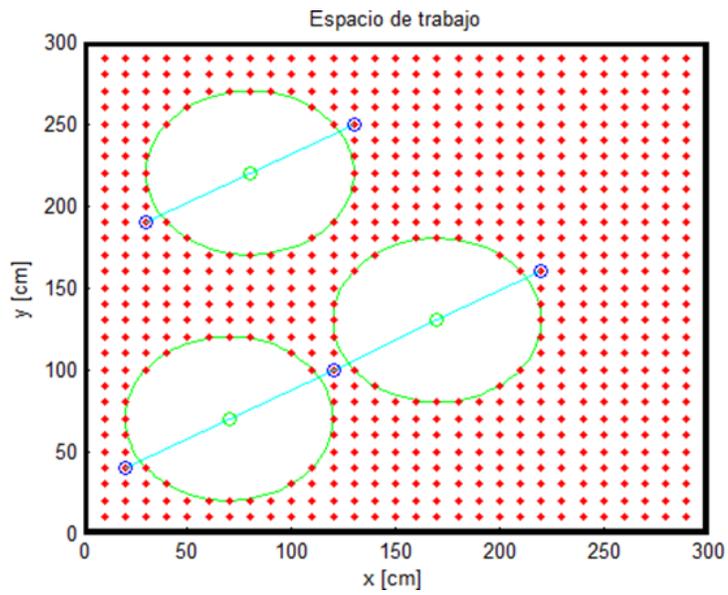


Figura 54 Ubicación óptima para hacer captura de imagen [Autores]

9. COMUNICACIÓN

Una de las partes más importantes del proyecto es la comunicación del vehículo autónomo terrestre no tripulado con el ordenador porque la información que obtiene cada uno de estos se complementa para llevar a cabo un buen funcionamiento. Al momento de realizar dicha comunicación fue necesario implementarla de forma inalámbrica ya que el VATNT estará en desplazamiento constante, esto permitirá guiar al robot, ordenarle movimientos y obtener la información del mismo.

9.1 Comunicación digital

Para llevar a cabo esta comunicación se implementó un par de módulos Xbee los cuales proporcionan una comunicación serial bidireccional entre ambas partes. Para este sistema se utilizó módulos Xbee Pro de la compañía Digi® con una antena integrada tipo chip y potencia de transmisión de 100mW y operan con una banda de frecuencia de 2.4 GHz. Para lograr comunicación entre coordinador modulo conectado al ordenador y router modulo conectado en el vehículo autónomo fueron emparejados previamente, esta configuración fue realizada mediante el software X-CTU.

Los módulos Xbee fueron conectados al ordenador mediante un puerto USB para ser leídos por el software X-CTU, para esto se abrió dos veces el programa para la lectura independiente de cada módulo, en la figura 56 se observa los pasos para conectar el modulo que se quiere configurar, seleccionando primero el puerto por el cual está conectado, seguido del botón “Test / Query” el cual abrirá la pestaña “Com test / Query Modem” donde se verifica que exista conexión con el modulo

seguidamente dar click en ok, pasar a la pestaña “Terminal” para verificar que responda correctamente.

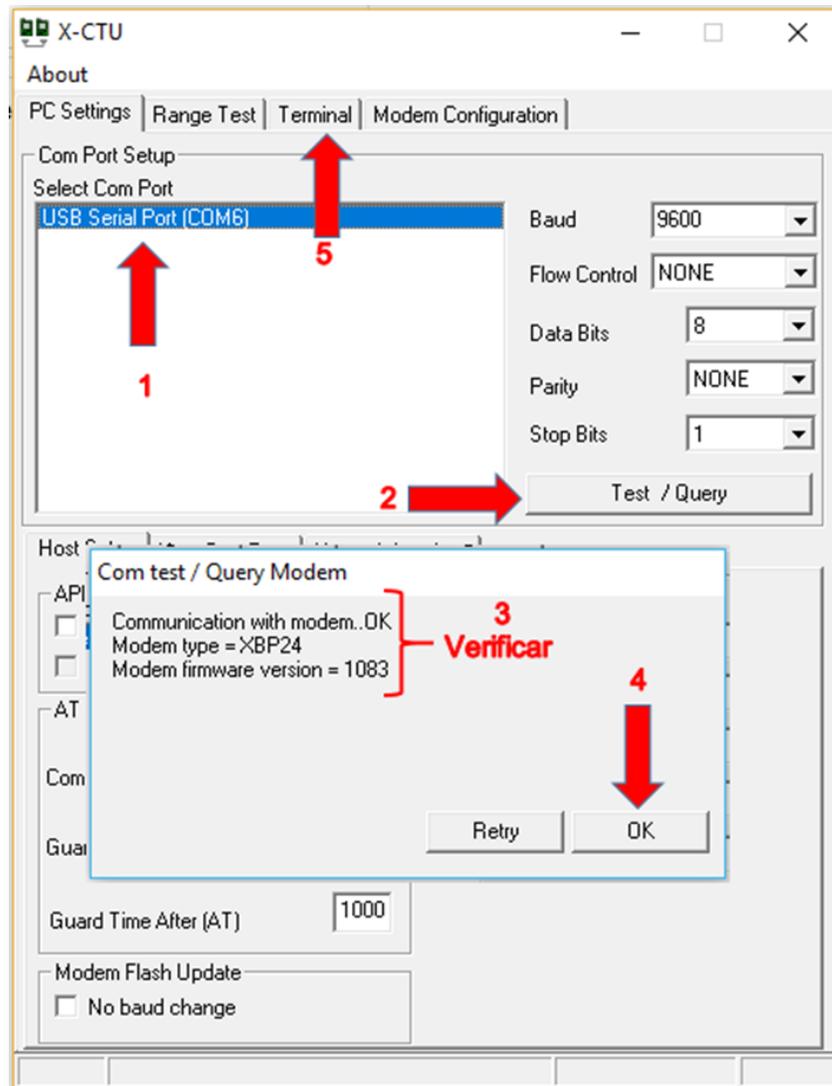


Figura 56 Lectura de Xbee [Autores]

Para observar si la Xbee está respondiendo correctamente se digita en el terminal tres (3) signos más (+) el cual aparecerá en color azul y se recibe como respuesta en color rojo “OK” respondiendo que tenemos comunicación con dicho modulo, se observa en la figura 57.



Figura 57 Respuesta Xbee [Autores]

Teniendo la comunicación en el módulo pasaremos a la pestaña “Modem Configuración” donde se procede a realizar la configuración de cada Xbee para hacer su emparejamiento. Primero se escoge si es “Coordinador” o “router” para esto se selecciona en “Función Set”. La configuración básica para el emparejamiento de los módulos es el cruce de seriales Serial Number Low (SL) y Serial Number High (Sh) de cada Xbee con Destination Address High (DH) y

Destination Address Low (DL) del otro modulo y el mismo ID en PANID para ambos, figura 58.

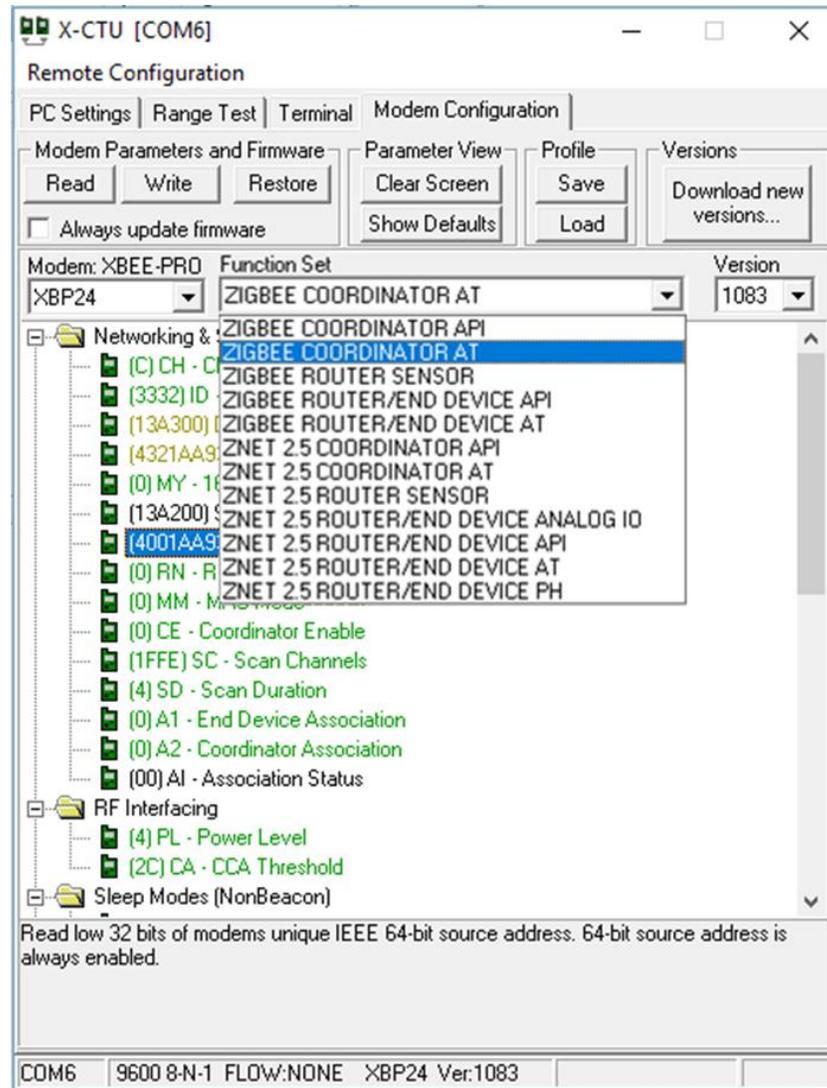


Figura 58 Configuración Xbee [Autores]

EL módulo Xbee configurado como router se encuentra conectado al puerto serial 3 de la placa arduino mega ADK, el cual recibirá del módulo configurado como coordinador la ubicación a la cual se debe dirigir enviando devuelta una respuesta

al momento de llegar a las coordenadas indicadas para que el software de identificación siga con su trabajo se observa en la figura 59.

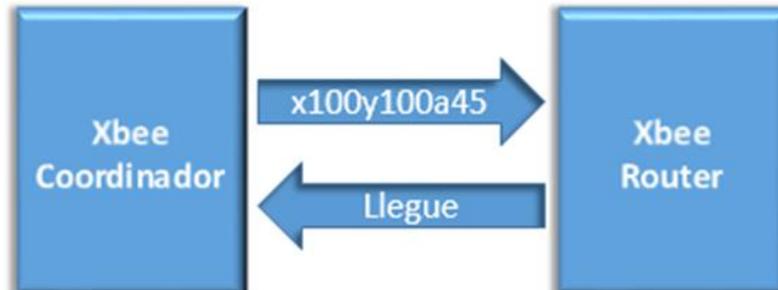


Figura 59 Metodología de comunicación entre Xbee [Autores]

El coordinador envía la ubicación y orientación final a la cual debe llegar el VATNT y cuando este cumple con su rutina el router devolverá como respuesta que ya llego esperando su próxima orden.

9.2 Comunicación análoga

Al no poderse hacer adquisición de imágenes mediante la placa controladora implementada en el VATNT se optó por hacer una comunicación inalámbrica análoga de video en tiempo real. Transmitiendo los datos de la cámara FPV mencionada anteriormente, con un transmisor de video TS832, se observa en la figura 60, que funciona mediante radiofrecuencia, a una frecuencia de 5.8GHz con un alcance máximo de 200m en exteriores y 30m en interiores.

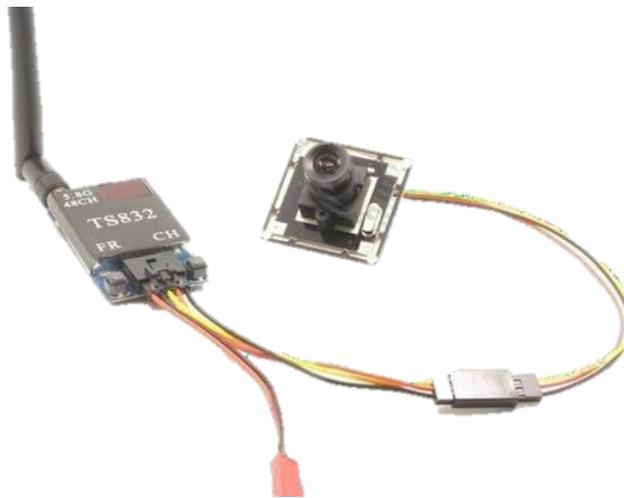


Figura 60 Cámara FPV y transmisor análogo [25]

La señal enviada por el TS832 es captada por el receptor RC832, los cuales deben estar configurados por el mismo canal y la misma frecuencia la cual se escogió canal ocho (8) y frecuencia tres (3) con el fin de obtener una mejor calidad de imagen ya que es la mayor frecuencia de trabajo del dispositivo, tabla 6.

FR \ CH		CH							
		CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR	FR1	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
	FR2	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
	FR3	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
	FR4	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M

Tabla 6 Frecuencias de operación para la comunicación análoga [27]

El dispositivo análogo implementado en el proyecto tiene ocho (8) canales y seis (6) frecuencias, y también tiene salida análoga la cual debe ser digitalizada para ser enviada al ordenador y esto se realiza mediante una capturadora de video, figura 61, la cual recibe la señal análoga y envía por puerto USB al ordenador.



Figura 61 Capturadora de video [Autores]

Se realizó un terminal encargado de recibir la información completa del vehículo autónomo no tripulado inalámbricamente, tanto análoga como digital el cual cuenta con la capturadora de video, el receptor RC832 plus mencionados anteriormente para la comunicación análoga, también con el módulo Xbee configurado como coordinador para la comunicación digital, un repetidor USB al cual se encuentra conectado la señal digitalizada de la imagen en tiempo real de la capturadora y la señal digital de la Xbee, figura 62.

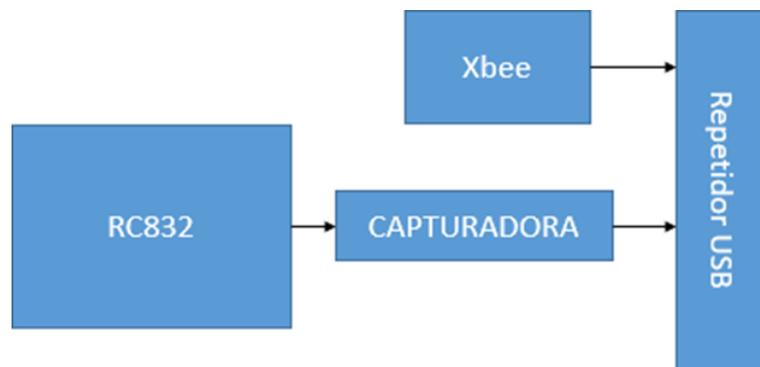


Figura 62 Diagrama de la conexión interna del terminal inalámbrico [Autores]

El terminal cuenta con un cooler encargado de extraer el calor generado dentro del mismo, ya que la capturadora de video tiende a subir su temperatura al igual que el receptor RC832 al encontrarse alimentados. Solo cuenta con una entrada de alimentación de 12v para alimentar el cooler y el RC832 y una salida USB para ser conectada al ordenador.

10. VISIÓN ARTIFICIAL

Sabiendo que la primera etapa para el reconocimiento de imágenes es la adquisición de estas, teniendo en cuenta las condiciones en las cuales son obtenidas. Para la captura de imágenes se usó una cámara ubicada en la parte superior del vehículo garantizando una toma total de los frutos presentes en el árbol para comprobar su maduración.

Se realizaron ajustes a la posición y orientación de la cámara con el objetivo de tomar una imagen optima, para hacer captura de la imagen la cámara se ubicó a 10cm de altura con una inclinación de 20° separada a 60cm del árbol de cacao como se observa en la figura 63. Las imágenes fueron capturadas con una resolución de 720x480 pixeles usando fondo blanco e iluminación LED blanca para facilitar el procesamiento.

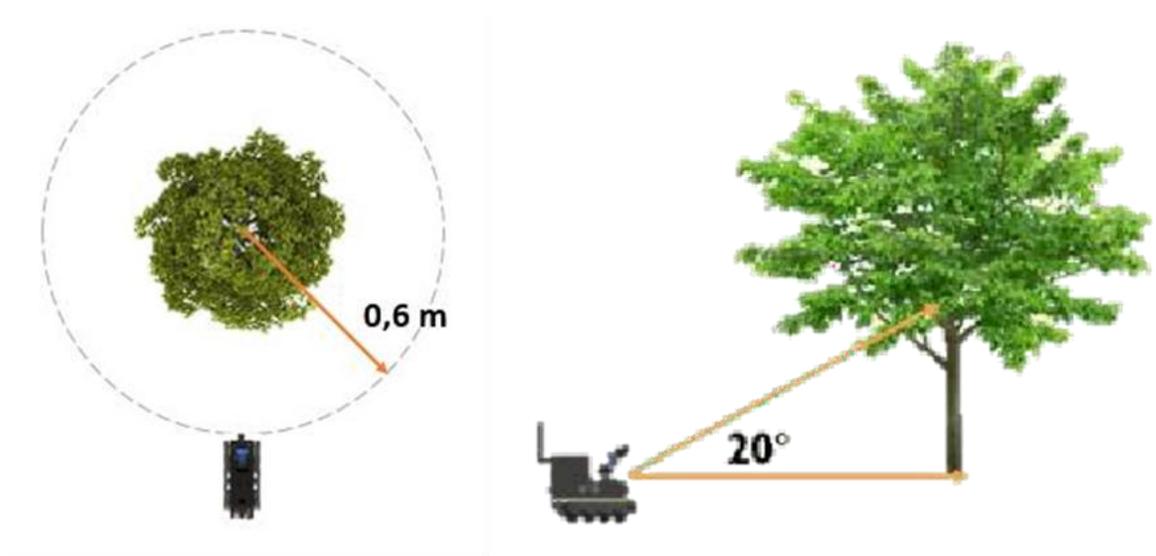


Figura 63 Ubicación y orientación de la cámara [Autores]

Luego de seleccionar el fondo para la captura de imágenes y ajustar la cámara pasamos a la siguiente etapa, el procesado de imagen donde se establece si un fruto de cacao está maduro o no.

El análisis de color cuenta de tres fases: segmentación, extracción de características y algoritmo de clasificación:

La primera fase es la segmentación que consiste en separar el fruto de los demás objetos presentes en la imagen debido a que en esta etapa solo necesitaremos la información concerniente a los frutos de cacao, al terminar esta etapa pasamos a la extracción de característica.

La segunda fase es la extracción de características en la cual se hizo conversión a diferentes espacios de color como $L^*a^*b^*$, HSV y YCbCr con los cuales se separa la información cromática, luminosidad y crominancia de la imagen, características más adecuadas para un reconocimiento que el espacio RGB al momento de analizar los colores de un objeto. Al ser tomada la foto en formato RGB se usaron operaciones matemáticas.

Al realizar las conversiones se obtuvieron nuevas matrices que representan la imagen original, separando las características de cada fruto debido a que esto es vital para determinar si un cacao está apto para ser cosechado. Las características seleccionadas de los diferentes espacios con los cuales trabajamos son las siguientes seis (6); crominancia azul y crominancia roja obtenidas del espacio YCbCr, la coordenada de color entre rojo y verde, coordenada de color entre amarillo y azul con el espacio de color CIE $L^*a^*b^*$ y la saturación y valor del espacio HSV.

La tercera fase es el clasificador donde se implementó un clasificador por red neuronal con una capa de entrada de seis (6) neuronas las cuales son cada las características mencionadas anteriormente sacadas de cada pixel de la imagen, dos (2) capas ocultas.

El proceso para pasar por el clasificador es ver las características de cada pixel existente en la imagen el cual entrará a la red neuronal y saldrá respuesta si pertenece a un fruto maduro o no maduro.

10.1 Clasificación por color

Se tomó una muestra de frutos de cacao figura 64, en la cual se le hizo reconocimiento de color con dos formas de segmentación diferentes, para separar el fruto maduro y no maduro mediante visión artificial.



Figura 64 Mazorcas de Cacao [3]

10.1.1 YCbCr

Al procesar la imagen se separó solo el fruto maduro y no maduro, diferenciado por su crominancia azul y roja, y se obtuvo la separación como se ve en la figura 65.



Figura 65 A) Fruto maduro YCbCr. B) Fruto no maduro YCbCr [Autores]

10.1.2 CIELAB

Mediante la segmentación de color CIELAB, se obtuvo la media de cada color presente en la imagen para poder hacer el código y diferenciar cada color y se obtuvo la separación como se ve en la figura 66.



Figura 66 A) Fruto maduro CIELAB. B) Fruto no maduro CIELAB [Autores]

10.2 Primeras pruebas

Unas de las primeras pruebas del vehículo autónomo terrestre no tripulado, al llegar el vehículo autónomo la posición en la que se encuentra ubicado el primer árbol, mediante técnicas de visión artificial al reconocer el árbol, tomara una foto y la recortara al tamaño del árbol para hacer más fácil la identificación de los frutos, quitando de la imagen lo que se pueda ver al fondo y algunas ramas u hojas que puedan quedar en esta para hacer más sencillo el reconocimiento. Figura 67.



Figura 67 Pruebas de clasificación de mazorcas de cacao [Autores]

Al hacer este proceso, el vehículo seguirá su trayectoria y tomara otra foto por el lado opuesto del árbol del cual tomo y hará el mismo procedimiento, con el fin de determinar la cantidad de frutos maduros y no maduros existen sacando el porcentaje, estimando si la planta debe ser cosechada o aun no es necesario ir a recoger sus frutos.

11. INTERFAZ GRÁFICA

La aplicación HMI desarrollada fue pensada de forma sencilla y de fácil manejo para el usuario, de tal forma que no es necesaria una capacitación al momento de implementarlo ya que su uso cuenta con un paso a paso en el cual pide información detallada del terreno, ubicación exacta de cada árbol y puertos de conexión.

El menú principal del interfaz cuenta con solo 3 botones, figura 68, los cuales tienen como función configuración del espacio de trabajo e inicio del vehículo autónomo no tripulado, registro de cada recorrido realizado por el VATNT el cual tiene fecha y hora exacta en que fue realizado, y por ultimo salir del sistema.

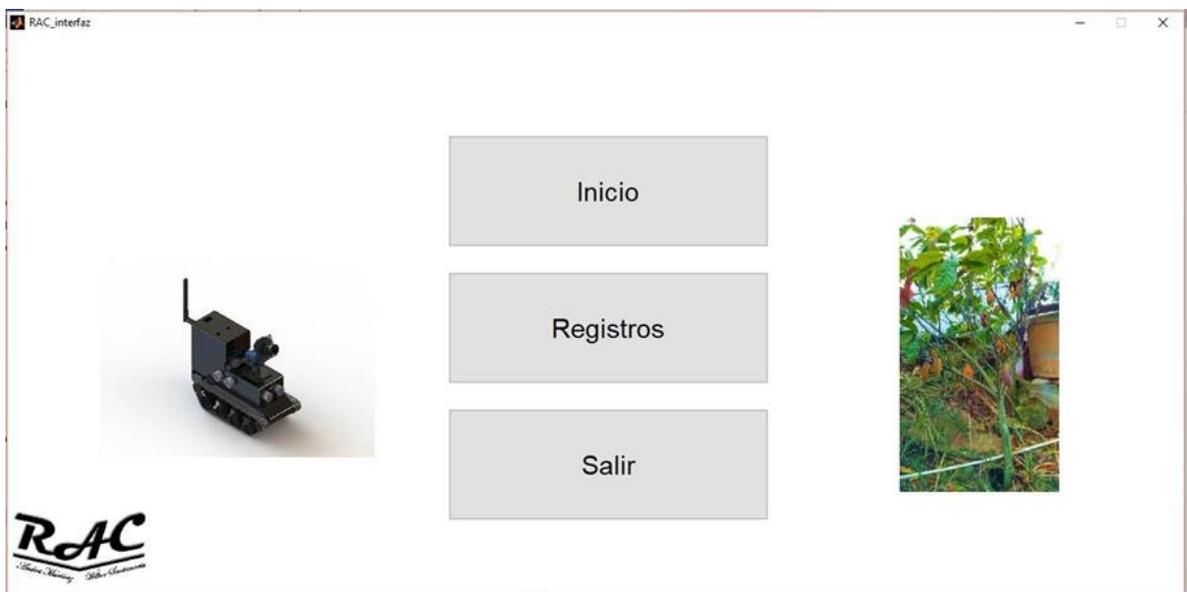


Figura 68 Aplicación HMI [Autores]

Su primer función es la configuración e implementación del sistema completamente, dando inicio a la trayectoria que tomara el vehículo autónomo una vez sean dadas

las ubicación de cada árbol de cacao, aquí se observa el recorrido que va a hacer el VATNT y los puntos a los cuales llegara para tomar la foto, figura 69.

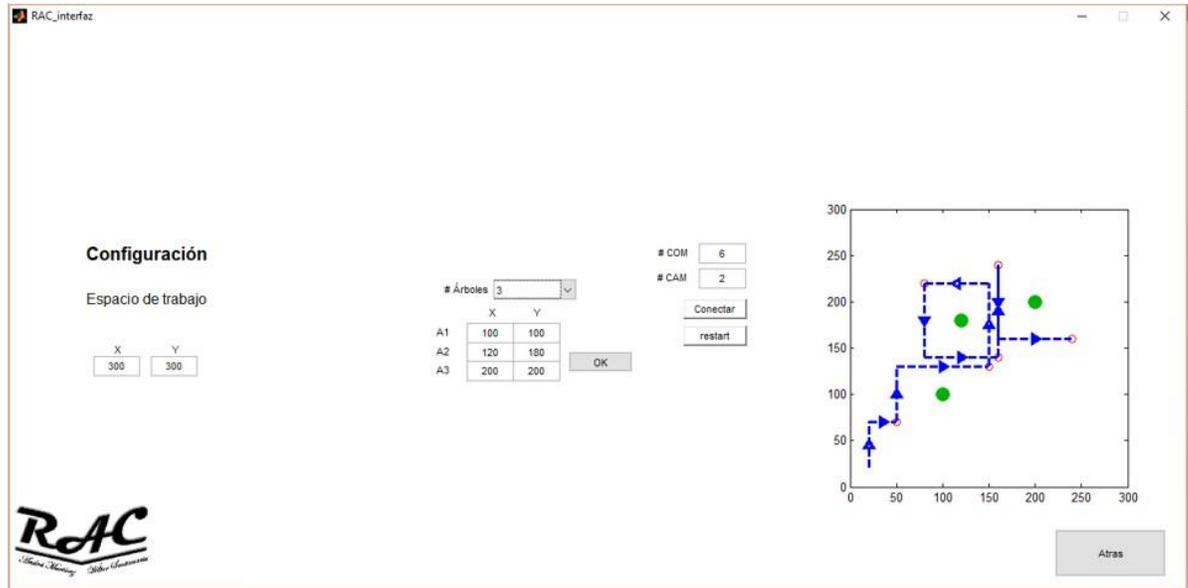


Figura 69 Función botón inicio [Autores]

Donde la parte de configuración es para establecer el área de trabajo del VATNT, seguido de la ubicación de árboles de cacao, donde se tendrá que seleccionar el puerto COM al cual está conectado el modula Xbee y el número de la cámara (por defecto en un ordenador de escritorio es 1 y en laptop es 2). Se observa un ejemplo en el gráfico de la trayectoria que realizara el vehículo autónomo para hacer captura de imágenes y enviarlas al terminal inalámbrico para ser guardadas en los registros de la aplicación.

El botón de registros tiene el historial de las imágenes adquiridas en todos los recorridos realizados por el vehículo autónomo terrestre no tripulado, figura 70. Donde además muestra los resultados del procesamiento de imágenes al momento se seleccionar el que se desea ver.

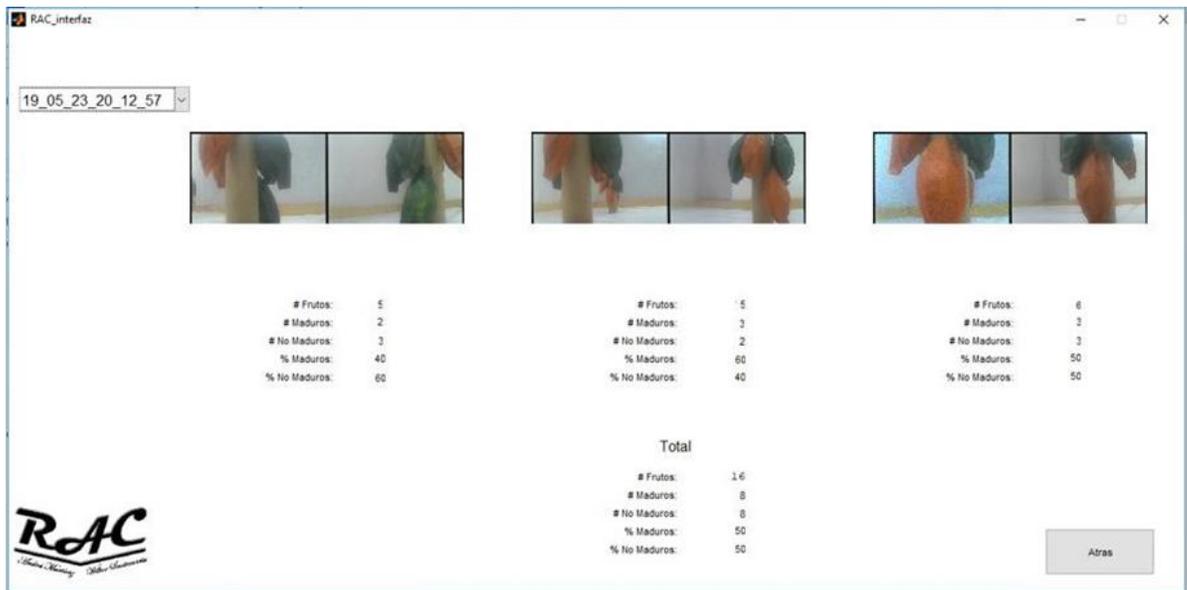


Figura 70 Función botón Registros [Autores]

Por último la función del botón “Salir” cierra las aplicaciones internas del programa que pudieron haber quedado en funcionamiento antes de cerrar la interfaz por completo.

12. RESULTADOS OBTENIDOS

Se realizaron diferentes pruebas con el objetivo de obtener el porcentaje de error total existente en el sistema al momento de hacer el reconocimiento de frutos, para esto se hicieron 15 trayectorias diferentes con 5 diferentes pruebas por trayectoria, cada prueba tenía una cantidad diferentes de frutos maduros y no maduros.

En la primera trayectoria se realizó una prueba con 3 frutos maduros y 1 fruto verde en cada árbol dando un porcentaje de error de 8% con un solo error en el segundo árbol, la segunda prueba se ubicaron 2 frutos maduros y 2 frutos verdes por árbol dando un porcentaje de error de 8% teniendo un error en el segundo árbol, la tercer prueba se ubicaron 3 frutos verdes y 1 maduro en el cual no hubo error en el reconocimiento de los frutos, la cuarta prueba tenía 4 frutos maduros en cada árbol y dio un porcentaje de error de 15% con error en el tercer árbol, la quinta y última prueba de esta trayectoria se realizó con 4 frutos verdes por árbol y dio un error en el segundo árbol de 8%, resultados en la tabla 7.

#1	A1	A2	A3	% Error
1V 3M	1V 3M	1V 4M	1V 3M	8%
2V 2M	2V 2M	1V 2M	2V 2M	8%
3V 1M	3V 1M	3V 1M	3V 1M	0%
4M	4M	4M	6M	15%
4V	4V	3V	4V	8%

Tabla 7 Resultados trayectoria uno (1) [Autores]

El porcentaje total de error presente en el sistema es aproximadamente de 7%, en la tabla 8 se observa el porcentaje obtenido en cada prueba de cada trayectoria realizada, algunas trayectorias realizadas tenían un excelente reconocimiento debido a que en esos puntos en que los arboles estaban ubicados no les afectaba mucho la iluminación del entorno.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Total
Trayectoria 1	8%	8%	0%	15%	8%	7,8%
Trayectoria 2	8%	8%	15%	8%	8%	9,4%
Trayectoria 3	15%	8%	0%	8%	8%	7,8%
Trayectoria 4	0%	0%	0%	0%	0%	0%
Trayectoria 5	8%	8%	15%	8%	8%	9,4%
Trayectoria 6	8%	8%	15%	8%	8%	9,4%
Trayectoria 7	8%	0%	8%	15%	8%	7,8%
Trayectoria 8	0%	0%	0%	0%	0%	0%
Trayectoria 9	8%	15%	0%	8%	8%	7,4%
Trayectoria 10	8%	8%	8%	15%	8%	9,4%
Trayectoria 11	8%	8%	15%	15%	0%	9,2%
Trayectoria 12	15%	8%	8%	8%	8%	9,4%
Trayectoria 13	8%	8%	0%	15%	8%	7,8%
Trayectoria 14	0%	0%	0%	0%	0%	0%
Trayectoria 15	0%	8%	8%	15%	8%	7,8%
						≈7%

Tabla 8 Porcentajes de error [Autores]

13. CONCLUSIONES

- La orientación del VATNT se puede ver afectada por la presencia de algún material magnético presente en el lugar de trabajo, para eliminar esta distorsión se hace un análisis del terreno, con el fin de disminuir el error en la lectura del sensor se implementó el modelo de calibración a la salida del magnetómetro para corregir los efectos causados por el hierro duro y suave.
- Se realizaron pruebas donde se logró evidenciar que con solo dos (2) fotos a 180° una de la otra son suficientes para obtener la información completa de la cantidad de frutos maduros y no maduros se encuentran presentes en el árbol de cacao, reduciendo así el tiempo y distancia recorrida del VATNT.
- La interfaz gráfica está diseñada de manera tal que cualquier usuario pueda entender su funcionamiento de forma ágil e intuitiva sin necesidad de una capacitación previa, haciendo de este un sistema favorable para todo agricultor de cacao.
- El sistema de adquisición fue desarrollado con una tarjeta controladora de la familia arduino porque soporta una gran cantidad de sensores y drivers entre los cuales se encuentran los necesarios para el buen funcionamiento del VATNT.
- El sistema de reconocimiento de imágenes es afectado por la iluminación del lugar en el cual se hace captura de imágenes, motivo por el cual es necesario hacer un pre-procesado para aclarar y tonificar colores mejorando el resultado final del análisis mediante visión artificial.

- Los algoritmos para generar la trayectoria tienen en cuenta todas las ubicaciones posibles desde el cual se puede lograr una buena captura de imágenes, teniendo en cuenta también si existe un punto en común entre los árboles mejorando así el tiempo de recorrido del VATNT.

14. TRABAJO FUTURO

Teniendo en cuenta los resultados obtenidos del sistema y de los aportes dados por los evaluadores y el director, tanto de la trayectoria del VATNT, como del reconocimiento de frutos de cacao mediante el software R.A.C. (Reconocimiento Autónomo de Cacao) y los requerimientos necesarios para garantizar su implementación en la industria de cacao, es necesario realizar los siguientes ajustes al sistema:

- Hacer más capturas de imágenes en cada punto con un bucle en el software para mejorar el reconocimiento mediante una media de las fotos tomadas.
- Optimizar el sistema de posicionamiento implementando antenas y tag RTLS (Real Time Location System) para poder llevarlo al cultivo real de cacao.
- Adquirir un magnetómetro de mayor precisión el cual se encuentre aislado de perturbaciones externas.
- Mejorar el sistema mecánico de forma más robusta que soporte un funcionamiento óptimo en terrenos no controlados.

15. BIBLIOGRAFÍA

- [1] Instituto Meteorológico Nacional de Costa Rica “ENOS (El niño-Oscilación del Sur)”. Dirección URL: <https://www.imn.ac.cr/enos>.
- [2] Sistema de Información Ambiental de Colombia (SIAC) “Fenómenos del niño y la niña”. Dirección URL: <http://www.siac.gov.co/ninoynina>.
- [3] “Caja de herramienta para cacao”. Guía 1: “El cacao en sistemas agroforestales”. Lutheran World Relief. Dirección URL: <http://cacaomovil.com/>.
- [4] Federación Nacional de Cacaoteros, Dirección URL: <http://www.fedecacao.com.co/portal/index.php/es/2015-04-23-20-00-33/118-en-san-vicente-de-chucuri-ya-hay-casos-de-exito-con-colombia-siembra>.
- [5] L. I. Garcia. “Modelado Cinemático y Control de Robots Móviles con Ruedas”. Ph.D. disertación, Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, 2010.
- [6] J. M. Medel. “Desarrollo De Una Plataforma De Monitorización Y Control De Un Minirobot Móvil Basada En Redes De Sensores Inalámbricos”. Universidad de Sevilla, Escuela Técnica Superior de Ingeniería.
- [7] Jhonny A. Valencia, Alejandro Montoya. “Modelo cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odometrica”. Universidad Tecnológica de Pereira. 2009.
- [8] J. I. Perez; S. Tovar; V. C. Ubaldo; G. H. Efren; J. C. Pedraza; J. E. Vascas. J.

M. Arreguin, O. A. Sotomayor. "Robot Móvil de Tracción Diferencial con Plataforma de Control Modular para Investigación y Desarrollo Ágil de Proyectos". 10th National Congress of mechatronics. Puerto Vallarta, Jalisco, 2011.

[9] J. Collazo; E. Gorrostieta; J. Pedraza; U. Villaseñor; R. Romero and M. Gonzalez. "Modelación de un Robot de dos Ruedas con Tracción Diferencial". 8th National Congress of mechatronics. Veracruz, 2009.

[10] L. Rios; M. Bueno; "Modelo Matemático Para Un Robot Móvil" Scientia Et Technica Año XIV. 2008.

[11] Jean Carlos Herrera Pérez, Silfri Manuel Medina Ortiz. "Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial". Universidad Autónoma del Caribe. 2015.

[12] Carlos Julio Arismendi Pereira. "Técnicas de análisis del patrón respiratorio y cardiorrespiratorio ara la extracción de índices en pacientes en proceso de extubación". Barcelona. 2007.

[13] Jorge Yazlle and Camilo Jadur. "Dinámica Simbólica: Apuntes de Cátedra". Salta: Departamento de Matemática de la Universidad Nacional de Salta. 2006.

[14] Jean Dickinson Gibbons and Subhabrata Chakraborti. "Nonparametric Statistical Inference". Tuscaloosa, Alabama: Marcel Decker Inc. 2003.

[15] Jaime Villamil, Jesús Delgado. "Entrenamiento de una red neuronal multicapa para la tasa de cambio euro - dólar (EUR/USD)". Universidad Nacional de Colombia. 2007.

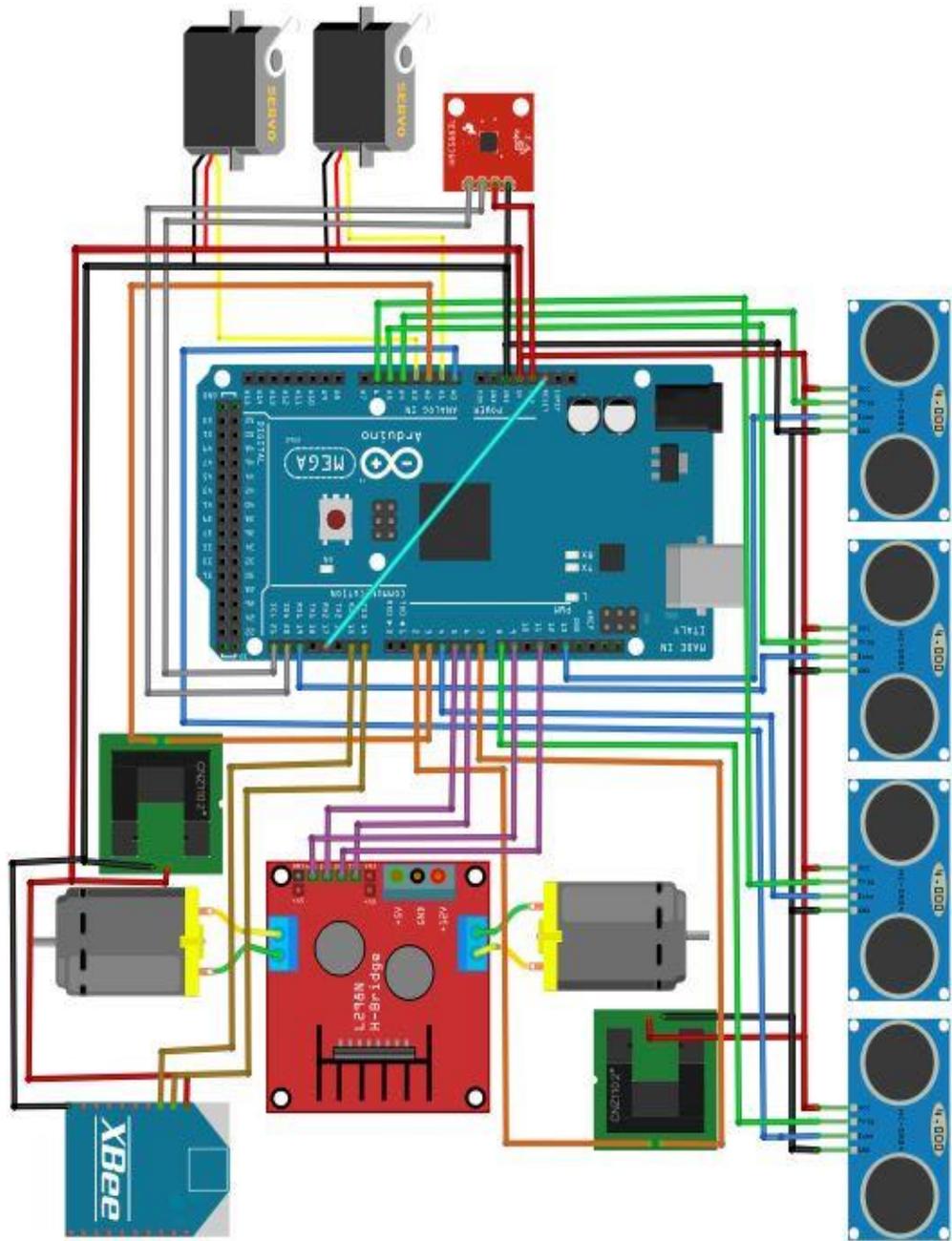
[16] "Caja de herramienta para cacao". Guía 8: "Cosecha, fermentación y secado del cacao". Lutheran World Relief. Dirección URL: <http://cacaomovil.com/>.

- [17] DHGate. Dirección URL: <https://es.dhgate.com/online-shopping/chassis-robot-diy-online.html>.
- [18] Pololu Robotics & Electronics. Dirección URL: <https://www.pololu.com/product/2214>.
- [19] yorobotics. Dirección URL: <https://yorobotics.co/producto/soporte-camara-para-servo-sg90/>.
- [20] Pololu Robotics & Electronics. Dirección URL: <https://www.pololu.com/product/3081>
- [21] PJRC.Electronic Projects. Dirección URL: https://www.pjrc.com/teensy/td_libs_Encoder.html
- [22] Naylamp mechatronics. Dirección URL: <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [23] VECTORNAV Technologies. Dirección URL: <https://www.vectornav.com/support/library/magnetometer>
- [24] Prometec. “LO QUE HAY QUE SABER PARA ELEGIR LA CAMARA FPV PARA UN CUADROPTERO”. Dirección URL: <https://www.prometec.net/elegir-camara-fpv-para-dron/>
- [25] Desde el aire RC. Dirección URL: <https://www.desdeelairerc.net/fpv/camara-fpv-cmos-700tv/>
- [26] XBee. “¿QUE ES XBEE?”. Dirección URL: <https://xbee.cl/que-es-xbee/>

- [27] "Manual de operacion RC832 y TS832" .2013
- [28] Moviltronics. "Modulo L298N(Puente H)". Dirección URL: <https://moviltronics.com.co/modulos/143-modulo-l298-rojo.html>.
- [29] Magnetic Declination. "Find the magnetic declination at your location". Dirección URL: <http://www.magnetic-declination.com/>. 2018.
- [30] Yuri Matselenak. DIY Drones. "Advanced hard soft iron magnetometer calibration". Dirección URL: <https://diydrone.com/profiles/blogs/advanced-hard-and-soft-iron-magnetometer-calibration-for-dummies?id=705844%3ABlogPost%3A1676387&page=4>. Julio 2014.
- [31] Felix Monasterio Helin; Alvaro Gutierrez. "Modelado experimental de un Motor DC real". 2016.

16. ANEXOS

ANEXO A Diagrama conexiones internas del VATNT



ANEXO B Código Arduino

```
#include "Wire.h"
#include "I2Cdev.h"
#include "HMC5883L.h"

int Xi=10, Yi=10, Ai=-4, XX=288, YY=288;
int ai=0, giro=0, dd=0, da=0, ret=0;

String ttt="", PWM="";
int c=0, s=0, f=0, xbee=0;
int spxf=Xi, spyf=Yi, spaf=Ai, llego=0;

long Distacia, t_Distacia;
int sadelante=0, satras=0, sderecha=0, sizquierda=0;

int esd=LOW, esi=LOW, easd=LOW, easi=LOW;
double phd=0.0, pld=0.0, tld=0.0, thd=0.0, td=0.0;
int contd=0, vd=0, svd=0, ttd=0, ssvd=0;
double phi=0.0, pli=0.0, tli=0.0, thi=0.0, ti=0.0, inv;
int conti=0, vi=0, svi=0, tti=0, ssvi=0;
volatile int vvd=0, vvi=0;

int spa=0, arreglo=0, ua=0, spy=0, uy=0;
float errorp=0, errory=0;

HMC5883L magnetometro;
int16_t mx, my, mz;
float declinacion=-8.22, angulo=0.0;
int angle=0;

int cond=0, nomove=1, a=0, x=0, y=0, af=0, xf=0, yf=0, sy=0, sx=0;

float Bx=-30.921, By=-5.918, Bz=-184.122, M11=2.124, M12=0.08, M13=0.015;
float M21=-0.211, M22=2.109, M23=0.112, M31=-0.088, M32=0.171, M33=1.98;
float xv=0.0, yv=0.0, zv=0.0, Xc=0.0, Yc=0.0, Zc=0.0;

int ang()//Angulo
{
  magnetometro.getHeading(&mx, &my, &mz);

  xv=mx-Bx;
  yv=my-By;
  zv=mz-Bz;
  Xc=M11*xv+M12*yv+M13*zv;
  Yc=M21*xv+M22*yv+M23*zv;
```

```

Zc=M31*xv+M32*yv+M33*zv;

angulo=atan2(mz, mx);
angulo=angulo*(180/M_PI);
angulo=angulo-declinacion;
if(angulo>180)
{
  arreglo=angulo-180;
  angulo=-180+arreglo;
}
return angulo;
}

void setup()
{
  attachInterrupt(digitalPinToInterrupt(2),Encd,RISING);
  attachInterrupt(digitalPinToInterrupt(3),Enci,RISING);
  Serial.begin(9600);
  Serial3.begin(9600);
  pinMode(19, INPUT);
  pinMode(10, INPUT);
  pinMode(4, INPUT);
  pinMode(A0, INPUT);
  pinMode(48, OUTPUT);
  pinMode(A5, OUTPUT);
  pinMode(A6, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A3, OUTPUT);
  Wire.begin();
  magnetometro.initialize();
}

void moveree(int pin,int aangulo)//Servo
{
  float pausa;
  pausa=aangulo*2000.0/180.0+500;
  digitalWrite(pin,HIGH);
  delayMicroseconds(pausa);
  digitalWrite(pin,LOW);
  delayMicroseconds(25000-pausa);
}

int ultra(int pinout, int pinin)//Ultrasonidos
{
  digitalWrite(pinout,LOW);

```

```

delayMicroseconds(5);
digitalWrite(pinout, HIGH);
delayMicroseconds(15);
digitalWrite(pinout,LOW);
t_Distacia=pulseIn(pinin, HIGH);
Distacia= int(0.0185*t_Distacia);
if(Distacia>XX)
{
  Distacia=XX;
}
return Distacia;
}

```

```

int Svd()//Velocidad Derecha
{
  esd=digitalRead(2);
  if(esd!=easd)
  {
    phd=pulseIn(2,HIGH,100000);
    pld=pulseIn(2,LOW,100000);
    tld=pld/1000000;
    thd=phd/1000000;
    td=thd*10+tld*10;
    vd=1.4/td;
    contd=0;
  }
  else
  contd++;
  if(contd==20 || vd>10000)
  vd=0;
  easd=esd;
  return vd;
}

```

```

int Svi()//Velocidad izquierda
{
  esi=digitalRead(3);
  if(esi!=easi)
  {
    phi=pulseIn(3,HIGH,100000);
    pli=pulseIn(3,LOW,100000);
    tli=pli/1000000;
    thi=phi/1000000;
    ti=thi*10+tli*10;
    vi=1.4/ti;
    conti=0;
  }
  else

```

```

    conti++;
    if(conti==20 || vi>10000)
    vi=0;
    easi=esi;
    return vi;
}
void Encd()
{
    if(digitalRead(7)==HIGH)
    vvd=-1;
    else
    vvd=1;
}
void Enci()
{
    if(digitalRead(A2)==HIGH)
    vvi=-1;
    else
    vvi=1;
}
int probando=0;
int por=70, uu=0;
int ser=0;
void loop()
{
    moveree(A1,90);//abajo
    moveree(A3,115);//arriba
    if(xbee==1)
    {
        Xi=10, Yi=10;
        xf=spxf;
        yf=spyf;
        ai=Ai;//siempre adelante
        af=spaf;
        xbee=0;
        nomove=0;
        cond=1;
    }
    angle=ang();
    if(angle>Ai-20 && angle<Ai+20)//Adelante
    {
        if(sy==0)
        {
            satras=ultra(A5,19);
            sadelante=0;
            y=satras;//+9
        }
    }
}

```

```

else
{
  sadelante=ultra(48,10);
  satras=0;
  y=YY-sadelante;
}
if(sx==0)
{
  sizquierda=ultra(8,A0);
  sderecha=0;
  x=sizquierda;
}
else
{
  sderecha=ultra(A6,4);
  x=XX-sderecha;
  sizquierda=0;
}
if(satras>(YY/2))
sy=1;
if(sadelante>(YY/2))
sy=0;
if(sizquierda>(XX/2))
sx=1;
if(sderecha>(XX/2))
sx=0;
}
if(angle>1 && angle<41)//Atras
{
  if(sy==0)
  {
    satras=ultra(A5,19);
    sadelante=0;
    y=YY-satras-9;
  }
  else
  {
    sadelante=ultra(48,10);
    satras=0;
    y=sadelante+9;
  }
  if(sx==0)
  {
    sderecha=ultra(A6,4);
    sizquierda=0;
    x=sderecha;
  }
}

```

```

else
{
  sizquierda=ultra(8,A0);
  sderecha=0;
  x=XX-sizquierda;
}
if(satras>(YY/2))
sy=1;
if(sadelante>(YY/2))
sy=0;
if(sizquierda>(XX/2))
sx=0;
if(sderecha>(XX/2))
sx=1;
}
if(angle>-115 && angle<-75)//Derecha
{
  if(sy==0)
  {
    sderecha=ultra(A6,4);
    sizquierda=0;
    y=sderecha;
  }
  else
  {
    sizquierda=ultra(8,A0);
    sderecha=0;
    y=YY-sizquierda;
  }
  if(sx==0)
  {
    satras=ultra(A5,19);
    sadelante=0;
    x=satras;//+9
  }
  else
  {
    sadelante=ultra(48,10);
    satras=0;
    x=XX-sadelante;
  }
  if(sderecha>(YY/2))
  sy=1;
  if(sizquierda>(YY/2))
  sy=0;
  if(satras>(XX/2))
  sx=1;
}

```



```

if(sadelante>(XX/2))
sx=0;
}
if(angle>-80 && angle<-40)//Izquierda
{
if(sy==0)
{
sizquierda=ultra(8,A0);
sderecha=0;
y=sizquierda;
}
else
{
sderecha=ultra(A6,4);
sizquierda=0;
y=YY-sderecha;
}
}
if(sx==0)
{
sadelante=ultra(48,10);
satras=0;
x=sadelante+9;
}
else
{
satras=ultra(A5,19);
sadelante=0;
x=XX-satras-9;
}
}
if(sizquierda>(YY/2))
sy=1;
if(sderecha>(YY/2))
sy=0;
if(sadelante>(XX/2))
sx=1;
if(satras>(XX/2))
sx=0;
}
if(cond==3)
{
spa=af;
errorp=spa-angle;
errorr=0;
if(errorp<5 && errorr>-5)
errorp=0, cond=4, llego=1;//avisar a matlab que ya llego
}
}
if(cond==2)

```

```

{
  if(dd==1)
    spy=xf, Xi=xf, errorr=spy-x;
  if(dd==2)
    spy=yf, Yi=yf, errorr=spy-y;
  if(spy>da)
    spa=ai-(88*giro);
  if(spy<da)
    spa=ai+(88*giro);
  if(spa>180)
  {
    arreglo=spa-180;
    spa=-180+arreglo;
  }
  if(spa<-180)
  {
    arreglo=abs(spa)-180;
    spa=180-arreglo;
  }
  errorp=spa-angle;
  if(errorr>-4 && errorr<4 && (errorp>-4 && errorp<4))
    cond=3;
}
if(cond==1)
{
  spa=ai;
  errorp=spa-angle;
  if((spa==Ai))
    spy=yf, dd=1, Yi=yf, da=Xi, errorr=spy-y;
  if((spa>(Ai+70) && spa<(Ai+110))||(spa<(Ai-70) && spa>(Ai-110)))
    spy=xf, dd=2, Xi=xf, da=Yi, errorr=spy-x;
  if((dd==1 && (spa>(-30) && spa<(15))) || (dd==2 && (spa<(Ai-70) && spa>(Ai-110))) ||
(dd==2 && yf<Yi))
    giro=1;
  else
    giro=-1;
  if((errorr>-4 && errorr<4) && (errorp>-4 && errorp<4))
    cond=2;
}

uy=20*(0.625*errorr);
if (uy<-por)
uy=-por;
if(uy>por)
uy=por;
ua=15*(0.2777*errorp);
if (ua<-por)

```

```

ua=-por;
if(ua>por)
ua=por;
if((angle>(Ai-20) && angle<(Ai+20)) || (angle>(-120) && angle<(-70)))//rango ade y der
ret=1;
else
ret=0;
if(ua<=0)
{
analogWrite(9,-(2.55*ua));//ci
analogWrite(5,0);//ci
analogWrite(6,0);//cd
analogWrite(11,-(2.55*ua));//cd
}
if(ua>0)
{
analogWrite(9,0);
analogWrite(5,(2.55*ua));
analogWrite(6,(2.55*ua));
analogWrite(11,0);
}
uu=uy;
if(uy<=0 && errorp<5 && errorp>-5 && nomove==0 && (errorp>2 || errorp<-2))
{
if(uy<=0)
uu=-uy;
if(ret==0)
{
analogWrite(5,0);//ci
analogWrite(9,(2.55*uu));//ci
analogWrite(11,0);//cd
analogWrite(6,(2.55*uu));//cd
}
if(ret==1)
{
analogWrite(5,(2.55*uu));
analogWrite(9,0);
analogWrite(11,(2.55*uu));
analogWrite(6,0);
}
}
if(uy>0 && errorp<5 && errorp>-5 && nomove==0 && (errorp>2 || errorp<-2))
{
if(uy<=0)
uu=-uy;
if(ret==0)
{

```

```

    analogWrite(5,(2.55*uu));
    analogWrite(9,0);
    analogWrite(11,(2.55*uu));
    analogWrite(6,0);
  }
  if(ret==1)
  {
    analogWrite(5,0);//ci
    analogWrite(9,(2.55*uu));//ci
    analogWrite(11,0);//cd
    analogWrite(6,(2.55*uu));//cd
  }
}
if (llego==1)
digitalWrite(12,HIGH), delay(1000), Serial3.println(llego), llego=0;
}
void serialEvent3()
{
  while(Serial3.available(>)>0)
  {
    //a90x0y60f
    xbee=1;
    llego=0;
    digitalWrite(12,LOW);
    ttt=Serial3.readString();
    c=ttt.indexOf('x');
    s=ttt.indexOf('y');
    f=ttt.indexOf('f');
    PWM=ttt.substring(1, c);
    spaf=PWM.toFloat();
    PWM=ttt.substring(c+1, s);
    spxf=PWM.toFloat();
    PWM=ttt.substring(s+1, f);
    spyf=PWM.toFloat();
  }
}
}

```

ANEXO C Código coordenadas

```
function [xf,yf,af]=in(x,y)
xa=x;
ya=y;

Xi=300;
Yi=300;
L=10;
angcar=-4;

arboles=size(xa);

if arboles(2)==3

%arbol 1
arb1x=xa(1);
arb1y=ya(1);
%arbol 2
arb2x=xa(2);
arb2y=ya(2);
%arbol 3
arb3x=xa(3);
arb3y=ya(3);

r=50;

i=1;
clear x y
for Px=L:L:Xi-L;
    for Py=L:L:Yi-L;
        if (sqrt((Px-arb1x)^2+(Py-arb1y)^2))>=r
            if (sqrt((Px-arb2x)^2+(Py-arb2y)^2))>=r
                if (sqrt((Px-arb3x)^2+(Py-arb3y)^2))>=r
                    x(i)=Px;
                    y(i)=Py;
                    i=i+1;
                end
            end
        end
    end
end
clear Px Py
i=size(x);
a1=1;
a2=1;
a3=1;
for j=1:i(:,:2)
    if (sqrt((x(j)-arb1x)^2+(y(j)-arb1y)^2))<=r+10
        if x(j)<=arb1x
```

```

    for l=1:1:i(:,2)
        if x(l)==2*arb1x-x(j) && y(l)==2*arb1y-y(j)
            pa1(a1,:)=x(j) y(j) 2*arb1x-x(j) 2*arb1y-y(j)];
            a1=a1+1;
        end
    end
end
end
end
if (sqrt((x(j)-arb2x)^2+(y(j)-arb2y)^2))<=r+10
    if x(j)<=arb2x
        for l=1:1:i(:,2)
            if x(l)==2*arb2x-x(j) && y(l)==2*arb2y-y(j)
                pa2(a2,:)=x(j) y(j) 2*arb2x-x(j) 2*arb2y-y(j)];
                a2=a2+1;
            end
        end
    end
end
end
if (sqrt((x(j)-arb3x)^2+(y(j)-arb3y)^2))<=r+10
    if x(j)<=arb3x
        for l=1:1:i(:,2)
            if x(l)==2*arb3x-x(j) && y(l)==2*arb3y-y(j)
                pa3(a3,:)=x(j) y(j) 2*arb3x-x(j) 2*arb3y-y(j)];
                a3=a3+1;
            end
        end
    end
end
end
clear i xx1 xx2 yy1 yy2 x y p2 j l r
a1=a1-1;
a2=a2-1;
a3=a3-1;
pa1=pa1(1:a1,:);%xi yi xf yf
pa2=pa2(1:a2,:);%xi yi xf yf
pa3=pa3(1:a3,:);%xi yi xf yf

au1=1;
au2=1;
au3=1;
for i=1:a1
    if pa1(i,1)==arb1x || pa1(i,3)==arb1x || pa1(i,2)==arb1y || pa1(i,4)==arb1y || pa1(i,1)==arb2x ||
pa1(i,1)==arb3x || pa1(i,3)==arb2x || pa1(i,3)==arb3x || pa1(i,2)==arb2y || pa1(i,2)==arb3y ||
pa1(i,4)==arb2y || pa1(i,4)==arb3y || pa1(i,1)==arb1x+10 || pa1(i,3)==arb1x+10 ||
pa1(i,2)==arb1y+10 || pa1(i,4)==arb1y+10 || pa1(i,1)==arb2x+10 || pa1(i,1)==arb3x+10 ||
pa1(i,3)==arb2x+10 || pa1(i,3)==arb3x+10 || pa1(i,2)==arb2y+10 || pa1(i,2)==arb3y+10 ||
pa1(i,4)==arb2y+10 || pa1(i,4)==arb3y+10 || pa1(i,1)==arb1x-10 || pa1(i,3)==arb1x-10 ||
pa1(i,2)==arb1y-10 || pa1(i,4)==arb1y-10 || pa1(i,1)==arb2x-10 || pa1(i,1)==arb3x-10 ||
pa1(i,3)==arb2x-10 || pa1(i,3)==arb3x-10 || pa1(i,2)==arb2y-10 || pa1(i,2)==arb3y-10 ||
pa1(i,4)==arb2y-10 || pa1(i,4)==arb3y-10

else

```

```

    a1u(au1,:)=pa1(i,:);
    au1=au1+1
end
end
clear i
for i=1:a2
    if pa2(i,1)==arb2x || pa2(i,3)==arb2x || pa2(i,2)==arb2y || pa2(i,4)==arb2y || pa2(i,1)==arb1x ||
pa2(i,1)==arb3x || pa2(i,3)==arb1x || pa2(i,3)==arb3x || pa2(i,2)==arb1y || pa2(i,2)==arb3y ||
pa2(i,4)==arb1y || pa2(i,4)==arb3y || pa2(i,1)==arb2x+10 || pa2(i,3)==arb2x+10 ||
pa2(i,2)==arb2y+10 || pa2(i,4)==arb2y+10 || pa2(i,1)==arb1x+10 || pa2(i,1)==arb3x+10 ||
pa2(i,3)==arb1x+10 || pa2(i,3)==arb3x+10 || pa2(i,2)==arb1y+10 || pa2(i,2)==arb3y+10 ||
pa2(i,4)==arb1y+10 || pa2(i,4)==arb3y+10 || pa2(i,1)==arb2x-10 || pa2(i,3)==arb2x-10 ||
pa2(i,2)==arb2y-10 || pa2(i,4)==arb2y-10 || pa2(i,1)==arb1x-10 || pa2(i,1)==arb3x-10 ||
pa2(i,3)==arb1x-10 || pa2(i,3)==arb3x-10 || pa2(i,2)==arb1y-10 || pa2(i,2)==arb3y-10 ||
pa2(i,4)==arb1y-10 || pa2(i,4)==arb3y-10

    else
        a2u(au2,:)=pa2(i,:);
        au2=au2+1;
    end
end
clear i
for i=1:a3
    if pa3(i,1)==arb3x || pa3(i,3)==arb3x || pa3(i,2)==arb3y || pa3(i,4)==arb3y || pa3(i,1)==arb1x ||
pa3(i,1)==arb2x || pa3(i,3)==arb1x || pa3(i,3)==arb2x || pa3(i,2)==arb1y || pa3(i,2)==arb2y ||
pa3(i,4)==arb1y || pa3(i,4)==arb2y || pa3(i,1)==arb3x+10 || pa3(i,3)==arb3x+10 ||
pa3(i,2)==arb3y+10 || pa3(i,4)==arb3y+10 || pa3(i,1)==arb1x+10 || pa3(i,1)==arb2x+10 ||
pa3(i,3)==arb1x+10 || pa3(i,3)==arb2x+10 || pa3(i,2)==arb1y+10 || pa3(i,2)==arb2y+10 ||
pa3(i,4)==arb1y+10 || pa3(i,4)==arb2y+10 || pa3(i,1)==arb3x-10 || pa3(i,3)==arb3x-10 ||
pa3(i,2)==arb3y-10 || pa3(i,4)==arb3y-10 || pa3(i,1)==arb1x-10 || pa3(i,1)==arb2x-10 ||
pa3(i,3)==arb1x-10 || pa3(i,3)==arb2x-10 || pa3(i,2)==arb1y-10 || pa3(i,2)==arb2y-10 ||
pa3(i,4)==arb1y-10 || pa3(i,4)==arb2y-10

    else
        a3u(au3,:)=pa3(i,:);
        au3=au3+1;
    end
end
au1=au1-1;
au2=au2-1;
au3=au3-1;
clear a1 a2 a3 i p12 p21 p22 p31 p32 p11 p12 p1a1 p2a1 p1a2 p2a2 p1a3 p2a3

%primer punto
gotox=a1u(1,1);
gotoy=a1u(1,2);
if arb1x>gotox
    goto0=-90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
else
    goto0=90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
end
gotox2=a1u(1,3);

```

```

gotoy2=a1u(1,4);
if arb1x>gotox2
    goto02=-90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
else
    goto02=90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
end

clear a1 i a1u
a2=size(a2u);
cona2=0;
for i=1:a2(1,1)
    if (a2u(i,1)==gotox && a2u(i,2)==gotoy) || (a2u(i,1)==gotox2 && a2u(i,2)==gotoy2)
        gotox3=a2u(i,1);
        gotoy3=a2u(i,2);
        gotox4=a2u(i,3);
        gotoy4=a2u(i,4);
        cona2=1;
    else
        if (a2u(i,3)==gotox && a2u(i,4)==gotoy) || (a2u(i,3)==gotox2 && a2u(i,4)==gotoy2)
            gotox3=a2u(i,3);
            gotoy3=a2u(i,4);
            gotox4=a2u(i,1);
            gotoy4=a2u(i,2);
            cona2=1;
        end
    end
end
clear i

if cona2==0
    gotox3=a2u(1,1);
    gotoy3=a2u(1,2);
    gotox4=a2u(1,3);
    gotoy4=a2u(1,4);
end
if arb2x>gotox3
    goto03=-90+(atan((arb2y-gotoy3)/(arb2x-gotox3))*180/pi);
else
    goto03=90+(atan((arb2y-gotoy3)/(arb2x-gotox3))*180/pi);
end
if arb2x>gotox4
    goto04=-90+(atan((arb2y-gotoy4)/(arb2x-gotox4))*180/pi);
else
    goto04=90+(atan((arb2y-gotoy4)/(arb2x-gotox4))*180/pi);
end

clear a2 i %a2u
a3=size(a3u);
cona3=0;
for i=1:a3(1,1)
    if (a3u(i,1)==gotox && a3u(i,2)==gotoy) || (a3u(i,1)==gotox2 && a3u(i,2)==gotoy2) ||
        (a3u(i,1)==gotox3 && a3u(i,2)==gotoy3) || (a3u(i,1)==gotox4 && a3u(i,2)==gotoy4)

```



```

    gotox5=a3u(i,1);
    gotoy5=a3u(i,2);
    gotox6=a3u(i,3);
    gotoy6=a3u(i,4);
    cona3=1;
else
    if (a3u(i,3)==gotox && a3u(i,4)==gotoy) || (a3u(i,3)==gotox2 && a3u(i,4)==gotoy2) ||
(a3u(i,3)==gotox3 && a3u(i,4)==gotoy3) || (a3u(i,3)==gotox4 && a3u(i,4)==gotoy4)
        gotox5=a3u(i,3);
        gotoy5=a3u(i,4);
        gotox6=a3u(i,1);
        gotoy6=a3u(i,2);
        cona3=1;
    end
end
end

if cona3==0
    gotox5=a3u(1,1);
    gotoy5=a3u(1,2);
    gotox6=a3u(1,3);
    gotoy6=a3u(1,4);
end

if arb3x>gotox5
    goto05=-90+(atan((arb3y-gotoy5)/(arb3x-gotox5))*180/pi);
else
    goto05=90+(atan((arb3y-gotoy5)/(arb3x-gotox5))*180/pi);
end
if arb3x>gotox6
    goto06=-90+(atan((arb3y-gotoy6)/(arb3x-gotox6))*180/pi);
else
    goto06=90+(atan((arb3y-gotoy6)/(arb3x-gotox6))*180/pi);
end

goto0=goto0+angcar;
goto02=goto02+angcar;
goto03=goto03+angcar;
goto04=goto04+angcar;
goto05=goto05+angcar;
goto06=goto06+angcar;

clearvars -except gotox gotox2 gotox3 gotox4 gotox5 gotox6 gotoy gotoy2 gotoy3 gotoy4 gotoy5
gotoy6 goto0 goto02 goto03 goto04 goto05 goto06
xf=struct('pt1',gotox,'pt2',gotox2,'pt3',gotox3,'pt4',gotox4,'pt5',gotox5,'pt6',gotox6);
yf=struct('pt1',gotoy,'pt2',gotoy2,'pt3',gotoy3,'pt4',gotoy4,'pt5',gotoy5,'pt6',gotoy6);
af=struct('pt1',goto0,'pt2',goto02,'pt3',goto03,'pt4',goto04,'pt5',goto05,'pt6',goto06);
else
if arboles(2)==2
%arbol 1
arb1x=xa(1);

```

```

arb1y=ya(1);
%arbol 2
arb2x=xa(2);
arb2y=ya(2);
i=1;
r=50;
clear x y
for Px=L:L:Xi-L;
    for Py=L:L:Yi-L;
        if (sqrt((Px-arb1x)^2+(Py-arb1y)^2))>=r
            if (sqrt((Px-arb2x)^2+(Py-arb2y)^2))>=r
                x(i)=Px;
                y(i)=Py;
                i=i+1;
            end
        end
    end
end
clear Px Py
i=size(x);
a1=1;
a2=1;
for j=1:1:i(:,2)
    if (sqrt((x(j)-arb1x)^2+(y(j)-arb1y)^2))<=r+10
        if x(j)<=arb1x
            for l=1:1:i(:,2)
                if x(l)==2*arb1x-x(j) && y(l)==2*arb1y-y(j)
                    pa1(a1,:)= [x(j) y(j) 2*arb1x-x(j) 2*arb1y-y(j)];
                    a1=a1+1;
                end
            end
        end
    end
    if (sqrt((x(j)-arb2x)^2+(y(j)-arb2y)^2))<=r+10
        if x(j)<=arb2x
            for l=1:1:i(:,2)
                if x(l)==2*arb2x-x(j) && y(l)==2*arb2y-y(j)
                    pa2(a2,:)= [x(j) y(j) 2*arb2x-x(j) 2*arb2y-y(j)];
                    a2=a2+1;
                end
            end
        end
    end
end
clear i xx1 xx2 yy1 yy2 x y p2 j l r
a1=a1-1;
a2=a2-1;
pa1=pa1(1:a1,:);%xi yi xf yf
pa2=pa2(1:a2,:);%xi yi xf yf

au1=1;
au2=1;

```

```

for i=1:a1
    if pa1(i,1)==arb1x || pa1(i,3)==arb1x || pa1(i,2)==arb1y || pa1(i,4)==arb1y || pa1(i,1)==arb2x ||
    pa1(i,3)==arb2x || pa1(i,2)==arb2y || pa1(i,4)==arb2y || pa1(i,1)==arb1x+10 || pa1(i,3)==arb1x+10 ||
    pa1(i,2)==arb1y+10 || pa1(i,4)==arb1y+10 || pa1(i,1)==arb2x+10 || pa1(i,3)==arb2x+10 ||
    pa1(i,2)==arb2y+10 || pa1(i,4)==arb2y+10 || pa1(i,1)==arb1x-10 || pa1(i,3)==arb1x-10 ||
    pa1(i,2)==arb1y-10 || pa1(i,4)==arb1y-10 || pa1(i,1)==arb2x-10 || pa1(i,3)==arb2x-10 ||
    pa1(i,2)==arb2y-10 || pa1(i,4)==arb2y-10

        else
            a1u(au1,:)=pa1(i,:);
            au1=au1+1;
        end
    end
clear i
for i=1:a2
    if pa2(i,1)==arb2x || pa2(i,3)==arb2x || pa2(i,2)==arb2y || pa2(i,4)==arb2y || pa2(i,1)==arb1x ||
    pa2(i,3)==arb1x || pa2(i,2)==arb1y || pa2(i,4)==arb1y || pa2(i,1)==arb2x+10 || pa2(i,3)==arb2x+10 ||
    pa2(i,2)==arb2y+10 || pa2(i,4)==arb2y+10 || pa2(i,1)==arb1x+10 || pa2(i,3)==arb1x+10 ||
    pa2(i,2)==arb1y+10 || pa2(i,4)==arb1y+10 || pa2(i,1)==arb2x-10 || pa2(i,3)==arb2x-10 ||
    pa2(i,2)==arb2y-10 || pa2(i,4)==arb2y-10 || pa2(i,1)==arb1x-10 || pa2(i,3)==arb1x-10 ||
    pa2(i,2)==arb1y-10 || pa2(i,4)==arb1y-10

        else
            a2u(au2,:)=pa2(i,:);
            au2=au2+1;
        end
    end
clear i

au1=au1-1;
au2=au2-1;
clear a1 a2 a3 i p12 p21 p22 p31 p32 p11 p12 p1a1 p2a1 p1a2 p2a2 p1a3 p2a3
%primer punto
gotox=a1u(1,1);
gotoy=a1u(1,2);
if arb1x>gotox
    goto0=-90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
else
    goto0=90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
end
gotox2=a1u(1,3);
gotoy2=a1u(1,4);
if arb1x>gotox2
    goto02=-90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
else
    goto02=90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
end
clear a1 i a1u
a2=size(a2u);
cona2=0;
for i=1:a2(1,1)
    if (a2u(i,1)==gotox && a2u(i,2)==gotoy) || (a2u(i,1)==gotox2 && a2u(i,2)==gotoy2)

```

```

    gotox3=a2u(i,1);
    gotoy3=a2u(i,2);
    gotox4=a2u(i,3);
    gotoy4=a2u(i,4);
    cona2=1;
else
    if (a2u(i,3)==gotox && a2u(i,4)==gotoy) || (a2u(i,3)==gotox2 && a2u(i,4)==gotoy2)
        gotox3=a2u(i,3);
        gotoy3=a2u(i,4);
        gotox4=a2u(i,1);
        gotoy4=a2u(i,2);
        cona2=1;
    end
end
end
if cona2==0
    gotox3=a2u(1,1);
    gotoy3=a2u(1,2);
    gotox4=a2u(1,3);
    gotoy4=a2u(1,4);
end
if arb2x>gotox3
    goto03=-90+(atan((arb2y-gotoy3)/(arb2x-gotox3))*180/pi);
else
    goto03=90+(atan((arb2y-gotoy3)/(arb2x-gotox3))*180/pi);
end
if arb2x>gotox4
    goto04=-90+(atan((arb2y-gotoy4)/(arb2x-gotox4))*180/pi);
else
    goto04=90+(atan((arb2y-gotoy4)/(arb2x-gotox4))*180/pi);
end

goto0=goto0+angcar;
goto02=goto02+angcar;
goto03=goto03+angcar;
goto04=goto04+angcar;

clearvars -except gotox gotox2 gotox3 gotox4 gotoy gotoy2 gotoy3 gotoy4 goto0 goto02 goto03
goto04
xf=struct('pt1',gotox,'pt2',gotox2,'pt3',gotox3,'pt4',gotox4);
yf=struct('pt1',gotoy,'pt2',gotoy2,'pt3',gotoy3,'pt4',gotoy4);
af=struct('pt1',goto0,'pt2',goto02,'pt3',goto03,'pt4',goto04);
else

if arboles(2)==1
%arbol 1
arb1x=xa(1);
arb1y=ya(1);

r=50;
i=1;
clear x1 y1 x2 y2 x3 y3 theta

```

```

for Px=L:L:Xi-L;
  for Py=L:L:Yi-L;
    if (sqrt((Px-arb1x)^2+(Py-arb1y)^2))>=r
      x(i)=Px;
      y(i)=Py;
      i=i+1;
    end
  end
end
clear Px Py i
i=size(x);
a1=1;
for j=1:1:i(:,2)
  if (sqrt((x(j)-arb1x)^2+(y(j)-arb1y)^2))<=r+10%((x(j)<=arb1x+r && x(j)>=arb1x-r) && (y(j)<=arb1y+r
&& y(j)>=arb1y-r))
    if x(j)<=arb1x
      for l=1:1:i(:,2)
        if x(l)==2*arb1x-x(j) && y(l)==2*arb1y-y(j)
          pa1(a1,:)=[x(j) y(j) 2*arb1x-x(j) 2*arb1y-y(j)];
          a1=a1+1;
        end
      end
    end
  end
end
clear i xx1 xx2 yy1 yy2 x y p2 j l r
a1=a1-1;
pa1=pa1(1:a1,:);%xi yi xf yf

au1=1;
for i=1:a1
  if pa1(i,1)==arb1x || pa1(i,3)==arb1x || pa1(i,2)==arb1y || pa1(i,4)==arb1y || pa1(i,1)==arb1x+10
|| pa1(i,3)==arb1x+10 || pa1(i,2)==arb1y+10 || pa1(i,4)==arb1y+10 || pa1(i,1)==arb1x-10 ||
pa1(i,3)==arb1x-10 || pa1(i,2)==arb1y-10 || pa1(i,4)==arb1y-10

    else
      a1u(a1,:)=pa1(i,:);
      au1=au1+1;
    end
end

clear i a1 a2 a3 i p12 p21 p22 p31 p32 p11 p12 p1a1 p2a1 p1a2 p2a2 p1a3 p2a3

%primer punto
gotox=a1u(1,1);
gotoy=a1u(1,2);
if arb1x>gotox
  goto0=-90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
else
  goto0=90+(atan((arb1y-gotoy)/(arb1x-gotox))*180/pi);
end
gotox2=a1u(1,3);

```

```

gotoy2=a1u(1,4);
if arb1x>gotox2
    goto02=-90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
else
    goto02=90+(atan((arb1y-gotoy2)/(arb1x-gotox2))*180/pi);
end

clear a1 i a1u
goto0=goto0+angcar;
goto02=goto02+angcar;

clearvars -except gotox gotox2 gotoy gotoy2 goto0 goto02
xf=struct('pt1',gotox,'pt2',gotox2);
yf=struct('pt1',gotoy,'pt2',gotoy2);
af=struct('pt1',goto0,'pt2',goto02);
end
end
end
end

```