

**IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN INALÁMBRICA Y
GENERACIÓN DE TRAYECTORIAS A TRAVÉS DE ENTORNOS DINÁMICOS
CONTROLANDO VARIABLES X , Y y θ DE LA PLATAFORMA MÓVIL P-MITO
(PLATAFORMA MOVIL PARA LA IMPLEMENTACIÓN DE TÉCNICAS DE
ODOMETRÍA)**

Santiago Arroyave Marín

Luis Miguel Bueno Ramírez

**IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN INALÁMBRICA Y
GENERACIÓN DE TRAYECTORIAS A TRAVÉS DE ENTORNOS DINÁMICOS
CONTROLANDO VARIABLES X, Y y θ DE LA PLATAFORMA MÓVIL P-MITO
(PLATAFORMA MOVIL PARA LA IMPLEMENTACIÓN DE TÉCNICAS DE
ODOMETRÍA)**

Santiago Arroyave Marín

Luis Miguel Bueno Ramírez

Trabajo de grado presentado como requisito
parcial para optar por el título de
Ingeniero Electricista

Pereira, Enero de 2018

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

Programa de Ingeniería Eléctrica



Implementación del sistema de comunicación inalámbrica y generación de trayectorias a través de entornos dinámicos controlando variables X , Y y Θ de la plataforma móvil P-MITO (Plataforma Móvil para la Implementación de Técnicas de Odometría)

Santiago Arroyave Marín

Luis Miguel Bueno Ramírez

Director: Ingeniero Luis Hernando Ríos Gonzalez

Pereira, Enero de 2018

Programa de Ingeniería Eléctrica

Universidad Tecnológica de Pereira

La Julita, Pereira (Colombia)

Tel: (+57) (6) 3137122

www.utp.edu.co

AGRADECIMIENTOS

Principalmente quiero darles las gracias a mis padres por haberme formado como la persona que soy ahora, todos mis logros se los debo a ustedes y en especial este. Me educaron con empeño y dedicación, siempre queriendo que yo fuera un gran hombre, acompañándome, guiándome y apoyándome siempre en los buenos momentos como también en los más difíciles de mi vida. Quiero agradecerle a mi hermana, la mujer incondicional que se convirtió en mi segunda madre, ella ha sentado en mis los deseos de superación convirtiéndose en ese espejo en el cual me quiero reflejar, pues sus virtudes, valores y gran corazón, me llevan a admirarla cada día más. Finalmente, quiero darle gracias a Dios por mi familia, por mis logros y por mis valiosos amigos que siempre han estado acompañándome en muchos de los momentos más alegres de mi vida.

Santiago Arroyave Marín

Agradezco a DIOS por darme el mejor regalo del mundo que es la vida. Agradezco a mis padres ya que gracias a su dedicación, enseñanzas, esfuerzos, acompañamiento y consejos soy un hombre con grandes valores. También agradezco a mi hermano ya que su amistad y consejos han influenciado en gran manera mi vida. Agradezco a mi esposa ya que gracias a esta gran mujer he superado varios obstáculos y me ha dado muchos momentos de felicidad. Agradezco a mi hija ya que DIOS me brindo este ángel en el momento preciso de mi vida, ya que ella me dio las fuerzas necesarias para terminar mis estudios de Pregrado. Agradezco a mis amigos ya que han estado siempre en los momentos que más los he necesitado. Finalmente agradezco a los familiares que me han brindado su apoyo incondicional.

Luis Miguel Bueno Ramírez

Finalmente, queremos agradecer a nuestro director de tesis, el ingeniero Luis Hernando Ríos, y al Ingeniero Julio Cesar Yepes por toda la disposición y ayuda incondicional brindada durante la elaboración de este proyecto. Sin la asesoría y apoyo brindado por ellos, no hubiéramos podido llevarlo a cabo.

DEDICATORIA

Quiero dedicar este proyecto a Mónica Velásquez Sierra, por su apoyo incondicional, por su acompañamiento, por la motivación y el empuje que dio en mí última etapa de preparación académica de esta carrera. Porque ha dejado a un lado muchos de sus gustos e intereses para compartir su vida conmigo y convertirse en ese fragmento que le hacía falta a mi alma. A mis amigos, que siempre creyeron en mí y han mantenido esta amistad sólida y fuerte durante más de 10 años.

Santiago Arroyave Marín

Dedico este proyecto a Dios, mis padres, hermano, esposa, a mi hija, amigos, familiares. Ya que gracias a las experiencias que he vivido con cada uno de ellos soy un ser humano con valores y principios.

Luis Miguel Bueno Ramírez

RESUMEN

En este proyecto de grado se presenta el proceso de implementación de la comunicación inalámbrica, el desarrollo necesario para lograr un adecuado sistema de control de velocidad de rotación de las ruedas, tratamiento de datos provenientes de sensores y actuadores, sus lecturas confiables mediante softwares como Arduino y MatLab y la adición de características como almacenamiento de datos en tiempo real y su correcta visualización para que se permitan un eficiente y adecuado comportamiento de la plataforma en entornos dinámicos y aleatorios. Todo esto, para el proyecto de grado “P-MITO - Plataforma Móvil para la Implementación de Técnicas de Odometría” realizado por Marlon Andrey Fernández Mesa y Julio Cesar Yepes Valencia.

Índice

| | |
|---|-----------|
| 1. INTRODUCCIÓN | 1 |
| 1.1. Planteamiento del Problema | 2 |
| 1.2. Justificación | 3 |
| 1.3. Objetivos | 4 |
| 1.3.1. Objetivo General | 4 |
| 1.3.2. Objetivos Específicos | 4 |
| 1.4. Estado del Arte | 5 |
| 1.5. Principales resultados | 7 |
| 1.6. Estructura del trabajo de grado | 7 |
| 2. DESCRIPCIÓN DE LA PLATAFORMA MÓVIL P-MITO | 9 |
| 2.1. Estado de la plataforma móvil P-MITO al iniciar el proyecto. | 9 |
| 2.1.1. Sensores de distancia | 9 |
| 2.1.2. Encoder | 11 |
| 2.1.3. Brújula | 11 |
| 2.1.4. Batería | 12 |
| 2.1.5. Motor | 13 |
| 2.1.6. Módulo control motores | 13 |
| 2.1.7. Controlador Arduino Mega 2560 | 14 |
| 2.2. Comunicación Inalámbrica: | 15 |
| 2.3. Filtro de Kalman | 18 |
| 2.4. Evasión de obstáculos: | 19 |
| 2.5. Filtrado de datos atípicos (outliers) de una distribución de datos con Matlab | 20 |
| 3. DESARROLLO DE LA PLATAFORMA ROBÓTICA P-MITO | 21 |
| 3.1. Introducción al desarrollo de la plataforma | 21 |
| 3.2. Desarrollo e implementación de algoritmos en las tarjetas Arduino de la P-MITO | 22 |
| 3.2.1. Tarjeta inalámbrica de conexión a computador | 23 |
| 3.2.2. Tarjeta Maestro | 24 |
| 3.2.3. Tarjeta Esclavo | 25 |
| 3.3. Ajustes al algoritmo en MatLab para el control principal de la P-MITO. | 25 |
| 3.4. Ajustes y re configuraciones en las características constructivas de la P-MITO | 27 |
| 3.5. Descripción de la interfaz de usuario (GUIE de MatLab) | 28 |
| 3.5.1. Ingresar coordenadas del P-MITO | 28 |
| 3.5.2. Recuadro de velocidad lineal, angular y distancia recorrida | 29 |
| 3.5.3. Ángulo de dirección y botones principales | 29 |

| | | |
|-----------|--|-----------|
| 3.5.4. | Gráfico de recorrido de la plataforma P-MITO | 30 |
| 3.5.5. | Gráfico de velocidad de las ruedas | 31 |
| 3.5.6. | Datos en sensores | 32 |
| 4. | RESULTADOS | 33 |
| 4.1. | Errores presentes en el eje Y+ | 34 |
| 4.2. | Errores presentes en el eje Y- | 35 |
| 4.3. | Errores presentes en el eje X- | 37 |
| 4.4. | Comportamiento y lecturas de los sensores ultrasónicos | 38 |
| 5. | CONCLUSIONES | 43 |
| 6. | ANEXOS | 43 |
| 7. | BIBLIOGRAFÍA | 44 |

1. INTRODUCCIÓN

El proyecto denominado “Implementación del sistema de comunicación inalámbrica y generación de trayectorias a través de entornos dinámicos controlando variables x , y y Θ de la plataforma móvil **P-MITO** (Plataforma Móvil para la Implementación de Técnicas de Odometría)” pretende realizar la comunicación inalámbrica, el ajuste del diseño de control realizado previamente y la implementación de un algoritmo que permita a la plataforma móvil una correcta y adecuada respuesta ante la presencia de obstáculos aleatorios en su entorno de trabajo.

En este documento se presenta el desarrollo de los procesos implementados para la continuidad del proyecto, con el cual se pretende darle mayor autonomía a la plataforma móvil P-MITO. Figura 1.

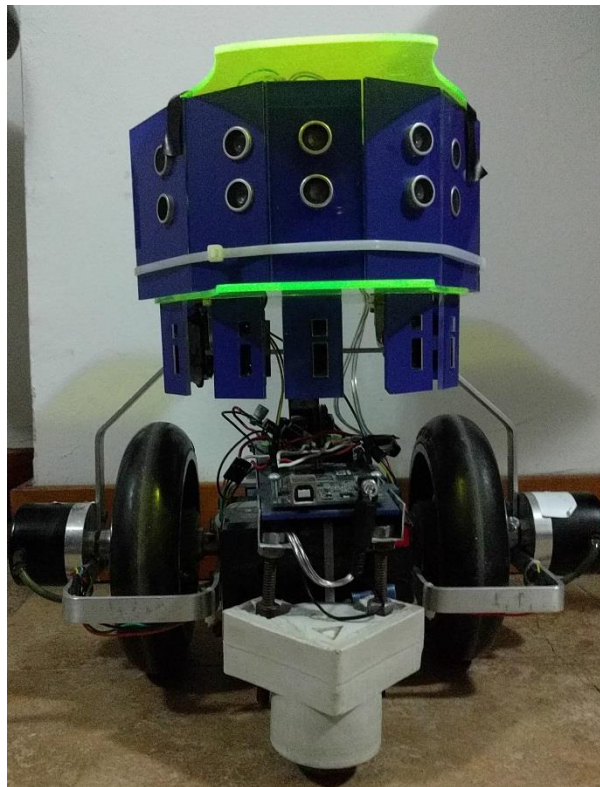


Figura 1. P-MITO, Plataforma Móvil para la Implementación de Técnicas de Odometría

1.1. Planteamiento del Problema

El evadir obstáculos y navegar por entornos dinámicos es uno de los objetivos de los robots móviles autónomos y sus desafíos continúan siendo de gran interés y desarrollo para lograr el mejor o continuo progreso del comportamiento de los robots.

Una de las cualidades de los robots móviles autónomos es la precisión con que estos perciben el entorno por donde se desplazan. De dicha precisión se puede garantizar una óptima interacción del robot móvil con su entorno de trabajo.

En la bibliografía se referencia robots móviles autónomos en entornos de trabajo transportando materiales y herramientas e interactuando con las personas para cumplir tareas específicas. Para la realización de dichas tareas se requiere la implementación de sofisticados sistemas, tanto de control como de instrumentación. [1] [2]

Con este proyecto de grado se implementará el sistema de comunicación inalámbrica entre la plataforma móvil y un PC que actuará como unidad central de proceso, para planificar y generar las trayectorias de navegación por diferentes entornos de trabajo.

El sistema de comunicación inalámbrico será el encargado de enviar a la plataforma móvil las coordenadas en X , Y y Θ y recibir en cada instante de tiempo de la plataforma móvil toda la información del entorno por el que se desplace a través de los datos captados por los diferentes módulos de sensores; Sensores odométricos, anillo de sensores ultrasónicos. [3] [4]

Como dicha comunicación inalámbrica se hará utilizando sensores X-Bee, la pregunta central del proyecto es: ¿Cuál es el mejor protocolo y configuración que permita lecturas e intercambio de datos de manera confiable, entre la plataforma P-MITO y computador mediante el software de Arduino y MatLab utilizando tarjeta Arduino y módulos XBee para lograr un apropiado comportamiento por entornos dinámicos?

1.2. Justificación

El uso de robots móviles está justificado en aplicaciones en las que se realizan tareas de riesgo para el ser humano. El transporte de material peligroso, las excavaciones mineras, la limpieza industrial o la inspección de plantas nucleares son ejemplos donde un robot móvil puede desarrollar su labor y evita exponer, la salud del trabajador.

También existen robots encargados de explorar en nuestro planeta, al interior de volcanes, en las profundidades del océano o cualquier otro lugar inaccesible por el hombre. [5]

Las características de las plataformas móviles, (cualquier dispositivo móvil que puede desplazarse por un entorno o región) empleadas para investigación en el área de la robótica, son la versatilidad y adaptabilidad. La versatilidad se entiende como la capacidad de lograr y ejecutar diferentes tareas, o ejecutar la misma tarea de distintas maneras.

La versatilidad se entiende como la capacidad de lograr y ejecutar diferentes tareas, o ejecutar la misma tarea de distintas maneras.

La adaptabilidad significa que después de implementada la parte sensorial y todo lo relacionado con la plataforma, esta deberá adaptarse y enfrentarse por sí sola ante un entorno.

Una plataforma de navegación debe estar en capacidad de responder ante ambientes estáticos y dinámicos, los ambientes estáticos son aquellos entornos donde los obstáculos permanecen fijos y los ambientes dinámicos son aquellos entornos donde los obstáculos pueden estar en movimiento. [6]

No basta con tener un excelente proceso de censado, es necesario además, contar con un sistema de control que sepa cómo manipular la información obtenida de los sensores, y que tome una decisión según su función, como por ejemplo recorrer una trayectoria mínima hasta llegar a un punto deseado o evitar los obstáculos existentes en el medio. [7]

Una correcta percepción de su entorno puede significar para el robot, un buen desempeño en sus funciones para lo cual es totalmente indispensable el uso de sistemas de cómputo de altas prestaciones(computadores, microcontroladores

o microprocesadores). Se puede decir que estos elementos son el cerebro de la plataforma. [8]

A esto se le suma que los datos que obtiene cada plataforma móvil y los datos de navegación, de alguna manera tienen que ser vistos y/o enviados por el ente de control, para su posterior análisis. Para esto se utilizan diferentes tipos de comunicación como la AM (modulación en amplitud) o la FM (modulación en frecuencia), las cuales permiten una buena comunicación entre diferentes puntos cuando la plataforma móvil se encuentra en operación.

Es importante que la comunicación entre la unidad central de proceso (computador) y la plataforma móvil sea una comunicación fluida y segura, y que se pueda adaptar a la dinámica de los entornos por los que la plataforma se desplaza. Dicha información debe proporcionar a la unidad de control la información necesaria para que ésta realice las acciones de control pertinentes que generen una navegación autónoma y confiable de la plataforma, no importando la dinámica del entorno por el que se desplace. Es por ello que la comunicación se debe dar en cada instante de tiempo.

1.3. Objetivos

1.3.1. Objetivo General

- Implementar la comunicación inalámbrica mediante dispositivos XBee, tarjeta de Arduino y software de MatLab, y algunos algoritmos de control a partir de la lectura del entorno, para la navegación por entornos dinámicos de la plataforma (P-MITO)

1.3.2. Objetivos Específicos

- Estudiar la bibliografía del estado del arte sobre proyectos implementados con comunicación inalámbrica, full Duplex, comunicación serial y técnicas de navegación autónoma por entornos estáticos y dinámicos.
- Implementar los protocolos de comunicación entre un PC y la plataforma móvil, desarrollando protocolos que mejor se adapten a los objetivos del proyecto.

- Desarrollar un algoritmo para la comunicación, que permita hacer una lectura confiable del entorno por el que cual se desplaza la plataforma. Estas lecturas de distancias, y ángulos de desplazamiento deben servir de base de información para el sistema de control.
- Verificar la correcta transmisión de datos entre la plataforma y el computador
- Ajustar el sistema de control de la velocidad de giro de los motores.
- Comprobar las variables K_p y K_i del sistema del control implementado para un correcto y adecuado funcionamiento del robot.
- Verificar el correcto desplazamiento de la plataforma móvil P-MITO en un ambiente aleatorio.
- Verificar el correcto funcionamiento del robot ante ambientes dinámicos y aleatorios.

1.4. Estado del Arte

La importancia de diseñar máquinas dotadas de un grado de autonomía que permita el desenvolvimiento natural en un ambiente dado y efectuar tareas, es la puerta a un mundo con un nivel tecnológico que hará más cómoda la existencia del ser humano.

A principios de los años sesenta se introducen en la industria, de modo significativo, los robots manipuladores como un elemento más del proceso productivo. Esta proliferación, motivada por la amplia gama de posibilidades que ofrecía, suscitó el interés de los investigadores para lograr manipuladores más rápidos, precisos y fáciles de programar. La consecuencia directa de este avance originó un nuevo paso en la automatización industrial, que flexibilizó la producción con el nacimiento de la noción de célula de fabricación robotizada.

Los trabajos desarrollados por los robots manipuladores consistían frecuentemente en tareas repetitivas, como la alimentación de las distintas máquinas componentes de la célula de fabricación robotizada. Ello exigía ubicarlas en el interior de un área accesible para el manipulador, caracterizada por la máxima extensión de sus articulaciones, lo cual podría resultar imposible a medida que la célula sufría progresivas ampliaciones. Una solución a este problema se logra al desarrollar un vehículo móvil sobre rieles para proporcionar un transporte eficaz de los materiales entre las distintas zonas de la cadena de

producción. De esta forma, aparecen los primeros vehículos guiados automáticamente (AGV's). Una mejora con respecto a su concepción inicial escriba en la sustitución de los raíles como referencia de guiado en la navegación por cables enterrados, reduciéndose con ello, los costes de instalación.

La posibilidad de estructurar el entorno industrial permite la navegación de vehículos con una capacidad sensorial y de razonamiento mínimas. De este modo, la tarea se estructura en una secuencia de acciones en la que a su término el vehículo supone que ha alcanzado el objetivo para el que está programado. [9]

Ante cualquier cambio inesperado en el área de trabajo que afecte el desarrollo normal de la navegación, el sistema de navegación del vehículo se encontrará imposibilitado para ejecutar acciones alternativas que le permitan reanudar su labor. Sin embargo, por sus potenciales aplicaciones fuera del ámbito industrial, donde resulta costoso o imposible estructurar el entorno, se les dotó, en la búsqueda de un vehículo de propósito general apto para desenvolverse en cualquier clase de ambiente, de un mayor grado de inteligencia y percepción. Una definición correcta de robot móvil plantea la capacidad de movimiento sobre entornos no estructurados, de los que se posee un conocimiento incierto, mediante la interpretación de la información suministrada a través de sus sensores y del estado actual del vehículo.

El uso de robots móviles está justificado en aplicaciones en las que se realizan tareas molestas o arriesgadas para el trabajador humano. Entre ellas, el transporte de material peligroso, las excavaciones mineras, la limpieza industrial o la inspección de plantas nucleares son ejemplos donde un robot móvil puede desarrollar su labor y evita exponer, gratuitamente, la salud del trabajador. Otro grupo de aplicaciones donde este tipo de robots complementa la actuación del operador lo componen las labores de vigilancia, de inspección o asistencia a personas discapacitadas. Asimismo, en aplicaciones de tele operación, donde existe un retraso sensible en las comunicaciones, resulta interesante el uso de vehículos con cierto grado de autonomía. [2]

El robot móvil autónomo se caracteriza por una conexión inteligente entre las operaciones de percepción y acción, que define su comportamiento y le permite llegar a la consecución de los objetivos programados sobre entornos con cierta incertidumbre. El grado de autonomía depende en gran medida de la facultad del robot para abstraer el entorno y convertir la información obtenida en órdenes, de tal modo que, aplicadas sobre los actuadores del sistema de locomoción, garantice la realización eficaz de su tarea. De este modo, las dos grandes

características que lo alejan de cualquier otro tipo de vehículo se relacionan a continuación:

- Percepción: Determina la relación del robot con su entorno de trabajo, mediante el uso de los sensores.
- Razonamiento: Determina las acciones que se han de realizar en cada momento, según el estado del robot y su entorno, para alcanzar las metas asignadas. [5]

Ahora bien, el uso de la electrónica digital va de la mano con la robótica, el hardware libre ARDUINO “está revolucionando la manera en que los diseñadores de sistemas implementan lógica digital. Se reduce radicalmente los costos y el tiempo de desarrollo para implementar miles de compuertas lógicas”. [5]

Para la comunicación de Robot o plataformas móviles, se utilizan diferentes tipos de comunicación entre usuario y máquina. Una de gran utilidad es el dispositivo X-bee que se usa para comunicaciones en transmisión de datos. [10]

1.5. Principales resultados

Los principales resultados de este proyecto de grado son:

- Un algoritmo de transmisión de datos de manera inalámbrica entre la plataforma móvil P-MITO y el computador.
- La correcta visualización de la transmisión de datos entre la plataforma y el computador.
- El sistema de control para la velocidad de rotación de las ruedas de la plataforma móvil P-MITO.
- Almacenamiento de los datos generados por la plataforma durante la ejecución de una trayectoria.

1.6. Estructura del trabajo de grado

Con el fin de responder al carácter investigativo del proyecto, alcanzar los objetivos y comprobar la veracidad del planteamiento del mismo, se tiene en

cuenta una metodología teórico - experimental implementada en cada uno de las etapas del proyecto, además de la utilización de la simulación y experimentación para detectar y resolver problemas técnicos que puedan dificultar el logro de los objetivos.

El proyecto de investigación se llevará a cabo en fases que delimitan y especifican cada una de las actividades a desarrollar.

- Fase I “Estudio de la técnica a implementar”. En esta etapa se realizó una recopilación de información. Para dicho proceso se estudió material académico, libros y artículos especializados para lograr obtener un correcto funcionamiento del robot.
- Fase II “Implementación de técnicas”. En esta etapa se diseñaron e implementaron técnicas necesarias para cumplir los objetivos propuestos. Al tener claro los conceptos teóricos de la fase I. Se resolvió el problema y se generaron distintas combinaciones de las técnicas propuestas.
- Fase III “Implementación de la comunicación inalámbrica”. En esta fase se diseñó e implementó un método para lograr la comunicación entre la plataforma móvil P-mito y un computador. Este método de comunicación se adaptó a los objetivos del proyecto.
- Fase IV “Ajuste del sistema de control” En esta fase se ajustó el método usado en el sistema de control de la velocidad de giro de los motores, teniendo en cuenta que el movimiento de la plataforma móvil P-mito debe ser limpio y suave.
- Fase V: “Corrección del algoritmo de censado de obstáculos” En esta fase se ajustó el algoritmo utilizado para el censado y detección de obstáculos en un ambiente aleatorio.
- Fase VI: “Pruebas en la plataforma móvil”. En esta etapa se usó la plataforma móvil en ambientes aleatorios para verificar todas las fases del proyecto, tales como el control, la comunicación y trayectorias de desplazamiento; se utilizó una interfaz gráfica entre en Software Matlab y el software Arduino para tomar datos del entorno y definir la posición y orientación de la plataforma móvil, además, se realizaron pruebas para verificar el error del robot con respecto a una trayectoria dada.
- Fase VII: “Presentación de resultados”. En esta última etapa del proyecto se presentará un documento final a la comunidad en general con todos los datos, conclusiones y resultados obtenidos. Esto con el fin de brindar nuevos conocimientos y dejar abiertas las posibilidades de un perfeccionamiento o continuación del proyecto.

2. DESCRIPCIÓN DE LA PLATAFORMA MÓVIL P-MITO

2.1. Estado de la plataforma móvil P-MITO al iniciar el proyecto.

Como se mencionó con anterioridad, este trabajo se lleva a cabo partiendo del trabajo de grado desarrollado por los Ingenieros Marlon Andrey Fernández y Julio Cesar Yepes Valencia. Los próximos párrafos ayudarán a ver el estado en el que se encontraba la plataforma móvil P-mito. [4]

2.1.1. Sensores de distancia

- La plataforma cuenta con Sensores ultrasónicos (HC-SR04) para medir distancias entre 2cm a 400cm (distancias largas de obstáculos), sensores infrarrojos Sharp (2Y0A21) para medir distancias entre 10 a 80 cm (distancias cortas de obstáculos), encoders como medidores de distancia recorrida por las ruedas (sensores odómetros) Y una brújula digital, como sensor de dirección y rumbo. [11] [12]

A continuación, se muestran imágenes de los sensores infrarrojos y de ultrasonido instalados en el robot:



Figura 2. HC-SR04 Sensor Ultrasónico.



Figura 3. Sensor Infrarrojo Sharp (2Y0A21).

Se ubica un anillo de 8 sensores de ultrasonido en la parte superior de la plataforma (ver figura 4). Estos sensores se utilizaron con la finalidad de detectar los obstáculos que se encuentren alrededor del robot, también incluye un anillo de 8 sensores infrarrojos situados debajo de los anillos de sensores ultrasónicos, estos dispositivos se usan para detectar objetos que se encuentren a una distancia de 0 a 20 cm.

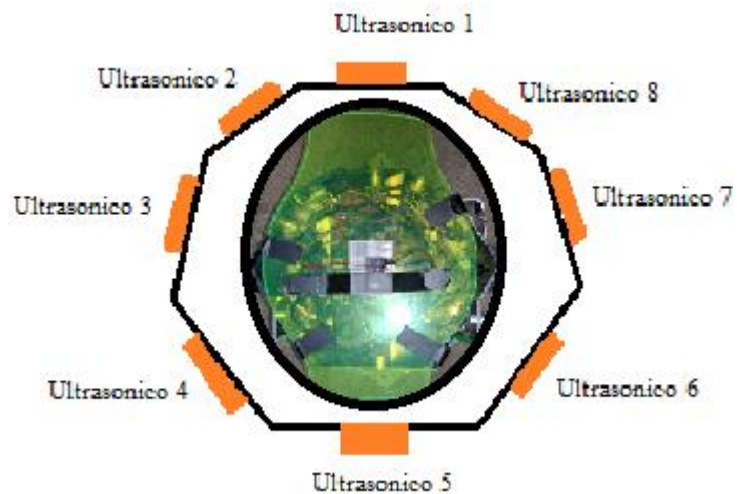


Figura 4. Posición de los sensores ultrasónicos en el robot.

2.1.2. Encoder

- Para la medición de la distancia recorrida por el robot, se tienen instalados Encoders Stegmann, los cuales son transductores rotativos que transforman un movimiento angular en una serie de pulsos digitales. Estos pulsos generados pueden ser utilizados para controlar los desplazamientos de tipo angular o lineal, siendo estos quienes permiten la medición de la distancia recorrida por el robot. El encoder instalado de muestra en la figura 5.

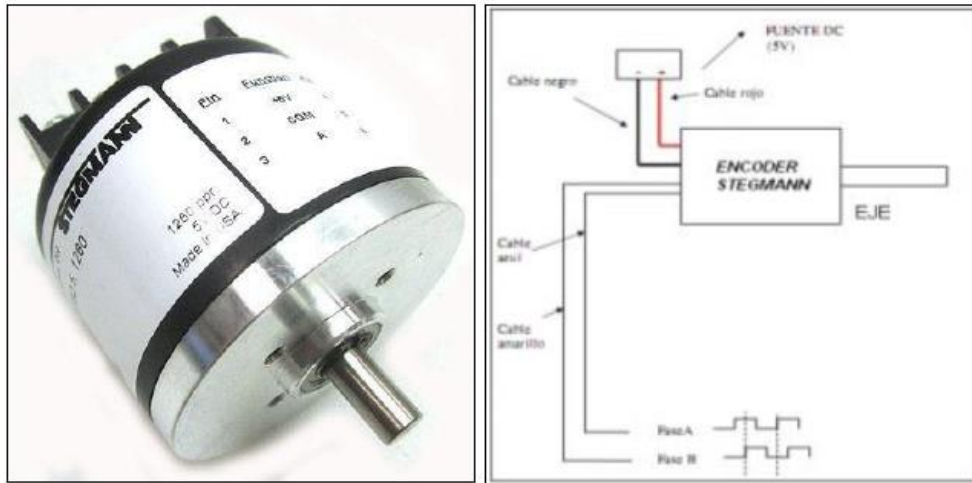


Figura 5. Encoders stegmann.

2.1.3. Brújula

- También se tiene instalada en la parte superior una brújula digital HCM5883L la cual permite detectar campos magnéticos terrestres. Mediante los campos magnéticos, la brújula es capaz de detectar y entregar datos de ubicación del Sur magnético y por consiguiente saber dónde está el Norte el Este y el Oeste, como también la ubicación geográfica de la misma. [13]

La brújula instalada en el robot se muestra en la figura 6.

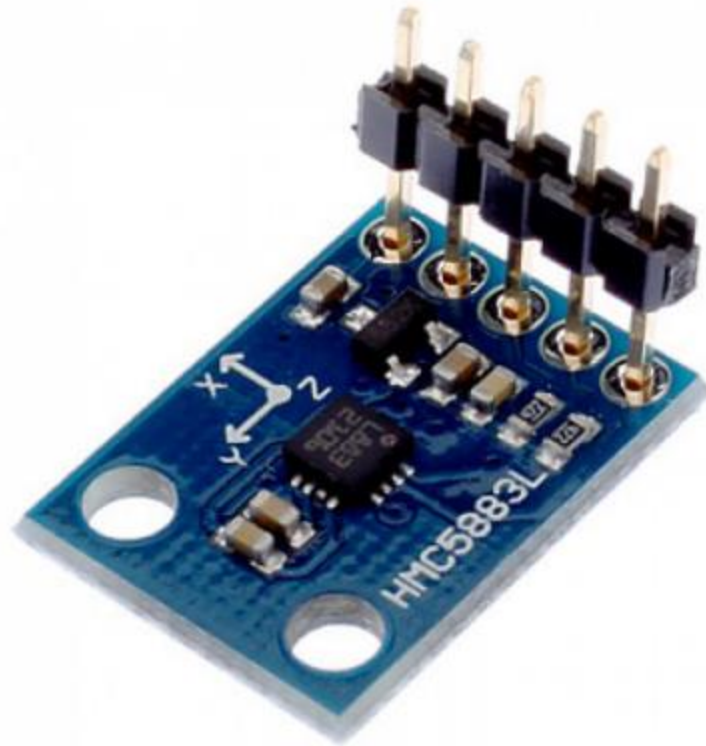


Figura 6. Brújula Digital HCM5883L

2.1.4. Batería

- Instalado en la parte inferior, se encuentra una batería “Tb-Plus TB12-1.3”, la cual proporciona 12v con capacidad de 1.3 AH, permitiendo la alimentación con corriente directa de todos los elementos y sensores puestos en el robot. [14]

La batería instalada en el robot se muestra en la figura 7.



Figura 7. Batería Tb-Plus TB12-1.3.

2.1.5. Motor

- Para el movimiento de las ruedas cuenta con unos motores DC de marca Dayton, estos son de gran utilidad cuando se requiere controlar velocidades y además entrega un par relativamente grande a velocidades reducidas. Este tipo de motores son alimentados a 12v, los cuales desarrollan una velocidad máxima de 18 metros por minutos y pueden cambiar la dirección de giro invirtiendo su polaridad. El motor instalado en el robot se muestra en la figura 8.



Figura 8. Motor DC Dayton.

2.1.6. Módulo control motores

- Cuenta con un módulo para el manejo de los motores DC, tanto para su velocidad como para su dirección. Estos son controlados por medio del Driver L298N que permite el control PWM de los dos motores hasta 2 amperios. Este módulo es conectado a una batería de 12v y a los 5v del Arduino Mega. El módulo de control de los motores se muestra en la figura 9.

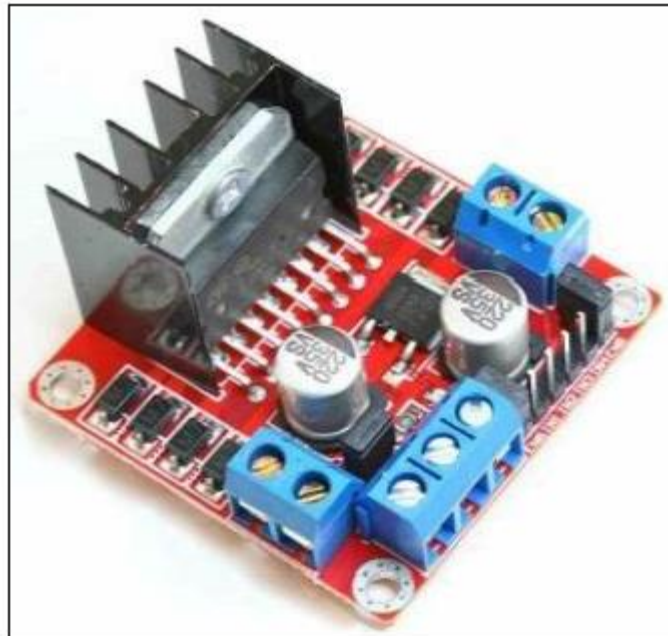


Figura 9. Driver motor L298n.

2.1.7. Controlador Arduino Mega 2560

- Para el procesamiento de los datos, el robot cuenta con un controlador Arduino Mega 2560, encargada del procesamiento de los datos de los sensores, así como la velocidad de giro de las ruedas y el envío de los datos obtenidos al computador que le controla. [15]

En la figura 10, se puede ver la imagen de la tarjeta instalada.

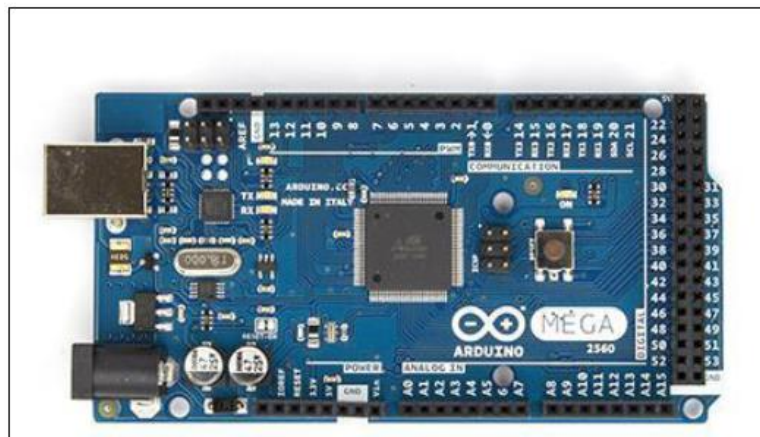


Figura 10. Arduino mega2560.

Los párrafos anteriores, fueron escritos con la finalidad de que el lector tenga presente el estado en el cual, se encontraba la plataforma móvil P-mito antes de iniciar este proyecto de grado.

2.2. Comunicación Inalámbrica:

De los principales objetivos de este proyecto, fue que la plataforma móvil P-mito pasara de una comunicación USB (Universal Serial Bus) a una comunicación inalámbrica Full Duplex, debido a que el control remoto inalámbrico de un robot móvil representa aplicaciones como la exploración de terrenos al aire libre, en interiores, entre otro tipo de aplicaciones.

La tecnología inalámbrica Full Duplex, se ha convertido en una de las nuevas soluciones en el campo de los sistemas inalámbricos para la comunicación. Esta tecnología es atractiva porque permite duplicar el ancho de banda sin expandir el ancho de banda inicial, sin embargo, hay problemas que deben abordarse. En la tecnología Full Duplex, la transmisión de señales se realiza en el mismo ancho de banda de frecuencia simultáneamente generando en algunas ocasiones problemas de ruido por interferencia. [16]

La comunicación Full dúplex genera una mejora en el rendimiento del sistema sobre la comunicación semidúplex convencional al permitir la transmisión y recepción simultáneamente por una misma banda de frecuencia. Sin embargo, dicha comunicación produce una fuerte autointerferencia ya que por el mismo ancho de banda se están ingresando dos señales, lo cual genera que la banda se divida y el camino que usa el envío y la recepción se disminuye pudiendo interferir en una señal en el camino de la otra. [17]

Los robots de exploración de terreno pueden operar en dos regímenes principales: autónomo o mediante control remoto (manual). Para ambos casos se necesita de comunicación inalámbrica para que los robots estén enviando datos de los terrenos que están explorando. En caso de régimen autónomo, los

robots deben tener varios sensores o dispositivos que permitan la detección de obstáculos (objetos) y algún tipo algoritmo para generar alguna ruta alterna que pueda evadir los obstáculos. En caso de modo de operación del controlador remoto, también necesita sensores para la detección de obstáculos, pero la decisión para la evadirlos la realiza el operador desde el control remoto. Hay otra posibilidad de operación, llamado modo semiautónomo, en el cual el robot opera de forma remota utilizando una aplicación con algunos parámetros predefinidos con respecto a las trayectorias y comportamiento y las decisiones se pueden tomar de manera autónoma o siendo el operador quien la defina. [18]

Existen muchos dispositivos en el mercado para realizar comunicaciones de dispositivos electrónicos, siendo los XBee's de muy buen precio y excelente utilidad y eficiencia, estos son pequeñísimos chips azules capaces de comunicarse de forma inalámbrica unos con otros. Pueden hacer cosas simples, como reemplazar un par de cables en una comunicación serial a distancias muy limitadas, a una comunicación a distancias considerables, lo cual es de gran utilidad cuando se desean desarrollar, por ejemplo, vehículos radio-controlados o robots de navegación autónoma. [19]

Para este proyecto, el control remoto inalámbrico de la P-MITO, se utilizan los módulos inalámbricos X-bee que operan sobre banda ISM (Industrial, Científica y Médica) (2.4GHz u 868MHz).

Los más comunes son el módulo S2 (2mW) con un rango de 120m y velocidad de datos de 250Kbps y S2B PRO (50mW) con un rango de 1.6Km. Se usaron módulos X-bee Pro s2b, que utiliza el protocolo de comunicación ZigBee. Los dos módulos X-bee están configurados uno como coordinador de nodo y el otro como enrutador. [18]

La Serie 2B trae mejoras respecto a la potencia de salida y protocolo de datos de los Pro Serie 2. Los módulos de Serie 2B le permiten crear complejas redes en malla basadas en el firmware de malla del XBee ZB ZigBee. Estos módulos permiten una comunicación muy fiable y simple entre microcontroladores, computadores, sistemas, ¡prácticamente cualquier cosa con un puerto serial! Soporta redes punto a punto y multi-punto.

Los módulos de Serie 2B usan una configuración de hardware diferente de los modelos anteriores, y si bien puede comunicarse con los módulos XBee Serie 2, estos no son compatibles con los módulos antiguos de Serie 2.5. [19]

Funcionalidades del módulo X-bee Pro s2b:

- 3.3V @ 295mA
- 250Kbps máxima tasa de datos
- Salida de 63mW (+17dBm)
- Rango de 1 milla (1.6 km)
- Antena alámbrica
- Certificación FCC
- 6 pines entrada CAD 10-bit
- 8 pines E/S digitales
- Encriptación 128-bits
- Configuración local o inalámbrica
- Set de comandos AT o API

Estos módulos fueron instalados en dos tarjetas de Arduino MEGA2560, uno para conectar en el robot y otro para conectar en la tarjeta que recibe los datos en el computador. En la figura 11 se puede observar la imagen de los módulos instalados.



Figura 11. Modulo XBee PRO S2B.

Las tarjetas de Arduino MEGA 2560 cuentan con pines para establecer una comunicación serial entre tarjetas o tarjetas y computador. La interfaz de conexión entre las tarjetas y los módulos XBEE (Figura 12), tienen una conexión directa a los pines PWM 0(RX) y 1(TX), los cuales internamente tienen una conexión a la comunicación USB entre tarjeta y computador. Esto generaba conflictos a la hora de enviar los datos desde el computador al robot y viceversa, ya que los datos de entrada y salida al momento de verificarlos presentaban incoherencias o se presentaban datos totalmente diferentes a los esperados. Para solucionar esto, fue necesario modificar la interfaz de conexión, anulando la conexión de la interfaz a los pines 0(RX) y 1(TX) y llevando estos a una conexión a los pines 18(TX1) y 19(RX1). [20]

Después de hacer esta modificación, se pudo tener una recepción y envío de datos totalmente limpia y sin problemas. Generando una comunicación apropiada entre robot y computador.



Figura 12. XBee shield V03.

2.3. Filtro de Kalman

En el transcurso de la ejecución del presente proyecto, se observaron datos atípicos con bastante frecuencia provenientes de los sensores ultrasónicos, principalmente del sensor frontal y, al momento de tomar decisiones con ellos, estos generaban un comportamiento ilógico en el robot, haciendo que este se

comportara de manera intermitente y con cambios bruscos en su trayectoria principal.

Es de gran interés el filtro Kalman, por la amplia variedad de aplicaciones que proporciona. Es iterativo y considera una serie de mediciones (que contiene ruido estadístico) a lo largo del tiempo, y luego estima la variable desconocida. La aplicación más común es la lectura de mapas y la regulación de los sistemas vehiculares. Además, el principio de filtrado de Kalman se aplica ampliamente en el campo de la econometría y el procesamiento de señales. El algoritmo implica dos pasos principales, a saber, la predicción y la actualización.

En el paso de predicción, el filtro calcula la variable de estado actual. La estimación se hace considerando las incertidumbres del estado. La actualización de la variable estimada se calcula utilizando un promedio ponderado, cuando se destaca el resultado de la medición siguiente.

El filtro se ejecuta en tiempo real, dependiendo únicamente de las mediciones de la entrada actual y el estado anterior o estimado previamente. No se requiere información pasada. Típicamente, el filtrado se aplica de manera repetida, utilizando el resultado anterior de la predicción fusionada como la entrada del nuevo ciclo. El algoritmo tiene la ventaja de ser liviano en la memoria, de que no necesitan mantener ningún historial que no sea el estado anterior. Además, el algoritmo de filtro de Kalman es muy rápido, lo que los hace adecuados para aplicaciones en tiempo real. [21]

Finalmente se implementó el código del filtro de Kalman mediante la librería disponible para Arduino "SimpleKalmanFilter" en la tarjeta principal (Maestro). [22] Este filtro se implementó de tal forma que, al tomar un dato del sensor, este es ingresado al filtro, y este devuelve un posible futuro dato, el cual con un nuevo dato proveniente del sensor es promediado para obtener un valor lo más aproximado al real. Este nuevo algoritmo permitió una mejora en el robot, evitando que este tuviera frenados y cambios bruscos en su trayectoria haciendo que se comportara de una forma aceptable en los diferentes ambientes.

2.4. Evasión de obstáculos:

Los sistemas de evasión de obstáculos se han utilizado en una amplia gama de diferentes áreas de robótica y han tenido un éxito extraordinario al minimizar el riesgo de colisiones. Los sistemas de evasión de obstáculos se aplican principalmente en sistemas de transporte como control de tráfico de aeronaves, vehículos autónomos y vehículos submarinos, etc. La prevención de colisiones

es un requisito crítico en la construcción de sistemas de robots móviles donde todos presentan algún tipo de técnicas de detección de obstáculos para evitar dos o más objetos de colisión (dos robots móviles o un robot con un obstáculo).

El objetivo principal de evitar obstáculos es obtener una trayectoria libre de colisiones desde el punto de partida hasta el objetivo en entornos de monitoreo. Los obstáculos se pueden dividir en dos tipos, estáticos donde el obstáculo está predefinido y tiene una posición fija y un obstáculo dinámico donde la posición no es reconocida y tiene patrones de movimiento inciertos (objetos en movimiento). La detección de obstáculos dinámicos es más exigente que la detección de obstáculos estáticos ya que la dinámica tiene una dirección variable y requiere una predicción de la posición dinámica del obstáculo en cada paso para lograr el requisito de una planificación de trayectoria crítica en el tiempo.

El algoritmo de obstáculo dinámico es mucho más complejo, ya que no solo detecta un objeto, sino que también considera algún tipo de medida con respecto a las dimensiones del obstáculo en movimiento y la distancia entre el robot móvil y el objeto. Tan pronto como se hayan determinado estas medidas, el algoritmo de detección de obstáculos debe guiar al robot alrededor del objeto y continuar hacia el objetivo. Sin embargo, estos procedimientos continuos deben realizarse mientras el robot se mueve hacia el objetivo original.

La posición de un obstáculo se puede detectar a través de una medición proveniente de diferentes módulos, como cámaras y GPS integrados con una variedad de sensores, como sensores ultrasónicos y sensores de infrarrojos. Basado en la fusión de estas lecturas, el robot puede tomar la decisión de detectar y prevenir colisiones mientras se mueve a lo largo de la ruta. [23]

2.5.Filtrado de datos atípicos (outliers) de una distribución de datos con Matlab

En las bases de datos y el análisis de imágenes, la detección de valores atípicos es un problema de investigación importante que tiene como objetivo encontrar datos que son considerablemente diferentes, excepcionales e inconsistentes con respecto a la mayoría de los puntos de datos. En muchos casos, se encuentra que los valores atípicos son más significativos y útiles que los datos típicos o esperados. Por lo tanto, la detección de valores atípicos se ha convertido en una tecnología muy útil para una amplia gama de aplicaciones prácticas en la industria, negocios, seguridad e ingeniería, etc. Por ejemplo, la detección de valores atípicos puede ayudar a identificar transacciones fraudulentas sospechosas para compañías de tarjetas de crédito y también puede utilizarse

para identificar señales cerebrales anormales que pueden indicar el desarrollo temprano de cánceres cerebrales. [24]

Es importante garantizar que las series de datos en el tiempo sean confiables y libres de valores atípicos, datos faltantes y ruido excesivo. Además, aunque los desarrollos recientes de dispositivos (sensores / actuadores) de bajo costo con redes han llevado a las Wireless Sensor Networks (WSN) a un alto nivel de aplicaciones potenciales para el muestreo de datos de series de datos en el tiempo, tales datos suelen ser notorios, corruptos y afectados por varios valores atípicos. [25]

Fue necesaria la implementación de un código de OULTIERS, este código fue implementado de acuerdo con un ejemplo presentado por Vicente Reyes. [26] Se implementó este código ya que, en los datos tomados de la distancia, se presentaban muchos datos atípicos. Esto se debía a que en el procesamiento se obtenían datos que cambiaban bruscamente con respecto al verdadero recorrido del robot y se veían datos que llegaban con una diferencia de hasta 10 metros con respecto al verdadero desplazamiento, produciendo que el recorrido para distancias cortas se tomara como concluido. En breve, se implementaron los Outliers en los códigos del robot, de tal forma que se almacenaran 30 datos en un vector para luego pasar por el proceso de eliminación de datos atípicos quedando solo los datos verdaderos o confiables, disminuyendo así el error que estos generaban.

El código que se implementó se puede observar en el Anexo

3. DESARROLLO DE LA PLATAFORMA ROBÓTICA P-MITO

3.1. Introducción al desarrollo de la plataforma

A partir de la documentación sobre comunicación inalámbrica y manejo de software Arduino y MatLab, se logró la implementación de algoritmos de control que permitieron un adecuado comportamiento de la Plataforma Móvil P-MITO en ambientes aleatorios, a partir del procesamiento de datos obtenidos del entorno por el que se desplaza dicha plataforma.

En los siguientes capítulos se mostrará todo el desarrollo del robot, desde el momento en que este fue recibido por parte de los ingenieros Marlon Andrey Fernández y Julio Cesar Yepes, donde previamente se mostró el estado en que se encontraba, y todos los ajustes que fueron necesarios para lograr que la

plataforma se comportara de manera adecuada y los datos que se intercambiaban fueran los más reales posible.

3.2. Desarrollo e implementación de algoritmos en las tarjetas Arduino de la P-MITO

Se inició con el entendimiento y análisis de los códigos cargados inicialmente en la plataforma, cuáles eran sus funciones y cuál era la función de cada tarjeta.

La plataforma cuenta con dos tarjetas Arduino Mega 2560, [15] configuradas en modo de comunicación Maestro-Esclavo, en donde la tarjeta configurada en modo Esclavo hace sus propios procesos y, los datos y/o procesos que ella ha tenido solo son enviados cuando la tarjeta configurada en modo Maestro le solicita los datos. [27]

La tarjeta configurada en modo Esclavo en la plataforma cumplía las funciones de hacer el muestreo de los pulsos enviados por los Encoders, para ésta tomar datos de la distancia recorrida por las ruedas, además, es quien se encarga de realizar el sistema de control para la velocidad de las ruedas y así darle la indicación a las mismas de la velocidad que deben llevar de acuerdo con la dirección que sea impuesta por quien este controlando el Robot y la ubicación enviada por la brújula desde la tarjeta Maestro. Otro de los procesos que tenía a cargo esta tarjeta, son los de parar el movimiento de las ruedas si éste presentaba objetos a menos de 5cm de distancia. [4]

La tarjeta configurada en modo Maestro es quien se encargaba de tomar los datos de distancia medidos por los sensores ultrasónicos y los sensores infrarrojos. Esta también tenía a cargo los datos de ubicación del robot dados por la brújula instalada en la parte superior que son enviados tanto al computador que controla el robot como a la tarjeta configurada en modo Esclavo. [4]

Los códigos que inicialmente se presentaron en la tesis de los Ingenieros Marlon Andrey Fernández y Julio Cesar Yépez, se fueron modificando de forma progresiva de acuerdo con las necesidades que se iban presentando por las implementaciones y desarrollos con el fin de llevar a cabo los objetivos propuestos.

Para poder tener una comunicación inalámbrica, fue necesaria la adición de una tercera tarjeta de Arduino, la cual siempre va a estar conectada al computador que este controlando el robot, que permite enviar y recibir todos los datos del computador y plataforma.

3.2.1. Tarjeta inalámbrica de conexión a computador

Esta tarjeta contiene el código de recepción de datos desde el robot y el envío de datos desde el computador a la plataforma. La tarjeta contiene como hardware la interfaz de conexión entre Xbee y Arduino y la Xbee PRO S2B. cómo se puede ver en la figura 13.

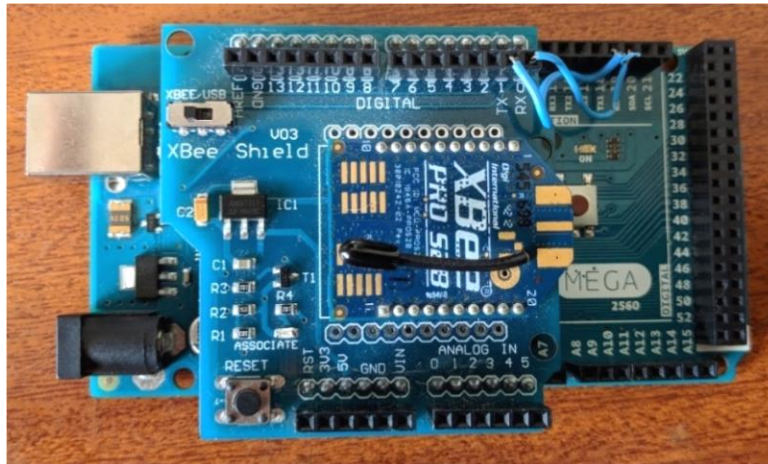


Figura 13. Tarjeta inalámbrica Arduino de conexión a PC.

En la figura 11, se puede apreciar la modificación mencionada en la sección 2.2.

Los datos que se reciben desde el robot son:

- DistA: Distancia de la rueda derecha.
- DistB: Distancia de la rueda izquierda.
- rpmA: Velocidad de la rueda derecha.
- rpmB: Velocidad de la rueda izquierda.
- Alpha: Angulo de posición del robot medido por la brújula.
- Cm1: Distancia medida por el sensor infrarrojo 1.
- Cm2: Distancia medida por el sensor infrarrojo 2.
- Cm3: Distancia medida por el sensor infrarrojo 3.
- Cm4: Distancia medida por el sensor infrarrojo 4.
- Cm5: Distancia medida por el sensor infrarrojo 5.
- Cm6: Distancia medida por el sensor infrarrojo 6.
- Cm7: Distancia medida por el sensor infrarrojo 7.
- Cm8: Distancia medida por el sensor infrarrojo 8.
- Distancia1: Distancia medida por el sensor ultrasónico 1.
- Distancia2: Distancia medida por el sensor ultrasónico 2.
- Distancia3: Distancia medida por el sensor ultrasónico 3.
- Distancia4: Distancia medida por el sensor ultrasónico 4.

- Distancia5: Distancia medida por el sensor ultrasónico 5.
- Distancia6: Distancia medida por el sensor ultrasónico 6.
- Distancia7: Distancia medida por el sensor ultrasónico 7.
- Distancia8: Distancia medida por el sensor ultrasónico 8.

El único dato que se envía al robot es el ángulo Theta generado por la posición final a la que se desea que el robot se dirija.

El código de la tarjeta se puede ver en el Anexo 3.

3.2.2. Tarjeta Maestro

El código de la tarjeta Maestro fue modificado completamente ya que, al momento de implementar la comunicación inalámbrica, este presentaba bastantes problemas con la medición de las variables. Además, al momento de recibir el robot, este presentaba fallas en las medidas de los sensores infrarrojos, por lo cual fueron retiradas las líneas de código de los sensores infrarrojos para obtener menos procesos y un menor grado de error en el nuevo algoritmo desarrollado.

En el anexo 2, se presenta la comparación entre los códigos anteriores y los nuevos códigos escritos para el procesamiento de las señales en la tarjeta Maestro, entre los procesos más destacados de esta tarjeta tenemos:

- El controla de los datos del anillo principal de sensores ultrasónicos, ver figura 3, y se hace el envío de los datos de estos tanto al esclavo como a la tarjeta inalámbrica conectada al computador.
- Tiene la función de obtener la medida de ubicación dada por la brújula, la procesa y la envía tanto a la tarjeta esclavo como a la tarjeta inalámbrica que se conecta al computador.
- Es quien le solicita los datos de distancia y velocidad de las ruedas para ser enviados a la tarjeta inalámbrica y luego al computador que le controla.

Las líneas de código de los sensores ultrasónicos fueron modificadas, ya que estos, según el DataSheet [11], pueden tomar medidas hasta de 4m y para los códigos estaban cargados inicialmente estos solo alcanzaban 25cm, por lo que luego de revisar la programación de los sensores, esta se cambió y se obtuvieron medidas hasta de un metro, siendo estas aceptables y funcionales para los requerimientos del robot. [28]

3.2.3. Tarjeta Esclavo

Esta puede ser la tarjeta más importante, ya que esta se encarga de los procesos de medición de la velocidad de las ruedas, básicamente, es quien permite que el robot se mueva.

A esta tarjeta se le cambiaron algunas líneas de código con el fin de que esta le permitiera girar al robot en su propio eje antes de empezar a caminar hacia la dirección que se le ordena, también se le cambio la sentencia de frenado por obstáculo, siendo esta cambiada por una condición de que este frena si el ángulo Theta enviado desde el computador es igual cero. Ya es el computador quien decide cuando el robot debe frenar y si hay algún tipo de objeto al frente, el computador hace a Theta igual a cero y se lo envía al robot en donde la tarjeta Esclavo se encarga de hacer cero la velocidad de las ruedas.

En el anexo 1, se puede observar el código que fue cargado en la tarjeta Esclavo.

3.3. Ajustes al algoritmo en MatLab para el control principal de la P-MITO.

Los ajustes al algoritmo planteado inicialmente por los Ingenieros Marlon Andrey Fernández y Julio Cesar Yepes fue el más tedioso. Acá se vio la necesidad de cambiar muchas de las configuraciones iniciales del robot.

En un principio los procesos asignados a MatLab era de tomar las lecturas planteadas en el capítulo 3.2.1. Allí MatLab se encargaba de promediar las velocidades de las ruedas, calcular la distancia recorrida por el robot y en una GUIE (Ver figura 14) mostrar estos datos, además de un gráfico en tiempo real donde se ve el recorrido de la plataforma, las medidas de los sensores ultrasónicos e infrarrojos y otro gráfico en tiempo real de la velocidad promedio de las ruedas.

El proceso estaba configurado de tal forma que los cálculos y gráficos de actualizaban cada 30 datos enviados por la plataforma.

Para los nuevos desarrollos, fue necesario que los datos no fueran adquiridos cada 30 veces, sino que estos debían estar listos para actuar uno a uno.

Esencialmente para poder generar la parada del robot en caso de presentarse algún tipo de objeto alrededor del robot. También, los datos de distancia del robot debían almacenarse en tiempo real en vectores para poder realizar el proceso de Outliers y evitar datos atípicos en la distancia recorrida por el robot.

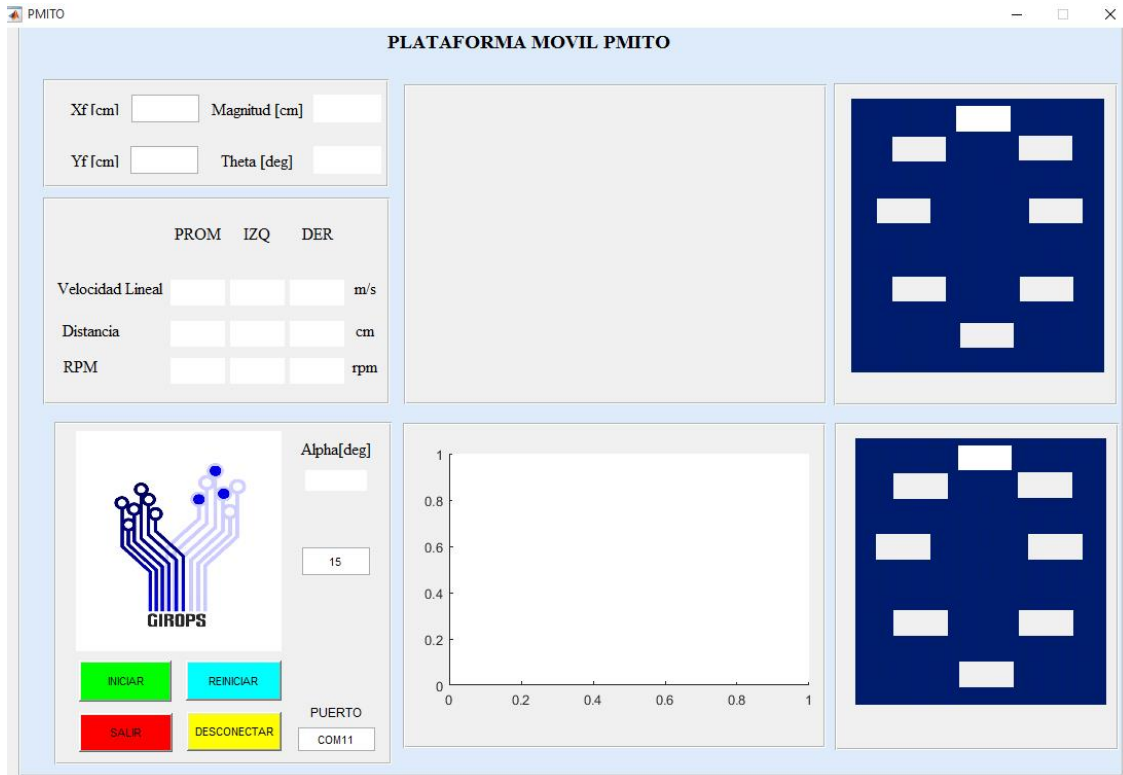


Figura 14. Guie de MatLab para el control de la P-MITO. [4]

En el anexo 4, se puede observar el código en MatLab desarrollado para la nueva fase de la plataforma móvil P-MITO, en donde se pueden resumir los cambios al código en los siguientes:

- Se le ingresaron líneas de código que se encargan de poner a Theta en cero en caso de tener algún obstáculo a 15cm o menos del robot.
- Se le ingresaron líneas de código que permitieron al robot hacer Outliers en el filtrado de datos atípicos en la distancia recorrida por el robot y evitar que el robot frene antes de haber recorrido la distancia ingresada.
- Fue necesario que los datos de los sensores fueran leídos en tiempo real para poder tomar decisiones cuando el robot presentaba objetos a menos de 15cm.
- Se le agregó la capacidad de almacenar los datos en Excel para su posterior trabajo con ellos después de algún recorrido ejecutado.

Los demás procesos no fueron modificados y se mantuvieron igual a como se presentaron en la tesis de grado de los ingenieros Marlon Andrey Fernández y Julio Cesar Yepes. [4]

3.4. Ajustes y re configuraciones en las características constructivas de la P-MITO

En sus características contractivas el robot no sufrió muchas transformaciones, a parte de la instalación de la tarjeta inalámbrica.

Su principal modificación se dio en la ubicación de los sensores ultrasónicos 4 y 6 los cuales presentaban cierta curvatura presentada en la figura 4, estos fueron puestos en forma paralela a las ruedas con el fin de detectar objetos que estén paralelos al movimiento del robot.

En las figuras 15 y 16 se puede observar la nueva ubicación de los sensores con el fin de tener medidas más certeras de los objetos que están alrededor de la plataforma.

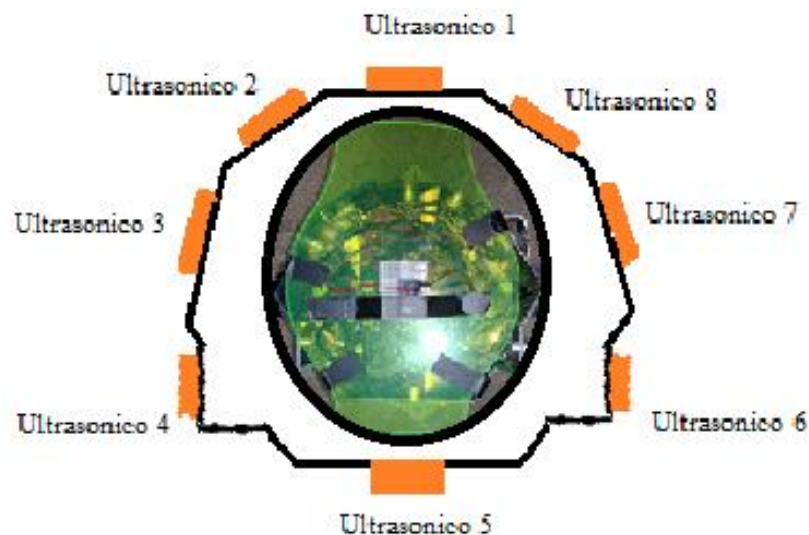


Figura 15. Nueva distribución de los sensores ultrasónicos 4 y 6.

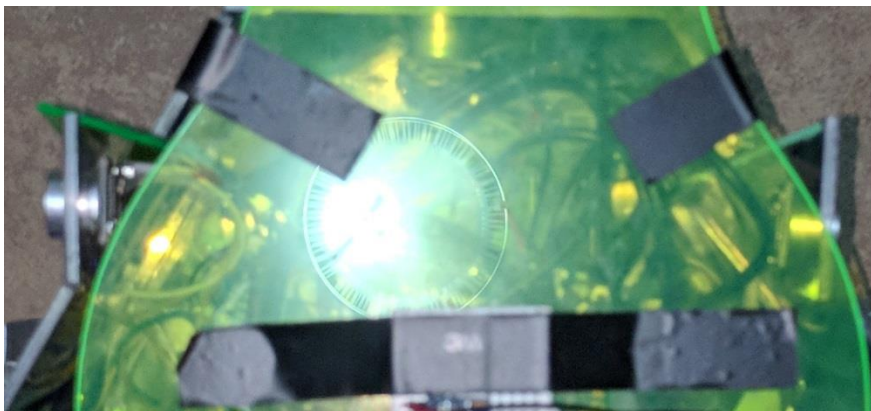


Figura 16. Sensores 4 y 6 en su nueva ubicación.

3.5. Descripción de la interfaz de usuario (GUIE de MatLab)

La GUIE realizada en MatLab como se ve en la Figura 14, consta de 7 recuadros principales, en esta sección se numerarán uno a uno con la intención de que el lector comprenda el funcionamiento y la visualización de la GUIE desarrollada [4] para el control de la plataforma móvil P-MITO.

3.5.1. Ingresar coordenadas del P-MITO

En el recuadro mostrado en la Figura 17, se ingresan los datos de las coordenadas a las que se quiere llegar con el robot; En los cuadros situados a la derecha de “Xf [cm] y Yf [cm] se ingresan los valores negativos o positivos. Se debe tener en cuenta que los cuadrantes del robot están divididos de acuerdo con los campos magnéticos de la tierra en donde el Eje Y+ es el Este y el Eje X+ es el Sur, teniendo en cuenta estos datos ya se le pueden ingresar fácilmente las coordenadas al robot. Es de aclarar que no se debe ingresar el valor del punto inicial, ya que el robot asume su punto inicial como el lugar en donde se encuentra.

| | | | |
|---------|----------------------------------|---------------|--------------------------------------|
| Xf [cm] | <input type="text" value="950"/> | Magnitud [cm] | <input type="text" value="1173.55"/> |
| Yf [cm] | <input type="text" value="689"/> | Theta [deg] | <input type="text" value="36"/> |

Figura 17. Recuadro de datos iniciales.

Los cuadros de Magnitud[cm] y Theta[deg], son calculados internamente por los procesos de MatLab y estos son la distancia que el robot debe

recorrer y el ángulo al cual se debe dirigir con respecto al eje X+(Sur magnético terrestre) y eje Y+ (Este).

3.5.2. Recuadro de velocidad lineal, angular y distancia recorrida

En este recuadro mostrado en la Figura 18 se presentan las velocidades lineales, angulares en RPM y la distancia recorrida por el robot tanto el promedio como la individual por rueda.

| | PROM | IZQ | DER | |
|------------------|---------|---------|---------|-----|
| Velocidad Lineal | 31.4159 | 31.4159 | 31.4159 | m/s |
| Distancia | 775 | 1330 | 1320 | cm |
| RPM | | 30 | 30 | rpm |

Figura 18. Velocidad angular, lineal y distancia de la P-MITO.

Todos estos datos son procesados al momento de ser recibidos por la plataforma móvil.

3.5.3. Ángulo de dirección y botones principales

En el cuadro de la figura 18 se puede observar un círculo con ángulos a su alrededor y una flecha en medio. Esta flecha es quien muestra la dirección angular que está teniendo el robot al momento de realizar una trayectoria definida previamente.

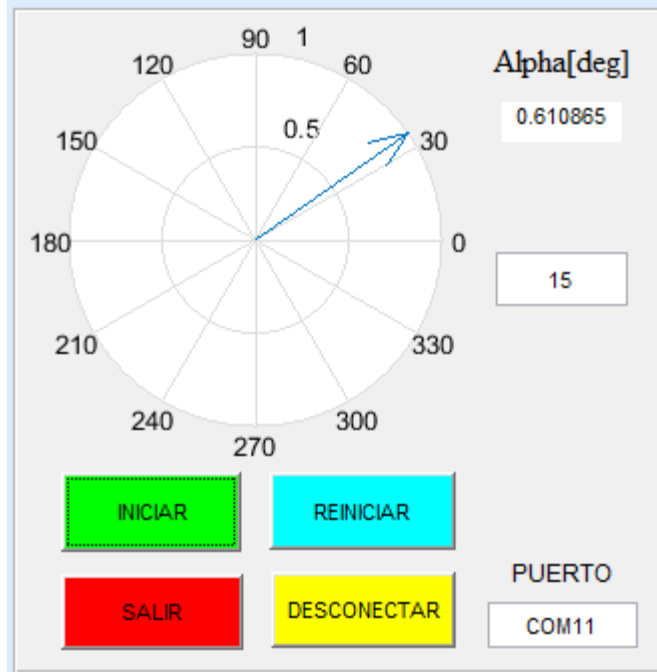


Figura 19. Dirección angular y botones principales.

También se puede observar un cuadro el cual es su parte superior tiene escrito “Alpha[deg]”, este presenta de manera numérica el ángulo en el que se direcciona el robot. Podemos ver otro cuadro en donde se ve el número 15 escrito, este número indica cuantas veces se reciben datos antes de tomar decisiones, en otras palabras, cada que se reciben por parte del robot 15 veces los datos, MatLab muestra datos de trayectorias, velocidades en los diferentes recuadros de la GUIE.

El botón verde “Iniciar” le indica a Matlab que debe iniciar el intercambio de datos y su posterior recorrido de la trayectoria indicada. El Botón azul “Reiniciar” es quien cancela el envío y recepción de datos en caso de presentar problemas, el botón rojo “Salir” sirve para cerrar la GUIE y el botón amarillo “Desconectar” se encarga de desconectar la comunicación entre MatLab y el puerto USB o la tarjeta inalámbrica.

El Ultimo cuadro, que se llama “Puerto” es el encargado de decirle a MAtLab cual es el puerto en el que se encuentra conectada la tarjeta de Arduino inalámbrica.

3.5.4. Gráfico de recorrido de la plataforma P-MITO

En el cuadro de la figura 20 se muestra un gráfico del recorrido del robot y se va punteando de color rojo las posiciones temporales que va obteniendo

durante el recorrido de la trayectoria. La Línea azul es el recorrido ideal que debería tener el robot. Las distancias se presentan tanto para el eje X como el Y en centímetros.

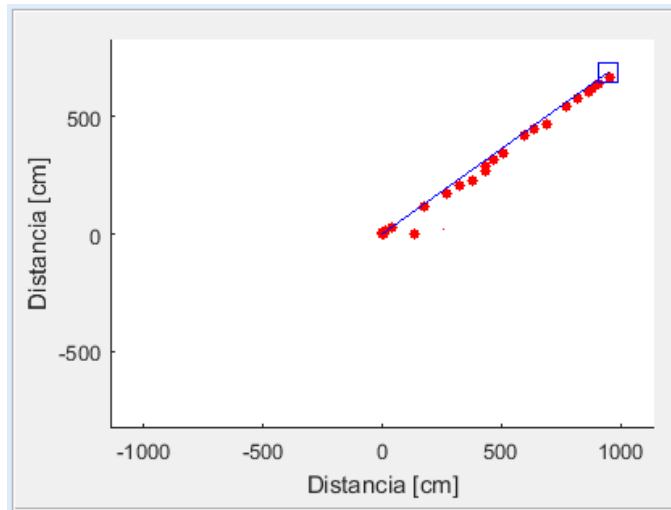


Figura 20. Gráfico del recorrido del robot.

3.5.5. Gráfico de velocidad de las ruedas

En la figura 21 la GUIE muestra en gráfico las velocidades individuales de las ruedas en rojo y amarillo y en azul se presenta la velocidad promedio de las dos ruedas.

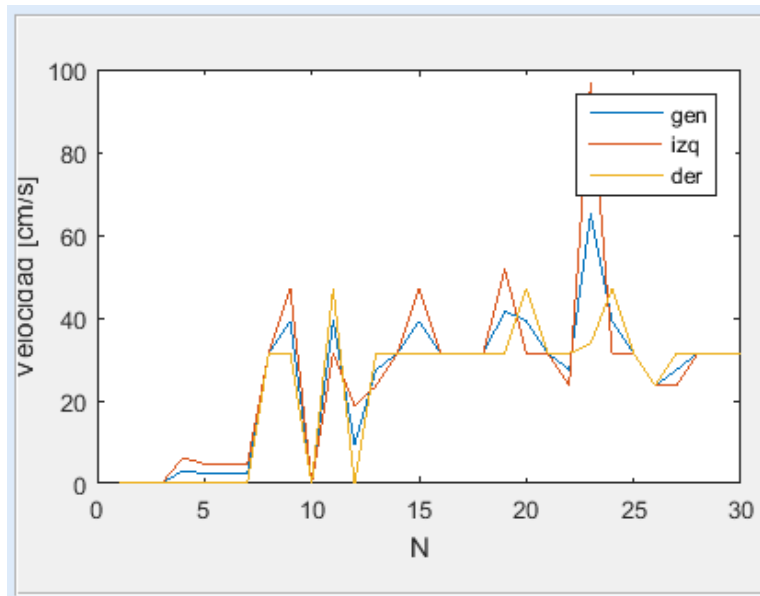


Figura 21. Gráfico de velocidades promedio e individuales de las ruedas.

3.5.6. Datos en sensores

Finalmente, los últimos dos cuadros mostrados en la figura 22, el cuadro de la parte superior presenta en tiempo real las medidas de los sensores ultrasónicos y el cuadro de la parte inferior presenta las medidas de los sensores infrarrojos. Es de recordar, que los sensores infrarrojos fueron eliminados de las rutinas del robot por lo que este cuadro siempre va a presentar datos en 0. También es de tener en cuenta, que las medidas de los sensores ultrasónicos no dependen del número de veces que se deban se deban recibir los datos antes de MatLab tomar alguna decisión, estos datos son en tiempo real y si se presenta algún obstáculo alrededor del robot, este podrá actuar inmediatamente.

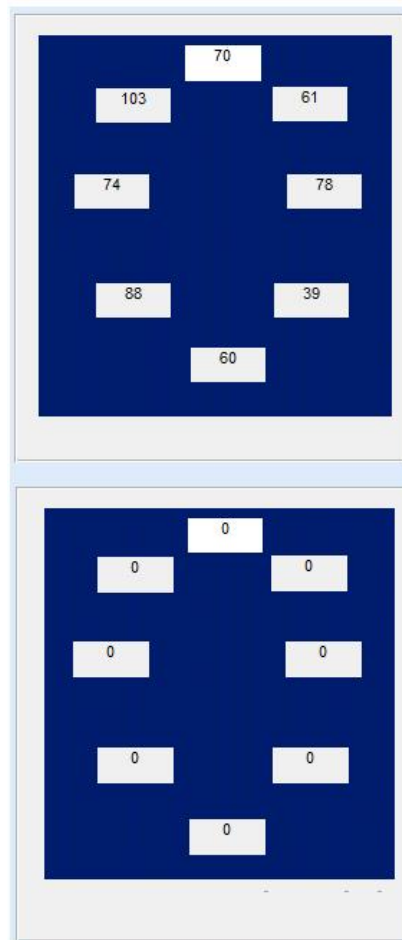


Figura 22. Medidas de los sensores ultrasónicos e infrarrojos.

También se puede notar que las posiciones de los cuadros que representan cada sensor coinciden con la ubicación presentada en la figura 15.

4. RESULTADOS

La plataforma móvil P-MITO, se sometió a varias pruebas para verificar su comportamiento. Las pruebas se hicieron de tal forma que el robot intentará irse de forma recta en sus ejes coordenados. Como ya se había mencionado, los ejes están asociados a los campos magnéticos de la tierra, en la figura 23 se muestra la posición de los ejes coordenados del robot.

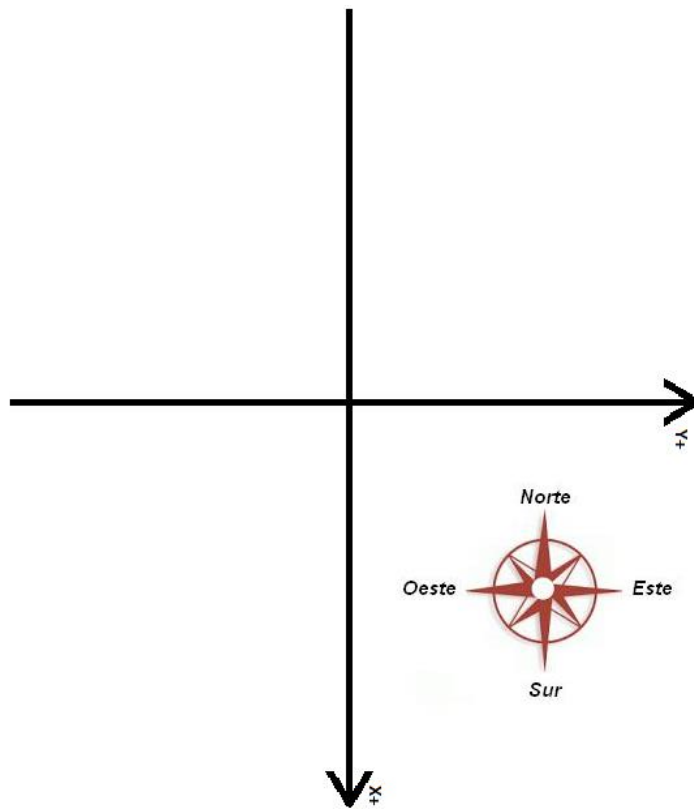


Figura 23. Ejes coordenados de la plataforma P-MITO.

Los resultados de las pruebas que se muestran a continuación fueron realizados en los Ejes X-, Y+ y Y- y para mostrar los errores que se presentaban, se hizo uso del error absoluto y el error relativo. Estos errores fueron calculados para X, Y y Φ de acuerdo con las pruebas de se fueron realizando.

Error absoluto: Este tipo de error se calculó con la siguiente formula:

$$\text{Error absoluto} = |\text{Valor real} - \text{Valor esperado}|$$

Error relativo: Este tipo de error se calculó con la siguiente formula:

$$\text{Error relativo} = \left| \frac{\text{Error absoluto}}{\text{Valor esperado}} \right|$$

4.1. Errores presentes en el eje Y+

Para este eje, se realizaron 13 pruebas asignándole al robot que se moviera 350cm en el eje Y y 0cm en el eje X esperando un ángulo de 90°. Los resultados fueron los siguientes:

Los datos en rojo representan medidas atípicas que se presentaron al momento de realizar las 13 pruebas de la plataforma.

| VALOR ESPERADO EN Y | VALOR ESPERADO EN X | ÁNGULO ESPERADO | VALOR REAL EN Y | VALOR REAL EN X | ÁNGULO REAL |
|---------------------|---------------------|-----------------|-----------------|-----------------|-------------|
| 350 | 0 | 90 | 647 | 40 | 86,46225977 |
| 350 | 0 | 90 | 582 | 57 | 84,40639655 |
| 350 | 0 | 90 | 510 | 23 | 87,41782231 |
| 350 | 0 | 90 | 465 | 14 | 88,27548672 |
| 350 | 0 | 90 | 528 | 2 | 89,78297157 |
| 350 | 0 | 90 | 625 | 52 | 85,24394514 |
| 350 | 0 | 90 | 640 | 39 | 86,51285054 |
| 350 | 0 | 90 | 502 | 54 | 83,8603176 |
| 350 | 0 | 90 | 502 | 24 | 87,26284369 |
| 350 | 0 | 90 | 619 | 23 | 87,87205659 |
| 150 | 0 | 90 | 383 | 35 | 84,77859568 |
| 350 | 0 | 90 | 543 | 121 | 77,43767946 |
| 350 | 0 | 90 | 411 | 161 | 68,60835561 |

Tabla 1. Datos del eje Y+.

Los errores obtenidos se presentan en la siguiente tabla

| ERROR ABSOLUTO EN Y | ERROR ABSOLUTO EN X | ERROR ABSOLUTO DEL ÁNGULO | ERROR RELATIVO EN Y | ERROR RELATIVO DEL ÁNGULO |
|---------------------|---------------------|---------------------------|---------------------|---------------------------|
| 297 | 40 | 3,537740226 | 84,857% | 3,931% |
| 232 | 57 | 5,593603449 | 66,286% | 6,215% |
| 160 | 23 | 2,582177687 | 45,714% | 2,869% |
| 115 | 14 | 1,724513279 | 32,857% | 1,916% |
| 178 | 2 | 0,21702843 | 50,857% | 0,241% |
| 275 | 52 | 4,756054856 | 78,571% | 5,285% |
| 290 | 39 | 3,487149461 | 82,857% | 3,875% |
| 152 | 54 | 6,1396824 | 43,429% | 6,822% |
| 152 | 24 | 2,737156305 | 43,429% | 3,041% |
| 269 | 23 | 2,127943407 | 76,857% | 2,364% |
| 233 | 35 | 5,22140432 | 155,333% | 5,802% |
| 193 | 121 | 12,56232054 | 55,143% | 13,958% |
| 61 | 161 | 21,39164439 | 17,429% | 23,768% |

Tabla 2. Errores relativos y absolutos de los datos obtenidos en la tabla 1.

Se puede ver que hay un error bastante considerable en el recorrido del robot el cual puede presentarse debido a la velocidad de procesamiento del computador en que se estén realizando las pruebas. Esto se presenta ya que al momento del robot haber recorrido de la distancia que se le ha ingresado previamente, el computador se queda haciendo bastantes procesos antes de enviar la orden de frenar a la plataforma, generando en el robot un desplazamiento extra que se da entre el momento que sobrepasa la distancia que debía recorrer y mientras el computador (MatLab) le envía la orden de frenar a la plataforma.

4.2. Errores presentes en el eje Y-

Para el eje Y-, se realizaron 18 pruebas asignándole al robot que se moviera los mismos 350cm en el eje Y- y 0cm en el eje X, esperando un ángulo de 270° para el posicionamiento de la plataforma. Los resultados fueron los siguientes:

| VALOR ESPERADO EN Y | VALOR ESPERADO EN X | ÁNGULO ESPERADO | VALOR REAL EN Y | VALOR REAL EN X | ÁNGULO REAL |
|---------------------|---------------------|-----------------|-----------------|-----------------|-------------|
| -350 | 0 | 270 | -508 | -98 | 280,9189893 |
| -350 | 0 | 270 | -510 | -8 | 270,8986836 |
| -350 | 0 | 270 | -475 | -33 | 273,9741632 |
| -350 | 0 | 270 | -498 | -21 | 272,4146565 |
| -350 | 0 | 270 | -561 | -9 | 270,9191048 |
| -350 | 0 | 270 | -520 | -1 | 270,1101841 |
| -350 | 0 | 270 | -508 | 43 | 274,8383063 |
| -350 | 0 | 270 | -518 | -48 | 275,294143 |
| -350 | 0 | 270 | -492 | -61 | 277,0676776 |
| -350 | 0 | 270 | -515 | -57 | 276,315769 |
| -350 | 0 | 270 | -523 | 31 | 273,3921481 |
| -350 | 0 | 270 | -488 | -12 | 271,4086287 |
| -350 | 0 | 270 | -508 | -57 | 276,4020796 |
| -350 | 0 | 270 | -625 | 8 | 270,7333459 |
| -350 | 0 | 270 | -542 | 10 | 271,0569978 |
| -350 | 0 | 270 | -483 | -156 | 287,8994898 |
| -350 | 0 | 270 | -598 | -64 | 276,1087374 |
| -350 | 0 | 270 | -430 | -221 | 297,2010559 |

Tabla 3. Datos en el eje Y-.

El cálculo de los errores se observa en la tabla 4

| ERROR ABSOLUTO EN Y | ERROR ABSOLUTO EN X | ERROR ABSOLUTO DEL ÁNGULO | ERROR RELATIVO EN Y | ERROR RELATIVO DEL ÁNGULO |
|---------------------|---------------------|---------------------------|---------------------|---------------------------|
| 158 | 98 | 10,91898933 | 45,143% | 4,044% |
| 160 | 8 | 0,898683621 | 45,714% | 0,333% |
| 125 | 33 | 3,974163212 | 35,714% | 1,472% |
| 148 | 21 | 2,414656521 | 42,286% | 0,894% |
| 211 | 9 | 0,919104784 | 60,286% | 0,340% |
| 170 | 1 | 0,110184056 | 48,571% | 0,041% |
| 158 | 43 | 4,83830629 | 45,143% | 1,792% |
| 168 | 48 | 5,294143007 | 48,000% | 1,961% |
| 142 | 61 | 7,067677584 | 40,571% | 2,618% |
| 165 | 57 | 6,315768994 | 47,143% | 2,339% |
| 173 | 31 | 3,392148077 | 49,429% | 1,256% |
| 138 | 12 | 1,408628735 | 39,429% | 0,522% |
| 158 | 57 | 6,40207958 | 45,143% | 2,371% |
| 275 | 8 | 0,733345929 | 78,571% | 0,272% |
| 192 | 10 | 1,056997777 | 54,857% | 0,391% |
| 133 | 156 | 17,89948984 | 38,000% | 6,629% |
| 248 | 64 | 6,108737414 | 70,857% | 2,262% |
| 80 | 221 | 27,20105591 | 22,857% | 10,074% |

Tabla 4. Errores relativos y absolutos de los datos obtenidos en la tabla 3.

Es de gran interés que en este eje, nuevamente los errores angulares son relativamente bajos, mientras que los que están presentes en el recorrido del robot siguen mostrando el mismo error aproximado, lo que hace más notorio que la velocidad de procesamiento del computador que se encuentre controlando la plataforma influye en el desplazamiento mismo.

En la figura 24 se puede ver el registro fotográfico de una de las pruebas que se le realizaron al robot.



Figura 24. Prueba realizada al robot.

El metro (línea amarilla) representa el eje Y y la línea roja representa el eje X. la parte inferior donde inicia el metro muestra el punto donde inició el robot y al fondo se puede ver donde termino el robot.

4.3. Errores presentes en el eje X-

La misma prueba se realizó también para el eje X para corroborar que el error se comportara de la misma manera en este eje. Para estas pruebas se tomaron un total de 13 medidas que se muestran en la tabla 5.

| VALOR ESPERADO EN Y | VALOR ESPERADO EN X | ÁNGULO ESPERADO | VALOR REAL EN Y | VALOR REAL EN X | ÁNGULO REAL |
|---------------------|---------------------|-----------------|-----------------|-----------------|-------------|
| 0 | -350 | 180 | 64 | -545 | 173,3023629 |
| 0 | -350 | 180 | 74 | -573 | 172,6412752 |
| 0 | -350 | 180 | 52 | -561 | 174,7042931 |
| 0 | -350 | 180 | 71 | -547 | 172,6044187 |
| 0 | -350 | 180 | 73 | -518 | 171,9783242 |
| 0 | -350 | 180 | 84 | -565 | 171,5436316 |
| 0 | -350 | 180 | 54 | -540 | 174,2894069 |
| 0 | -350 | 180 | 14 | -532 | 178,4925642 |
| 0 | -350 | 180 | 97 | -595 | 170,7407981 |
| 0 | -350 | 180 | 73 | -522 | 172,039002 |
| 0 | -350 | 180 | -92 | -558 | 189,3623878 |
| 0 | -350 | 180 | 143 | -553 | 165,5015076 |
| 0 | -350 | 180 | 174 | -563 | 162,8257854 |

Tabla 5. Datos en el eje Y-.

Los errores se muestran en la tabla 6

| ERROR ABSOLUTO EN Y | ERROR ABSOLUTO EN X | ERROR ABSOLUTO DEL ÁNGULO | ERROR RELATIVO EN X | ERROR RELATIVO DEL ÁNGULO |
|---------------------|---------------------|---------------------------|---------------------|---------------------------|
| 64 | 195 | 6,697637132 | 55,71% | 3,721% |
| 74 | 223 | 7,358724759 | 63,71% | 4,088% |
| 52 | 211 | 5,295706896 | 60,29% | 2,942% |
| 71 | 197 | 7,39558132 | 56,29% | 4,109% |
| 73 | 168 | 8,021675758 | 48,00% | 4,456% |
| 84 | 215 | 8,456368431 | 61,43% | 4,698% |
| 54 | 190 | 5,710593137 | 54,29% | 3,173% |
| 14 | 182 | 1,507435759 | 52,00% | 0,837% |
| 97 | 245 | 9,259201943 | 70,00% | 5,144% |
| 73 | 172 | 7,960998049 | 49,14% | 4,423% |
| 92 | 208 | 9,36238781 | 59,43% | 5,201% |
| 143 | 203 | 14,49849238 | 58,00% | 8,055% |
| 174 | 213 | 17,17421462 | 60,86% | 9,541% |

Tabla 6. Errores relativos y absolutos de los datos obtenidos en la tabla 5.

Como era de esperarse, los errores en el recorrido del robot se asemejan en todos los ejes manteniendo el comportamiento con un error aproximado del 56% para todas las medidas.

Los ángulos tienen el mismo comportamiento en todos los ejes con un error muy pequeño.

Siempre es bueno tener presente la ruta aproximada que debe seguir el robot y que, al momento de generar una trayectoria, el recorrido que realice no sea un dato atípico, ya que en las tres pruebas siempre se tuvieron presentes datos atípicos.



Figura 25. Prueba realizada al robot.

4.4. Comportamiento y lecturas de los sensores ultrasónicos

Para verificar el comportamiento de los sensores, se realizaron dos series de pruebas. Las pruebas consistían en poner a funcionar el robot de forma paralela a una pared y cada 0,5 metros marcar la posición por donde había pasado el robot (Ver figura 26) para luego tomar la medida de este contra la pared y compararlo con las medidas que se almacenaban en el computador. Se realizaron estas medidas para los sensores paralelos a las ruedas 4 y 6 (Ver figura 15). A los demás sensores (Sensores 2, 3, 5, 7 y 8), no se les realizaron pruebas ya que estos por su ubicación era difícil seguir una trayectoria que permitiera tener lecturas de los mismos. El sensor 1 (Sensor frontal) respondió

de forma correcta en todas las pruebas que se hicieron al momento de poner algún obstáculo al frente.



Figura 26. Muestra de las pruebas realizadas a los sensores ultrasónicos.

Se le realizaron 3 pruebas al sensor S4, los resultados obtenidos, se muestran en las tablas 7, 8 y 9.

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 18,5 | 3,5 |
| 1 | 39,5 | 28,7 | 10,8 |
| 1,5 | 39,5 | 40,6 | 1,1 |
| 2 | 43 | 53,3 | 10,3 |
| 2,5 | 47,5 | 48,9 | 1,4 |
| 3 | 47 | 39,6 | 7,4 |
| 3,5 | 59 | 57,5 | 1,5 |

Tabla 7. Datos de la prueba 1 al sensor S4.

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 25,3 | 3,3 |
| 1 | 34,5 | 27,2 | 7,3 |
| 1,5 | 33 | 25,5 | 7,5 |
| 2 | 34 | 34,4 | 0,4 |
| 2,5 | 31 | 29,3 | 1,7 |
| 3 | 33 | 25,2 | 7,8 |
| 3,5 | 35 | 29,0 | 6,0 |
| 4 | 41 | 38,1 | 2,9 |

Tabla 8. Datos de la prueba 2 al sensor S4.

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 24,1 | 2,1 |
| 1 | 37 | 37,5 | 0,5 |
| 1,5 | 36 | 37,8 | 1,8 |
| 2 | 38,5 | 42,1 | 3,6 |
| 2,5 | 39 | 36,3 | 2,7 |
| 3 | 40 | 36,2 | 3,8 |
| 3,5 | 47 | 51,0 | 4,0 |
| 4 | 20,5 | 20,1 | 0,4 |

Tabla 9. Datos de la prueba 3 al sensor S4.

Se puede observar que los datos obtenidos por el robot y los datos obtenidos con las marcas en el suelo son muy aproximados, arrojando valores que se pueden utilizar para medir caminos estrechos en los que la plataforma se encuentre o en caso de intentar evadir algún obstáculo, poder saber el ángulo al cual podría girar para evadirlo.

Al sensor S6 ubicado al lado derecho del robot, se le realizaron las mismas tres pruebas. En las tablas 10, 11 y 12 se presentan los resultados obtenidos:

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 24,1 | 2,1 |
| 1 | 30,5 | 31,0 | 0,5 |
| 1,5 | 31,5 | 33,0 | 1,5 |
| 2 | 27 | 30,6 | 3,6 |
| 2,5 | 33,5 | 30,6 | 2,9 |
| 3 | 39,5 | 35,7 | 3,8 |
| 3,5 | 40 | 44,0 | 4,0 |
| 4 | 42 | 41,6 | 0,4 |

Tabla 10. Datos de la prueba 1 al sensor S6.

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 18,5 | 3,5 |
| 1 | 25 | 14,2 | 10,8 |
| 1,5 | 22 | 23,1 | 1,1 |
| 2 | 15 | 25,3 | 10,3 |
| 2,5 | 23 | 24,4 | 1,4 |
| 3 | 25,5 | 18,1 | 7,4 |
| 3,5 | 27 | 25,5 | 1,5 |
| 4 | 30,5 | 33,3 | 2,8 |

Tabla 11. Datos de la prueba 2 al sensor S6.

| DISTANCIA RECORRIDA | DISTANCIA ESPERADA ENTRE PARED Y SENSOR | DISTANCIA REAL ENTRE PARED Y SENSOR | ERROR ABSOLUTO |
|----------------------------|--|--|-----------------------|
| 0 | 22 | 25,3 | 3,3 |
| 1 | 29 | 21,7 | 7,3 |
| 1,5 | 28,5 | 21,0 | 7,5 |
| 2 | 27,5 | 27,9 | 0,4 |
| 2,5 | 33 | 31,3 | 1,7 |
| 3 | 38 | 30,2 | 7,8 |
| 3,5 | 40 | 34,0 | 6,0 |
| 4 | 43 | 40,1 | 2,9 |
| 4,5 | 42 | 43,5 | 1,5 |

Tabla 12. Datos de la prueba 3 al sensor S6.

En la figura 27, se pueden observar las marcas del robot durante una prueba realizada al sensor S6.



Figura 27. Prueba contra pared realizada al sensor S6.

Podemos apreciar que el robot tiene muy bajo margen de error con los sensores ultrasónicos, generando medidas confiables con las cuales se pueden trabajar y generar algoritmos de evasión de obstáculos que sirvan para un apropiado comportamiento del robot.

5. CONCLUSIONES

- ✓ Se logró la implementación de la comunicación inalámbrica entre la plataforma P-MITO y un computador central mediante XBee, permitiendo un intercambio de datos bidireccional, hay que tener en cuenta que la comunicación no es FULL DUPLEX, ya que la banda que utiliza el XBee solo permite ser usada para enviar o recibir datos de forma separada, siendo esta una sola vía de comunicación. Aunque no funciona de manera FULL DUPLEX, el cambio de envío a recibo o viceversa es tan rápido, que las lecturas se logran realizar en menos de un segundo, permitiendo tener un buen control sobre el robot y hacer que las lecturas sean confiables.
- ✓ Se implementó un código para la operación de filtrado Kalman, la eficacia del algoritmo de filtrado se ha probado proporcionando diferentes tipos de entradas a la estructura. Los resultados muestran una reducción de errores en los datos, lo que proporciona mejores resultados a la hora de tomar decisiones sobre la plataforma.
- ✓ Se implementó al código desarrollado en MatLab, la capacidad de almacenar datos históricos del robot con la finalidad de analizar cada recorrido que se le asignaba, esto con la intención de que con cada recorrido se puedan medir sus errores y sus datos atípicos y posteriormente, ejercer mecanismos de reducción de error y eliminar datos que generen malas lecturas.

6. ANEXOS

Los anexos se presentan en otro archivo llamado "AnexosP-MITO.PDF"

7. BIBLIOGRAFÍA

- [1] R. Siegwart y I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Cambridge, Massachusetts: Massachusetts Institute of Technology, 2004.
- [2] C. A. Agudelo y D. R. Arango, *Implementación de una plataforma móvil utilizando hardware reconfigurable (FPGA'S) para navegación en entornos dinámicos*, Pereira, Colombia, 2011.
- [3] Y. Tarutoko, K. Kobayashi y W. Kajiro, «Topological Map Generation based on Delaunay Triangulation for Mobile Robot,» de *SICE-ICASE International Joint Conference 2006*, Bexco, Busan, Korea, 2006.
- [4] J. C. Yepes y M. A. Fernández, *P-MITO Plataforma móvil para la investigación de técnicas de odometría*, Pereira, Colombia: Universidad Tecnológica de Pereira, 2016.
- [5] J. d. L. Asiaín, «Emisión y recepción de infrarojos,» Universidad Politécnica de Madrid, Madrid, 2001.
- [6] J. M. Días Peña, *Navegación visual del robot Pioneer*, España: Universidad Rey de Juan Carlos, 2004-2005.
- [7] E. Akbas y E. M. Esin, «A simulational comparison of intelligent control,» *Robotic and autonomous system*, vol. 49, nº 49, pp. 236-244, 2004.
- [8] M. Muñoz y V. F., *Planificación de trayectorias para robots móviles*, Málaga: Universidad de Málaga, 1995.
- [9] J. L. Blanco, J. Gonzalez y J. A. Fernández Madrigal, «Extending Obstacle Avoidance Methods through Multiple Parameter-Space Transformations,» *Autonomous Robots*, vol. 24, nº 1, pp. 29-48, 2008.
- [10] J. Titus, «Two-Way Communications with Xbee Modules,» de *The Hands-on Xbee Lab Manual*, 2012, pp. 139-156.
- [11] microkits, «Sensor Ultrasonido,» [En línea]. Available: <https://www.microkitselectronica.com/media/attachment/file/h/c/hcsr04.pdf>. [Último

acceso: 6 Julio 2017].

- [12] SHARP, «Sharp Corporation,» 2005. [En línea]. Available: <http://www.heppg.de/ikg/box/adapter/datenblatt/GP2Y0A21YK-DATA-SHEET.PDF>. [Último acceso: 2017 Julio 6].
- [13] Honeywell, «Magnetic Sensors,» 2013. [En línea]. Available: https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf. [Último acceso: 7 Julio 2017].
- [14] Tecnobaterías. [En línea]. Available: http://esco.tiendastore.net/hermes_ima/tecnobaterias/files/Fichas_tecnicas_Baterias/Ficha_Tecnica_TB_12-1.3_12V-1.3AH.pdf. [Último acceso: 7 Julio 2017].
- [15] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMega>. [Último acceso: 14 Julio 2017].
- [16] E. V. Rogozhnikov, A. S. Koldomov, D. A. Pokamestov y Y. V. Kryukov, «Full Duplex Wireless Communication System, Analog and Digital Cancellation, Experimental Research,» de *International Siberian Conference on Control Communication (SIBCON)*, 2017.
- [17] N. H. Mahmood, M. G. Sarret, G. Berardinelli y P. Mogensen, «Full Duplex Communications in 5G Small Cells,» de *Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, España, 2017.
- [18] R. M. Teodorescu, I. B. Cioc y B. A. Vochin, «LABVIEW Application Used for Remote Control of a Mobile Robot whit XBee Communication,» de *Electronics, Computers and Artificial Intelligence (ECAI)*, Ploiesti, Romania, 2016.
- [19] Xbee, «Xbee,» 2017]. [En línea]. Available: <http://xbee.cl/category/modxbee/xbee/>. [Último acceso: 9 Agosto 2017].
- [20] Arduino, «Arduino,» 2017. [En línea]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>. [Último acceso: 26 Agosto 2017].
- [21] H. Haritha, C. Seethal y R. Gandhiraj, «Realization of Kalman Filter in GNU Radio,» de *Computer Communication and Infomatics (ICCCI)*, Coimbatore, India, 2017.
- [22] Denyssense, «GitHub,» 5 Enero 2017. [En línea]. Available:

<https://github.com/denyssene/SimpleKalmanFilter>. [Último acceso: 4 Noviembre 2017].

- [23] A. M. Alajlan, M. M. Almasri y K. M. Elleithy, «Multi-Sensor Based Collision Avoidance Algorithm for Mobile Robot,» de *Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, USA, 2015.
- [24] Z. Huang, S. Long, Y. Jiang y Q. Chen, «An Outlier Detection Method Based on PageRank Algorithm,» de *International Conference on Computational Intelligence and Security*, Wuxi, China, 2016.
- [25] A. Cardoso, P. Gil, J. Lourenco, J. Leitao y A. Santos, «Demonstration of an application for remote,» de *Remote Engineering and Virtual Instrumentation (REV)*, Porto, Portugal, 2014.
- [26] V. Reyes, «5 Minutos de MatLab,» 2013. [En línea]. Available: <http://5minutosdematlab.blogspot.com.co/2013/09/retirar-datos-atipicos-outliers-de-una.html>. [Último acceso: 22 Diciembre 2017].
- [27] Arduino, «Arduino,» 2015. [En línea]. Available: <https://www.arduino.cc/en/Tutorial/MasterWriter>. [Último acceso: 2017 Agosto 15].
- [28] ElectronTools, «Etools,» 2017. [En línea]. Available: <http://www.electrontools.com/Home/WP/2016/04/01/como-funciona-el-sensor-ultrasonico-hc-sr04/>. [Último acceso: 27 Diciembre 2017].