



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELÉCTRICA, AUTOMATIZACIÓN Y
CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE:**

INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

**TEMA: “ADAPTACIÓN DEL SISTEMA ROBÓTICO MÓVIL
AUTÓNOMO COOPERATIVO SMELLROBSENSE PARA MAPEO 2D DE
OBSTÁCULOS, TEMPERATURA Y CONCENTRACIÓN DE GASES EN
UN AMBIENTE CONTROLADO”**

AUTOR: ILLANES TIBANLOMBO, JUAN CARLOS

DIRECTOR: ARCENALES VITERI, ANDRES RICARDO

SANGOLQUÍ

2018



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

CERTIFICACIÓN

Certifico que el trabajo de titulación, *“ADAPTACIÓN DEL SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO SMELLROBSENSE PARA MAPEO 2D DE OBSTÁCULOS, TEMPERATURA Y CONCENTRACIÓN DE GASES EN UN AMBIENTE CONTROLADO”* realizada por el señor *Illanes Tibanlombo Juan Carlos*, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 5 de diciembre del 2018

Dr. Arcentales Viteri Andrés Ricardo

C.C.: 17122480746



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

AUTORÍA DE RESPONSABILIDAD

Yo, *Illanes Tibanlombo Juan Carlos*, declaro que el contenido, ideas y criterios del trabajo de titulación: ***“ADAPTACIÓN DEL SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO SMELLROBSENSE PARA MAPEO 2D DE OBSTÁCULOS, TEMPERATURA Y CONCENTRACIÓN DE GASES EN UN AMBIENTE CONTROLADO”***, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 5 de diciembre del 2018

Illanes Tibanlombo Juan Carlos

C.C.: 1721559027



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

AUTORIZACIÓN

Yo, *Illanes Tibanlombo Juan Carlos*, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación ***“ADAPTACIÓN DEL SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO SMELLROBSENSE PARA MAPEO 2D DE OBSTÁCULOS, TEMPERATURA Y CONCENTRACIÓN DE GASES EN UN AMBIENTE CONTROLADO”***, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 5 de diciembre del 2018

A handwritten signature in blue ink, which appears to read 'Juan Carlos Illanes Tibanlombo', is written over a horizontal dashed line.

Illanes Tibanlombo Juan Carlos

C.C.: 1721559027

DEDICATORIA

Quiero dedicar este proyecto a mis padres Wilson y Graciela, ellos son el pilar fundamental de mi formación, tanto personal como académica, han sido el apoyo más grande que he recibido en el transcurso de mi carrera profesional, me han enseñado que la vida es un camino en el cual debo luchar día a día para alcanzar todos mis objetivos, mi madre brindándome todo su cariño y mi padre la fuerza necesaria para salir adelante a pesar de las adversidades, ellos son las personas más importantes en mi vida y a ellos en especial, les dedico este trabajo.

A mi hermana Cristina, que, a pesar de la distancia, siempre ha estado pendiente de mi apoyándome incondicionalmente en cada momento de mi vida. También va dedicado para mi sobrino ahijado Leo, del cual quiero ser el mejor ejemplo en su vida.

Para ustedes con mucho cariño.

AGRADECIMIENTO

Quiero agradecer a mi familia, que siempre ha estado junto a mí en las buenas y en las malas, en especial a mis primos Henry, Paulina, Michelle y Christian, con ustedes he compartido momentos gratos incluso en los momentos más difíciles.

Gracias a todos mis amigos de la Universidad, con quienes sin lugar a duda he pasado algunos de los mejores años de mi vida, quiero agradecer especialmente a mi amigo Carlos que siempre estuvo alentándome para no rendirme, a Diego y Alexis con quienes empezamos la carrera y hasta ahora mantenemos una gran amistad.

Para finalizar quiero agradecer a los Ingenieros del departamento que han forjado mis conocimientos y han permitido formarme como un profesional, en especial quiero agradecer a mi tutor Dr. Andrés Arcentales, quien a pesar de todo pudo confiar en mí, encaminándome en mis primeras prácticas pre profesionales y guiándome en el desarrollo de mi proyecto de titulación, gracias por su dedicación y paciencia.

ÍNDICE DE CONTENIDOS

CERTIFICADO DEL DIRECTOR.....	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS.....	1
RESUMEN.....	xiii
ABSTRACT	xiv
CAPÍTULO I.....	1
INTRODUCCIÓN.....	1
1.1. Antecedentes.....	6
1.2. Justificación e Importancia.....	8
1.3. Alcance del Proyecto	11
1.4. Objetivos.....	13
1.4.1. Objetivo general	13
1.4.2. Objetivos específicos.....	13
CAPÍTULO II.....	15
ESTADO DEL ARTE	15

2.1.	Introducción.....	15
2.2.	Agente inteligente.....	15
2.2.1.	Clasificación de agentes inteligentes.....	16
2.3.	Robótica móvil	17
2.3.1.	Aplicaciones de la robótica móvil	18
2.3.2.	Clasificación de robots móviles por su estructura	22
2.4.	Robótica cooperativa	25
2.5.	Robótica móvil cooperativa.....	26
2.5.1.	Planeación de movimientos.....	26
2.5.2.	Autonomía	27
2.5.3.	Comunicación.....	27
2.5.4.	Modelos de control	29
2.6.	Algoritmos de navegación, mapeo y localización (SLAM)	32
2.6.1.	Algoritmos basados en técnicas geométricas	32
	CAPÍTULO III	36
	DISEÑO E IMPLEMENTACIÓN DE LAS PLATAFORMAS ROBÓTICAS	36
3.1.	Introducción.....	36
3.2.	Antecedentes.....	36
3.3.	Requerimientos de funcionamiento.....	37
3.4.	Sistema mecánico	41
3.4.1.	Plataforma robótica móvil DG012-ATV 4WD	42
3.4.2.	Diseño CAD	43
3.5.	Sistema encargado del control.....	46

3.5.1. Teensy 3.6	46
3.5.2. Raspberry Pi 3 modelo B.....	50
3.6. Sistema de sensado	51
3.6.1. Odometría	51
3.6.2. Obstáculos y fronteras	54
3.6.3. Sensado químico.....	56
3.6.4. Sensado de temperatura.....	59
3.7. Sistema de potencia	59
3.7.1. Motores Dagu DG02S	60
3.7.2. Driver L298N	61
3.8. Conexión y montaje de la plataforma robótica móvil	62
3.9. Integración de hardware	67
CAPÍTULO IV	69
SISTEMA DE NAVEGACIÓN Y COMUNICACIÓN	69
4.1. Introducción.....	69
4.2. Descripción del sistema de comunicación y control	69
4.3. Topología de red.....	71
4.4. Entorno de navegación y pruebas	73
4.5. Algoritmo para solución de la cinemática directa	74
4.6. Algoritmo de navegación	79
4.6.1. Algoritmo para solución de cinemática inversa	81
4.7. Desarrollo del sistema de comunicación	86
4.7.1. Comunicación entre Teensy 3.6 y Raspberry Pi 3	86

4.7.2. Comunicación inalámbrica entre Raspberry Pi 3 y ordenador centralizado	88
4.7.3. Trama de retorno	94
4.8. Definición del sistema robótico móvil autónomo	94
4.9. Interacción del algoritmo de navegación y comunicación	96
CAPÍTULO V	97
VISUALIZACIÓN DE MAPAS, PRUEBAS Y RESULTADOS	97
5.1. Integración del sistema de control centralizado	97
5.2. Interfaz gráfica de control centralizado	98
5.2.1. Distribución de la pantalla.....	99
5.3. Visualización de mapa de obstáculos	103
5.4. Visualización de mapas de sensado químico y temperatura.....	105
5.5. Pruebas de sistemas independientes	106
5.5.1. Recorrido	107
5.5.2. Muestreo y estimación de parámetros	108
5.5.3. Normalización de sensores químicos	109
5.5.1. Comunicación serial USB	111
5.5.2. Comunicación inalámbrica.....	112
5.6. Pruebas y resultados del sistema general.....	113
5.6.1. Escenarios	114
5.7. Resultados del sistema robótico cooperativo	118
5.7.1. Pruebas del escenario No. 1.....	118
5.7.2. Pruebas del escenario No. 2.....	121
5.7.3. Pruebas del escenario No. 3.....	125

5.7.4. Tiempos de navegación	129
CAPÍTULO VI	130
CONCLUSIONES Y RECOMENDACIONES	130
6.1. Diseño e implementación del hardware	130
6.2. Desarrollo de software de control, navegación y sensado.....	133
6.3. Pruebas y resultados experimentales	136
6.4. Trabajos futuros.....	138
REFERENCIAS BILIOGRÁFICAS	140

ÍNDICE DE TABLAS

Tabla 1 Ventajas y desventajas de los modelos	29
Tabla 2 Fases para determinar los comportamientos de un robot móvil	31
Tabla 3 Especificaciones técnicas de la plataforma DG012-ATV 4WD.....	43
Tabla 4 Configuración de pines de la Teensy 3.6	48
Tabla 5 Configuración de pines de los conversores lógicos bidireccionales.....	55
Tabla 6 Especificaciones técnicas de los actuadores que integran al robot móvil.....	60
Tabla 7 Especificaciones técnicas del driver L298N.....	61
Tabla 8 Configuración de pines del driver L298N	62
Tabla 9 Condiciones para determinar el ángulo de giro	77
Tabla 10 Primer caso para determinar la cinemática inversa de un robot	83
Tabla 11 Segundo caso para determinar la cinemática inversa de un robot	83
Tabla 12 Tercer caso para determinar la cinemática inversa de un robot.....	84
Tabla 13 Cuarto caso para determinar la cinemática inversa de un robot	84
Tabla 14 Quinto caso para determinar la cinemática inversa de un robot	84
Tabla 15 Sexto caso para determinar la cinemática inversa de un robot	85
Tabla 16 Séptimo caso para determinar la cinemática inversa de un robot.....	85
Tabla 17 Octavo caso para determinar la cinemática inversa de un robot.....	85
Tabla 18 Variables de la interfaz gráfica de usuario y sistema de control centralizado	93
Tabla 19 Corrección de ecuaciones para estimar distancias recorridas.....	107
Tabla 20 Cálculo de valores proporcionales de concentración de gases	110
Tabla 21 Configuración del escenario No. 1	114

Tabla 22 Configuración del escenario No. 2	114
Tabla 23 Configuración del escenario No. 3	115
Tabla 24 Tiempos de navegación	129

RESUMEN

El proyecto de investigación desarrollado explica el proceso de diseño e implementación de un sistema robótico autónomo cooperativo conformado por tres plataformas robóticas móviles de configuración diferencial, un ordenador central y un concentrador de datos; estas plataformas robóticas interactúan entre sí mediante un algoritmo de navegación, control y comunicación programado en el ordenador central el cual cumple con la función de controlador centralizado. Cada plataforma consta de los siguientes subsistemas: sistema mecánico, sistema de control, sistema de sensado y sistema de potencia. El objetivo del sistema de sensado es determinar mediante exploración autónoma, distancias hacia obstáculos o fronteras mediante sensores de estimación de distancias y determinar los niveles de temperatura y concentración de gases en un entorno mediante sensores de temperatura y sensores químicos, toda la adquisición de datos de cada agente robótico es llevada a cabo en una tarjeta programable Teensy 3.6, la información es procesada en una Raspberry Pi 3 y posteriormente es enviada mediante comunicación inalámbrica hacia una interfaz gráfica implementada en el ordenador central donde toda la información es visualizada en tiempo real y se procesa nuevamente para permitir la visualización de mapas de obstáculos, temperatura y concentración de gases.

PALABRAS CLAVES:

- **ROBÓTICA COOPERATIVA**
- **MAPEO**
- **ODOMETRÍA**
- **AUTÓNOMO**
- **SENSADO QUÍMICO**

ABSTRACT

The research project developed explains the process of design and implementation of a robotic autonomous cooperative system consisting of three mobile robotic platforms of differential configuration, a central computer and a data concentrator; These robotic platforms interact with each other through a navigation, control and communication algorithm programmed in the central computer which fulfills the centralized controller function. Each platform consists of the following subsystems: mechanical system, control system, sensing system and power system. The purpose of the sensing system is to determine, by autonomous exploration, distances to obstacles or borders by means of distance estimation sensors and to determine the temperature and concentration levels of gases in an environment by temperature sensors and chemical sensors, all data acquisition of each robotic agent is carried out in a Teensy 3.6 programmable card, the information is processed in a Raspberry Pi 3 and later it is sent by wireless communication to a graphical interface implemented in the central computer where all the information is visualized in real time and it is processed again to allow the visualization of obstacle maps, temperature and concentration of gases.

KEYWORDS:

- **COOPERATIVE ROBOTICS**
- **MAPPING**
- **ODOMETRY**
- **AUTONOMOUS**
- **CHEMICAL SENSING**

CAPÍTULO I

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Visión esquemática de un agente inteligente.....	15
<i>Figura 2.</i> Esquema de un agente inteligente de reflejo simple.....	16
<i>Figura 3.</i> Esquema de un agente inteligente basado en metas.	17
<i>Figura 4.</i> RoboScrub, para aspirar y limpiar pisos.....	18
<i>Figura 5.</i> Vehículo prototipo MDARS-E.....	19
<i>Figura 6.</i> Robot explorador lunar Lunokhod 1.....	19
<i>Figura 7.</i> Robot soldado TALON.....	20
<i>Figura 8.</i> Asfaltadora Cedarapids® utilizada en Ingeniería Civil.....	21
<i>Figura 9.</i> Robot hexápodo bioinspirado en el Tenebrio.....	22
<i>Figura 10.</i> Robot móvil diferencial.....	23
<i>Figura 11.</i> Robot móvil SUMMIT XL Robotnik®.....	23
<i>Figura 12.</i> Robotino® de Festo.....	24
<i>Figura 13.</i> Robot de configuración síncrona.....	24
<i>Figura 14.</i> Robot móvil de configuración Ackermann.....	25
<i>Figura 15.</i> Grupos de robots encargados de levantar columnas.....	26
<i>Figura 16.</i> Representación esquemática de la búsqueda de un plan.....	30
<i>Figura 17.</i> Ejemplo de arquitectura reactiva de un agente inteligente.....	30
<i>Figura 18.</i> Grafo de visibilidad.....	33
<i>Figura 19.</i> Grafo de Voronoi.....	34

Figura 20. Grafo por descomposición de celdas.	35
Figura 21. Subsistemas de la arquitectura de un robot móvil autónomo.	41
Figura 22. Robots móviles de configuración diferencial.	42
Figura 23. Plataforma robótica móvil DG012-ATV (izq.), vista interna del chasis (der.)	42
Figura 24. Diseño CAD del prototipo de un robot móvil autónomo	44
Figura 25. Ubicación de los sensores de estimación de distancias	45
Figura 26. Estructura semicircular para sensores de distancia.....	45
Figura 27. Tarjeta de programación Teensy 3.6.....	47
Figura 28. Diseño PCB para Teensy 3.6.	50
Figura 29. Optointerruptor FC-03 (izq.), encoder de 20 agujeros (der.).....	52
Figura 30. IMU BNO05.	53
Figura 31. Esquemático de ángulos de navegación: yaw (ψ), pitch (θ), roll (ϕ).....	53
Figura 32. Ultrasonico HC-SR04.	54
Figura 33. Conversor lógico.....	55
Figura 34. Sensor óptico de distancia SHARP 2Y0A21.....	56
Figura 35. Diseño PCB para matriz de sensores químicos (izq.), Vista interna de la nariz electrónica (der.).....	58
Figura 36. Vista exterior de la nariz electrónica.	59
Figura 37. Batería Lipo de 7.4 V (izq.), power bank 20.000 mAh (der.).....	60
Figura 38. Motores DG02S 48:1.	61
Figura 39. Driver L298N.....	61
Figura 40. Primer nivel correspondiente al robot móvil autónomo.	63
Figura 41. Segundo nivel correspondiente al robot móvil autónomo.	64

Figura 42. Tercer y cuarto nivel correspondientes al robot móvil autónomo.	66
Figura 43. Diseño CAD del robot móvil autónomo.	66
Figura 44. Vista superior del robot móvil No. 1.	67
Figura 45. Vista frontal del robot móvil No. 1.	67
Figura 46. Vista lateral del robot móvil No. 1.	68
Figura 47. Vista posterior del robot móvil No. 1.	68
Figura 48. Diagrama de bloques correspondiente al software del sistema.	71
Figura 49. Topología de red del sistema robótico cooperativo.	72
Figura 50. Entorno de navegación y pruebas.	73
Figura 51. Referencia cartesiana del robot móvil.	74
Figura 52. Diagrama de flujo para valor del ángulo de giro de un robot móvil autónomo.	77
Figura 53. Posición final de un robot móvil.	78
Figura 54. Distancia de un obstáculo.	79
Figura 55. Navegación de robot móviles determinando zonas no exploradas.	81
Figura 56. Casos para determinar cinemática inversa del robot móvil autónomo.	82
Figura 57. Diagrama de flujo de envío y recepción de datos entre Teensy 3.6 y Raspberry.	87
Figura 58. Trama de datos entre Teensy 3.6 y Raspberry Pi 3.	88
Figura 59. Diagrama de flujo de la configuración para comunicación inalámbrica bidireccional entre Raspberry y ordenador centralizado.	90
Figura 60. Flujo del envío de datos desde Raspberry y recepción en ordenador centralizado.	90
Figura 61. Diagrama de flujo del envío de datos desde ordenador centralizado y recepción en Raspberry.	91
Figura 62. Trama de datos entre Raspberry Pi 3 y ordenador centralizado.	92

Figura 63. a) Trama en caso de haber detectado un obstáculo frente a un robot móvil, b) Trama en caso de haber detectado un obstáculo frente a un robot móvil.	94
Figura 64. Esquema del robot móvil como agente inteligente basado en metas.....	95
Figura 65. Diagrama de flujo para interacción de algoritmos de control y comunicación.	96
Figura 66. Pantalla de inicio, acceso de usuario y contraseña.	100
Figura 67. Pantalla de configuración de posiciones iniciales.....	101
Figura 68. Pantalla de mapeo en tiempo real.	102
Figura 69. Pantalla de resultados.	103
Figura 70. Puntos detectados como obstáculos y fronteras (izq.), Mapa procesado mediante grafos de visibilidad (der.).....	104
Figura 71. Zonas mapeadas con sus niveles de temperatura (izq.), Mapa 2D procesado mediante interpolación 2D y filtro espacial de media móvil (der.).....	106
Figura 72. Flujo para sensado general.....	109
Figura 73. Prueba de comunicación bidireccional serial USB.....	111
Figura 74. a) Escenario No. 1, b) Escenario No. 2, c) Escenario No. 3.....	117
Figura 75. Escenario No. 1: a) Navegación y mapeo de obstáculos en tiempo real, b) Mapeo de obstáculos mediante gráfica de píxeles, c) Mapeo de obstáculos mediante grafos de visibilidad.....	118
Figura 76. Escenario No. 1: a) Mapeo de temperatura, b) Mapeo 2D de temperatura mediante interpolación 2D y filtro espacial de media móvil.....	119
Figura 77. Escenario No. 1, sensor TGS2610: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.	120

Figura 78. Escenario No. 1, sensor TGS2600: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.	120
Figura 79. Escenario No. 1, sensor TGS2602: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.	121
Figura 80. Mapeo de obstáculos en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona	122
Figura 81. Mapeo de temperatura en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona ...	122
Figura 82. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona	123
Figura 83. Mapeo de concentración de gases mediante sensor químico TGS2600 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona	123
Figura 84. Mapeo de concentración de gases mediante sensor químico TGS2602 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona	124
Figura 85. Mapeo de obstáculos en Escenario 3: a) Alcohol b) Tolueno, C) Acetona	126
Figura 86. Mapeo de temperatura en Escenario 3: a) Alcohol b) Tolueno, C) Acetona	126
Figura 87. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario 3: a) Alcohol b) Tolueno, C) Acetona	127
Figura 88. Mapeo de concentración de gases mediante sensor químico TGS2600 en Escenario 3: a) Alcohol b) Tolueno, C) Acetona	127
Figura 89. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario 3: a) Alcohol b) Tolueno, C) Acetona	128

INTRODUCCIÓN

1.1. Antecedentes

En el ámbito del desarrollo tecnológico, la evolución de la Robótica se ha convertido en un aspecto central para disminuir la intervención de los seres humanos en tareas repetitivas, algunas de las cuales se tornan peligrosas por el entorno de trabajo en el cual se desempeñan. (Valencia & González, 2014). De este modo los robots vienen a ser herramientas esenciales que cumplen tareas precisas y específicas (Erazo, Matamoros, Minchala, & Paillacho, 2009), pudiendo destacar aplicaciones dentro del campo de la medicina, de construcción e incluso manejo de desastres. En los últimos años la presencia de los robots no solo se ve necesaria dentro de laboratorios, centros de investigación o en la industria, sino también dentro de los hogares (Wang, 2013).

En la actualidad, la robótica es una de las disciplinas que surgen a partir de la necesidad de minimizar trabajos difíciles, repetitivos y peligrosos para el ser humano, es el estudio de los robots autónomos o máquinas que son capaces de tomar sus propias decisiones para cumplir con un objetivo específico (Pons, 2012). De esta manera, se ha puesto un especial énfasis en los robots móviles conformados por un conjunto de ruedas conectadas a los motores y un procesador de control (Ajay & Venkataraman, 2013).

La investigación enfocada en la robótica, se ha transformado en la última década, pasando de los entornos simples y controlados a entornos exteriores con dinámica variable e impredecible. Para cumplir con esta transformación, se han desarrollado sistemas más robustos y eficaces tanto en sensores como en actuadores (Merino, Capitán, & Ollero, 2009).

Los robots móviles principalmente cumplen funciones de mapeo, localización y planificación de caminos. Esta actividad presenta una serie de problemas dentro de la robótica, lo cual ha

estimulado a introducir nuevas aplicaciones y soluciones a los mismos. En la temática de navegación y exploración de sistemas multi-robot es comúnmente usado el método de mapeo y localización simultánea *Simultaneous Localization and Mapping* (SLAM) (Atanasov, Ny, Daniilidis, & Pappas, 2015) definido como la construcción de un mapa correspondiente a un entorno desconocido y ocurre a la vez que el robot calcula su posición (Escalante, 2016).

Por su parte, se define a la odometría como el estudio de la estimación de la posición de vehículos con ruedas durante su navegación y para ello se usa información como la rotación de las ruedas o también mediante secuencias de imágenes por una o varias cámaras (Cruz, 2016), siendo la odometría uno de los métodos más utilizados para la localización de vehículos terrestres (Pretto, 2009). A pesar de su simplicidad y bajo costo, la odometría ha permitido mediante el uso de sensores especializados tales como brújulas y acelerómetros facilitar la navegación; y mediante encoders contar las revoluciones de las ruedas de un robot móvil lo cual es un factor clave para determinar desplazamientos, posición y orientación de un robot móvil (Fischer, Nitsche, & Pedre, 2014). Sin embargo, la odometría tiene una serie de problemas en cuanto al crecimiento sin control del error relativo (Azcón, 2003).

Los primeros proyectos e investigaciones relevantes sobre robótica distribuida y cooperativa surgieron durante la década de los 90's, dichas investigaciones trataron temas en sistemas de robots móviles múltiples incluyendo: robots celulares o reconfigurables, planeación de movimientos de multi-robots y arquitecturas para cooperación. Previo a este tiempo, la investigación se había concentrado en robots simples o sistemas de solución en problemas distribuidos (López Pérez, Mata Herrera, & García Pérez, 2001). Es importante destacar la robótica cooperativa como una herramienta útil en construcción de mapas o planos en ambientes desconocidos, misiones de búsqueda y rescate, operación de seguridad y sobrevivencia. Sin embargo, el reto se enfoca en la

necesidad de conocer la posición de cada uno de los robots que conforman el sistema cooperativo y el modo de comunicación más viable que se adapte a dicho sistema (Villa & Vázquez, 2014).

Existen aplicaciones de gran alcance que han sido desarrolladas en los últimos años, tales como: la minería, donde un robot sea capaz de movilizarse dentro de un túnel, el cual, y mediante sensores colocados dentro del mismo logre realizar un mapa describiendo caminos, extensión y variables ambientales (Cabrera & Delgado, 2014); ambientes semiestructurados, donde el robot presente la capacidad de moverse tanto en ambientes conocidos como desconocidos (Carimatto, Verrastro, & Gómez, 2013); entornos de desastre, donde el robot es capaz de navegar de forma remota (teleoperada) o en modo autónomo en entornos post-desastre de eventos como sismos, derrumbes y explosiones, realizando la búsqueda de víctimas humanas (Sánchez, 2013).

1.2. Justificación e Importancia

Actualmente está en desarrollo el proyecto de investigación 2016-PIC-009 titulado “LOCALIZACIÓN DE TNT Y PÓLVORA BASE DOBLE A TRAVÉS DE SENSADO QUÍMICO EN UN ENTORNO CONTROLADO MEDIANTE ROBÓTICA COOPERATIVA. (SmellRobSense)”, aprobado por la Universidad de las Fuerzas Armadas ESPE y dirigido por la Dra. Ana Guamán. Este proyecto se encuentra orientado en aplicaciones militares de seguridad y defensa permitiendo mediante robótica cooperativa, explorar zonas con explosivos y por ende peligrosas. Para lo cual se realiza un sensado de variables tales como: la temperatura, la densidad del humo, la concentración de sustancias explosivas y/o tóxicas, y obstáculos (Moya, 2017).

El proyecto se enmarca dentro de una de las áreas de investigación, desarrollo tecnológico e innovación de las Fuerzas Armadas como lo es la Soberanía y Defensa, y dentro de la línea de investigación de la Universidad de las Fuerzas Armadas ESPE en cuanto a Automática y Control.

En relación al proyecto mencionado se han desarrollado varios temas de investigación para la obtención del título de grado, dentro de los cuales se puede mencionar el tema de tesis “SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO PARA MAPEO 2D DE OBSTÁCULOS Y NIVELES DE TEMPERATURA EN UN AMBIENTE CONTROLADO” (Moya, 2017), cuyo objetivo fue mediante un sistema multi-robot explorar un entorno controlado y realizar un mapeo de obstáculos del mismo en el cual se visualice los niveles de temperatura en tiempo real.

El presente trabajo de titulación se enfoca en optimizar el comportamiento de los robots desarrollados en la tesis previamente citada en donde además de mejorar el rendimiento del sistema multi-robot, se plantea el rediseño de los algoritmos de navegación, posición y orientación utilizando nuevas plataformas robóticas con mejor desempeño. Además, el presente trabajo apoyará en las tareas propuestas en el proyecto 2016-PIC-009 mediante la integración de sensores capaces de captar la concentración de gases en un ambiente controlado, lo cual permitiría posteriormente a un robot especializado identificar químicos explosivos en las zonas exploradas.

Mediante la investigación sobre las tendencias y nuevas tecnologías utilizadas, se ha determinado la importancia de una implementación en este ámbito para el desarrollo de sistemas capaces de navegar autónomamente en zonas peligrosas y entornos desconocidos, esto determina un desafío de importante impacto en un gran número de ramas aplicables en situaciones reales tales como: tareas de búsqueda, rescate, exploración submarina y minera, así como en vigilancia y reconocimiento militar.

La robótica colectiva se ha enfocado principalmente en resolver las situaciones mencionadas; en consecuencia, las investigaciones se enfatizan en diseñar sistemas compuestos de varios robots capaces de resolver problemas conjuntamente. Los robots que forman parte de un sistema multi-robot son simples en términos de diseño y control, y menos costosos que los sistemas de un sólo robot especializado. Los sistemas multi-robot están orientados a resolver problemas en los cuales la participación de un solo robot no es suficiente o resulta ser muy costosa, en términos de diseño y tiempo, a manera de ejemplo se puede mencionar el transporte de objetos voluminosos, el manejo de material peligroso, la exploración y cobertura de terreno (Rodrigo, 2006). En la actualidad, dentro de la robótica móvil se desarrolla un creciente interés por resolver problemas de percepción los cuales son: conocer el entorno (mapeo del área) y saber dónde está el robot dentro del entorno (localización) (Ponnu & George, 2016).

La fusión e intercambio de información entre los diferentes robots favorece en primer lugar, la optimización y ahorro de recursos y en segundo lugar facilita el trabajo global del problema a resolver ya que, por ejemplo, mientras uno de ellos puede estar recopilando datos del terreno, otro puede irse desplazando por él y otro a su vez, ir moviendo obstáculos (Carrasco, 2015).

El presente proyecto plantea las bases fundamentales para el desarrollo de la robótica en el país, lo que servirá como una plataforma abierta para nuevos investigadores dentro y fuera de la Universidad de las Fuerzas Armadas ESPE.

La idea fundamental de utilizar robótica cooperativa en la detección de fuentes de olor radica en la utilización de robots dispuestos a cooperar en tareas especializadas como lo es la búsqueda y detección de fuentes de olor. Es por eso que la realización de la presente investigación se sustenta en trabajos previos presentes en la literatura, en las cuales se han obtenido diversos resultados en la temática de detección de fuentes de olor de sustancias químicas utilizando plataformas robóticas.

Por tal motivo es necesario tomar las investigaciones realizadas y dar continuidad a este tipo de estudios y su desarrollo.

La detección de fuentes de olor mediante la utilización de la robótica puede ser aplicada en un gran número de sectores, tales como: militar, donde se pueda localizar explosivos o drogas; sector alimenticio y farmacéutico, para mejoras de la calidad de los productos; entre otros. Esto contribuirá a la seguridad de las personas y procesos en los distintos ambientes y actividades.

Existen estudios relacionados con el uso de robots singulares y el uso de múltiples robots, donde se ha comprobado que la robótica cooperativa para estas aplicaciones presenta menores tiempos de desarrollo y mayor eficiencia. Esto ha generado estudios sobre la efectividad de grupos multi-robots comparada con versiones de robots singulares, teniendo aspectos a ser determinados por la variedad de enfoques disponibles dentro de localización, mapeo, detección, entre otras. La robótica cooperativa es empleada actualmente en la ejecución de varias investigaciones debido a la facilidad para trabajar con varios robots, los cuales son coordinados para la detección de fuentes de olor reduciendo tiempos de búsqueda. Desarrollándose además diversos sistemas de control, arquitecturas de comunicación que permiten disponer al conjunto de robots configuraciones centralizadas o distribuidas.

1.3. Alcance del Proyecto

Dentro del proyecto de titulación “SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO PARA MAPEO 2D DE OBSTÁCULOS Y NIVELES DE TEMPERATURA EN UN AMBIENTE CONTROLADO” (Moya, 2017), surgieron una serie de problemas tales como: desestabilidad de la brújula, transmisión de datos interrumpida, largos periodos de mapeo, inconsistencia en la calibración para la medición de pulsos generados por la rotación de los

encoders, la ausencia de un acelerómetro y problemas aleatorios ocasionados por la estructura de la plataforma.

En el presente proyecto, se plantea rediseñar el sistema propuesto por Carlos Moya (Moya, 2017) lo cual mejorará el desempeño en general del mismo, y reducirá los errores de muestreo y posicionamiento. El sistema será mudado a nuevas plataformas robóticas móviles de cuatro ruedas fijas y repotenciado con: sensores químicos que permitirán la medición de concentración de gases y acelerómetros que brindarán nuevos datos para obtener una mayor precisión en el posicionamiento de los robots, todo en tiempo real y dentro de un ambiente controlado.

Se reexaminará la función del controlador para que los autómatas cumplan las tareas mencionadas con el mejor comportamiento posible. La evaluación de desempeño significará una importante reducción de costos tras una disminución en el gasto de energía o incluso una mayor precisión en el procedimiento a ejecutar logrando una mayor eficiencia y eficacia en el sistema robótico (Pastor & Ayesa, 2011).

Para cumplir este propósito, se implementará un prototipo de plataforma robótica y posteriormente se desarrollará réplicas de dicha plataforma robótica, las cuales tendrán iguales capacidades y habilidades que el prototipo implementado.

Cada plataforma estará estructurada mediante sensores especializados en odometría tales como IMU's (sensores de medición inercial), sensores de distancia y sensores ópticos, que interpretarán la ubicación y orientación de los robots en un plano de dos dimensiones.

El sistema permitirá la visualización de valores sensados a tiempo real en una interfaz gráfica GUI (*Graphical User Interface*) basada normas y estándares de diseño, el usuario visualizará el acondicionamiento de los valores de temperatura y concentración de gases, así como el mapeo del entorno de navegación, todo dentro de un ordenador central de control del proceso.

Al ser un sistema online, los valores obtenidos serán transmitidos hacia el ordenador central de control mediante comunicación inalámbrica Wi-Fi, estos datos serán conocidos por todos los robots que conformen el sistema cooperativo con el objetivo de evitar redundancia en la navegación de los robots.

Se realizará pruebas con varios algoritmos de navegación y control para conseguir la identificación de un modelo adecuado y eficaz que reduzca el tiempo de escaneo y transmisión de valores relacionados al entorno tales como niveles de temperatura y concentración de gases captados el sistema multi-robot cooperativo.

1.4. Objetivos

1.4.1. Objetivo general

- Repotenciar el sistema multi robot móvil autónomo para el mapeo SMELLROBSENSE 2D de obstáculos, temperatura y niveles de concentración de gases en un ambiente controlado.

1.4.2. Objetivos específicos

- Adecuar en plataformas robóticas móviles, un sistema de control de navegación, posición y orientación mejorando respecto al proyecto de investigación previo, su desempeño en cuanto a tiempo de escaneo, adquisición de datos, y reducción de error.
- Integrar sobre las plataformas robóticas, sensores químicos que permitan detectar la concentración de gases en un entorno controlado.

- Realizar una búsqueda de nuevos sensores y compararlos con los sensores implementados en las plataformas robóticas desarrolladas en el proyecto de investigación previo, de manera que disminuya el error y aumente la precisión de los valores obtenidos por el nuevo sistema.
- Rediseñar la interfaz gráfica en un ordenador central de control que permita visualizar la posición de los robots y el mapa en dos dimensiones, tanto de los obstáculos, como de los niveles de temperatura y concentración de los gases.
- Integrar un sistema de comunicación inalámbrica mejorado y eficiente que permita al servidor tener la información adquirida por las plataformas robóticas en tiempo real y con la mínima pérdida de información.
- Procesar los datos obtenidos durante el sensado de robots para la visualización y caracterización del entorno respecto a los cambios de temperatura y concentración de gases.

CAPÍTULO II

ESTADO DEL ARTE

2.1. Introducción

La robótica es una rama de la ciencia e ingeniería cuyo objetivo es el diseño y construcción de robots para realizar tareas repetitivas que implican riesgo, aislando la mano de obra humana. En términos generales, se define como la integración de conocimientos electrónicos, informáticos, mecánicos y de ingeniería utilizados para construir dispositivos o máquinas que pueden programarse con el objetivo de interactuar con el entorno de forma automática (Pineda, 2009).

2.2. Agente inteligente

En esta investigación, el termino agente inteligente se utiliza indistintamente para referirse a entidades que sean capaces de generar acciones de forma razonada y autónoma; en una definición simple, un agente es una entidad que actúa y percibe sobre su entorno (ver Figura 1) , dicha definición es respaldada por numerosos investigadores del tema (Julián & Botti, 2000).

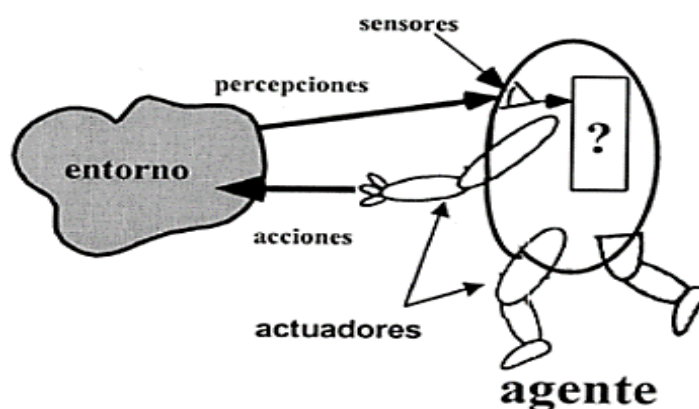


Figura 1. Visión esquemática de un agente inteligente.
Fuente: (Julián & Botti, 2000)

2.2.1. Clasificación de agentes inteligentes

A continuación, se detalla, como se clasifican los agentes inteligentes basados en su comportamiento para lograr objetivos predefinidos dentro de un determinado entorno.

- Agentes de reflejo simple: Son aquellos que se desempeñan de manera adecuada cuando las acciones son tomadas de manera correcta en base a la percepción de las condiciones del momento, es decir funcionan bajo reglas de condición-acción. En la Figura 2 se muestra un esquema correspondiente al funcionamiento de un agente de reflejo simple (Pineda, 2009).

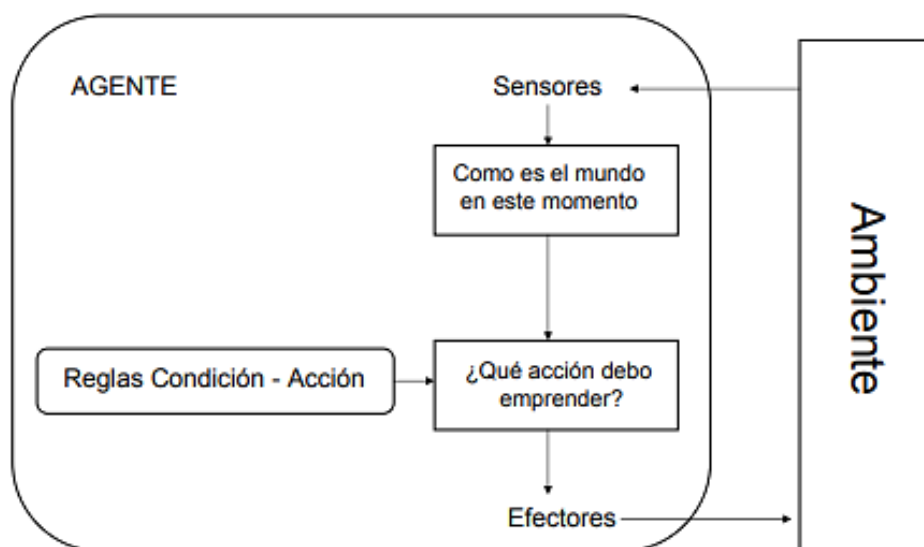


Figura 2. Esquema de un agente inteligente de reflejo simple.

Fuente: (Pineda, 2009)

- Agentes basados en metas: Los agentes basados en metas se definen en este estudio como aquellos que son capaces de modificar su comportamiento basándose en los estímulos que reciben, de este modo no es suficiente con conocer el entorno, también es fundamental determinar qué acciones deben ejecutarse para conseguir una meta u objetivo (Linares & Sánchez, 2008). En la Figura 3 se muestra un esquema correspondiente al funcionamiento de un agente de reflejo simple.

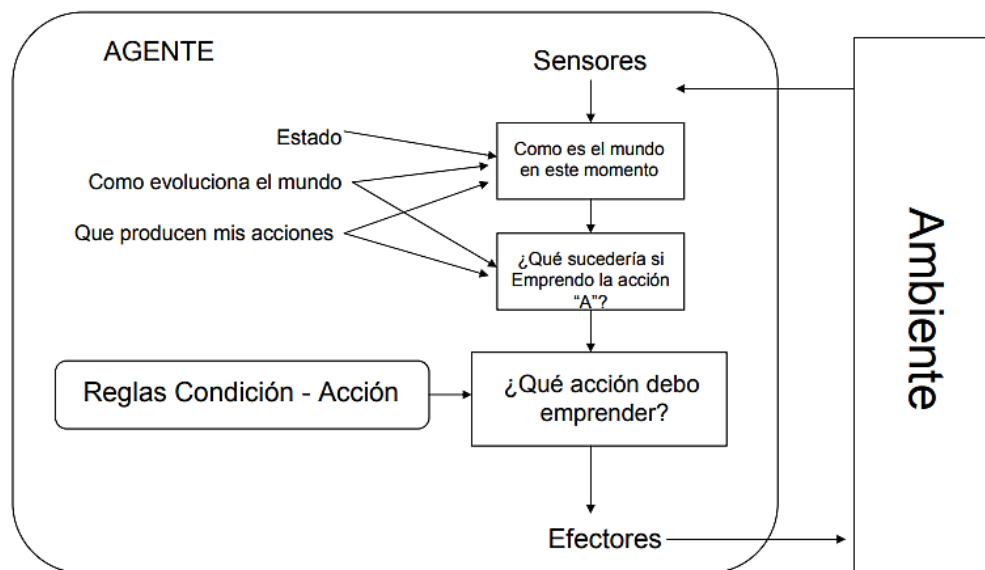


Figura 3. Esquema de un agente inteligente basado en metas.

Fuente: (Pineda, 2009)

- Agentes basados en utilidad: Se caracterizan por generar secuencias más óptimas para alcanzar una meta, en donde la utilidad se define como una función (f_U) que vincula un estado con un número booleano, el cual define el grado de satisfacción del sistema (0: meta no alcanzada, 1: meta alcanzada) (Linares & Sánchez, 2008).

2.3. Robótica móvil

La robótica móvil es un área que cobra cada vez más importancia en el estudio de sistemas centralizados desde un computador, para de esta manera, ser capaces de establecer un medio de comunicación con los humanos, al mismo tiempo interactuar con el entorno de navegación en el cual se encuentran y donde podrán llevar a cabo instrucciones predeterminadas en aplicaciones de carácter riesgoso, en este sentido cabe destacar a los robots teleoperados que han sido utilizados en acciones de reconocimiento y limpieza de zonas de desastre como los ocurridos en las plantas nucleares de Three Mile Island y Chernobyl (Bermudez, 2002).

2.3.1. Aplicaciones de la robótica móvil

Existen varios diseños de robots móviles desarrollados a lo largo de la historia los cuales están distribuidos en varios campos de aplicación y fines particulares, de esta manera se perfecciona nuevos algoritmos de navegación autónoma y guiado sensorial. A continuación, se detallan algunas de las principales aplicaciones en el campo de la robótica móvil:

- Limpieza: Las primeras aplicaciones de la robótica móvil, se encuentran en el campo de la limpieza, de este modo se introduce robots que tienen la función de aspirar pisos de gran extensión en las industrias (ver Figura 4); como consecuencia se programa patrones de navegación a través del espacio y algoritmos de navegación autónoma basados en el uso transductores infrarrojos que permiten detectar la presencia de obstáculos para posteriormente evitar chocar con ellos (Bermudez, 2002).



Figura 4. RoboScrub, para aspirar y limpiar pisos.

Fuente: (Bermudez, 2002)

- Seguridad: Esta clase de robots cuentan con características específicas que determinan su sistema de navegación tales como: GPS diferencial, encoders en las ruedas, giroscopios, radares, visión estéreo, ultrasonidos y balizas; en este sentido se acopla un sistema de seguridad que permite detección de intrusos basado en visión y en radar (Shoop, Johnston,

Goehring, Moneyhun, & Skibba, 2006). En cuanto a sistemas de seguridad se puede destacar al robot móvil (ver Figura 5) MDARS – E (MDARS – Exterior).



Figura 5. Vehículo prototipo MDARS-E.

Fuente: (Shoop, Johnston, Goehring, Moneyhun, & Skibba, 2006)

- Exploración: Desde la perspectiva de exploración de entornos desconocidos, es importante evaluar los peligros que se asumen al realizar ciertas tareas, en este sentido no resulta conveniente que sean realizadas por humanos; de esta situación se concibe introducir robots espaciales flexibles caracterizados por contar con brazos robóticos para la manipulación y locomoción para la exploración. Pese a que existen robots móviles terrestres con condiciones similares, los robots móviles espaciales (ver Figura 6) difieren de estos por su estructura, la cual debe soportar condiciones exteriores extremas (Navarro, 2005).



Figura 6. Robot explorador lunar Lunokhod 1.

Fuente: (Navarro, 2005)

- Militar: El propósito de gran parte de los robots móviles es salvaguardar la vida de las personas; frente a esta situación se logra el desarrollo de robots que tienen como propósito, remover pequeñas bombas y granadas que no detonan, pero constituyen un inminente riesgo para quienes se encuentran alrededor (ver Figura 7), de esta manera cabe destacar el éxito que tienen estos robots en navegación autónoma, búsqueda en espiral, levantamiento de mapas y transporte de objetos probablemente peligrosos (Bermudez, 2002). En la actualidad la tendencia en el sector militar es sustituir con robots a toda una infantería, de este modo los robots cumplen con funciones de vigilancia o combate, detectan la presencia de enemigos sin requerir actividad humana y fundamentalmente evitan una baja en vidas humanas (Arranz, y otros, 2006).



Figura 7. Robot soldado TALON.

Fuente: (Arranz, y otros, 2006)

- Construcción: En términos generales se obtiene grandes ventajas de la automatización en el campo de la construcción debido a la repetibilidad de tareas y la complejidad del entorno. A pesar de esto el nivel de automatización en el sector productivo de la construcción sigue siendo uno de los más bajos, por lo tanto, el proceso aún se caracteriza por centrar su trabajo

en la mano de obra. De esta manera, en los últimos años los estudios se enfocan principalmente en los campos de la edificación y obra civil en donde cabe destacar la automatización en construcción de túneles, carreteras y puentes mediante asfaltadoras (ver Figura 8) y apisonadoras dotadas por GPS, sensores de compactación y sensores de densidad, lo que permite tener movilidad autónoma con guía sensorial de gran precisión (Arranz, y otros, 2006).



Figura 8. Asfaltadora Cedarapids® utilizada en Ingeniería Civil.

Fuente: (Arranz, y otros, 2006)

- Investigación: La robótica móvil despierta un gran interés en tratar temas de investigación que no son completamente abordados y que logran obtener resultados satisfactorios utilizando sistemas multi-robots autónomos distribuidos. Por esta razón, la robótica móvil cooperativa comienza después de la formulación de un nuevo modelo de control basado en el comportamiento fundado en inspiraciones biológicas.

Dentro del análisis de varios investigadores, sus estudios evalúan las características sociales de los animales, de allí es posible destacar varios hallazgos en base al comportamiento de colonias de insectos (ver Figura 9), de esta manera se consigue reglas de control de varias

sociedades biológicas de abejas, hormigas, escarabajos, etc. (López Pérez, Mata Herrera, & García Pérez, 2001).

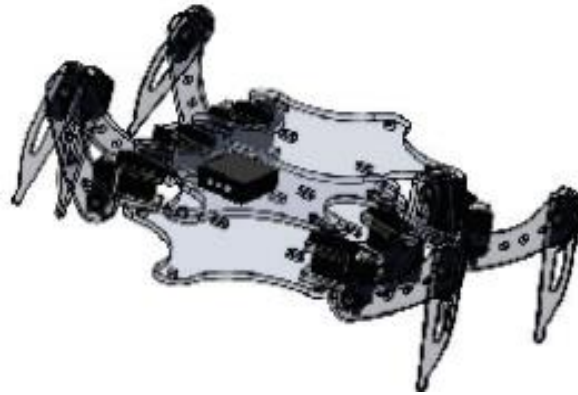


Figura 9. Robot hexápodo bioinspirado en el Tenebrio.

Fuente: (Rodríguez, Ramos, & Peña, 2015)

2.3.2. Clasificación de robots móviles por su estructura

Para la fundamentación teórica del presente proyecto de investigación, esta clasificación se limita únicamente a robots terrestres dotados con ruedas, dichos agentes tienen la ventaja de poseer mayor estabilidad mecánica y de control, de esta manera se presentan cuatro estructuras básicas explicadas a continuación:

- **Diferencial:** Es la más simple ya que se caracteriza por tener dos ruedas motrices que se ubican de manera simétrica permitiendo al robot avanzar, frenar o retroceder (ver Figura 10). En cuanto a la locomoción, es necesario utilizar puntos de apoyo adicionales o implementar un sistema de control de estabilidad y equilibrio, asimismo debe contar con una superficie homogénea para conseguir la misma alimentación en ambos motores y por consiguiente la misma tracción en ambas ruedas (Gómez, 2011).



Figura 10. Robot móvil diferencial.

Fuente: (Moya, 2017)

Existen también robots móviles de configuración diferencial que constan de 4 ruedas idénticas ubicadas simétricamente (ver Figura 11), en donde cada una consta de su propio motor; en este caso ya no es necesario buscar puntos de apoyo o equilibrio adicional como en las plataformas diferenciales de dos ruedas ya que al poseer 4 ruedas, las mismas son más que suficientes para dar el equilibrio necesario al robot.



Figura 11. Robot móvil SUMMIT XL Robotnik®.

Fuente: (robotnik, 2015)

- **Omnidireccional:** Permite realizar movimientos complejos cambiando de dirección sin la necesidad de cambiar de orientación, entre dichos movimientos figuran: giros de trescientos sesenta grados, desplazamientos hacia delante, atrás, izquierda o derecha (Moya, 2017), además consta de diferentes configuraciones que pueden ser de 3 o 4 ruedas (ver Figura 12), dichas ruedas son conocidas como omnidireccionales o ruedas suecas las cuales son un

requisito fundamental de la estructura y permiten los movimientos o desplazamientos en todos los sentidos.



Figura 12. Robotino® de Festo.
Fuente: (Zambrano, 2015)

- Síncona: Se caracteriza por el hecho de que, al girar, el chasis o superficie no cambia de orientación (ver Figura 13), es decir la orientación logra ser totalmente independiente de la ubicación del robot (Gómez, 2011).



Figura 13. Robot de configuración síncrona.
Fuente: (Gómez, 2011)

- Ackermann: Se describe convencionalmente como un robot móvil que consta de cuatro ruedas al igual que los automóviles actuales comunes (ver Figura 14), en donde las ruedas delanteras están gobernadas por un sistema de dirección que a su vez coordina el nivel de

curvatura en la trayectoria del robot, las ruedas posteriores constan de un sistema de tracción que permite formular la velocidad de traslación (Moyolema & Pereira, 2014).



Figura 14. Robot móvil de configuración Ackermann.
Fuente: (Moyolema & Pereira, 2014)

2.4. Robótica cooperativa

La robótica cooperativa cobra cada día más relevancia dado que su estudio aborda el desarrollo de sistemas de optimización en acciones de localización, mapeo y navegación autónoma entre dos o más robots. Estos grupos multi-robots a comparación de las versiones de robots singulares, comprenden un sistema de comunicación que puede ser centralizado o descentralizado, esto apoya a la toma de decisiones con redundancia y confiabilidad en las acciones que los denominados agentes inteligentes pueden proporcionar del entorno en el cual operan; su propósito principal es dimensionar sistemas de varios robots que sean capaces de asumir y resolver problemas de manera conjunta (ver Figura 15), de esta manera se trasciende sobre las acciones desarrolladas por un solo robot especializado logrando alcanzar objetivos con un menor costo en cuestión de tiempo y diseño (Rodrigo, 2006).

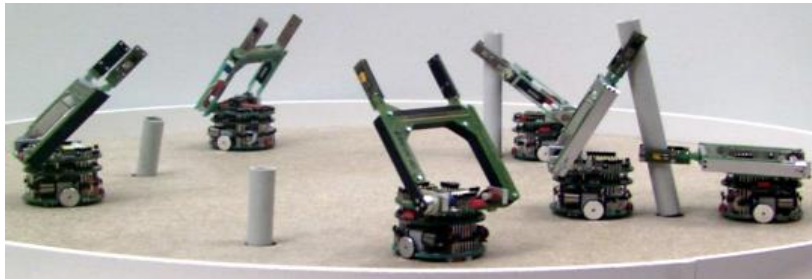


Figura 15. Grupos de robots encargados de levantar columnas.

Fuente: (Rodrigo, 2006)

2.5. Robótica móvil cooperativa

Se basa en la utilización de una configuración integrada por dos o más robots móviles que realizan tareas programadas o predefinidas mediante interacción y comunicación entre ellos, están estructurados mecánicamente de tal forma que se permita desplazamientos en un entorno de navegación controlado o desconocido.

Los sistemas robóticos cooperativos están formados por 2 componentes que interactúan directamente: agentes inteligentes y entorno de navegación, el entorno determina a partir de detección sensorial, que acciones son tomadas por parte de los robots móviles y de qué manera es procesada la información recolectada por cada agente. Entre las principales características que determinan la configuración de un sistema robótico móvil cooperativo se encuentran: planeación de movimientos, nivel de autonomía, tipo de comunicación, sistemas de control y modelos de control utilizados.

2.5.1. Planeación de movimientos

El concepto de planeación o planificación de movimientos de un robot móvil se posiciona como uno de los principales sistemas de control; en la mayoría de los casos la navegación de los robots

móviles es ejecutada en ambientes controlados ya que de esta manera es posible lograr secuencias de movimientos que sean interpretados correctamente por el agente o el sistema multi-robot. Dentro del análisis del entorno de navegación cabe resaltar el nivel de autonomía necesario y a su vez la definición de los sensores, actuadores y controladores que dimensionen los algoritmos más óptimos de navegación (Holman, Edwar, & Martínez, 2015).

2.5.2. Autonomía

La autonomía en la robótica móvil se define generalmente como un sistema de navegación de manera automática que es capaz de asumir acciones de precepción del entorno, planificación de rutas y control (Baturone, 2005); la autonomía se logra con la menor supervisión posible por parte del ser humano y considerando que este campo se comprende de cuantiosos desafíos, se proyecta obtener la mayor percepción del entorno para determinar el posicionamiento, navegación y cálculo de trayectorias (Fernández, Fernández, & Valmaseda, 2010).

2.5.3. Comunicación

Hasta la actualidad se evalúa nuevas tendencias y tecnologías de comunicación inalámbrica entre dispositivos, cabe destacar diferentes tipos de comunicación como Bluetooth, infrarrojos o Wifi, mismos que a pesar de no aportar las mismas prestaciones de una comunicación por cable, presentan velocidades de transmisión aceptables en el campo de la robótica (Roca, 2012).

La comunicación es un ámbito primordial en la robótica cooperativa tomando en cuenta cual es la capacidad que tienen los robots para intercambiar información entre ellos, de esta manera mediante la aplicación a la cual este destinado el sistema multi-robot, se puede establecer qué tipo

de comunicación es el más óptimo caracterizándose por 2 tipos: control centralizado y control distribuido.

2.5.3.1. Control centralizado

Se basa en una sucesión de procesos en los cuales los sensores se encargan de recibir la información, un controlador la procesa y evalúa el próximo movimiento a ejecutarse, basándose en las tareas inconclusas, las nuevas tareas y los obstáculos reconocidos en el entorno (Michalczewsky & Fillottrani, 2001). Los sistemas multi-robot cooperativos basados en control centralizado poseen módulos que se comunican con un ordenador central en donde se supervisa todos los problemas de configuración y movimiento con algoritmos altamente elaborados pero que limitan la autonomía de los módulos (Moreno, 2010).

2.5.3.2. Control distribuido

Se define como un conjunto de procesamientos del sistema multi-robot, en donde cada procesamiento es un agente diferente que trabaja independiente del resto y en paralelo a los demás robots (Báez, 2012). Dentro del análisis en el ámbito de comunicación en robótica móvil, el control distribuido se caracteriza por el hecho de que cada robot posea su módulo con su propio algoritmo de control y a su vez sea capaz de comunicarse con los demás robots para realizar un propósito cooperativo en común, por estas razones el control distribuido es el formato más usado de control ya que evita problemas como el embotellamiento en las comunicaciones y brinda mayor autonomía a los módulos (Moreno, 2010).

2.5.4. Modelos de control

Debido a la necesidad de obtener mecanismos enfocados en la funcionalidad e inteligencia de un agente o robot móvil, surge tres paradigmas de diseño que determinan el modelo de control. Cada paradigma o modelo de control posee sus ventajas y desventajas tal como se indica en la Tabla 1 (Berná, Pérez, Gilart, Ramos, & Ferrándiz, 2008).

Tabla 1

Ventajas y desventajas de los modelos

Modelo de control	Ventajas	Desventajas
Deliberativo	Robustos en problemas específicos	Mientras mayor sea la complejidad, mayor es la planificación del agente inteligente
Reactivo	Capaz de afrontar situaciones nuevas	Requiere un modelo de mundo cerrado lo que genera incertidumbre en las acciones
Híbridos	Rápida reacción ante comportamientos complejos	Complicada coordinación entre nivel deliberativo y reactivo.

Fuente: (Berná, Pérez, Gilart, Ramos, & Ferrándiz, 2008).

2.5.4.1. Control Deliberativo

Los sistemas de control deliberativo o jerárquico tienen su origen en las primeras aplicaciones de sistemas multi-robot móviles. El término de control deliberativo se basa en la adquisición de un modelo exacto y preciso del entorno de navegación del robot, es decir que dicho robot cuenta con un sistema de espacio de estados en donde a cada estado le corresponde una meta u objetivo (Pérez, 2003), de esta manera las acciones del robot ya se encuentran preestablecidas como un plan o una receta en la cual las acciones siguen un orden específico. Los autores (Molina & Matellán, 1996) definen a un planificador como un agente capaz de gestionar un plan de forma automática mediante una sucesión o secuencia de tareas que tienen como objetivo el logro satisfactorio de una meta, esta definición concuerda con la base del control deliberativo, tal como se indica en la Figura 16.

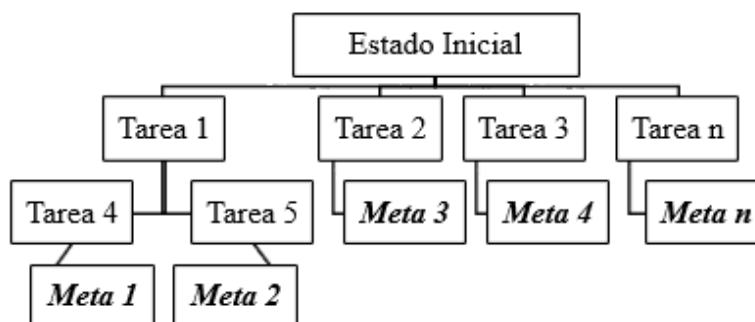


Figura 16. Representación esquemática de la búsqueda de un plan.

Fuente: (Molina & Matellán, 1996)

2.5.4.2. Control reactivo

La arquitectura de control reactivo en robótica móvil cooperativa se describe convencionalmente como aquellos agentes inteligentes que son capaces de reaccionar ante un estímulo en donde si el estímulo desaparece, de igual manera desaparece la reacción. Como consecuencia de este comportamiento los agentes se enfrentan a dos problemas en un medio real los cuales son: la dificultad de localizar a un robot móvil dentro de un entorno con coordenadas precisas y la incertidumbre sensorial de la exploración en dicho entorno.

En la Figura 17 se tiene el ejemplo de un robot móvil que identifica dos posibles acciones, seguir un objeto determinado o evitar obstáculos del entorno, como consecuencia, si se asume que la acción de seguir un objeto debe ser ejecutada, por consiguiente la acción de evitar obstáculos es bloqueada (Molina & Matellán, 1996).

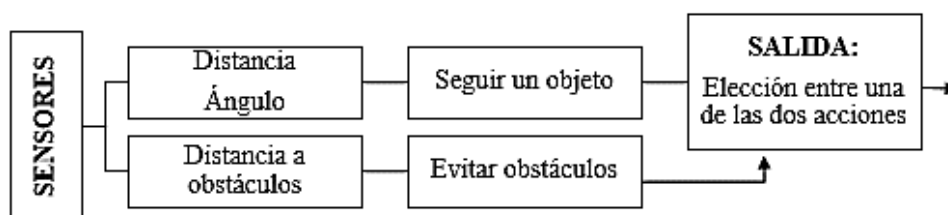


Figura 17. Ejemplo de arquitectura reactiva de un agente inteligente.

Fuente: (Molina & Matellán, 1996)

De igual manera se puede destacar al control reactivo como un comportamiento conformado por un conjunto de estímulos sensoriales en donde a cada estímulo le corresponde un patrón de acciones motoras necesarias para llevar a cabo una tarea, por lo tanto, es importante determinar todos los comportamientos que el agente inteligente debe asumir en un determinado entorno, en la Tabla 2 detallada a continuación se indican las fases para determinar los comportamientos de un robot móvil.

Tabla 2

Fases para determinar los comportamientos de un robot móvil

1. Especificación y Análisis	Descripción de las acciones
	Descripción del tipo de robot móvil
	Descripción del entorno
2. Diseño	Descripción de la reacción del robot móvil con el entorno o ambiente de navegación
3. Implementación	Establecimiento e implementación de comportamientos
	Prueba de cada comportamiento implementado
4. Integración	Acople de todos los comportamientos
	Prueba de todos los comportamientos acoplados

Fuente: (eva.fing, 2013)

2.5.4.3. Control híbrido

Esta fusión contempla diferentes tipos de restricciones temporales entre las tareas deliberativas y reactivas; las tareas reactivas cumplen con limitaciones temporales estrictas mientras que las tareas deliberativas no necesitan de un control temporal minucioso, es decir, si no cumple sus tiempos de ejecución el sistema deliberativo no se expone a riesgos de falla ya que cuenta con un sistema reactivo que se encarga de llevar los procesos bajo control (eva.fing, 2013).

Los agentes con un sistema de control híbrido se dimensionan bajo dos operaciones: reactivas dirigidas por los sensores y deliberativas para planificar operaciones y objetivos, de esta manera los robots logran planear la mejor forma de proyectar las tareas y dividirlas en subtareas, después

identifican mediante información proporcionada por los sensores, cuáles son los comportamientos más adecuados para ejecutar cada subtarea, dicha información sensorial además de formular los comportamientos, también está disponible para el proceso de planificación en la construcción de un modelo del entorno (Posadas, y otros, 2018).

2.6. Algoritmos de navegación, mapeo y localización (SLAM)

En el diseño de un agente inteligente existen diversos métodos y algoritmos de navegación que permiten la extracción de información del entorno de trabajo, dichos métodos se centran en la utilización de sistemas dotados de sensores de ultrasonidos y sistemas de visión artificial.

Entre los requerimientos fundamentales de los algoritmos de navegación se encuentra la estimación de la localización y orientación de un robot móvil autónomo en cada instante de tiempo; por estas razones, en el campo de la robótica móvil se usa métodos incrementales entre los cuales destaca la odometría, cuyo problema principal radica en la acumulación de errores durante la trayectoria, sin embargo, pueden ser resueltos mediante control de ajuste y corrección de errores (Armingol, 1997).

2.6.1. Algoritmos basados en técnicas geométricas

Durante el proceso de navegación de un robot móvil autónomo, dicho agente obtiene la representación geométrica de la información asociada al entorno en el cual se encuentra mediante un sistema sensorial sofisticado, posteriormente dicha información es procesada mediante modelos de integración basados en cinemática directa e inversa que permitan acoplar toda la información.

De este modo existen métodos geométricos que consisten en la construcción de grafos que interconecten el origen con el destino para posteriormente buscar el camino más óptimo o por lo

general la distancia más corta al objetivo (Espitia, 2011). A continuación, se detallan los diferentes métodos.

2.6.1.1. Grafos de visibilidad

En un entorno de dos dimensiones, cada nodo representa una configuración factible del robot y los obstáculos son representados por polígonos (ver Figura 18); si no se interpone ningún obstáculo entre dos nodos, dichos puntos se pueden unir mediante un segmento de recta lo cual se conoce como un grafo de conectividad entre vértices que son visibles, esto permite generar la mejor ruta de navegación, pero con la consecuencia de generar posibles rutas muy cercanas a obstáculos (López, 2012).

Para utilizar este método como algoritmo de navegación, el robot debe tener la capacidad de poder girar sobre su propio eje tal y como lo hacen los robots móviles de configuración diferencial, esto se debe a que el robot avanza en línea recta por el camino trazado entre dos nodos y cada que llega a un vértice, se orienta sin abandonar dicho vértice (Espitia, 2011).

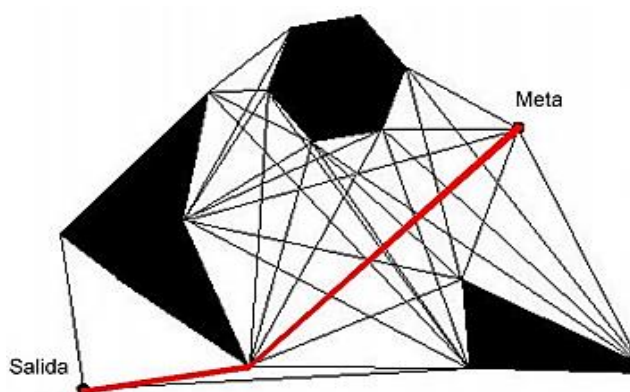


Figura 18. Grafo de visibilidad.

Fuente: (López, 2012)

2.6.1.2. Diagramas de Voronoi

Este método busca generar líneas de ruta para navegación lo más lejanas posible de los obstáculos presentes en el entorno (ver Figura 19), el grafo se construye mediante segmentos rectilíneos o parabólicos que dependen de la forma de los obstáculos y la distancia que equidisten entre ellos (López, 2012); si se elige este método como algoritmo de navegación, el robot móvil debe tener gran capacidad de maniobra para permitir giros abruptos con la finalidad de mantenerse alejado de los obstáculos en los vértices que así lo dispongan (Espitia, 2011).

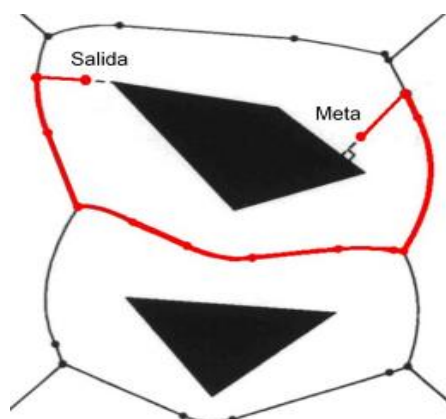


Figura 19. Grafo de Voronoi.
Fuente: (López, 2012)

2.6.1.3. Descomposición en celdas

El propósito de este método es descomponer los espacios vacíos en celdas desde el punto inicial hasta el punto final de la ruta a trazar (ver Figura 20), de esta manera se estima encontrar una sucesión de celdas que no presente discontinuidad, y mediante un grafo se estima la conectividad de dichas celdas; este tipo de algoritmo puede ser de dos tipos: si se crea celdas de forma pre establecida donde la unión de las mismas no es precisamente un espacio libre, el método es

aproximado, al contrario, si se descompone el espacio libre en celdas donde la unión corresponde a un espacio libre, el método es exacto (Espitia, 2011).

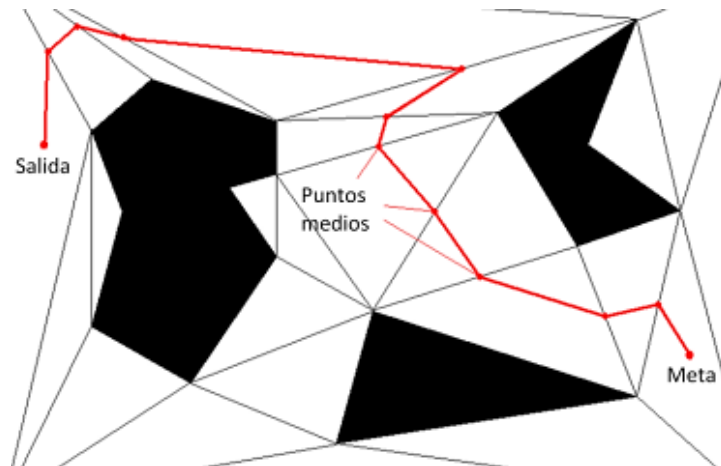


Figura 20. Grafo por descomposición de celdas.

Fuente: (López, 2012)

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DE LAS PLATAFORMAS ROBÓTICAS

3.1. Introducción

En este capítulo se explica el proceso de diseño e implementación de las plataformas robóticas móviles, autónomas y cooperativas encargadas de resolver la problemática descrita en los objetivos del capítulo 1. Se detalla características de diseño la arquitectura de los robots tales como: la plataforma a utilizar, controladores, sensores, actuadores, módulos de comunicación, fuentes de alimentación y montaje.

3.2. Antecedentes

A continuación se detallan las características más importantes del proyecto de investigación previamente desarrollado, denominado “Sistema robótico móvil autónomo cooperativo para mapeo 2d de obstáculos y niveles de temperatura en un ambiente controlado” (Moya, 2017).

- El sistema general se conformó por 3 plataformas robóticas móviles de configuración diferencial con dos ruedas y un punto de apoyo.
- Cada plataforma robótica se implementó con una torreta conformada por un servomotor y dos sensores ultrasónicos, cuya función se basa en el escaneo de obstáculos en 360 grados mediante giros continuos de la torreta.
- Para la detección y evasión de obstáculos frente al robot móvil, se utilizó un sensor ultrasónico adicional.

- Para la detección de giros, en cada plataforma robótica se implementó un magnetómetro o brújula utilizando como referencia el norte magnético.
- Para la transmisión de datos se utilizó un módulo Wifi ESP8266, del cual, mediante comandos AT y protocolo TCP, se realizó la comunicación inalámbrica hacia un ordenador central.
- Se utilizaron dos tarjetas programables (Arduino Mega y Arduino Nano), encargadas de la adquisición de datos, procesamiento, comunicación y activación de actuadores.
- Para la adquisición de la temperatura ambiente de un entorno, se instaló en cada plataforma un sensor DS18B20.
- La estimación de la odometría se realizó mediante la adquisición de pulsos incrementales de encoders instalados en sus ruedas.

3.3. Requerimientos de funcionamiento

Se ha planteado ciertos parámetros de funcionamiento para determinar qué tipo de plataforma robótica es la que mejor se ajusta a las necesidades y requerimientos del sistema, de igual forma para determinar qué tipo de sensores, actuadores, módulos y alimentación son los requeridos con la finalidad de repotenciar las características de las plataformas desarrolladas en el proyecto de investigación previamente desarrollado y que fueron mencionadas en el apartado anterior. De forma general, se detalla a continuación cuales son las necesidades de las nuevas plataformas robóticas móviles:

- Es necesario disponer de una plataforma robótica móvil de reducido tamaño y lo suficientemente robusta como para soportar el peso de sensores, actuadores, controladores, baterías y una nariz electrónica fabricada de aluminio.
- El robot móvil debe ser capaz de realizar movimientos rectilíneos constantes entre cada escaneo, la distancia fija que debe recorrer se establece dependiendo de las dimensiones de la plataforma, así como las dimensiones del espacio de trabajo o zona de navegación.
- Se debe implementar un sistema para detección y evasión de obstáculos o fronteras alrededor de todo el robot móvil, de la misma manera debe ser posible determinar cuál ha sido la distancia recorrida en caso de haber detectado un obstáculo frente al robot y el mismo obligue a detener su navegación.
- Es indispensable conocer el valor de los ángulos de giro que realiza el robot, tanto a la derecha como a la izquierda dependiendo de las condiciones del entorno.
- El sistema de sensado debe obtener en cada escaneo del robot móvil el valor de la temperatura ambiente, de igual forma, mediante la utilización de una matriz de sensores químicos ubicados dentro de la nariz electrónica, se debe obtener los valores de concentración de gases correspondientes al espacio explorado en el entorno de navegación.
- El tiempo de autonomía requerido para cada robot móvil es de 1 hora, durante la cual el comportamiento de los motores respecto a velocidades y vencimiento de torque debe ser constante, se considera que deja de ser constante cuando a pesar de aún tener carga en las baterías, los motores no vencen el torque para producir movimiento o existe variación en las velocidades predefinidas.

- Para un correcto funcionamiento es importante distinguir la etapa de adquisición de datos mediante sensado, activación de actuadores y la etapa de procesamiento de información, para ello es necesario analizar el uso de dos o más controlares que conformen al sistema robótico cooperativo.

Para el diseño e implementación es necesario considerar algunos puntos importantes para brindar la mayor eficacia y optimización posible; entre las características primordiales analizadas se encuentran: dimensión del robot, equilibrio, funcionalidad, eficiencia, adaptabilidad, maniobrabilidad y la arquitectura mecánica con la finalidad de crear un robot que sea capaz de navegar autónomamente y levantar mapas de un entorno con temperatura variable y dotado de muestras de alcohol, tolueno y acetona. A continuación, se describen las características de diseño:

- **Arquitectura mecánica:** al tratarse de un tema de investigación en el campo de la robótica móvil, es viable utilizar robots terrestres dotados de ruedas cuya plataforma sea capaz de soportar el peso correspondiente a los controladores, sensores, actuadores y fuentes de alimentación necesarias.
- **Equilibrio:** las plataformas deben poseer una distribución de peso adecuada con el objetivo de conseguir desplazamientos en línea recta, para ello también es necesario mantener velocidades de tracción idénticas en todas las ruedas.
- **Funcionalidad:** el diseño del robot móvil debe ofrecer facilidad, utilidad y comodidad en su funcionamiento el cual se basa en la capacidad de sensar la concentración de temperatura y gases en el ambiente, levantar un mapa del entorno y poder visualizarlo a tiempo real en una interfaz gráfica amigable al usuario.

- Eficiencia: la robótica móvil cooperativa busca cumplir adecuadamente objetivos con redundancia en los resultados esperados y en el menor tiempo posible, de esta manera es necesario la construcción de tres robots con características físicas iguales y mismo montaje.
- Adaptabilidad: los robots deben proyectarse a investigaciones futuras asumiendo la capacidad de poder adaptar nuevos componentes o retirar los ya instalados, además de la posibilidad de navegar en nuevos entornos de trabajo.
- Fiabilidad: los robots deben contemplar una precisión adecuada en sus medidas, por lo cual es necesario el uso de sensores de buena resolución que permitan obtener un sistema más óptimo en comparación a las plataformas robóticas mencionadas en el capítulo 1 y desarrolladas por Carlos Moya (Moya, 2017).
- Maniobrabilidad: el espacio de navegación de los robots móviles es un entorno controlado, sin embargo, es desconocido para los agentes, por esta razón es importante que se coordinen con una gran maniobrabilidad para evitar chocar contra obstáculos y entre ellos.

Para cumplir con lo requerido se plantea una plataforma con las siguientes características y subsistemas (ver Figura 21): sistema mecánico, sistema encargado del control, sistema de sensado y sistema de potencia, los cuales se integran para cumplir con las tareas de navegación autónoma, sensado, alimentación autónoma, control y procesamiento de datos recolectados.

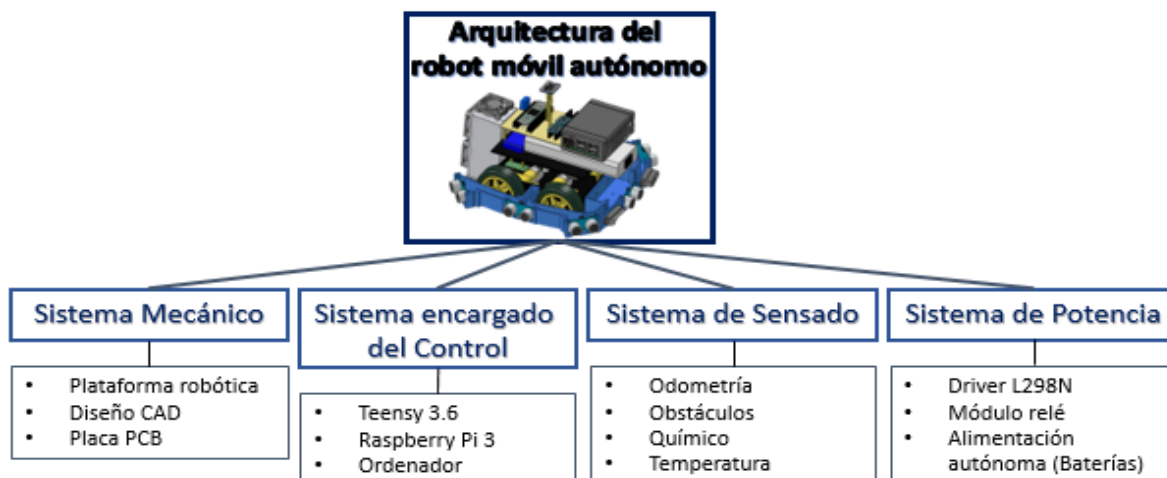


Figura 21. Subsistemas de la arquitectura de un robot móvil autónomo.

3.4. Sistema mecánico

El objetivo principal del presente proyecto de titulación es la optimización del “Sistema robótico móvil autónomo cooperativo para mapeo 2D de obstáculos y niveles de temperatura en un ambiente controlado” (Moya, 2017), en el cual se utilizaron plataformas de configuración diferencial conformadas por dos ruedas con sus respectivos motores y una tercera rueda utilizada como punto de apoyo (ver Figura 22), la cual ocasionaba problemas para desplazamientos en línea recta, y por ende una mala apreciación de la orientación. Para continuar con la investigación, se optó por utilizar las plataformas DG012-ATV 4WD de configuración diferencial las cuales están conformadas por cuatro ruedas en lugar de dos; esta arquitectura es la más viable para utilizarse en el sistema robótico móvil cooperativo planteado en el capítulo 1 por tener la ventaja de generar equilibrio sin la necesidad de puntos de apoyo, además de ser un sistema fácil de implementar y de diseño simple.

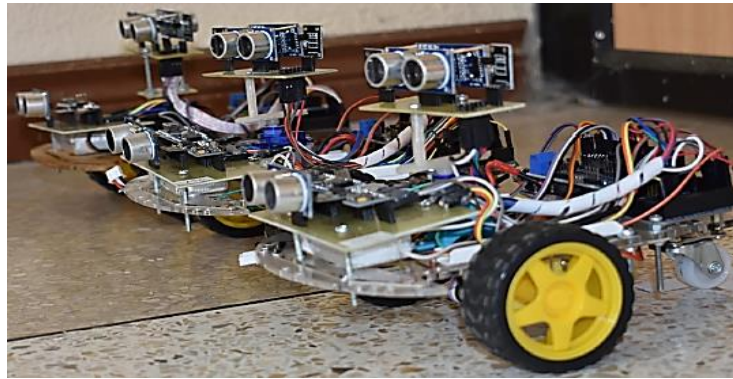


Figura 22. Robots móviles de configuración diferencial.
Fuente: (Moya, 2017)

3.4.1. Plataforma robótica móvil DG012-ATV 4WD

Se implementó 3 robots móviles autónomos cuya estructura se construyó teniendo como base a la plataforma robótica DG012 versión ATV 4WD (ATV en inglés *all-terrain vehicle*, 4WD en inglés *Four Wheel Drive*) de la familia DAGU, la Figura 23 muestra cómo se compone la plataforma en su interior y ya armada.



Figura 23. Plataforma robótica móvil DG012-ATV (izq.), vista interna del chasis (der.)
Fuente: (Robots, 2018)

La plataforma DG012-ATV está constituida por 4 ruedas ATV lo cual la hace ideal y maniobrable para ser operada mediante un sistema Arduino; su chasis resistente y liviano está fabricado de aluminio y posee varios puntos perforados para permitir el montaje de estructuras

extra que permitan la instalación de baterías, controladores, sensores y actuadores (Robots, 2018).

En la Tabla 3, se detallan las principales características técnicas de la plataforma DG012-ATV 4WD.

Tabla 3

Especificaciones técnicas de la plataforma DG012-ATV 4WD

Dimensiones	178 x 145 x 78 mm
Peso	580 g
Carga de trabajo	2 kg máx.
Tipo de transmisión	ATV ruedas (4WD)
Diámetro de la rueda	78 mm
Motores	Dagu DG02S
Par motor	0,8 kg.cm
Relación de engranajes	48:1
Velocidad sin carga	320 rpm a 7.2 V
Consumo sin carga	140 mA
Suministro	6 - 9 Vcc (recomendable 7,2 Vcc)

Fuente: (Robots, 2018)

3.4.2. Diseño CAD

En la fase inicial de desarrollo del proyecto de investigación, se diseñó en el software CAD *Solidworks* el prototipo de un robot móvil autónomo mediante modelado 3D el cual consiste en la estructura física de cada plataforma robótica detallando la ubicación de todos sus componentes tales como: controladores, sensores, actuadores, baterías, nariz electrónica, entre otros, de esta manera se tiene una idea previa de la arquitectura de cada agente y su comportamiento, en la Figura 24 se indica la versión del prototipo de la plataforma robótica.

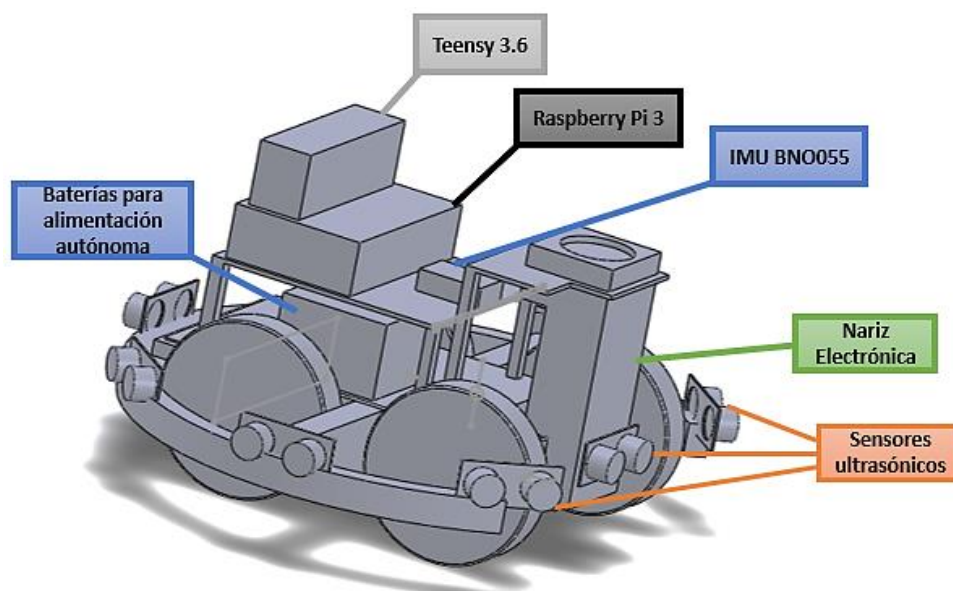


Figura 24. Diseño CAD del prototipo de un robot móvil autónomo

Se diseñó una estructura que contiene a los sensores de estimación de distancias en base a mediciones exactas del espacio que ocupan dichos sensores y la adaptabilidad en la instalación con la plataforma Dagu DG012-ATV, de este modo se consigue un diseño apto para imprimir en 3D las piezas necesarias para la construcción de los robots móviles autónomos.

El objetivo principal de la estructura es colocar cada sensor en posiciones que formen un semicírculo cada 45° alrededor de la plataforma robótica, de esta manera se logra obtener una estimación de distancias hacia obstáculos o fronteras del entorno en un campo visual de aproximadamente 360 grados; en cuanto a efectividad la estructura lograba su objetivo, sin embargo sus dimensiones eran excesivas para cumplir el propósito final del proyecto, de esta manera se cambió la estructura semicircular por una semiovalada de menor dimensión, tal como se indica en la Figura 25.

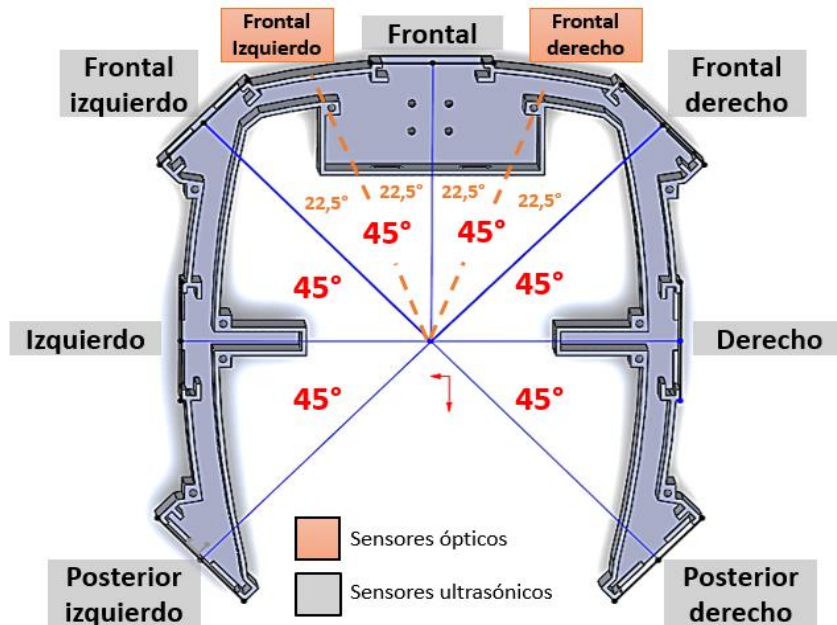


Figura 25. Ubicación de los sensores de estimación de distancias

Una vez desarrollado el diseño final de la estructura, se procedió a realizar las impresiones 3D mediante el software MatterControl, se dividió en 2 partes para alcanzar las dimensiones admitidas por la impresora *SeeMeCNC*, el material de la impresión es de ácido poliláctico (PLA) el cual es lo suficientemente robusto como para permitir el cableado e instalación de los sensores químicos y ópticos.



Figura 26. Estructura semicircular para sensores de distancia

3.5. Sistema encargado del control

El controlador de un robot móvil autónomo es el encargado de captar toda la información recopilada por los sensores, para posteriormente procesarla en tiempo real y de esta manera permitir la toma de decisiones con el fin de alcanzar el objetivo del sistema robótico cooperativo. En cuanto a procesamiento de información, cada robot móvil se construyó con dos controladores identificando a cada uno con una funcionalidad diferente.

El primer controlador es una tarjeta Teensy 3.6 la cual desempeña la función de activación y desactivación de actuadores, además de captar toda la información de los sensores que componen al robot, el segundo controlador es una Raspberry Pi 3 la cual recibe toda la información proveniente de la Teensy 3.6 para ser procesada con cálculos algebraicos y geométricos, traduciendo la información recibida en datos de navegación, posición y orientación del robot; toda la información procesada es enviada a un ordenador central en donde se puede visualizar mediante una interfaz gráfica de usuario los resultados de la navegación tales como: mapa de obstáculos, niveles de temperatura y concentración de gases en tiempo real. A continuación, se detallan las principales características técnicas y funcionales de los controladores utilizados.

3.5.1. Teensy 3.6

La tarjeta Teensy 3.6 (ver Figura 27) es una potente plataforma open-hardware basada en un procesador ARM Cortex M4 de 32 bits, ideal para proyectos embebidos que funcionan óptimamente con un reducido tamaño diseñado en forma de placa para que sea posible el montaje en protoboards, tiene la ventaja de poder ser programada en lenguaje Arduino con el IDE de código abierto de Arduino mediante el add-on Teensyduino. Esta placa electrónica cuenta con 62 pines

I/O, 25 de ellos se pueden usar como entradas analógicas de 13 bits, 22 salidas PWM y 4 puertos I2C. Una de las mayores ventajas que innova la Teensy 3.6 sobre un Arduino Uno o un Arduino Mega comúnmente usados en este tipo de proyectos, es su velocidad de procesamiento; mientras un Arduino Mega corre a una velocidad de 16 MHz que se traduce a 16 millones de operaciones por segundo, la Teensy 3.6 siendo de menor tamaño puede correr aproximadamente 11 veces más que un Arduino, hasta 180 MHz (Electronic Projects Components Available Worldwide, 2018).

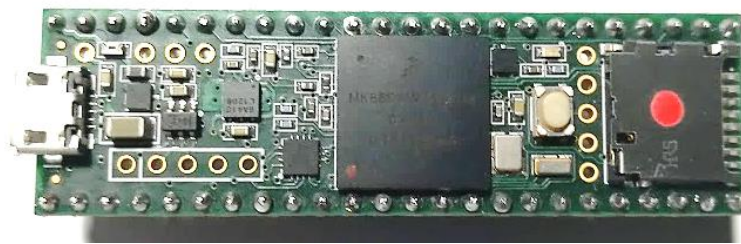


Figura 27. Tarjeta de programación Teensy 3.6.

La tarjeta Teensy 3.6 mediante programación de interrupciones, realiza un conteo incremental de los pulsos adquiridos en los encoders por la lectura de optointerruptores, esto permite estimar desplazamientos fijos de 20 cm en línea recta con intervalos de 7 segundos durante los cuales el robot se detiene, también realiza la activación y desactivación de ventiladores para aspirar aire en el ambiente y sensor la concentración de un gas en específico además del nivel de temperatura en dicho punto; durante la navegación en caso de que un robot detecte frente a él, un obstáculo o frontera mediante los sensores de estimación de distancias, el robot gira un ángulo específico, el cual es recibido desde la Raspberry Pi 3 en donde se realiza el cálculo dicho ángulo basado en giros hacia las zonas no exploradas del entorno, posteriormente el valor de ángulo que gira el agente robótico es captado por la IMU BNO055. Cada robot móvil cuenta con una Teensy 3.6 para operar los siguientes parámetros:

- Pulsos provenientes de los encoders y optointerruptores FC-03 de las ruedas posteriores derecha e izquierda, los pulsos son adquiridos mediante interrupciones en los pines 15 y 16.
- Temperatura ambiente del entorno de navegación mediante un sensor LM35 y obtenida por lectura analógica en el pin A12.
- Concentración de gases en el ambiente mediante sensores de la familia FIGARO TGS y obtenidos por lectura analógica desde el pin A18 hasta el pin A20.
- Ángulo de giro del robot móvil mediante la IMU BNO055 y adquirido mediante comunicación I2C en los pines 18 (SDA0) y 19 (SCL0).
- Distancias del robot hacia las fronteras y obstáculos del entorno de navegación, adquiridas mediante sensores ultrasónicos HC-SR04 y sensores SHARP 2Y0A21.
- Control de 4 motores Dagu DG02S 48:1 correspondientes a las ruedas delanteras y posteriores de la plataforma robótica móvil, el PWM de los motores es controlado por un driver L298N.
- Control de 3 ventiladores correspondientes a la nariz electrónica que permiten aspirar el aire del ambiente en el entorno de trabajo.

Todas las conexiones con la tarjeta programable Teensy 3.6, se implementaron tal como se detalla en la Tabla 4.

Tabla 4

Configuración de pines de la Teensy 3.6

Pin de la Teensy 3.6	Dispositivo conectado	Pin del dispositivo conectado
D0 (salida)	Sensor ultrasónico frontal	Trig
D1 (entrada)	Convertor lógico bidireccional No. 1	LV1
D2 (salida)	Sensor ultrasónico frontal derecho	Trig
D3 (entrada)	Convertor lógico bidireccional No. 1	LV2

CONTINÚA 

D4 (salida)	Sensor ultrasónico frontal izquierdo	Trig
D5 (entrada)	Convertor lógico bidireccional No. 1	LV3
D6 (salida)	Sensor ultrasónico derecho	Trig
D7 (entrada)	Convertor lógico bidireccional No. 1	LV4
D8 (salida)	Sensor ultrasónico izquierdo	Trig
D9 (entrada)	Convertor lógico bidireccional No. 2	LV1
D10 (salida)	Sensor ultrasónico posterior derecho	Trig
D11 (entrada)	Convertor lógico bidireccional No. 2	LV2
D10 (salida)	Sensor ultrasónico posterior izquierdo	Trig
D24 (entrada)	Convertor lógico bidireccional No. 2	LV3
D16 (entrada)	FC-03 de rueda posterior derecha	D0
D17 (entrada)	FC-03 de rueda posterior izquierda	D0
D20 (entrada)	Driver L298N	IN1
D21 (entrada)	Driver L298N	IN2
D22 (entrada)	Driver L298N	IN3
D23 (entrada)	Driver L298N	IN4
D14 (entrada)	Sharp 2Y0A21 frontal derecho	V _o
D15 (entrada)	Sharp 2Y0A21 frontal izquierdo	V _o
A18 (entrada)	Sensor químico TGS822	Vout
A19 (entrada)	Sensor químico TGS826	Vout
A20 (entrada)	Sensor químico TGS2600	Vout
A21 (entrada)	Sensor químico TGS2602	Vout
A22 (entrada)	Sensor químico TGS2610	Vout
SDA0	Resistencia 4,7K Ω - IMU BNO055	SDA
SCLA0	Resistencia 4,7K Ω - IMU BNO055	SCL
A12	Sensor de temperatura: LM35	Out
3,3V	Módulos FC-03	V _{cc}
3,3V	Resistencia 4,7K Ω - DS18B20	S
GND	Módulos FC-03	Gnd
GND	Sensor de temperatura: DS18B20	I
GND	Batería Lipo 7.4V	Gnd

Además, se implementó una placa de circuito impreso de dimensiones 7.5×9.25 cm (ver Figura 28) que se ajusta a las dimensiones de la tarjeta programable teensy 3.6 y convertidores lógicos bidireccionales, la placa sirve como tercer nivel del robot móvil autónomo en donde se realiza las conexiones de todos los sensores y actuadores del sistema robótico

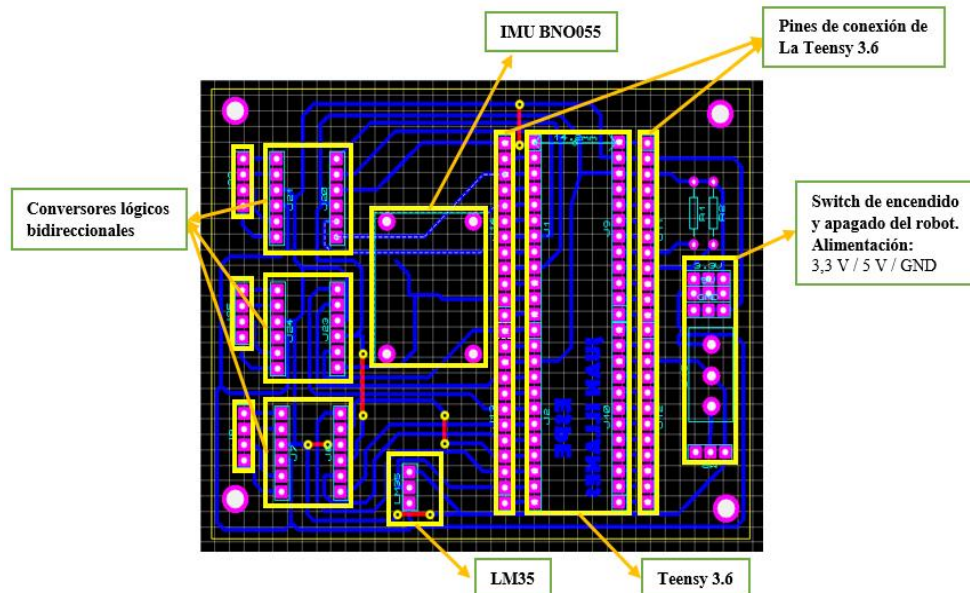


Figura 28. Diseño PCB para Teensy 3.6.

3.5.2. Raspberry Pi 3 modelo B

Se ha escogido la Raspberry Pi 3 modelo B para realizar la etapa de procesamiento de datos, ya que la misma ofrece las ventajas de una pequeña y potente computadora de bajo costo, su software se basa en código abierto con sistema operativo Raspbian basado en Linux, ofrece la facilidad de programación en lenguaje Python, considerado como uno de los mejores para procesamiento de cálculos algebraicos y matemáticos, de esta manera fue posible realizar toda la cinemática directa del robot mediante cálculos matriciales y la cinemática inversa del robot mediante cálculos trigonométricos.

La raspberry Pi 3 modelo B consta de 40 pines de entrada y salida, sin embargo, la mayor ventaja que se aprovecha en el actual proyecto de investigación es su conexión Wifi la cual es utilizada para establecer una conexión inalámbrica mediante protocolo UDP entre cada robot móvil autónomo y el ordenador central donde se encuentra el sistema de control centralizado y la interfaz

gráfica de usuario GUI. Cada robot móvil cuenta con una Raspberry Pi 3 para procesar los datos recibidos desde la teensy 3.6 y obtener la siguiente información:

- Posición en (X, Y) del robot móvil durante cada intervalo en el que se realiza el escaneo del entorno de navegación.
- Posiciones en (X, Y) de los obstáculos y fronteras detectados por los sensores ultrasónicos durante cada intervalo en el que se realiza el escaneo del entorno de navegación.
- Posición en (X, Y) de la nariz electrónica durante cada intervalo en el que se realiza el escaneo del entorno de navegación.

3.6. Sistema de sensado

Para la acción de adquisición de datos mediante sensores integrados en el sistema robótico autónomo cooperativo, se establece la siguiente clasificación: sensado para odometría, sensado de obstáculos o fronteras, sensado químico y sensado de temperatura.

3.6.1. Odometría

Para obtener odometría aplicada al sistema robótico, es necesario conocer cómo se comportan los movimientos realizados por los robots tanto en desplazamientos como en giros, este objetivo se resolvió mediante el uso de encoders, optointerruptores y sensores de medición inercial.

3.6.1.1. Módulo FC-03

El FC-03 (ver Figura 29) es un módulo encargado de determinar la velocidad o posición de un motor, su funcionamiento se basa en el uso de un sensor optointerruptor infrarrojo MOCH22A

conformado por un emisor IR y un receptor IR encargados de captar pulsos provenientes de un disco encoder, los pulsos son digitalizados en pulsos TTL e interpretados por un microcontrolador en donde por lo general se utiliza interrupciones por flancos de subida o bajada (Naylamp Mechatronics, 2018).

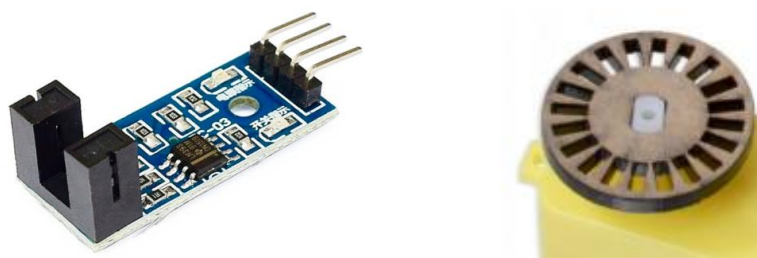


Figura 29. Optointerruptor FC-03 (izq.), encoder de 20 agujeros (der.)

Cada rueda posterior de todos los robots móviles está adaptada con un módulo FC-03 y un encoder con una resolución de 20 ppr., estos cumplen la función de establecer desplazamientos de distancias fijas y calcular cuánto se ha desplazado un robot en caso de que se haya detectado un obstáculo o frontera y se detenga antes de cumplir la distancia preestablecida.

3.6.1.2. IMU BNO055

El IMU (en inglés *inertial measurement unit*) BNO055 (ver Figura 30) es un sensor de medición inercial basado en un potente microcontrolador ARM Cortex-M0 encargado de abstraer la fusión de todos los sensores que lo integran y procesar los cálculos aritméticos necesarios para obtener los datos de orientación absoluta. El IMU BNO055 está constituido por: un acelerómetro, un magnetómetro, un giroscopio y un sensor de temperatura.

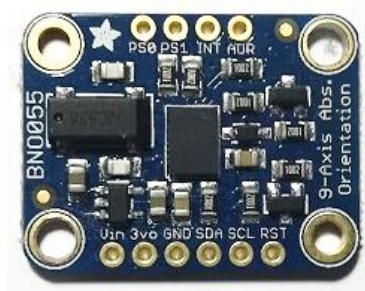


Figura 30. IMU BNO05.

Cada robot móvil autónomo está construido con un sensor IMU BNO055 el cual está encargado de calcular la orientación de la plataforma en cada instante de la navegación, el sensor está ubicado en el nivel más alto del robot con el objetivo de evitar interferencias con la inducción de los motores y ventiladores.

El BNO055 está formado por un giroscopio que corre a una velocidad de 100 Hz y cuyo funcionamiento está basado en vectores y ángulos de Euler (ver Figura 31), esto permite obtener los datos de orientación de tres ejes basados en una esfera de 360 grados en donde constan los ángulos de navegación: roll (eje X), pitch (eje Y), yaw (eje Z).

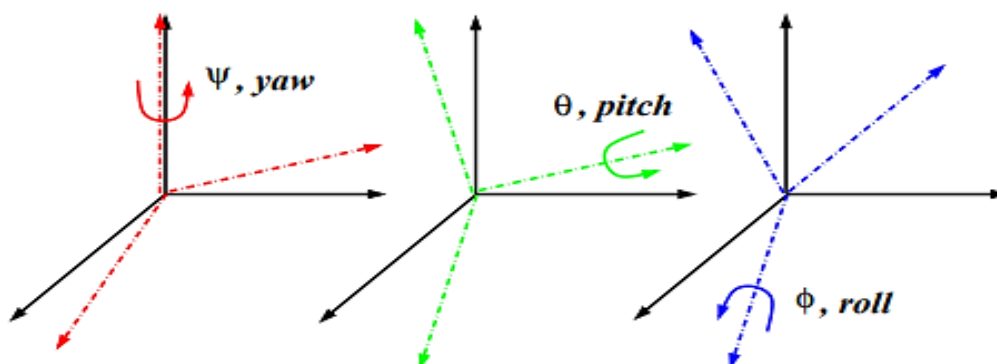


Figura 31. Esquemático de ángulos de navegación: yaw (ψ), pitch (θ), roll (ϕ).

Fuente: (Alemi & Bridges, 2010)

3.6.2. Obstáculos y fronteras

Para obtener un mapa en el cual se pueda visualizar la ubicación de obstáculos y fronteras, es necesario conocer en cada escaneo de los robots móviles, a qué distancia se encuentran dichos obstáculos, de esta manera se ha empleado sensores ultrasónicos e infrarrojos encargados de estimar las distancias para que posteriormente sean procesadas mediante cinemática directa y de esta manera obtener la ubicación de los obstáculos y fronteras.

3.6.2.1. Sensor ultrasónico HC-SR04

El HC-SR04 (ver Figura 32) es un sensor cuya función es detectar objetos a distancias en un rango de 2 cm a 450 cm mediante ultrasonido, su funcionamiento se basa en el envío de un pulso de arranque hasta que choque con el cuerpo sólido más cercano y mediante la anchura del pulso de retorno o tiempo de retorno se estima la distancia.

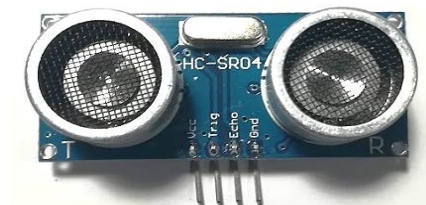


Figura 32. Ultrasónico HC-SR04.

Cada robot móvil está construido con 7 sensores ultrasónicos HC-SR04 ubicados estratégicamente cada 45° formando una semicircunferencia, de esta manera se logra tener una estimación del entorno por todos los frentes del robot a excepción del posterior ya que allí se encuentra instalada la nariz electrónica; a pesar de que el rango de medición del sensor permite captar distancias hasta 450 cm, solamente se considera la lectura de distancias menores a 40 cm, de esta manera se evitan errores de estimación de distancias con objetos o fronteras lejanas al robot.

Para realizar el acondicionamiento de voltajes entre sensores ultrasónicos y Teensy 3.6, se utilizaron módulos de conversión bidireccional (ver Figura 33), los cuales se encargan de transformar una señal lógica de 5V a 3.3V y viceversa, su función es adaptar las señales de entrada para la tarjeta programable Teensy 3.6 provenientes del sensor ultrasónico HC-SR04 (echo) cuyo nivel lógico TTL es de 5V, cada módulo consta de 4 entradas y 4 salidas, razón por la cual es necesario la implementación de 2 módulos conversores para los 7 sensores ultrasónicos, en la se detallan las conexiones entre conversores lógicos y sensores ultrasónicos.



Figura 33. Conversor lógico.

Tabla 5

Configuración de pines de los conversores lógicos bidireccionales

Dispositivo	Pin del dispositivo	Dispositivo conectado	Pin del dispositivo conectado
Conversor lógico bidireccional No. 1	HV1 (entrada)	Sensor ultrasónico frontal	echo
	HV2 (entrada)	Sensor ultrasónico frontal derecho	echo
	HV3 (entrada)	Sensor ultrasónico frontal izquierdo	echo
	HV4 (entrada)	Sensor ultrasónico derecho	echo
	LV1 (salida)	Teensy 3.6	D1 (entrada)
	LV2 (salida)	Teensy 3.6	D3 (entrada)
	LV3 (salida)	Teensy 3.6	D5 (entrada)
	LV4 (salida)	Teensy 3.6	D7 (entrada)
Conversor lógico bidireccional No. 2	HV1 (entrada)	Sensor ultrasónico frontal izquierdo	echo
	HV2 (entrada)	Sensor ultrasónico posterior derecho	echo
	HV3 (entrada)	Sensor ultrasónico posterior izquierdo	echo
	LV1 (salida)	Teensy 3.6	D9 (entrada)
	LV2 (salida)	Teensy 3.6	D11 (entrada)
	LV3 (salida)	Teensy 3.6	D24 (entrada)

3.6.2.2. Sensor óptico SHARP 2Y0A21

El SHARP 2Y0A21 (ver Figura 34) es un sensor que utiliza un emisor infrarrojo y receptor, cuya función es detectar objetos a distancias en un rango de 10 a 80 cm, su funcionamiento se basa en el método de triangulación y consiste medir el ángulo que forma el triángulo emisor–objeto–receptor, es decir a mayor distancia, menor es el ángulo (World, 2018).



Figura 34. Sensor óptico de distancia SHARP 2Y0A21.

Cada robot móvil está construido con 2 sensores ópticos SHARP 2Y0A21 ubicados en el frente principal del robot cuya función es únicamente captar obstáculos o fronteras en los puntos ciegos que los sensores ultrasónicos no logran reconocer, de esta forma los sensores ópticos se complementan a los sensores ultrasónicos frontales para lograr una correcta navegación autónoma.

3.6.3. Sensado químico

Los sensores de la familia FIGARO TGS (*Taguchi Gas Sensor*) están compuestos por óxido metálico de película gruesa, ideal para generar una gran sensibilidad ante la presencia de gases, ya sean tóxicos, explosivos o compuestos orgánicos volátiles para medir la calidad del aire (FIGARO, 2018). Entre los principales gases a los cuales son sensibles los sensores químicos Figaro TGS se encuentran: gases naturales (isopreno, pineno, limoneno) y gases artificiales (benceno, tolueno,

nitrobenceno). En el presente proyecto, cada robot móvil se construyó con una nariz electrónica la cual está conformada por una matriz de 3 sensores de gas detallados a continuación.

- TGS2610: Es un sensor de tamaño reducido y respuesta rápida, se caracteriza por tener alta sensibilidad al gas licuado de petróleo (GLP) y butano lo cual lo hace ideal para verificar fugas de gas, requiere una corriente de calentamiento de 56 mA para obtener lecturas confiables durante su uso.
- TGS2600: Es un sensor de tamaño reducido, respuesta rápida y bajo consumo de potencia, se caracteriza por tener alta sensibilidad a contaminantes del aire.
- TGS2602: Es un sensor de tamaño reducido, respuesta rápida y bajo consumo de potencia, se caracteriza por tener alta sensibilidad a contaminantes del aire, gases odoros, compuestos orgánicos volátiles liberados por quema de combustibles como gasolina, madera, carbón o gas natural, también liberados por disolventes o pinturas.

Para obtener el nivel de concentración de un gas representado mediante el valor del voltaje de salida (V_{OUT}), se realiza un divisor de tensión dado por la ecuación 1 (FIGARO, 2018).

$$V_{OUT} = \frac{V_C \times R_L}{R_S + R_L} \quad (1)$$

V_{OUT} : Tensión de alimentación [$5V \pm 0.2V(DC)$]

R_S : Resistencia interna al sensor variable ante la presencia de un gas

R_L : Resistencia de carga

Se diseñó 3 narices electrónicas, cuya arquitectura se basa en una caja cerrada de aluminio esterilizable con dimensiones de $10,5 \times 5 \times 6 \text{ cm}$. La nariz electrónica está conformada por 3

ventiladores ubicados de tal forma que permitan el ingreso de aire a la caja, los ventiladores funcionan bajo un control On-Off comandado por un módulo relé que se activa cada vez que el robot se detiene para realizar el sensado de gases en el entorno y se desactiva mientras el robot se encuentre en movimiento.

Además, se desarrolló una placa de circuito impreso (ver Figura 35) que integra una matriz de 5 sensores químicos de la familia Figaro TGS, sin embargo, mediante un análisis de los sensores que tienen mejor sensibilidad ante sustancias químicas, se optó por utilizar los 3 sensores mencionados anteriormente, la placa de dimensiones $10,4 \times 4,9 \text{ cm}$ se ubica dentro de la nariz electrónica ubicada en la parte posterior de la plataforma.

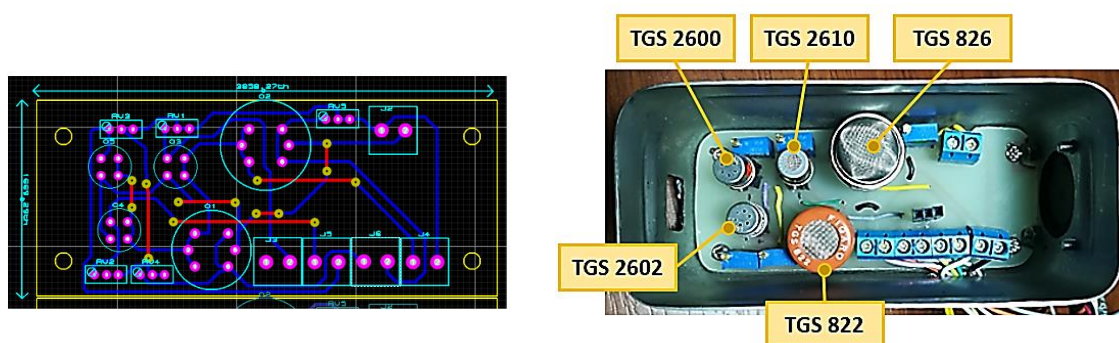


Figura 35. Diseño PCB para matriz de sensores químicos (izq.), Vista interna de la nariz electrónica (der.).

Una vez instalada la nariz electrónica, es importante recalcar la ubicación de la misma, la caja posee un orificio en la parte inferior, siendo el lugar por el cual entra el aire que es aspirado desde el entorno hacia el interior de la caja, dicho orificio se ubica a 1 cm del piso con el objetivo de que las muestras dotadas de sustancias químicas estén lo más cercanas posibles a la matriz de sensado, en la Figura 36 se muestra la ubicación de la nariz electrónica en la plataforma robótica móvil.

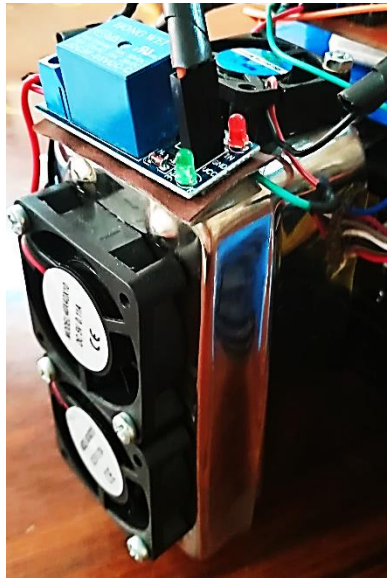


Figura 36. Vista exterior de la nariz electrónica.

3.6.4. Sensado de temperatura

Se ha instalado en cada robot móvil autónomo un sensor LM35 cuya función es medir la temperatura en un rango de -55°C a 150°C , su funcionamiento se basa en la interpretación de una salida lineal de tensión que es proporcional a la temperatura.

3.7. Sistema de potencia

El sistema de potencia engloba la integración de baterías y módulos necesarios para la alimentación de sensores, actuadores y controladores que conforman al robot móvil autónomo.

Cada robot móvil está integrado por una batería Lipo de 7.4 V a 1.200 mAh encargada de alimentar a varios componentes que conforman a los agentes robóticos, el módulo driver L298N se encarga de proveer una tensión menor o igual a 5 V , misma que es acondicionada para la activación de los motores DC Dagu DG02S, mediante el regulador de voltaje integrado al driver

se provee energía a los sensores cuya tensión de alimentación sea de 5 Vcc. En cuanto a la alimentación autónoma de la Raspberry Pi 3 y la Teensy 3.6, se utiliza un power bank de 20.000 mAh; la raspberry Pi 3 se alimenta mediante el power bank vía USB, de igual forma la Raspberry Pi 3 provee la tensión necesaria a la Teensy 3.6 mediante conexión USB, misma que se utiliza para transmisión de datos. En la Figura 37 se aprecia las baterías utilizadas para la implementación de las plataformas robóticas.



Figura 37. Batería Lipo de 7.4 V (izq.), power bank 20.000 mAh (der.)

3.7.1. Motores Dagu DG02S

Cada robot móvil está constituido por 4 motores DC Dagu DG02S 48:1 (Figura 38); los motores son controlados para generar la tracción de 4 ruedas ATV y realizar los respectivos desplazamientos de la plataforma robótica móvil. En la Tabla 6 se indican las principales características técnicas de los actuadores que se encuentran en cada robot móvil.

Tabla 6

Especificaciones técnicas de los actuadores que integran al robot móvil

Dagu DG02S	Dimensiones	43 x 45 x 15 mm
	Tensión de alimentación	3 - 6 Vcc
	Corriente sin carga	200 mA
	Velocidad sin carga	65 rpm

Fuente: (Robots, 2018)



Figura 38. Motores DG02S 48:1.

3.7.2. Driver L298N

El driver L298N (ver Figura 39) está encargado del control para activación y desactivación de los motores Dagu DG02S; a cada motor DG02S se le suministra un valor PWM el cual permite el control de la velocidad y sentido de giro en las ruedas de la derecha e izquierda de la plataforma. En la Tabla 7 se detallan las principales características técnicas del driver L298N.

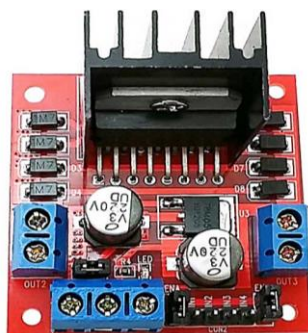


Figura 39. Driver L298N.

Tabla 7

Especificaciones técnicas del driver L298N

Driver L298N	Dimensiones	55 x 60 x 43 mm
	Tensión de alimentación	5 Vcc - 35 Vcc
	Tensión de alimentación para motores (mín.)	5 Vcc
	Tensión de alimentación para motores (máx.)	46 Vcc

CONTINÚA 

Corriente pico de operación	4 A
Corriente constante de operación	2 A
Circuito integrado	L298N
Consumo	20 W
Regulador de voltaje	Salida de 4.5 Vcc – 5 Vcc
Terminales de control	IN1, IN2, IN3, IN4, ENA, ENB
Salidas	OUT1, OUT2, OUT3, OUT4

Fuente: (Robots, 2018)

El driver L298N, al contar con un regulador de voltaje permite suministrar al sistema una tensión continua de 5 V, necesaria para alimentar componentes tales como: sensores químicos, sensores ultrasónicos, sensores ópticos y sensor de temperatura. A continuación, en la Tabla 8 se detalla la disposición de conexiones del driver L298N.

Tabla 8
Configuración de pines del driver L298N

Pin del driver L298N	Dispositivo conectado	Pin del dispositivo conectado
IN1	Teensy 3.6	D20 (entrada)
IN2	Teensy 3.6	D21 (entrada)
IN3	Teensy 3.6	D22 (entrada)
IN4	Teensy 3.6	D23 (entrada)
OUT A	Motores Dagu DG02S (parte derecha)	Pines de polarización
OUT B	Motores Dagu DG02S (parte izquierda)	Pines de polarización
12+ (entrada)	Batería Lipo 7.4V	Vcc
GND	Teensy 3.6	GND
GND	Batería Lipo 7.4V	Gnd
5V	Sensores ultrasónicos	Vcc
5V	Sensores ópticos	Vcc

3.8. Conexión y montaje de la plataforma robótica móvil

A continuación, se detallan los pasos efectuados para construir un robot móvil autónomo.

- Instalación de los motores DAGU DG02S y ruedas en el primer nivel (ver Figura 40) correspondiente a la plataforma robótica móvil, las uniones de cada rueda con los ejes de los motores fueron engrasadas para evitar posibles daños.
- Adaptación de discos encoder de 20 mm de diámetro con 20 agujeros en los ejes de los motores correspondientes a las ruedas posteriores de la plataforma robótica.
- Instalación de la estructura que contiene sensores ópticos y ultrasónicos en el primer nivel correspondiente la plataforma robótica móvil.
- Instalación de una línea en común para Vcc y GND de los 2 sensores ópticos y 7 sensores ultrasónicos con el objetivo de implementar el menor cableado posible.
- Cableado de los 7 sensores ultrasónicos y 2 sensores ópticos: conexiones de las líneas trigger y echo de los sensores ultrasónicos y V0 de los sensores ópticos.

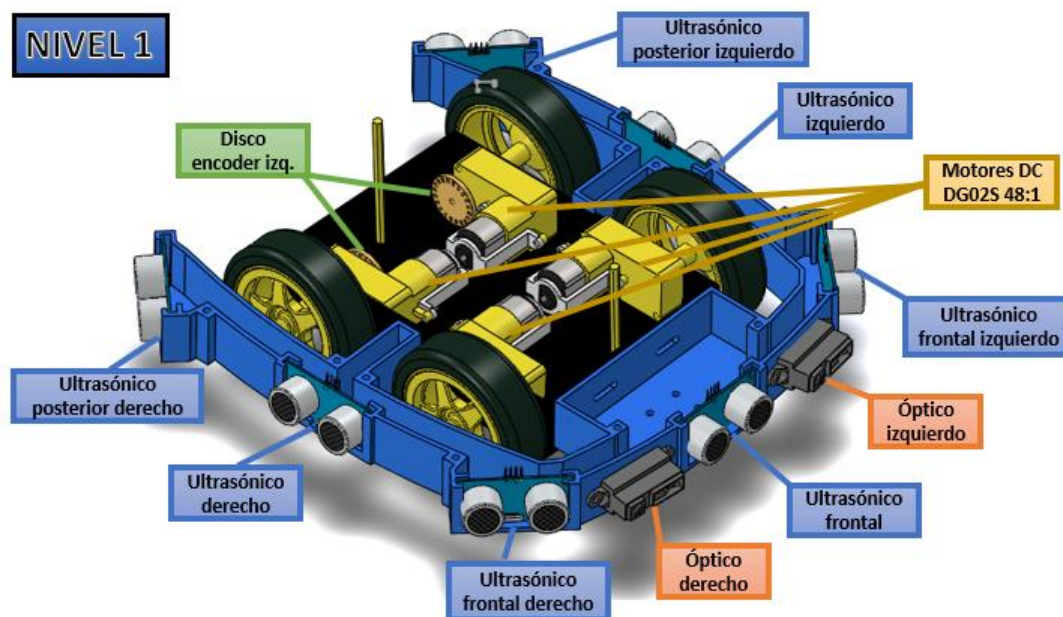


Figura 40. Primer nivel correspondiente al robot móvil autónomo.

- Instalación del driver L298N en la parte inferior de la tapa de la plataforma robótica DG012-ATV correspondiente al segundo nivel del robot móvil (ver Figura 41), polarización de los motores DAGU DG02S con las salidas del driver (OUT A, OUT B).
- Instalación de dos módulos optointerruptores FC-03 mediante 2 torres de 30 mm adaptadas desde la parte inferior de la tapa de la plataforma robótica DG012-ATV hasta los discos encoder de 20 agujeros instalados en los ejes de los motores posteriores.
- Instalación de una línea en común para Vcc y GND de los módulos optointerruptores FC-03 con el objetivo de implementar el menor cableado posible.
- Cableado de los módulos optointerruptores FC-03: conexión de las líneas D0.
- Instalación del segundo nivel del robot móvil correspondiente a la tapa de la plataforma robótica móvil, se utilizaron dos torres metálicas de 60 mm de distancia desde el primer nivel hacia la tapa.

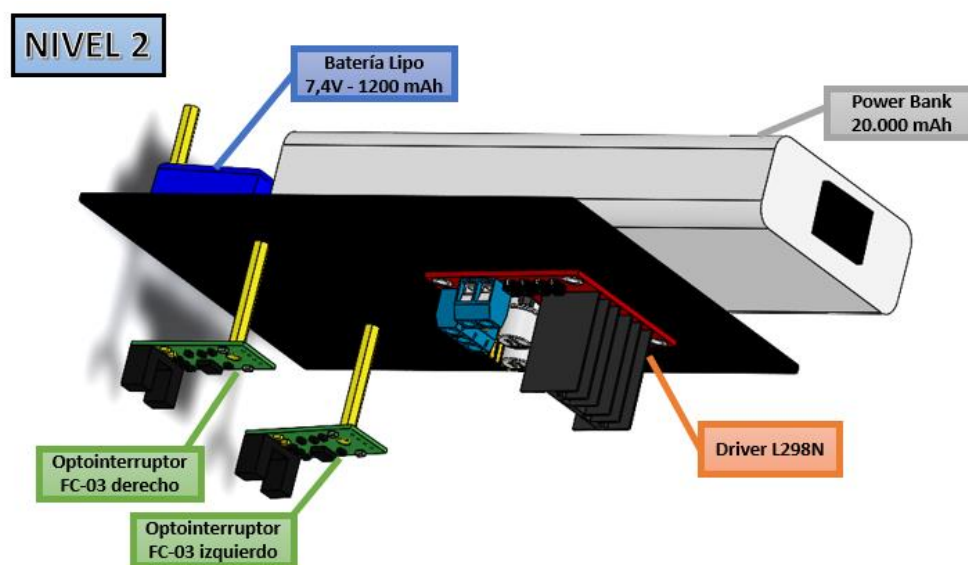


Figura 41. Segundo nivel correspondiente al robot móvil autónomo.

- Instalación del tercer nivel del robot móvil que corresponde a la placa electrónica que contiene a la Teensy 3.6, se utilizaron dos torres metálicas de 30 mm de distancia desde el segundo nivel hacia la placa electrónica.
- Instalación del cuarto nivel del robot móvil correspondiente al sensor IMU BNO055, se utilizaron dos torres metálicas de 70 mm de distancia desde el segundo nivel hacia la IMU.
- Instalación de las fuentes de alimentación de la Teensy 3.6 y la Raspberry Pi 3: el power bank de 20.000 mAh y la batería Lipo de 7.4V son ubicados en el segundo nivel del robot móvil por debajo de la placa electrónica.
- Instalación de la Raspberry Pi 3 en la parte frontal del tercer nivel (ver Figura 42) del robot móvil por encima del power bank.
- Instalación de la nariz electrónica en la parte posterior del robot móvil mediante una estructura metálica en forma de L adaptada al segundo nivel.
- Conexión de ventiladores correspondientes a la nariz electrónica con el módulo relé.
- Conexión de todo el cableado realizado con la Teensy 3.6: sensores ópticos, sensores ultrasónicos, módulos FC-03, sensor de temperatura, sensores químicos, módulo relé, Vcc, GND, alimentación autónoma.

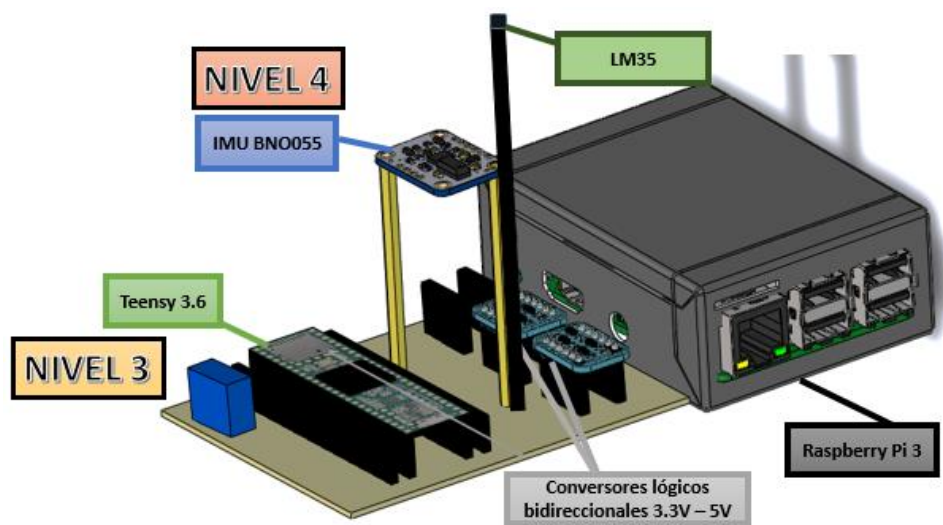


Figura 42. Tercer y cuarto nivel correspondientes al robot móvil autónomo.

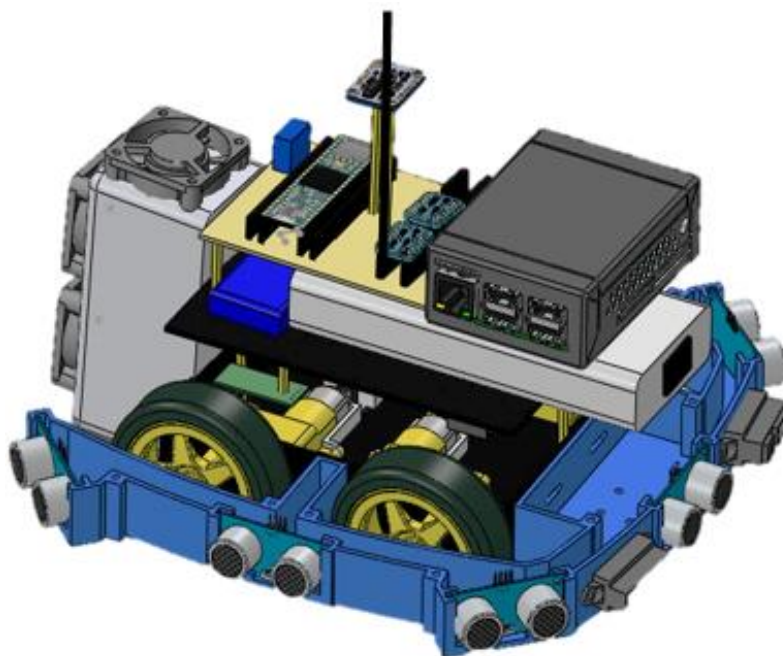


Figura 43. Diseño CAD del robot móvil autónomo.

3.9. Integración de hardware

Una vez especificados de manera independiente los subsistemas que conforman a un robot móvil autónomo, se procedió a la integración e instalación de cada uno de ellos; en la Figura 44, Figura 45, Figura 46 y Figura 47 se muestra la disposición de componentes mediante la vista superior, frontal, posterior y lateral respectivamente de un agente robótico implementado.

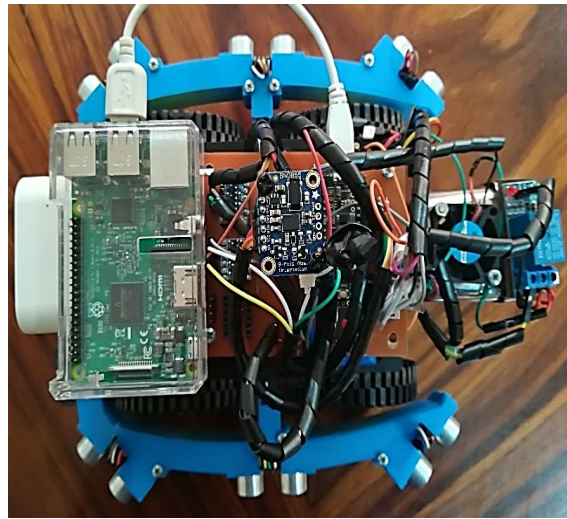


Figura 44. Vista superior del robot móvil No. 1.

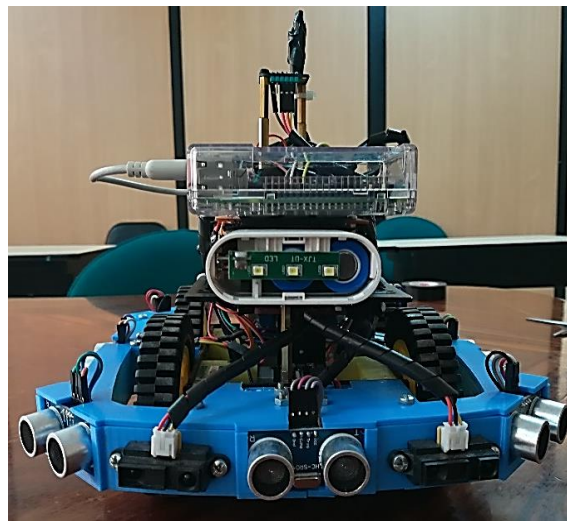


Figura 45. Vista frontal del robot móvil No. 1.

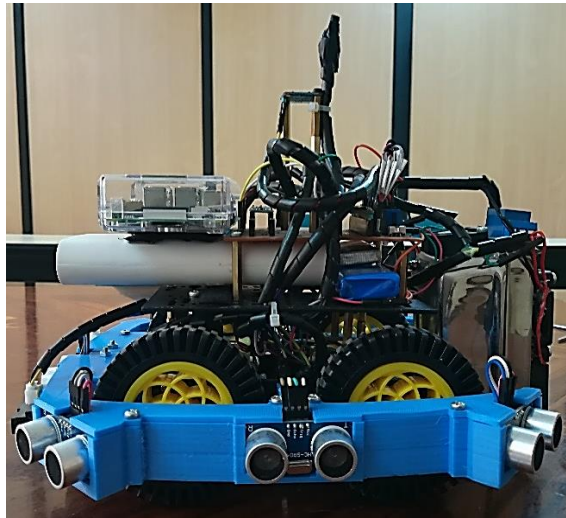


Figura 46. Vista lateral del robot móvil No. 1.

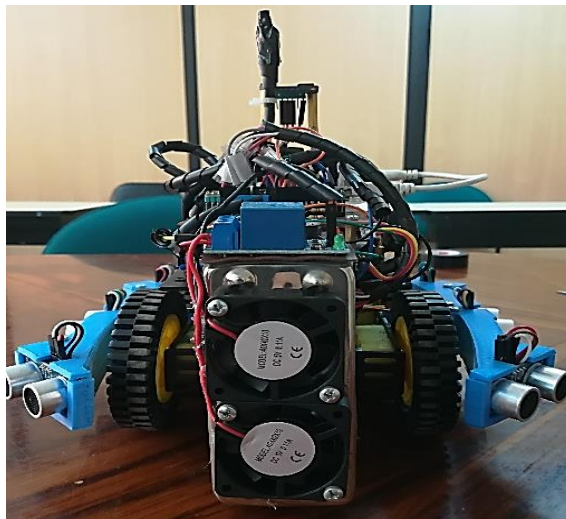


Figura 47. Vista posterior del robot móvil No. 1.

CAPÍTULO IV

SISTEMA DE NAVEGACIÓN Y COMUNICACIÓN

4.1. Introducción

En este capítulo se explica el proceso de diseño y programación del software de control centralizado, navegación y sensado. Se detalla fórmulas y cálculos necesarios para la obtención de la cinemática directa e inversa de un robot móvil autónomo, lo cual tiene como objetivo principal determinar su posición durante el proceso de navegación y los movimientos necesarios que debe efectuar cada agente inteligente para trasladarse hacia zonas no exploradas.

4.2. Descripción del sistema de comunicación y control

Se ha establecido requerimientos de funcionamiento para determinar la forma en la cual se llevarían a cabo las funciones de navegación, sensado y cooperación robótica entre tres agentes robóticos móviles, de esta manera se establece qué tipos de lenguajes de programación son necesarios, el tipo de comunicación para la transferencia de datos y qué tipo de algoritmo es el más óptimo para cumplir con los objetivos propuestos. De forma general, se detalla a continuación cuales son las necesidades del software de control, navegación y sensado:

- Es necesario disponer de un ordenador central en el cual se recepta toda la información proveniente de los 3 agentes robóticos, y de esta manera sea posible acoplar un controlador centralizado encargado de analizar toda la información recolectada por dichos agentes que cumplen funciones de robótica móvil cooperativa.

- El controlador centralizado es el encargado de analizar zonas exploradas y no exploradas en el entorno de navegación con el objetivo de retornar dicha información a cada agente para determinar que giros se deben realizar y de esta manera navegar hacia las zonas no exploradas.
- Es necesario un controlador encargado de la adquisición de datos correspondientes a la temperatura, concentración de gases y distancias hacia obstáculos o fronteras, para ello se debe tomar una determinada cantidad de muestras por cada escaneo que realiza el robot, garantizando de esta manera la confiabilidad, precisión y exactitud de cada valor sensado.
- Se debe implementar funciones de programación para determinar los movimientos de un robot móvil tales como: avanzar, detenerse, girar, sensar y enviar información.
- Es indispensable conocer las posiciones de cada agente, obstáculos y narices electrónicas, para ello es necesario programar algoritmos correspondientes a cálculos cinemáticos de robótica convencional (cinemática directa e inversa).
- La comunicación entre robots móviles y el ordenador central debe ser inalámbrica y la información recibida por el ordenador debe sincronizarse con cada agente en tiempo real, para ello se debe implementar el protocolo, topología y tipología de comunicación que sea más óptimo para este caso.
- El ordenador central debe integrar una interfaz gráfica de usuario GUI en conjunto con el controlador central, de esta manera se permite al usuario visualizar toda la información de la navegación cooperativa en tiempo real.

Para cumplir correctamente con todos los requerimientos de software, la programación se ha dividido en tres procesadores. En el capítulo anterior se han detallado los controladores escogidos

para el procesamiento de datos del sistema los cuales son: la tarjeta programable Teensy 3.6 y la Raspberry Pi 3, adicionalmente el software escogido para la programación del controlador centralizado es Matlab R2017b, por ser considerado una potente herramienta informática capaz de acoplarse al sistema robótico correspondiente al proyecto de investigación. En la Figura 48, se muestra cómo se integran los procesadores de datos del sistema.

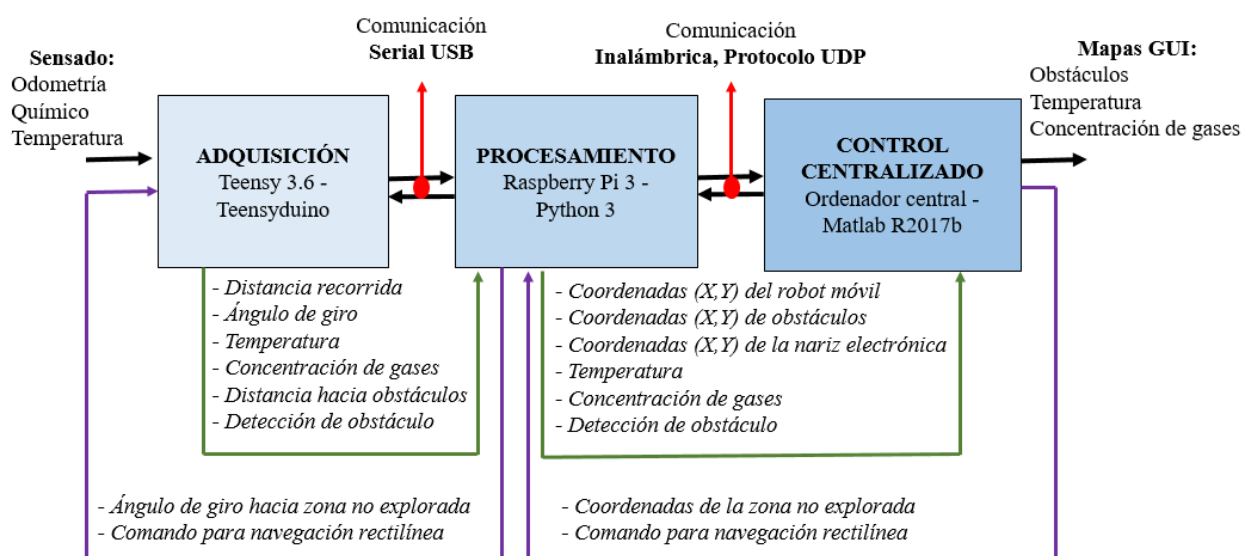


Figura 48. Diagrama de bloques correspondiente al software del sistema.

4.3. Topología de red

El presente proyecto de investigación cuenta con un total de 5 equipos: un router como concentrador de datos, una computadora portátil como sistema de control e interfaz centralizada y 3 agentes robóticos móviles autónomos, cada uno una Raspberry Pi 3 y una Teensy 3.6; el sistema robótico utiliza una topología tipo estrella en donde cada nodo (robots móviles y computadora portátil) se conectan a un nodo central (router) encargado de actuar como un concentrador capaz

de captar y reenviar las transmisiones de todos los nodos periféricos correspondientes a la red. En la Figura 49 se detalla la arquitectura o topología de red del sistema robótico cooperativo.

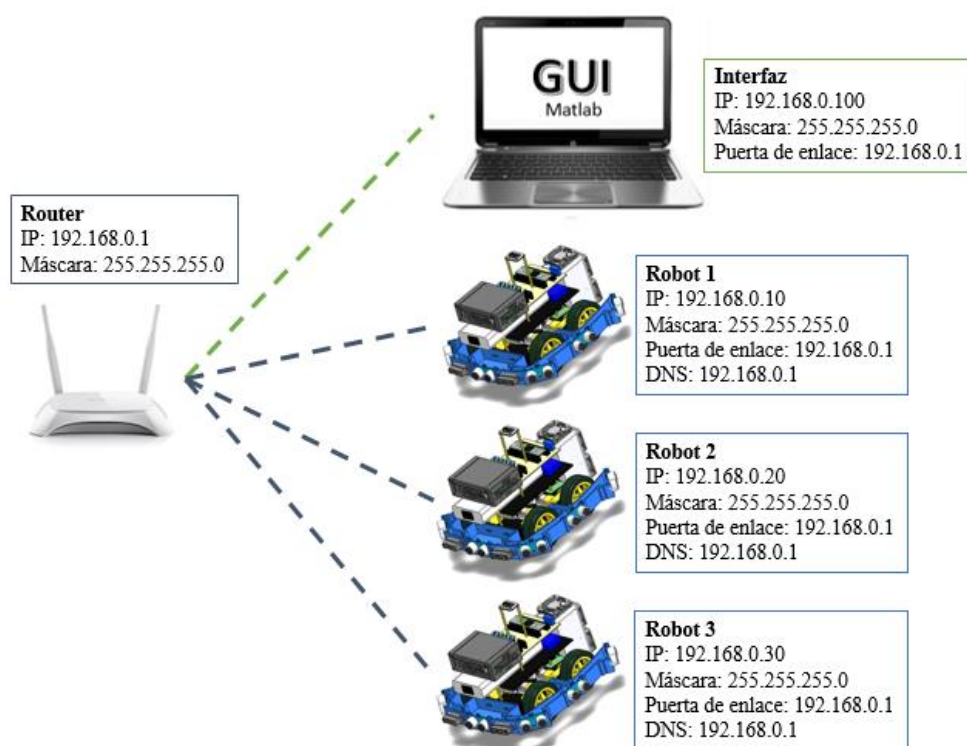


Figura 49. Topología de red del sistema robótico cooperativo.

En la capa de red del modelo OSI correspondiente al sistema robótico cooperativo, se maneja el protocolo IPv4 para transmisión de datos en medios no orientados a la conexión según el estándar RFC 791; en la capa de transporte para transmisión bidireccional de datos según el estándar RFC 768, se utilizaron sockets definidos como métodos de establecimiento de un enlace de conexión entre dos procesos en TCP/IP, de esta manera es posible manejar dos protocolos de comunicación: TCP y UDP, sin embargo mediante pruebas y un análisis de eficiencia se ha considerado que el protocolo más viable a utilizarse es mediante comunicación UDP.

4.4. Entorno de navegación y pruebas

Para establecer algoritmos de navegación, posición y orientación es necesario comprender el entorno de navegación controlado, en este caso, cuáles son sus dimensiones y como se divide.

En el laboratorio de pruebas químicas con robots en la Universidad de las Fuerzas Armadas ESPE, se ha preparado un área de navegación de 240 cm de largo correspondientes al eje de las ordenadas (Y) y 170 cm de ancho correspondiente al eje de las abscisas (X), además se ha dividido en cuadrículas de $20 \times 10\text{ cm}$ de color rojo y cuadrículas de $20 \times 20\text{ cm}$ de color negro (ver Figura 50), de esta manera, la misma cuadrícula es programada para visualizarse en la interfaz gráfica de control centralizado en el software Matlab. Además, cabe resaltar que el laboratorio esta sellado para evitar corrientes de aire que afecten el sensado de gases y niveles de temperatura.



Figura 50. Entorno de navegación y pruebas.

4.5. Algoritmo para solución de la cinemática directa

Al tratarse de un sistema en 2D, cada robot consta de 2 grados de libertad en el cual se tiene 2 tipos de movimientos: traslaciones a lo largo del eje de las ordenadas (eje Y), y rotaciones alrededor del eje Z (ver Figura 51). Una vez definidos los parámetros de entrada del sistema; mediante cinemática directa es posible conocer la posición actual (coordenadas en X y Y) de los robots móviles autónomos, para ello se utiliza cálculos matriciales definidos de la siguiente manera:

- **Desplazamientos a lo largo de eje Y (Matriz de traslación):**

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\begin{aligned} d_x &= 0 \\ d_z &= 0 \end{aligned}$$

- **Giros alrededor del eje Z (Matriz de rotación):**

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

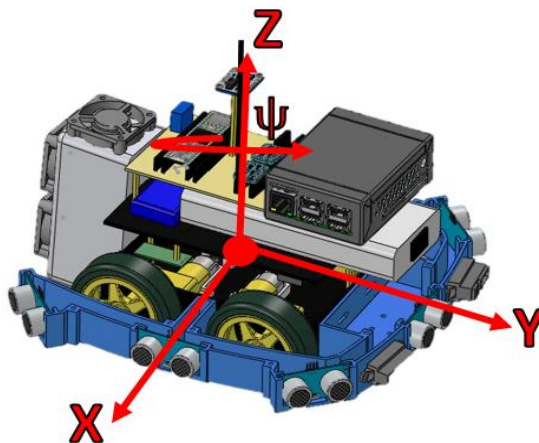


Figura 51. Referencia cartesiana del robot móvil.

Donde (dx) , (dy) y (dz) corresponden a desplazamientos a lo largo de los ejes X , Y y Z respectivamente y (ψ) corresponde al ángulo de giro alrededor del eje Z . Mediante multiplicaciones matriciales definidas por la ecuación 2 y ecuación 3 (Sánchez J. H., 2013) se obtiene las coordenadas en (X,Y) respecto al origen del sistema, tal como se muestra a continuación:

$$I \cdot T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$I \cdot R(\psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$I \cdot R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & P_x \\ \sin(\psi) & \cos(\psi) & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El algoritmo de solución para la cinemática directa de un robot móvil se basa en un sistema que consta de dos entradas (distancia recorrida y ángulo de giro) y dos salidas correspondientes a la posición en (X,Y) del robot respecto a la matriz implementada en el entorno de navegación. La tarjeta programable Teensy 3.6 se encarga de mantener recorridos lineales constantes de 20 *cm* entre cada escaneo o hasta encontrar obstáculos a una distancia menor a 15 *cm*. Para lograr el objetivo mencionado, se ha dotado de odometría a los robots móviles cuyos movimientos se detectan mediante programación de encoders colocados en las ruedas posteriores de cada plataforma.

La programación consta de ecuaciones (Moya, 2017) que permiten conocer cómo se mueven los robots móviles en función del giro de sus ruedas en donde se obtiene un valor incremental de los pulsos sensados por el optointerruptor FC-03, se analiza un factor de conversión en función del diámetro de la rueda (D) y la resolución total de los encoders instalados (R_T) tal como se indica a continuación:

$$\text{Factor de conversión} = \frac{\pi \times D}{R_T} \quad (6)$$

Cada plataforma cuenta con dos encoders y cada encoder instalado cuenta con una resolución de 20 ppr., por lo tanto se obtiene un resolución total de 40, el diámetro de las ruedas es de 10 cm, al remplazar estos valores en la ecuación 6 se obtiene el siguiente factor de conversión:

$$\text{Factor de conversión} = \frac{\pi \times 10 \text{ cm}}{40 \text{ pul/rev}} = 0.785 \text{ cm/rev} \quad (7)$$

Una vez calculado el factor de conversión, se analiza la distancia recorrida (d_i, d_d) por cada rueda en función de los pulsos detectados por sus respectivos optointerruptores (P_i, P_d) y finalmente se calcula la distancia total recorrida por la plataforma (d_T) como un promedio de las distancias recorridas por las dos ruedas.

$$d_i = \text{Factor de conversión} \times P_i = 0.785 \times P_i \quad (8)$$

$$d_d = \text{Factor de conversión} \times P_d = 0.785 \times P_d \quad (9)$$

$$d_T = \frac{d_i + d_d}{2} = \frac{0.785(P_i + P_d)}{2} = 0.395(P_i + P_d) \quad (10)$$

De esta manera la distancia recorrida en cualquier instante de tiempo se obtiene de la ecuación 10 definida como $0.395(P_i + P_d)$. Para conocer el segundo parámetro de entrada del sistema,

correspondiente al ángulo en el que se encuentra el robot móvil, se utiliza el valor absoluto del ángulo yaw (ψ) obtenido mediante el giroscopio incorporado en la IMU BNO055. Los valores correspondientes a la distancia recorrida y el ángulo son enviados hacia la Raspberry Pi 3, el ángulo de giro es calculado en función del ángulo anterior (a_{n-1}), el ángulo actual (a_n) y la bandera de giro (B) la cual indica si el giro se ha realizado hacia la derecha o a la izquierda. La programación para obtener el ángulo de giro se detalla en el diagrama de flujo de la Figura 52.

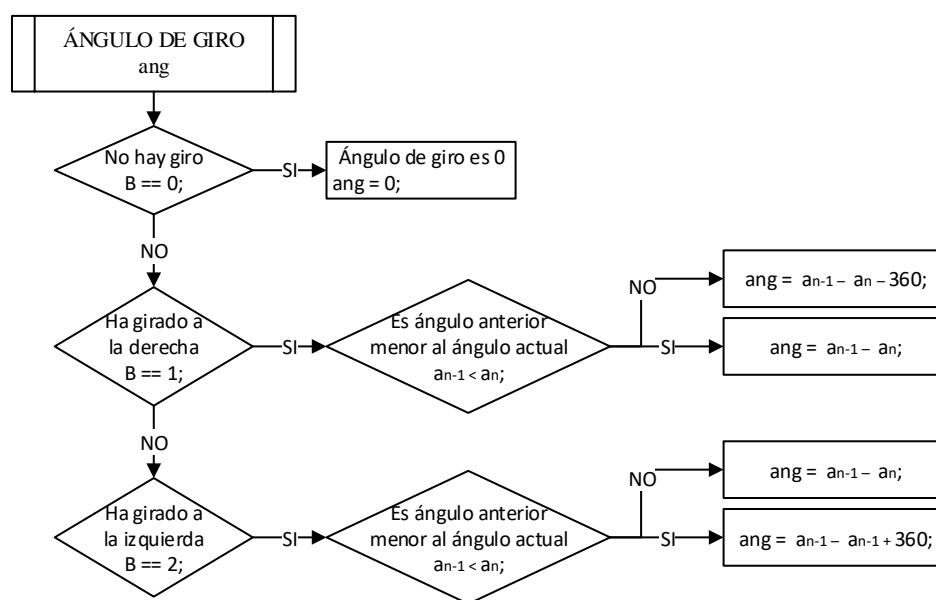


Figura 52. Diagrama de flujo para valor del ángulo de giro de un robot móvil autónomo.

a_n : Ángulo actual
 a_{n-1} : Ángulo anterior
 ang : Ángulo de giro calculado

Tabla 9

Condiciones para determinar el ángulo de giro

Giro	Condición	Ángulo de giro
Izquierda	$a_{n-1} < a_n$	$ang = a_{n-1} - a_n$ (11)
	$a_{n-1} > a_n$	$ang = a_{n-1} - a_n - 360$ (12)

CONTINÚA

Derecha	$a_{n-1} < a_n$	$ang = a_{n-1} - a_n + 360$	(13)
	$a_{n-1} > a_n$	$ang = a_{n-1} - a_n$	(14)

En un caso a manera de ejemplo (ver Figura 53), en el que un robot móvil de coordenadas iniciales (0,0), se traslada 20 cm, gira 90 grados a la derecha, y avanza 10 cm más, las coordenadas P_x y P_y se calcularían de la siguiente manera:

$$I \cdot T(0,20,0) \cdot R(-90^\circ) \cdot T(0,10,0) = \begin{bmatrix} * & * & * & P_x \\ * & * & * & P_y \\ * & * & * & P_z \\ * & * & * & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 20 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(-90) & -\sin(-90) & 0 & 0 \\ \sin(-90) & \cos(-90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 10 \\ -1 & 0 & 0 & 20 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

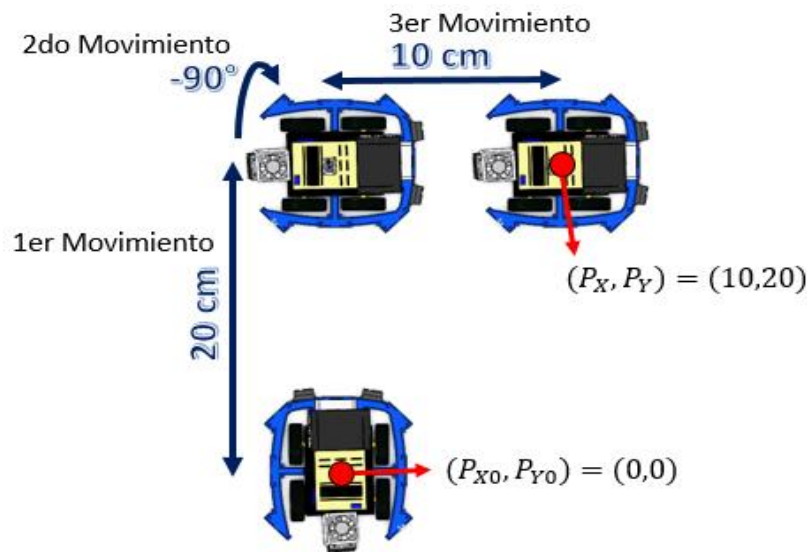


Figura 53. Posición final de un robot móvil.

De la misma manera se calcula la posición de la nariz electrónica y de los obstáculos o fronteras detectados por cada sensor ultrasónico en donde el parámetro dy corresponde a la suma de la

distancia desde el centro del robot hasta la ubicación del sensor más la distancia obtenida por el sensor. En la Figura 54 se muestra un ejemplo en el cual el sensor ultrasónico frontal derecho detecta un obstáculo a una distancia de 5 cm, por lo tanto, dy tiene el valor de 18.2 cm y un ángulo de giro de -45° .

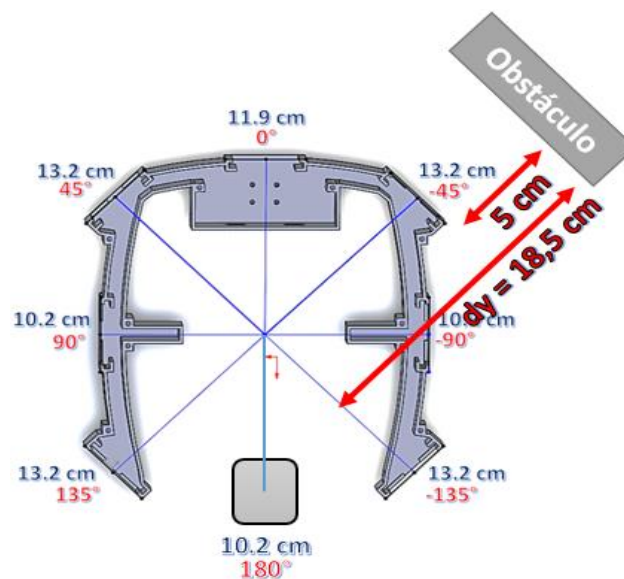


Figura 54. Distancia de un obstáculo.

4.6. Algoritmo de navegación

Una vez calculadas las coordenadas de cada robot móvil, las mismas son enviadas mediante comunicación inalámbrica hacia Matlab, para graficar los puntos obtenidos, guardarlos y posteriormente procesarlos con el objetivo de identificar zonas no exploradas.

En Matlab se tiene una matriz de 240×180 cuyas dimensiones se asemejan al espacio de exploración, la matriz se encuentra dividida en zonas de 40×40 , y cada zona se divide en subzonas de 5×5 , de cada subzona se predefine las coordenadas de su centro, de esta manera en caso de que un robot se encuentre con un obstáculo, el mismo sepa hacia dónde dirigirse.

Cada que Matlab recibe las coordenadas actuales de un robot, el espacio explorado corresponde a una cuadrícula de 20×20 cuyo centro corresponde a dichas coordenadas, si uno de esos puntos se encuentra dentro de una subzona, la misma se considera como explorada, de igual forma con los puntos correspondientes a obstáculos o fronteras.

Si el robot detecta en su frente, un obstáculo a una distancia menor a 15 cm , el sistema de control centralizado programado en Matlab se encarga de analizar la zona más lejana con mayor cantidad de subzonas no exploradas, de dicha zona se analiza la distancia desde las coordenadas actuales hacia la subzona no explorada más lejana y dichas coordenadas predefinidas son enviadas vía inalámbrica hacia la raspberry Pi 3, en donde mediante cinemática inversa se calcula el ángulo que debe girar el robot para llegar a la subzona no explorada.

Todos los robots generan movimientos simultáneamente después de cada escaneo, sean estos movimientos de traslación o movimientos de rotación, sin embargo, el orden de procesamiento de la información se realiza en el siguiente orden: primero robot 1, segundo robot 2 y tercero robot 3.

En caso de que más de un robot se encuentre con un obstáculo, el primero determinará la zona menos explorada más lejana, el siguiente la segunda zona menos explorada y de igual forma el tercero con la tercera zona menos explorada, de esta manera se garantiza que dos o tres robots no se dirijan a una misma zona.

En la Figura 55 se ejemplifica un caso hipotético en el cual la mayoría del entorno ya ha sido explorado a excepción de las zonas cercanas al origen $(0, 0)$ que se toma como referencia en la matriz, en este caso se puede apreciar 5 zonas con 5 cantidades de subzonas no exploradas, el robot 1 es el primero en detectar un obstáculo, por lo tanto el sistema determina que debería dirigirse hacia la zona con 8 subzonas no exploradas, el robot 2 es el segundo en detectar frente a él un obstáculo por lo tanto debería dirigirse hacia la zona con 16 subzonas no exploradas y el tercer

robot al encontrarse en la misma situación debería dirigirse hacia la zona con 25 subzonas no exploradas, en el último caso existen dos zonas con igual cantidad de subzonas no exploradas, sin embargo se determina la zona que tenga mayor distancia.

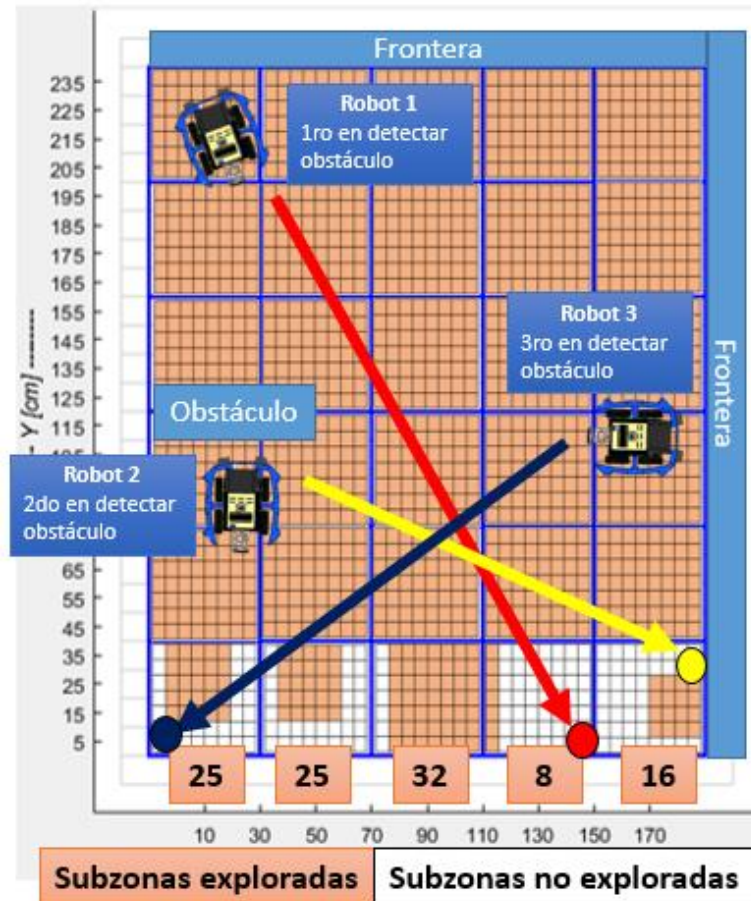


Figura 55. Navegación de robot móviles determinando zonas no exploradas.

4.6.1. Algoritmo para solución de cinemática inversa

El algoritmo de solución para la cinemática inversa de un robot móvil se basa en un sistema que consta de dos entradas (posición en X y Y) y dos salidas correspondientes al ángulo de giro del

robot y la distancia desde el punto de referencia hasta las coordenadas de entrada del sistema, todo respecto a la matriz implementada en el entorno de navegación.

La primera salida correspondiente a la distancia, ya fue calculada en el algoritmo de navegación, donde para encontrar la zona no explorada más lejana, se realiza una comparación entre todas las distancias hacia las subzonas no exploradas y se toma la de mayor valor; para calcular la segunda salida correspondiente al ángulo de giro hacia la subzona no explorada más lejana se identificó 64 casos (ver Figura 56) en función de dos pendientes de una recta: la primera calculada desde las coordenadas anteriores hasta las coordenadas actuales y la segunda calculada desde las coordenadas actuales hasta las coordenadas a las cuales se quiere llegar. A continuación, se detallan cada uno de los casos analizados para conseguir la cinemática inversa del robot móvil:

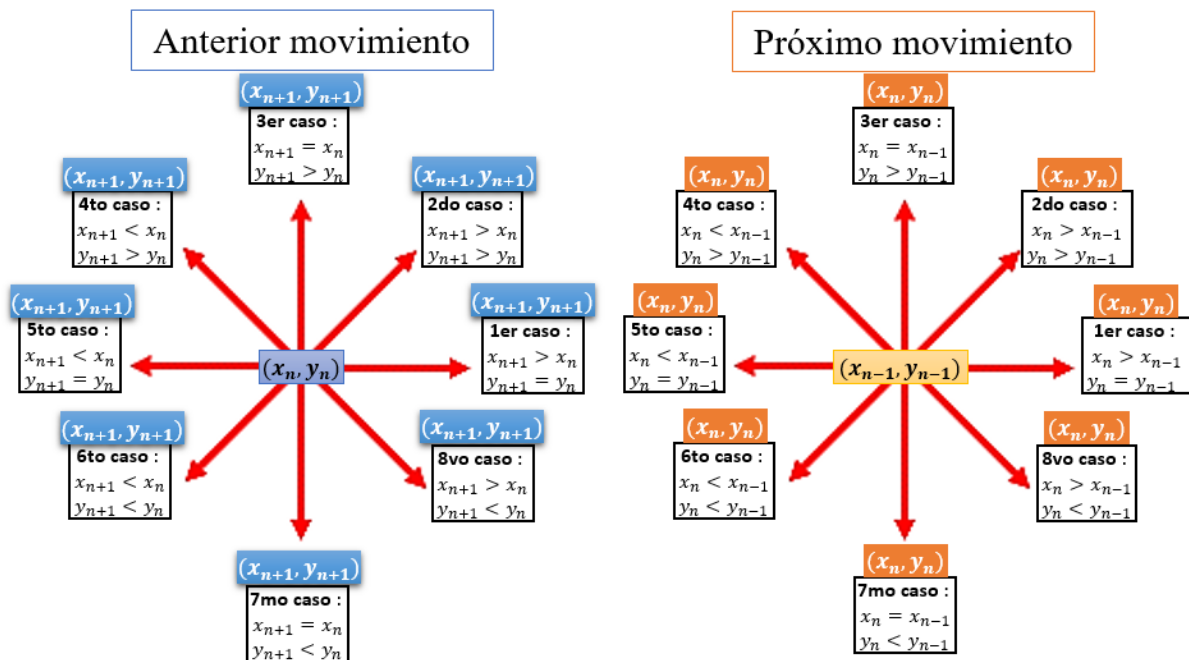


Figura 56. Casos para determinar cinemática inversa del robot móvil autónomo.

$$m_1 = \arctan\left(\frac{y_n - y_{n-1}}{x_n - x_{n-1}}\right) \quad (15)$$

$$m_2 = \arctan\left(\frac{y_{n+1} - y_n}{x_{n+1} - x_n}\right) \quad (16)$$

x_n : Coordenada actual en X y_n : Coordenada actual en Y
 x_{n-1} : Coordenada anterior en X y_{n-1} : Coordenada anterior en Y
 x_{n+1} : Coordenada futura en X y_{n+1} : Coordenada futura en Y

Tabla 10

Primer caso para determinar la cinemática inversa de un robot

Primer caso		$x_n > x_{n-1}$	$y_n = y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	0	(17)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$360 - m_2$	(18)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	270	(19)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$180 - m_2$	(20)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	180	(21)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$180 - m_2$	(22)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	90	(23)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$-m_2$	(24)

Tabla 11

Segundo caso para determinar la cinemática inversa de un robot

Segundo caso		$x_n > x_{n-1}$	$y_n > y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	m_2	(25)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$m_1 - m_2$	(26)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	$270 + m_2$	(27)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$180 - m_1 - m_2$	(28)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	$180 + m_1$	(29)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$180 + m_1 - m_2$	(30)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	$90 + m_1$	(31)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$m_1 - m_2$	(32)

Tabla 12

Tercer caso para determinar la cinemática inversa de un robot

Tercer caso		$x_n = x_{n-1}$	$y_n > y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	90	(33)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$90 - m_2$	(34)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	0	(35)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$270 - m_2$	(36)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	270	(37)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$270 - m_2$	(38)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	90	(39)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$-m_2$	(40)

Tabla 13

Cuarto caso para determinar la cinemática inversa de un robot

Cuarto caso		$x_n < x_{n-1}$	$y_n > y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	$180 + m_1$	(41)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$180 - m_1 - m_2$	(42)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	$90 + m_1$	(43)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$m_1 - m_2$	(44)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	$360 + m_1$	(45)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$360 + m_1 - m_2$	(46)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	$270 + m_1$	(47)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$180 + m_1 - m_2$	(48)

Tabla 14

Quinto caso para determinar la cinemática inversa de un robot

Quinto caso		$x_n < x_{n-1}$	$y_n = y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	180	(49)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$180 - m_2$	(50)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	90	(51)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$-m_2$	(52)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	0	(53)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$360 - m_2$	(54)

CONTINÚA 

$x_{n+1} = x_n$	$y_{n+1} < y_n$	270	(55)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$180 - m_2$	(56)

Tabla 15

Sexto caso para determinar la cinemática inversa de un robot

Sexto caso		$x_n < x_{n-1}$	$y_n < y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	$180 + m_1$	(57)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$180 + m_1 - m_2$	(58)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	$90 + m_1$	(59)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$m_1 - m_2$	(60)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	m_2	(61)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$m_1 - m_2$	(62)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	$270 + m_1$	(63)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$180 + m_1 - m_2$	(64)

Tabla 16

Séptimo caso para determinar la cinemática inversa de un robot

Séptimo caso		$x_n = x_{n-1}$	$y_n < y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	270	(65)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$270 - m_2$	(66)
$x_{n+1} = x_n$	$y_{n+1} > y_n$	180	(67)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$90 - m_2$	(68)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	90	(69)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$90 - m_2$	(70)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	0	(71)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$270 - m_2$	(72)

Tabla 17

Octavo caso para determinar la cinemática inversa de un robot

Octavo caso		$x_n > x_{n-1}$	$y_n < y_{n-1}$
Comparación de coordenadas en X	Comparación de coordenadas en Y	Ángulo de giro hacia zonas no exploradas	
$x_{n+1} > x_n$	$y_{n+1} = y_n$	$360 + m_1$	(73)
$x_{n+1} > x_n$	$y_{n+1} > y_n$	$360 + m_1 - m_2$	(74)

CONTINÚA 

$x_{n+1} = x_n$	$y_{n+1} > y_n$	$270 + m_1$	(75)
$x_{n+1} < x_n$	$y_{n+1} > y_n$	$180 + m_1 - m_2$	(76)
$x_{n+1} < x_n$	$y_{n+1} = y_n$	$180 + m_1$	(77)
$x_{n+1} < x_n$	$y_{n+1} < y_n$	$180 + m_1 - m_2$	(78)
$x_{n+1} = x_n$	$y_{n+1} < y_n$	$90 + m_1$	(79)
$x_{n+1} > x_n$	$y_{n+1} < y_n$	$m_1 - m_2$	(80)

4.7. Desarrollo del sistema de comunicación

4.7.1. Comunicación entre Teensy 3.6 y Raspberry Pi 3

4.7.1.1. Comunicación serial USB

Se dispuso una comunicación serial USB para el envío y recepción de datos entre la tarjeta programable Teensy 3.6 y Python (Raspberry Pi 3). El add-on Teensyduino ofrece un objeto utilizado para imprimir información en el monitor serial, similar al proporcionado por un Arduino, con la diferencia de que todas las comunicaciones que generan un objeto Teensy serial, siempre son ejecutadas a velocidad nativa USB de 12Mbits/sec.

4.7.1.2. Envío y recepción: Teensy 3.6 – Raspberry Pi 3

Para permitir la interacción entre la Teensy 3.6 y Python o Matlab, debe utilizarse la comunicación `/dev/ttyACM` definiéndose como un dispositivo de comunicación USB (CDC en inglés *communication device*) del subtipo de "modelo de control abstracto" (ACM en inglés *abstract model control*) (Electronic Projects Components Available Worldwide, 2018). En la Figura 57 se muestra el diagrama de flujo correspondiente al código programado en la Teensy 3.6 y en Python (Raspberry Pi 3) para lograr una comunicación bidireccional de envío y recepción de datos.

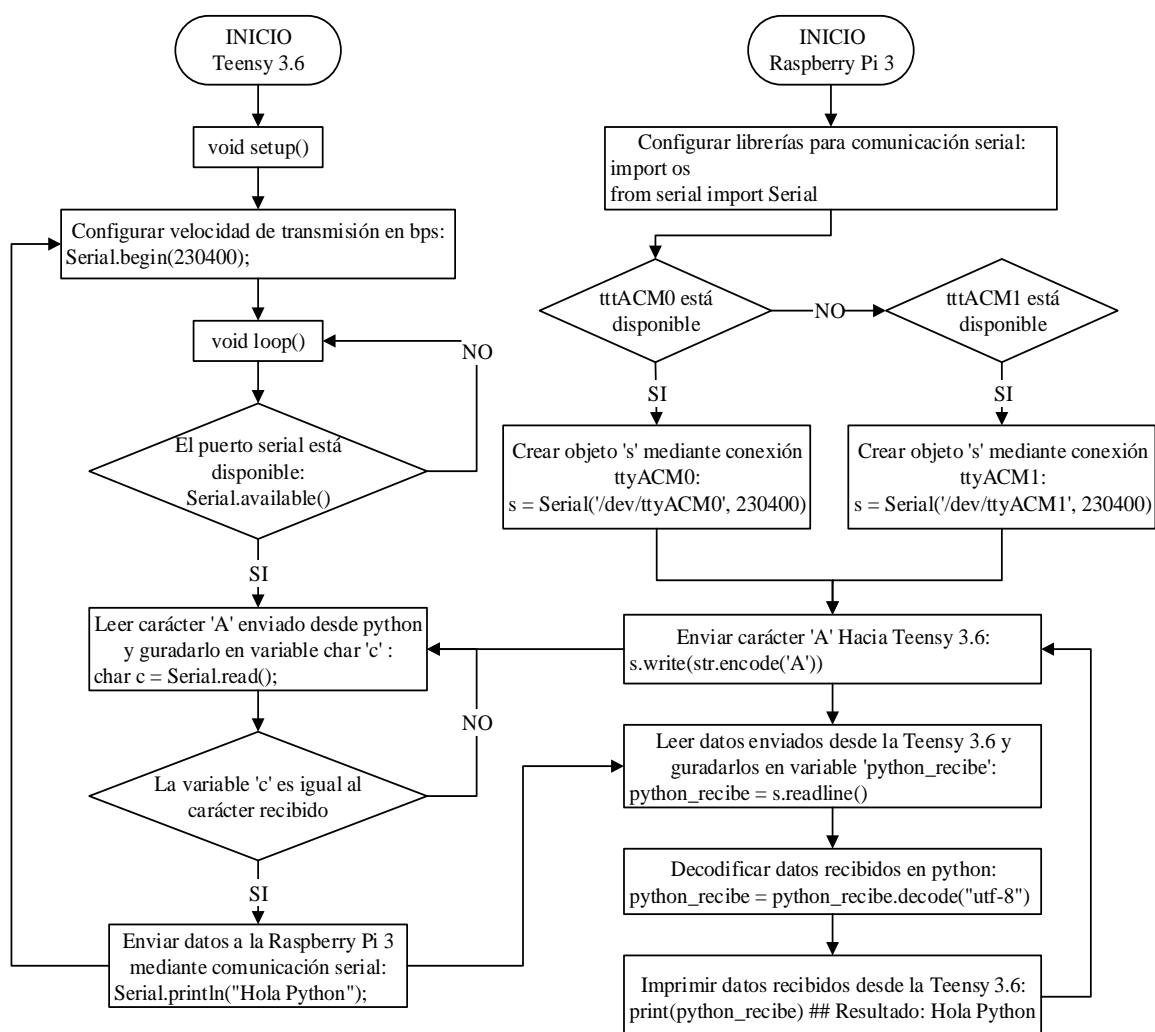


Figura 57. Diagrama de flujo de envío y recepción de datos entre Teensy 3.6 y Raspberry.

4.7.1.3. Trama Teensy 3.6 – Raspberry Pi 3

En la tarjeta programable Teensy 3.6 se realiza toda la adquisición de sensores y su interacción directa con el entorno, posteriormente cada dato es enviado en una trama tal como se indica en la Figura 58, los datos son delimitados por una coma ‘,’ con el objetivo de conocer donde comienza y donde termina cada valor enviado y recibido, de esta manera se permite la desconcatenación de la trama mediante lenguaje de programación Python en la Raspberry Pi 3.

Obst	Giro	Ángulo	Temperatura	Gas	Distancia 1	Distancia 2
------	------	--------	-------------	-----	-------------	-------------

Figura 58. Trama de datos entre Teensy 3.6 y Raspberry Pi 3.

- **Obst:** Mediante una bandera O, se indica si el robot ha detectado frente a él, la presencia de un obstáculo, cuando existe el obstáculo $O = 1$, si es giro es hacia la derecha $O = 0$.
- **Giro:** mediante una bandera B, se indica si el robot ha girado hacia la derecha o izquierda, cuando el giro es hacia la izquierda $B = 1$, si es giro es hacia la derecha $B = 0$.
- **Ángulo:** indica el ángulo de giro determinado por la IMU BNO055, se utiliza el valor absoluto correspondiente al ángulo ψ (yaw) el cual representa giros de 0 a 360 grados alrededor del eje Z.
- **Temperatura:** nivel de temperatura en grados celsius ($^{\circ}\text{C}$) captado por el sensor LM35.
- **Gas:** Indica el porcentaje de concentración de gas captado por los sensores químicos de la familia Figaro TGS.
- **Distancia 1:** indica la distancia recorrida por el robot móvil calculada a partir de los pulsos generados en las revoluciones del encoder y sensados por el módulo optointerruptor FC-03.
- **Distancia 2:** indica la distancia captada por un sensor ultrasónico donde el dato de la distancia obtenida por cada sensor es enviado por separado en cada trama para evitar problemas de sobrecarga de información en la comunicación serial USB.

4.7.2. Comunicación inalámbrica entre Raspberry Pi 3 y ordenador centralizado

4.7.2.1. Protocolos de transporte

La elección de un protocolo de transporte depende de varios requerimientos en la comunicación tales como la fiabilidad y la eficiencia. El protocolo de datagramas de usuario UDP (en inglés *User*

Datagram Protocol) se caracteriza por ser un método no orientado a la conexión y no fiable, al contrario del protocolo de control de transmisión TCP (en inglés *Transmission Control Protocol*) que es un método fiable y orientado a la conexión.

Uno de los requerimientos del sistema robótico móvil y autónomo es que sea un servicio fiable con adquisición de datos en tiempo real; por lo tanto, es viable el uso del protocolo TCP, sin embargo, al tomar en cuenta la cantidad de datos que son adquiridos por las tres plataformas robóticas y la necesidad de visualizar los datos en tiempo real, es más eficiente utilizar el protocolo UDP, de esta manera se descarta el tiempo necesario para establecer y liberar la conexión de cada robot móvil y a pesar de no poseer fiabilidad, el sistema se puede implementar con mecanismos de fiabilidad en la transmisión de datos, esto apoya la razón por la cual gran cantidad de sistemas de transmisión multimedia sobre IP tales como voz o video, utilizan el protocolo de transporte UDP.

4.7.2.2. Envío y recepción: Raspberry Pi 3 – Ordenador centralizado

En el presente proyecto de titulación, la comunicación inalámbrica entre los robots móviles y el ordenador en el cual se encuentra la interfaz gráfica de usuario GUI, se implementó mediante el protocolo UDP correspondiente a la capa de transporte del modelo OSI para la transmisión bidireccional de datos según el estándar RFC 768; se utilizaron sockets definidos como métodos de establecimiento de un enlace de conexión entre dos procesos en TCP/IP, para cada enlace es necesario establecer: el protocolo de transporte, dirección IP del destino, puerto del destino, dirección IP de origen y puerto del origen. En la Figura 59, Figura 60 y Figura 61 se muestran los diagramas de flujo correspondientes a los códigos programados en Python (Raspberry Pi 3) y Matlab para lograr una comunicación bidireccional de envío y recepción de datos.

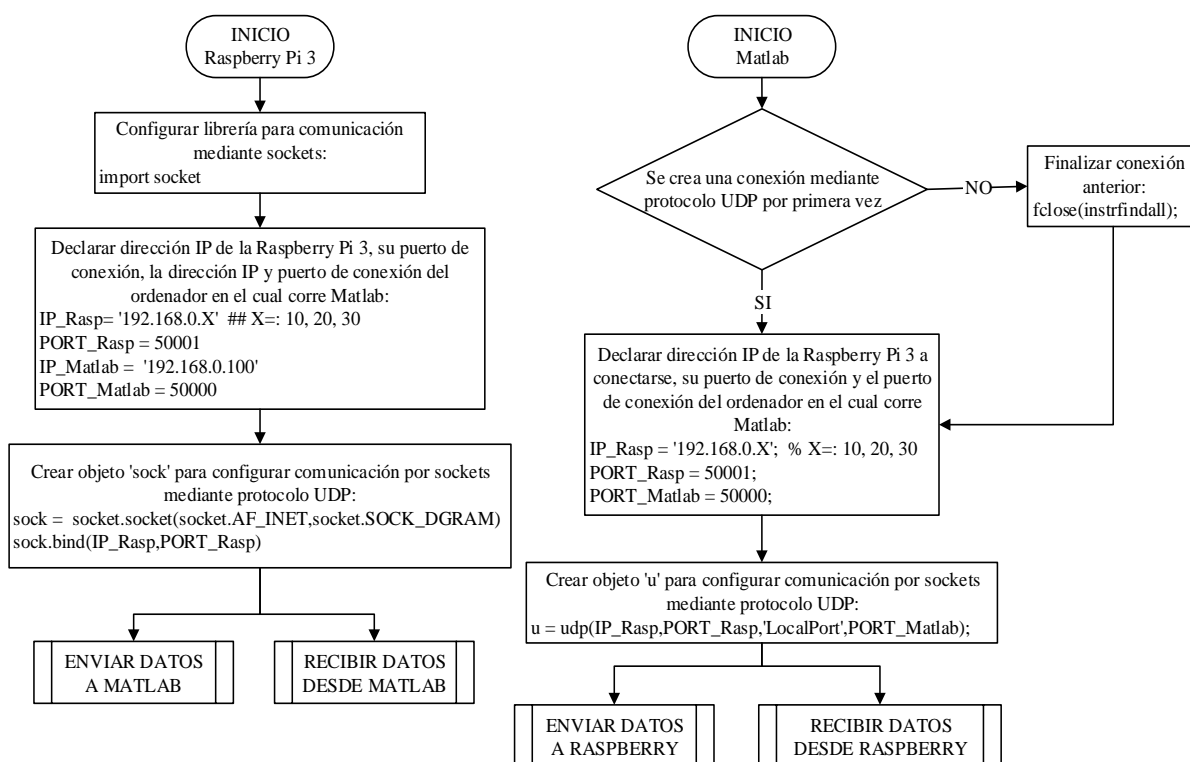


Figura 59. Diagrama de flujo de la configuración para comunicación inalámbrica bidireccional entre Raspberry y ordenador centralizado.

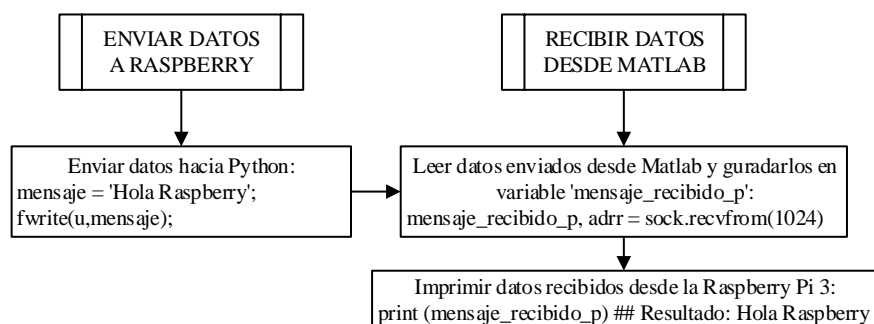


Figura 60. Flujo del envío de datos desde Raspberry y recepción en ordenador centralizado.

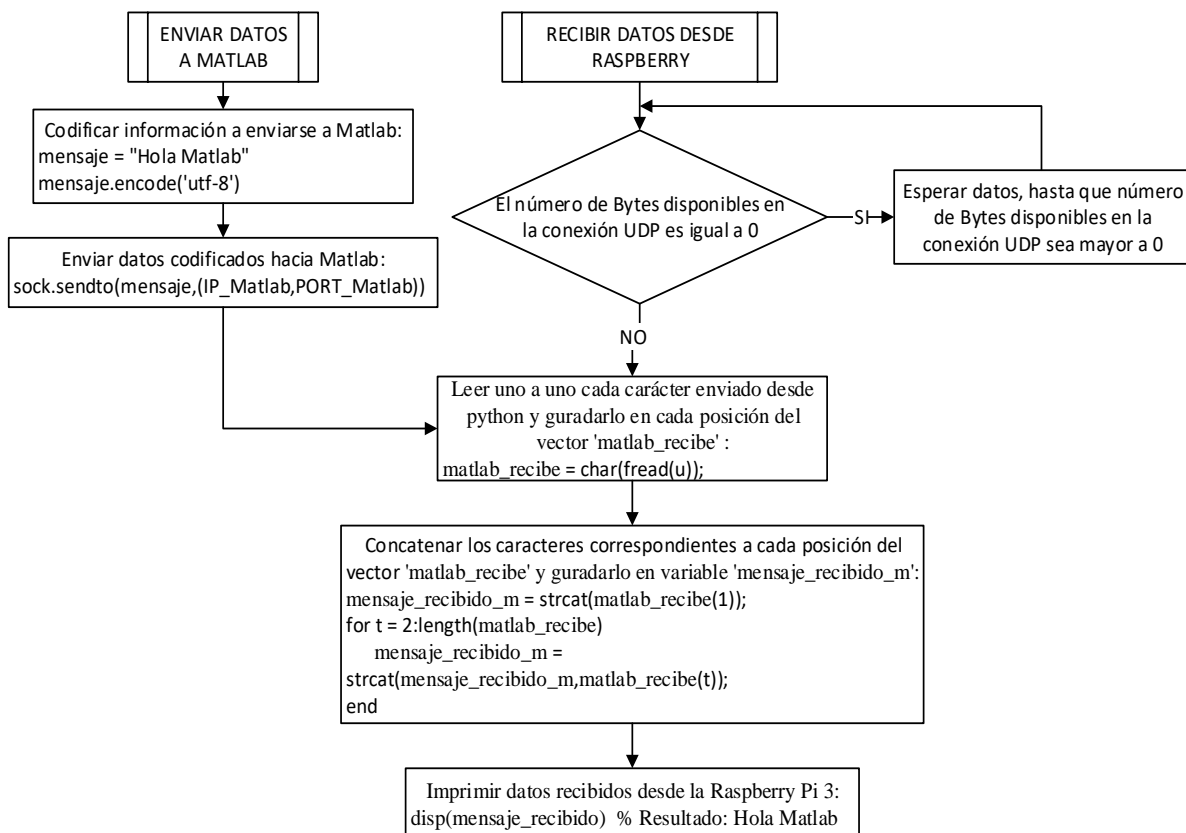


Figura 61. Diagrama de flujo del envío de datos desde ordenador centralizado y recepción en Raspberry.

4.7.2.3. Trama Raspberry Pi 3 – Ordenador centralizado

En la Raspberry Pi 3 se realiza el procesamiento de los datos adquiridos por la tarjeta programable Teensy 3.6 para obtener información relacionada a la navegación, posición del robot, obstáculos y fronteras del entorno mediante coordenadas en el plano XY, posteriormente cada dato es enviado en una trama tal como se indica en la Figura 62, los datos son delimitados por una coma ‘,’ con el objetivo de conocer donde comienza y donde termina cada valor enviado y recibido, de esta manera se permite la desconcatenación de la trama mediante programación en Matlab.

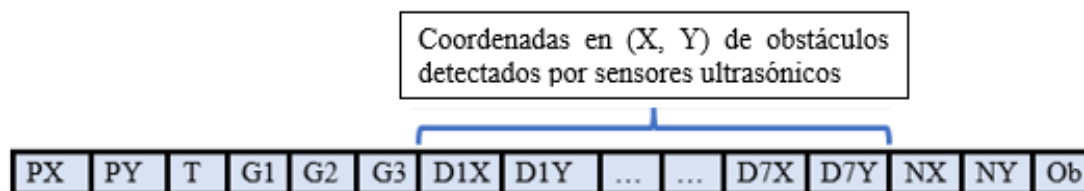


Figura 62. Trama de datos entre Raspberry Pi 3 y ordenador centralizado.

- PX: indica la coordenada en X correspondiente a la posición en tiempo real del robot móvil autónomo en un plano 2D (XY).
- PY: indica la coordenada en Y correspondiente a la posición en tiempo real del robot móvil autónomo en un plano 2D (XY).
- T: indica el nivel de temperatura en grados celsius ($^{\circ}\text{C}$) captado por el sensor LM35.
- G1: Indica el porcentaje de concentración de gas captado por el sensor químico TGS2610.
- G2: Indica el porcentaje de concentración de gas captado por el sensor químico TGS2600.
- G3: Indica el porcentaje de concentración de gas captado por el sensor químico TGS2602.
- Posiciones en X, Y (Ultrasónicos): indican las coordenadas en (X, Y), correspondientes a las posiciones en tiempo real de los obstáculos o fronteras del entorno de navegación captadas por los sensores ultrasónicos frontal, frontal derecho, frontal izquierdo, derecho, izquierdo, posterior derecho y posterior izquierdo.
- NX: indica la coordenada en X correspondiente a la posición en tiempo real de la posición de la nariz electrónica.
- NY: indica la coordenada en Y correspondiente a la posición en tiempo real de la posición de la nariz electrónica.
- Ob: Mediante una bandera O, se indica si el robot ha detectado frente a él la presencia de un obstáculo, cuando existe el obstáculo $O = 1$, cuando no existe obstáculo $O = 0$, en caso de

que exista un obstáculo Matlab procesa todos los datos adquiridos por el sistema para determinar las coordenadas correspondientes al punto no explorado más lejano y retornar dichos valores a la Raspberry Pi 3, en donde se determina el ángulo necesario para llegar a dicho punto, caso contrario se retorna un comando para que el robot continúe su navegación en línea recta.

Todos los datos de la trama proveniente de la Raspberry Pi 3 son codificados para ser utilizados en la programación de la interfaz gráfica de usuario y el controlador centralizado, en la Tabla 18 se enlista cada una de estas codificaciones.

Tabla 18

Variables de la interfaz gráfica de usuario y sistema de control centralizado

Etiqueta	Descripción
cx_rb1	Posición en X del robot móvil 1
cy_rb1	Posición en Y del robot autónomo móvil 1
cx_rb2	Posición en X del robot autónomo móvil 2
cy_rb2	Posición en Y del robot autónomo móvil 2
cx_rb3	Posición en X del robot autónomo móvil 3
cy_rb3	Posición en Y del robot autónomo móvil 3
temp_rb1	Temperatura detectada por el robot autónomo móvil 1
temp_rb2	Temperatura detectada por el robot autónomo móvil 2
temp_rb3	Temperatura detectada por el robot autónomo móvil 3
tgs1_rb1	Concentración de gas detectado por el sensor TGS2610 del robot 1
tgs2_rb1	Concentración de gas detectado por el sensor TGS2600 del robot 1
tgs3_rb1	Concentración de gas detectado por el sensor TGS2602 del robot 1
tgs1_rb2	Concentración de gas detectado por el sensor TGS2610 del robot 2
tgs2_rb2	Concentración de gas detectado por el sensor TGS2600 del robot 2
tgs3_rb2	Concentración de gas detectado por el sensor TGS2602 del robot 2
tgs1_rb3	Concentración de gas detectado por el sensor TGS2610 del robot 3
tgs2_rb3	Concentración de gas detectado por el sensor TGS2600 del robot 3
tgs3_rb3	Concentración de gas detectado por el sensor TGS2602 del robot 3

4.7.3. Trama de retorno

Tras haber realizado todo el procesamiento del controlador centralizado, como respuesta existen 2 opciones: orden de navegación en línea recta o una orden de giro. Como se ha mencionó en el algoritmo de navegación, las opciones dependen de la presencia de un obstáculo frente a un robot móvil. En la Figura 63 se muestra las 2 opciones de trama de retorno.

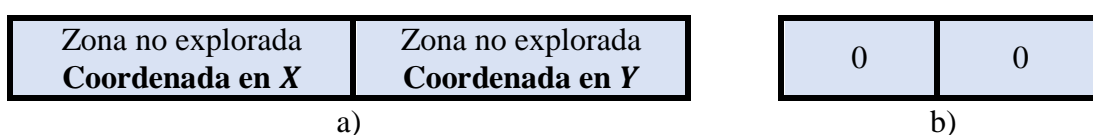


Figura 63. a) Trama en caso de haber detectado un obstáculo frente a un robot móvil, b) Trama en caso de haber detectado un obstáculo frente a un robot móvil.

Al detectarse un obstáculo se envían hacia la Raspberry Pi 3, las coordenadas en (X, Y) de la zona no explorada calculada por el controlador central, para posteriormente realizar la solución de la cinemática inversa donde a partir de las coordenadas se calcula un ángulo de giro que se retorna a la Teensy 3.6, en caso de no detectarse obstáculo, se envía por duplicado el carácter '0', debido al tamaño de la trama que representan los valores de las coordenadas en (X, Y) de las zonas no exploradas, el carácter '0' en la Teensy 3.6 también representa la orden de navegación rectilínea.

4.8. Definición del sistema robótico móvil autónomo

En base a la descripción funcional descrita en este capítulo, se define a cada robot móvil como un agente inteligente basado en metas (ver Figura 64) dado que, durante la navegación son capaces de modificar sus movimientos considerando como objetivo, la exploración de zonas no exploradas del entorno mediante un sistema de comunicación de control centralizado para el trabajo

cooperativo integrando la información obtenida por cada uno de los robots. Además, el comportamiento del sistema se basa en un control híbrido considerando que sus acciones se dimensionan bajo operaciones reactivas dirigidas por los sensores y operaciones deliberativas para planificar objetivos, como es el caso de la detección de obstáculos mediante sensores ultrasónicos y planificación de la mejor ruta hacia zonas no exploradas cuyos cálculos se realizan en el ordenador central y raspberry Pi 3.

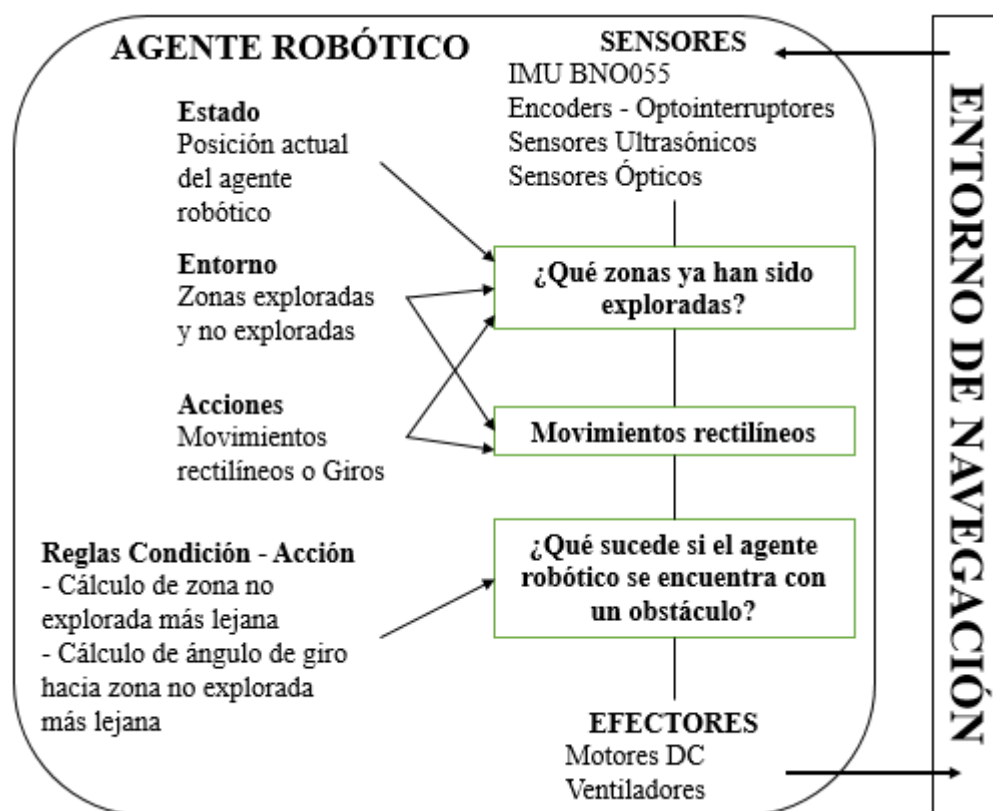


Figura 64. Esquema del robot móvil como agente inteligente basado en metas

4.9. Interacción del algoritmo de navegación y comunicación

Una vez definidos los métodos de comunicación bidireccional se proceden a explicar cómo se integran tres sistemas (ver Figura 65) logrando ejecutar un total de 7 programas: control centralizado en Matlab, procesamiento de datos en Python (1 programa por cada robot móvil autónomo), y adquisición de datos en Teensyduino (1 programa por cada robot móvil autónomo).

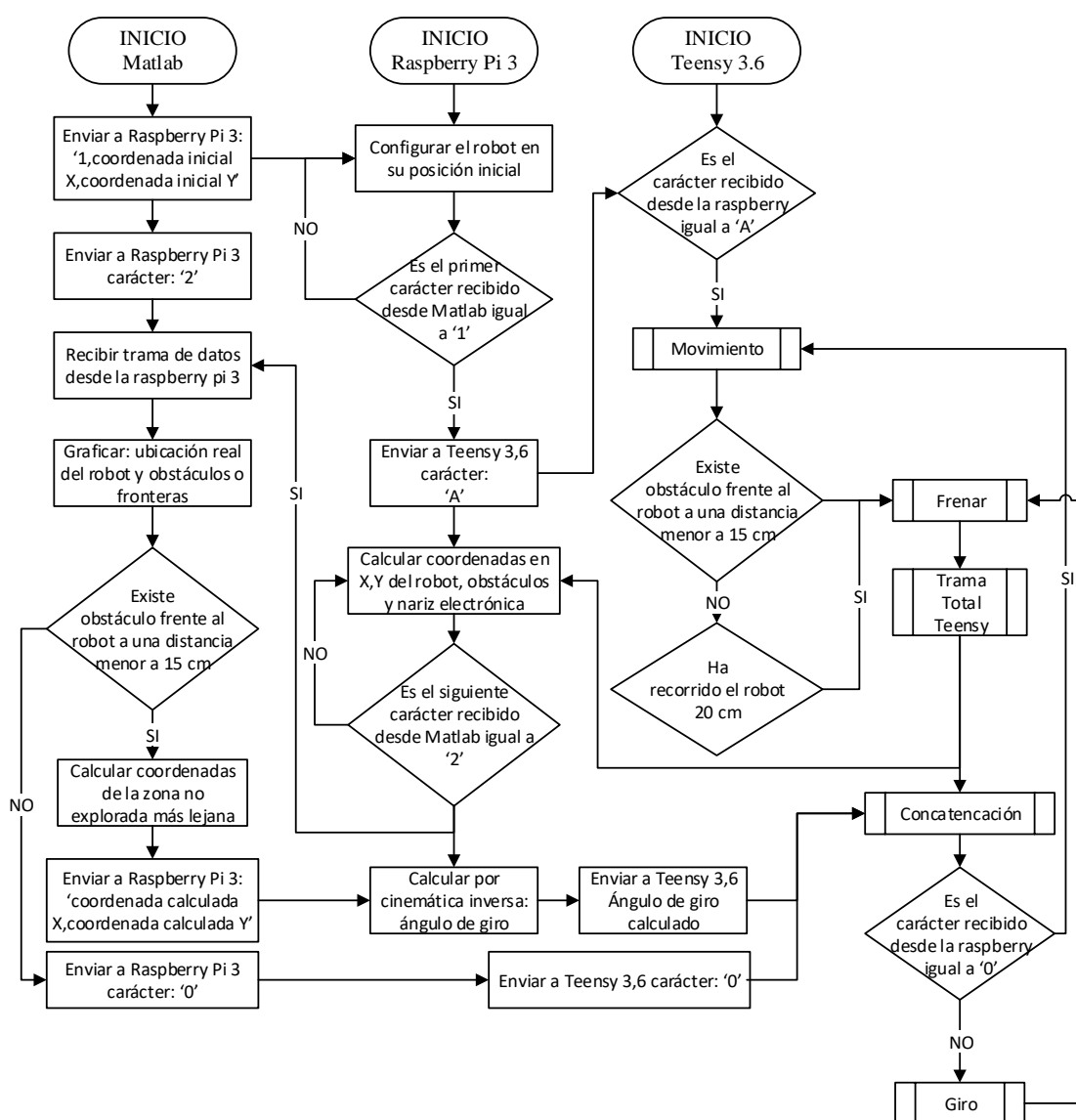


Figura 65. Diagrama de flujo para interacción de algoritmos de control y comunicación.

CAPÍTULO V

VISUALIZACIÓN DE MAPAS, PRUEBAS Y RESULTADOS

5.1. Integración del sistema de control centralizado

La integración de todo el sistema robótico cooperativo se realizó mediante una interfaz gráfica de control centralizado la cual permite al usuario visualizar toda la información de navegación y los resultados del sistema robótico cooperativo mediante mapas procesados de obstáculos o fronteras, niveles de temperatura y niveles concentración de gases, a continuación, se detalla los parámetros requeridos para alcanzar una correcta integración de sistemas.

- Monitorización de la posición en tiempo real, de los 3 agentes robóticos mediante un gráfico 2D delimitado por una cuadrícula similar al espacio de trabajo que se encuentra en el laboratorio de pruebas químicas con robots en la Universidad de la Fuerzas Armadas ESPE.
- Generación de los mapas correspondientes a la ubicación de obstáculos o fronteras detectados por cada robot móvil autónomo y el sistema robótico cooperativo en el entorno de navegación.
- Generación del mapa correspondiente a los niveles de temperatura sensados por cada robot móvil autónomo y el sistema robótico cooperativo en el entorno de navegación.
- Generación del mapa correspondiente a los niveles de concentración de gases sensados por cada robot móvil autónomo y el sistema robótico cooperativo en el entorno de navegación.

5.2. Interfaz gráfica de control centralizado

Uno de los objetivos fundamentales del proyecto de investigación, es poder visualizar en tiempo real el mapeo del entorno de navegación generado durante la exploración de los robots móviles autónomos cooperativos, para ello se implementó una interfaz gráfica de usuario GUI programada en el software de desarrollo matemático Matlab R2017b.

La interfaz gráfica cumple con criterios de diseño, simetría y proporcionalidad basados en la guía GEDIS y la norma ISA 101 para lograr una correcta distribución de la pantalla, uso de colores, e información a visualizarse, tal como se detalla a continuación:

- Manejo de información: los datos y parámetros más relevantes del proceso se deben mostrar mediante indicadores visuales dentro de la interfaz principal.
- Manejo del color: la información considerada de mayor importancia, se le asigna cierta intensidad de colores llamativos para captar la atención de los parámetros principales del proceso.
- Visibilidad: cada componente, objeto, números o letras ubicadas dentro de la interfaz gráfica, deben tener el tamaño y colores adecuados para evitar molestia visual por parte del usuario.
- Perceptibilidad: el usuario tiene la capacidad de conocer en cada instante, el estado actual del proceso para permitir una correcta toma de decisiones o análisis de resultados al momento del monitoreo.
- Interactividad: El usuario puede elegir, activar y desactivar comandos desde la interfaz gráfica que permitan una interacción directa con el proceso.

5.2.1. Distribución de la pantalla

Las diferentes pantallas de la interfaz gráfica de usuario para el sistema robótico móvil cooperativo constan de una distribución del espacio basado en normas y estándares establecidos para el diseño de las mismas. A continuación, se describen como se encuentra distribuidas las diferentes pantallas creadas para dicho sistema.

5.2.1.1. Pantalla de Inicio

La pantalla de inicio (ver Figura 66) tiene un tamaño de [83 × 34] y está conformada por los siguientes elementos:

- Logo: Presenta la imagen correspondiente al logotipo de la Universidad de las Fuerzas Armadas ESPE, ubicado en la parte superior de la pantalla de inicio.
- Título: Texto que presenta el título del proyecto de investigación, ubicado en la parte superior de la pantalla de inicio por debajo del logotipo de la universidad.
- Usuario: Cuadro de texto en el cual se ingresa el nombre de usuario y que junto con la contraseña permiten el ingreso al sistema, el usuario está configurado como “Juan Illanes”.
- Contraseña: Cuadro de texto cifrado en el cual se ingresa la contraseña y que junto al usuario permiten el ingreso al sistema, la contraseña está configurada como “1234”.
- Acceso de usuario: Botón que permite ingresar al sistema en caso de que el usuario y la contraseña ingresada sean correctas, en caso de que sean incorrectas, se abre un *pop-up* indicando el error de *login* en el sistema.
- Autor: Texto que presenta el nombre del autor del proyecto de investigación, ubicado en la parte inferior derecha de la pantalla de inicio.



Figura 66. Pantalla de inicio, acceso de usuario y contraseña.

5.2.1.2. Pantalla de configuración

La pantalla de configuración (ver Figura 67) tiene un tamaño de [24 × 74] y está conformada por los siguientes elementos:

- Cantidad de robots: Presenta un campo conformado por 3 botones, permite seleccionar la cantidad de robots encargados de realizar el mapeo: un robot, dos robots o tres robots.
- Posiciones iniciales: Permite ingresar dentro de campos de texto las coordenadas iniciales (X,Y) donde han sido colocados los robots dentro del entorno de navegación.
- Regresar: Botón que permite retornar a la configuración anterior.
- Empezar: Botón que permite abrir la siguiente ventana de mapeo en tiempo real y activa los robots encargados de realizar el mapeo.



Figura 67. Pantalla de configuración de posiciones iniciales.

5.2.1.3. Pantalla de mapeo en tiempo real

La pantalla de mapeo en tiempo real (ver Figura 68) tiene un tamaño de [113 × 58] y está conformada por los siguientes elementos:

- Mapa: Figura (configuración *meshgrid*) que presenta una representación del entorno de navegación ubicado en el laboratorio de pruebas químicas con robots en la Universidad de las Fuerzas Armadas ESPE.
- Datos Actuales: Permite visualizar en tiempo real los datos recolectados por los robots móviles autónomos: posición en X, posición en Y, temperatura y nivel de concentración de gases (%) detectado por los sensores químicos TGS.
- Resultados: Botón que permite cortar la comunicación con los robots móviles autónomos, guardar la información recolectada durante el mapeo y abrir la siguiente ventana de resultados.

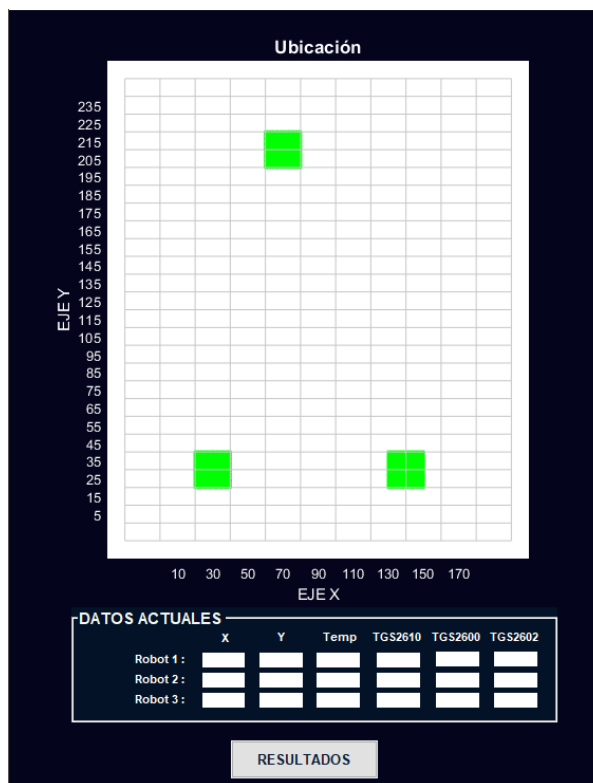


Figura 68. Pantalla de mapeo en tiempo real.

5.2.1.4. Pantalla de resultados

La pantalla de resultados (ver Figura 69) tiene un tamaño de $[30 \times 208]$ y está conformada por los siguientes elementos:

- **Obstáculos y fronteras:** Presenta un campo conformado por botones correspondientes a los robots móviles autónomos que han realizado el mapeo, permite visualizar dos mapas con los obstáculos o fronteras, el primer mapa muestra una gráfica de los puntos sensados y el segundo el procesamiento de dichos puntos mediante el método de grafos de visibilidad.
- **Temperatura:** Presenta un campo conformado por botones correspondientes a los robots móviles autónomos que han realizado el mapeo, permite visualizar dos graficas: la primera gráfica muestra las zonas por las cuales se ha navegado indicando su respectiva temperatura

sensada y la segunda gráfica indica el procesamiento de la primera gráfica mediante interpolación en 2d y un filtro espacial de media móvil, la temperatura es representada mediante un código de colores RGB en el cual, colores fríos representan niveles de temperatura bajos, y colores cálidos representan niveles de temperatura altos.

- **Gases:** Presenta un campo conformado por botones correspondientes a los robots móviles autónomos que han realizado el mapeo, permite visualizar dos graficas: la primera gráfica muestra las zonas por las cuales se ha navegado indicando su respectivo nivel de concentración de gases sensado y la segunda gráfica indica el procesamiento de la primera gráfica mediante interpolación en 2d y un filtro espacial de media móvil, el nivel de concentración de gases es representado mediante un código de colores RGB en el cual, colores fríos representan niveles de concentración bajos, y colores cálidos representan niveles de concentración altos.



Figura 69. Pantalla de resultados.

5.3. Visualización de mapa de obstáculos

Todos los puntos correspondientes a los obstáculos y fronteras recolectados durante el mapeo de los robots móviles, son guardados en una matriz para posteriormente ser procesados mediante

el método de grafos de visibilidad, para ello se basa en el concepto geométrico analítico de una recta entre dos puntos cuya fórmula se indica en la ecuación 81, de esta manera se verifica punto a punto distancias menores a 25 cm para poder graficarlas, esto permite visualizar la representación de obstáculos o fronteras únicamente de aquellos puntos que son cercanos entre sí. En la Figura 70 se presenta el ejemplo de una prueba real de mapeo de obstáculos y fronteras en el cual se puede apreciar el mapa dibujado mediante pixeles que representan los las ubicaciones en las cuales se ha detectado un obstáculo y la gráfica correspondiente al procesamiento del mapa anterior mediante grafos de visibilidad.

$$Distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (81)$$

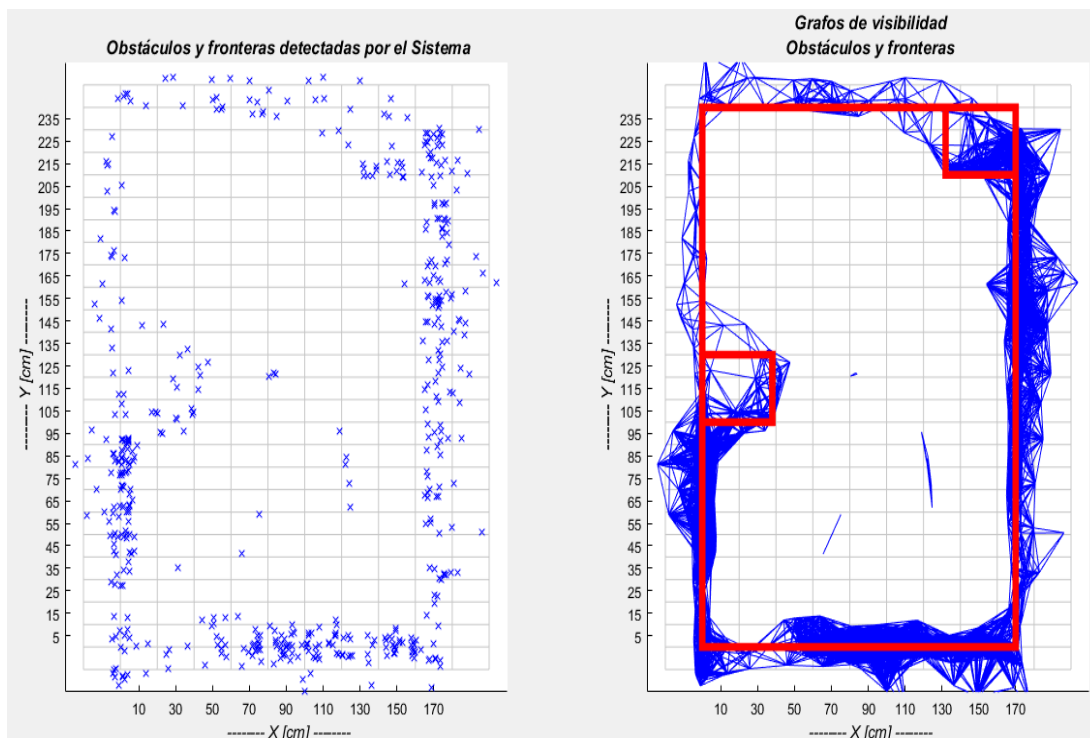


Figura 70. Puntos detectados como obstáculos y fronteras (izq.), Mapa procesado mediante grafos de visibilidad (der.).

5.4. Visualización de mapas de sensado químico y temperatura

Una vez finalizado la navegación por parte de los agentes robóticos, el sistema de control centralizado se encarga de generar mapas de niveles de temperatura y concentración de gases, en primera instancia se obtiene un mapa de 170×240 *pixeles* correspondiente a las dimensiones del área de trabajo, en dicho mapa se visualiza cuadrículas coloreadas de 20×20 *cm* correspondientes a las dimensiones aproximadas de un robot móvil, los niveles de temperatura y concentración de gases se representan mediante un código RGB en el cual colores fríos representan niveles de concentración bajos, y colores cálidos representan niveles de concentración altos.

Este mapa es procesado mediante un filtro espacial de media móvil en donde cada pixel es promediado con los valores de una cuadrícula de 10×10 *pixeles* a su alrededor, después se realiza un interpolación en 2D mediante funciones propias de Matlab y se vuelve a aplicar 14 veces más el filtro espacial debido a que en cada filtro en cual se procesa la matriz, se logra obtener una mejor dispersión de los valores alrededor de los datos adquiridos inicialmente, de esta manera se logra curvas más suaves en las gráficas a visualizarse.

En la Figura 71 se presenta el ejemplo de una prueba real de mapeo de niveles de temperatura en el ambiente correspondiente a un entorno controlado, en esta prueba se ha colocado un calentador eléctrico en la posición $P_C(90, 220)$ y un ventilador en la posición $P_V(90, 20)$, de esta manera se puede apreciar claramente como la temperatura presenta los niveles más altos en la dirección hacia la cual apunta el calentador, mientras que las zonas que se encuentran con niveles bajos de temperatura se encuentran cerca a la posición del ventilador.

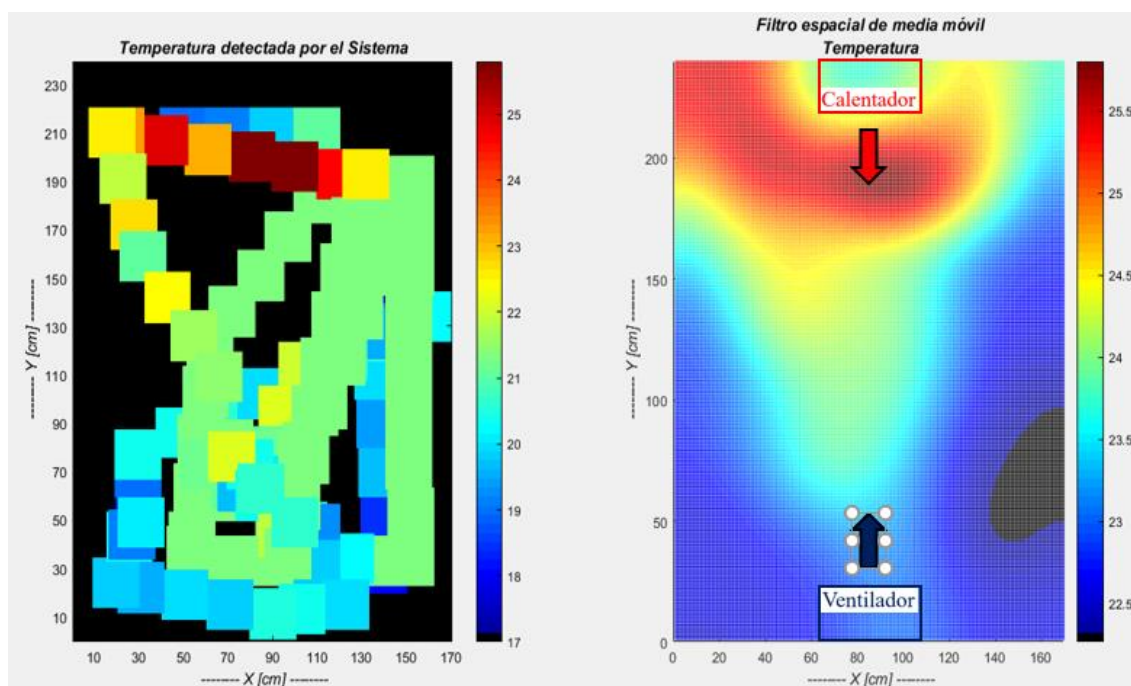


Figura 71. Zonas mapeadas con sus niveles de temperatura (izq.), Mapa 2D procesado mediante interpolación 2D y filtro espacial de media móvil (der.).

5.5. Pruebas de sistemas independientes

Los sistemas fueron probados por separado mediante la creación de programas independientes para comprobar su correcto funcionamiento. Una vez construidas las plataformas robóticas móviles en su versión final con todos los sistemas integrados se volvieron a probar uno a uno para determinar la existencia de variaciones en su comportamiento, de esta manera fue necesario la realización de ciertas correcciones que aseguren la efectividad en cada uno de requerimientos funcionales con todos los sistemas conectados entre sí.

5.5.1. Recorrido

Antes de realizar las pruebas de navegación fue necesario realizar una recalibración del recorrido constante entre cada escaneo y de las velocidades de los motores para permitir movimientos totalmente rectilíneos. Según la ecuación 10 del apartado 4.5 correspondiente al cálculo de la cinemática directa, es posible conocer la distancia recorrida por un robot móvil en cualquier instante de tiempo en función de un factor de conversión y los pulsos detectados por los encoders instalados en las ruedas posteriores izquierda y derecha. De esta forma se realizó un programa en la Teensy 3.6 con el objetivo de que cada agente robótico realice desplazamientos de 20 cm con intervalos de 5 segundos durante los cuales el robot se detiene; para comprobar que se realicen desplazamientos correctos se realizó pruebas sobre un piso marcado por cuadrículas de 20×20 cm, para corregir las distancias se determinó una relación inversamente proporcional entre el factor de conversión (F) y la distancia total recorrida (d_T), en caso de que el recorrido realizado sea mayor a 20 cm se disminuye el valor del factor de conversión y en caso de que el recorrido realizado sea menor a 20 cm se aumenta el valor del factor de conversión. En la Tabla 19 se describen las ecuaciones corregidas para determinar distancias recorridas de cada uno de los agentes robóticos.

Tabla 19

Corrección de ecuaciones para estimar distancias recorridas

Ecuación para determinar distancia total recorrida	$\frac{F(P_i + P_d)}{2}$	(10)
Ecuación calculada	$0.395(P_i + P_d)$	(10)
Ecuación corregida (Robot 1)	$0.387(P_i + P_d)$	(82)
Ecuación corregida (Robot 2)	$0.412(P_i + P_d)$	(83)
Ecuación corregida (Robot 3)	$0.392(P_i + P_d)$	(84)

5.5.2. Muestreo y estimación de parámetros

Los parámetros principales de sensado en el proyecto de investigación se enfocan en la adquisición de: distancias hacia obstáculos o fronteras, niveles de temperatura y concentración de gases del entorno de navegación. En este caso se desconoce si existe homogeneidad o heterogeneidad de la distribución espacial ya que los robots cumplen con la función de explorar y mapear un entorno controlado pero desconocido, para obtener datos confiables se utiliza el método de muestreo aleatorio sistemático cuyo procedimiento se basa en tomar muestras en espacios determinados aleatoriamente y en los cuales se conoce la distancias que existe entre dichas muestras, se toma un total de 10 muestras por cada sensor, cada muestra o dato es tomado por la tarjeta programable teensy 3.6 cada 10 ms, una vez obtenidas todas la muestras se procede a la estimación de parámetros (temperatura, concentración de gases, distancia hacia obstáculos o fronteras), para ello se calcula mediante la ecuación 85, la media del conjunto de muestras tomadas. En la Figura 72 se indica el diagrama de flujo para la obtención de dichos valores.

$$Media(M) = \bar{m} = \frac{\sum_{i=1}^N M_i}{N} \quad (85)$$

M : Conjunto de muestras sensadas
 N : Número total de muestras

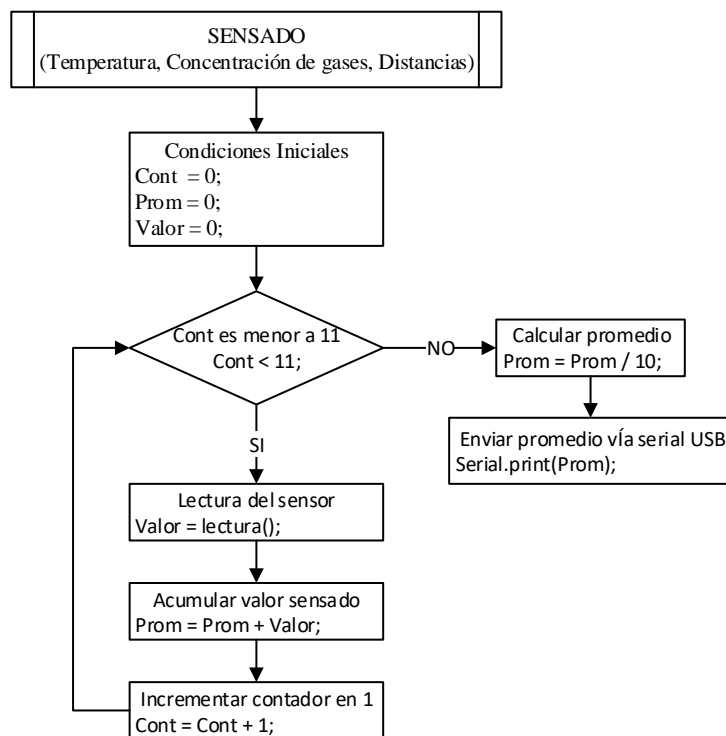


Figura 72. Flujo para sensado general.

5.5.3. Normalización de sensores químicos

Los valores correspondientes a la concentración de gases se obtienen a partir de la ecuación 1 del apartado 3.6.3 correspondiente a sensado químico, dicho valor es recalibrado variando la resistencia carga (R_L) de tal manera que la respuesta del voltaje de salida no sea superior a 3,3 [V]. Dado que los sensores TGS2610, TGS2600 y TGS2602 son altamente sensibles al alcohol, se ha realizado pruebas para determinar cuáles son los valores máximos y mínimos sensados ante la presencia de dicha sustancia; para obtener los posibles valores máximos, de cada robot móvil se colocó la nariz electrónica directamente sobre una muestra con 30 ml de alcohol, después se tomó un total de 50 muestras y se calculó la media del conjunto de datos adquiridos, de la misma forma

se realizó el cálculo de la media para obtener el valor mínimo alejando la muestra a unos 15 cm de la nariz electrónica.

El valor que debe visualizarse en la interfaz gráfica de usuario y mapas debe ser proporcional del 1 – 100%, el valor sentido V_{SQ} corresponde a la respuesta del voltaje de salida de los sensores químicos, dichos valores son procesados en la Raspberry Pi 3 mediante dos casos: cuándo V_{SQ} es menor al valor medio y cuando V_{SQ} es mayor al valor medio, en la Tabla 20 se detallan las fórmulas que determinan el porcentaje de concentración de un determinado gas.

Tabla 20

Cálculo de valores proporcionales de concentración de gases

Robot	Sensor Químico	Media de un conjunto de 50 muestras		Valor medio [mV]:	Valor proporcional	
		Valor mínimo [mV]	Valor máximo [mV]		$V_{SQ} < \text{Valor medio}$	$V_{SQ} > \text{Valor medio}$
					1%	100%
Robot 1	TGS2610	497,54	3194,74	1348,60	$\frac{V_{SQ}}{497,54}$ (86)	$\frac{100 \times V_{SQ}}{497,54}$ (95)
	TGS2600	366,17	1408,40	521,12	$\frac{V_{SQ}}{366,17}$ (87)	$\frac{100 \times V_{SQ}}{366,17}$ (96)
	TGS2602	307,69	2814,95	1253,63	$\frac{V_{SQ}}{307,69}$ (88)	$\frac{100 \times V_{SQ}}{307,69}$ (97)
Robot 2	TGS2610	403,93	3024,38	1310,23	$\frac{V_{SQ}}{403,93}$ (89)	$\frac{100 \times V_{SQ}}{403,93}$ (98)
	TGS2600	609,28	2175,60	783,16	$\frac{V_{SQ}}{609,28}$ (90)	$\frac{100 \times V_{SQ}}{609,28}$ (99)
	TGS2602	303,20	2935,59	1316,20	$\frac{V_{SQ}}{303,20}$ (91)	$\frac{100 \times V_{SQ}}{303,20}$ (100)
Robot 3	TGS2610	359,10	2897,13	1269,02	$\frac{V_{SQ}}{359,10}$ (92)	$\frac{100 \times V_{SQ}}{359,10}$ (101)
	TGS2600	60,14	2246,71	1093,29	$\frac{V_{SQ}}{60,14}$ (93)	$\frac{100 \times V_{SQ}}{60,14}$ (102)
	TGS2602	216,26	2518,40	1151,07	$\frac{V_{SQ}}{216,26}$ (94)	$\frac{100 \times V_{SQ}}{216,26}$ (103)

5.5.1. Comunicación serial USB

Se realizó dos programas de prueba, uno en la tarjeta Teensy 3.6 y otro en la Raspberry Pi 3 (ver Figura 73), en ambos programas se establece comunicación serial USB de forma bidireccional, el programa de la Raspberry Pi 3 consiste en el envío de dos comandos mediante los caracteres ‘A’ y ‘B’, cuando la Teensy 3.6 recibe la letra ‘A’ se realiza el encendido del led por defecto de la Teensy en el pin 13, después se imprime vía serial la frase “led encendido”, en el otro caso al recibir la letra ‘B’ se realiza el apagado del led y se imprime vía serial la frase “led apagado”, al mismo tiempo en la Raspberry Pi 3 se reciben todos los caracteres que ha sido enviados, en este caso las frases de led encendido o led apagado que son visualizadas en el monitor serial del IDLE de Python.

Esta prueba sirvió para posteriormente integrar en el sistema la activación y desactivación de motores dependiendo el tipo orden recibida en el tramo de retorno, de esta manera la Teensy 3.6 se encargaría de las acciones de navegación en línea recta o giro del ángulo recibido, así mismo el envío de las frases sirvieron para representar las tramas de datos sensados enviados desde la Teensy hacia la Raspberry.

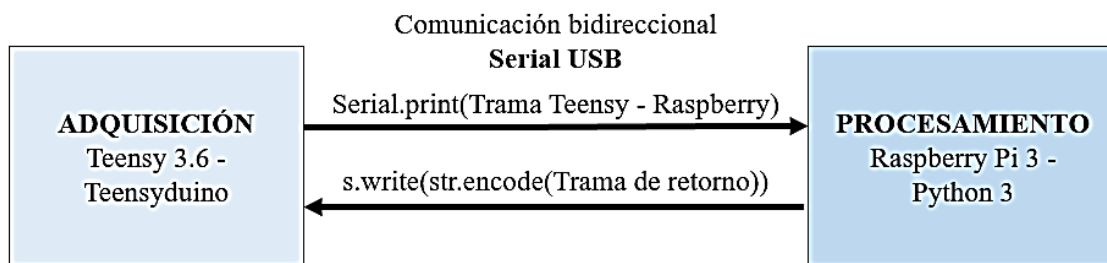


Figura 73. Prueba de comunicación bidireccional serial USB.

La trama recibida en la Raspberry se recibe completa mediante un solo comando, sin embargo, esto no sucede en la trama de retorno recibida en la Teensy, ya que se recibe carácter por carácter, para ello se ha realizado la función *concatenación*, la cual permite recibir de uno en uno cada carácter de la trama de retorno, cada dato es recibido en intervalos de 15 ms y se termina la concatenación cuando se detecte un salto de línea, es decir el fin de la trama.

5.5.2. Comunicación inalámbrica

Las pruebas iniciales del sistema robótico cooperativo provocaban tiempos de latencia mayores a 10 segundos, dada esta situación fue necesario buscar una solución para optimizar la comunicación respecto a los tiempos de envío y recepción de datos, pero realizar las pruebas de comunicación, la mismas resultaban complicadas debido al descarga de las baterías, colocación de los robots en sus posiciones iniciales, puesta en marcha de los programas, etc.

De esta manera fue necesario buscar una alternativa al envío y recepción de datos mediante simulaciones; para simular el procesamiento de datos entre los agentes robóticos y el ordenador central se han implementado cuatro programas: uno en la Raspberry Pi 3 (total 3, uno por cada agente robótico) y otro en Matlab que funciona como ordenador central, en todos los programas se establece comunicación inalámbrica de forma bidireccional. En cada Raspberry se predefinió una trama de similares características a las enviadas por los robots móviles, en la cual se genera un dato aleatorio entre 0 y 1 correspondiente a la bandera de obstáculo; en Matlab, se realiza la recepción de la trama, y tal como se ha explicado en el apartado 4.7.3 correspondiente a la trama de retorno, se decide en que caso se encuentra y se retorna a la Raspberry los valores de las coordenadas hacia zonas no exploradas o el comando para navegación rectilínea, todo dentro de un bucle infinito.

De esta forma ya no fue necesario la puesta en marcha de la navegación y activación de los agentes robóticos, además, mediante estas pruebas se logró optimizar el tiempo de latencia de datos entre cada escaneo, para ello se realizó la identificación del número máximo de caracteres enviados en la trama raspberry – ordenador central con un total de 140 caracteres correspondientes a 140 *bytes* y el máximo de caracteres enviado en la trama de retorno con un total de 16 caracteres correspondientes a 16 *bytes*, de esta manera se programó el tamaño de las tramas considerando todos los caracteres posibles, en un principio se había programado la recepción tanto en la raspberry como el Matlab con el tamaño por defecto correspondiente a 1024 *bytes*, es decir 1024 caracteres, esto ocasionaba un gasto computacional innecesario, provocando tiempos de latencia de 6 a 10 segundos. Todas las pruebas mencionadas fueron realizadas con un router TP-Link TLWR840N cuya velocidad de transmisión podía llegar a 300 *Mbps* en la banda de 2.4 *GHz*; en busca de optimizar de mejor manera al sistema de comunicación inalámbrico se optó por cambiar de router, para ello se adaptó un router Cisco RV130, cuya velocidad de transmisión podía llegar a 800 *Mbps*, generando por lo tanto, una mejora en el rendimiento del sistema, con tiempos de latencia menores a 5 segundos.

5.6. Pruebas y resultados del sistema general

Una vez realizadas las pruebas independientes de cada requerimiento del sistema robótico cooperativo, se desarrollaron los programas de la Teensy 3.6, Raspberry Pi 3 y Matlab siguiendo la lógica de funcionamiento detallada en el diagrama de flujo de la Figura 65 correspondiente a la integración de sistemas de control y comunicación.

También se propuso 3 escenarios con diferentes configuraciones respecto a: posiciones iniciales de los agentes robóticos móviles, disposición de obstáculos, ubicación de muestras con 3 diferentes sustancias químicas y diferentes ubicaciones del ventilador y calentador.

5.6.1. Escenarios

Para realizar las pruebas de sensado se utilizaron 3 sustancias químicas: alcohol, tolueno y acetona, de cada sustancia química se dispuso a realizar una prueba por cada sustancia en cada escenario propuesto, dando un total de 9 pruebas de navegación autónoma y mapeo de entorno. Para los obstáculos se utilizaron cajas cuyas dimensiones son de $38 \times 40 \times 30$ cm.

Tabla 21

Configuración del escenario No. 1

	Posición en X	Posición en Y
Posición inicial del Robot 1	30	30
Posición inicial del Robot 2	140	60
Posición inicial del Robot 3	70	210
Obstáculo 1	220	20
Obstáculo 2	20	220
Obstáculo 3	150	140
Muestra 1	50	40
Muestra 2	80	100
Muestra 3	140	200
Muestra 4	110	20
Calentador	20	220
Ventilador	220	20

Tabla 22

Configuración del escenario No. 2

	Posición en X	Posición en Y
Posición inicial del Robot 1	30	30
Posición inicial del Robot 2	140	30
Posición inicial del Robot 3	30	210

CONTINÚA 

Obstáculo 1	20	120
Obstáculo 2	150	220
Obstáculo 3	-	-
Muestra 1	70	30
Muestra 2	70	140
Muestra 3	130	140
Muestra 4	-	-
Calentador	150	220
Ventilador	20	-20

Tabla 23

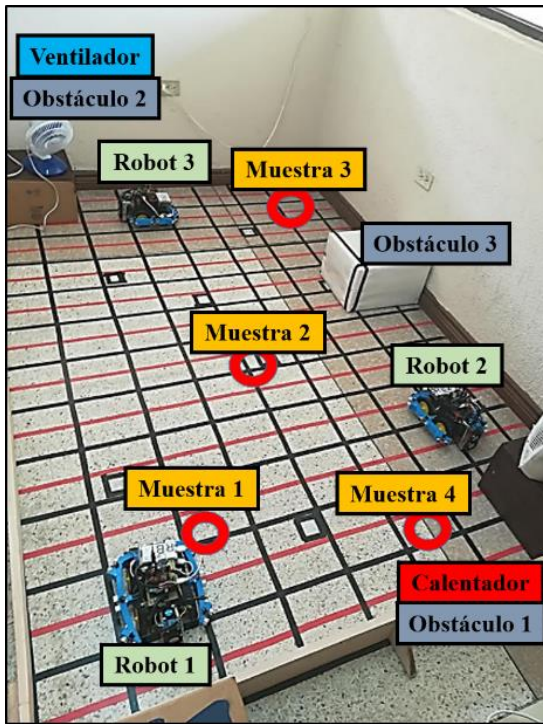
Configuración del escenario No. 3

	Posición en X	Posición en Y
Posición inicial del Robot 1	30	30
Posición inicial del Robot 2	140	30
Posición inicial del Robot 3	70	210
Obstáculo 1	20	220
Obstáculo 2	90	20
Obstáculo 3	150	220
Muestra 1	20	60
Muestra 2	20	180
Muestra 3	140	180
Muestra 4	140	60
Calentador	20	220
Ventilador	150	220

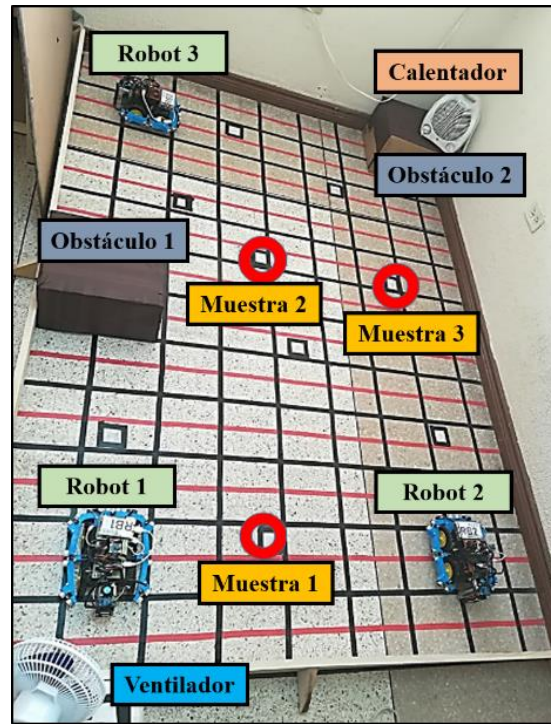
El escenario No. 1 se dispuso de tal forma que el Robot 2 sea el primero en encontrar un obstáculo y el primero en identificar la zona menos explorada más lejana; por la ubicación inicial del Robot 1 cercana a la recta $X = 0$, se espera obtener la mejor visualización del mapa generado en la frontera izquierda del entorno, de igual forma, por la ubicación inicial del Robot 3 cercana a la recta $Y = 240$, se espera obtener la mejor visualización del mapa generado en la frontera superior del entorno, también se espera obtener la mejor visualización de la muestra 3, puesto que se encuentra en dirección de los desplazamientos iniciales del robot 3.

El escenario No. 2 se dispuso de tal forma que el Robot 1 sea el primero en encontrar un obstáculo y el primero en identificar la zona menos explorada más lejana; por la ubicación inicial del Robot 2 cercana a la recta $X = 170$, se espera obtener la mejor visualización del mapa generado en la frontera derecha del entorno, de igual forma, por la ubicación inicial del Robot 3 cercana a la recta $Y = 240$, se espera obtener la mejor visualización del mapa generado en la frontera superior del entorno, también se espera obtener la mejor visualización de la muestra 3, puesto que se encuentra en dirección de los desplazamientos iniciales del robot 2.

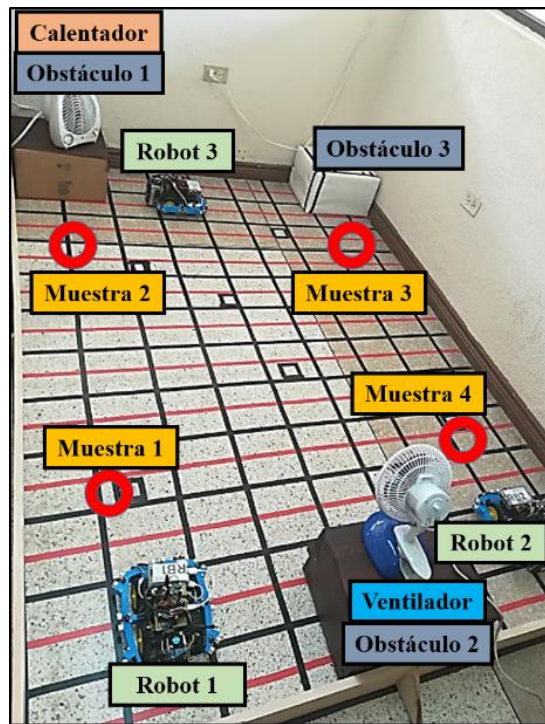
El escenario No. 3 se dispuso de tal forma que el Robot 3 sea el primero en encontrar un obstáculo y el primero en identificar la zona menos explorada más lejana; por la ubicación inicial del Robot 1 cercana a la recta $X=0$, se espera obtener la mejor visualización del mapa generado en la frontera izquierda del entorno, de igual forma, por la ubicación inicial del Robot 2 cercana a la recta $Y=170$, se espera obtener la mejor visualización del mapa generado en la frontera derecha del entorno, también se espera obtener la mejor visualización de las muestras 1 y 2 por el robot 1 y las muestras 3 y 4 por el robot 2, puesto que se encuentran en dirección de sus desplazamientos iniciales.



a)



b)



c)

Figura 74. a) Escenario No. 1, b) Escenario No. 2, c) Escenario No. 3.

5.7. Resultados del sistema robótico cooperativo

5.7.1. Pruebas del escenario No. 1

En la Figura 75 se muestra los resultados de mapeo de obstáculos de una prueba realizada en el escenario No.1 el cual se encuentra conformado mediante 3 obstáculos ubicados en las siguientes posiciones: $P_{O1}(220,20)$, $P_{O2}(220,20)$ y $P_{O3}(150,140)$, donde para poder comparar los resultados del mapeo con el entorno real, se ha programado una gráfica en color rojo indicando la disposición de los obstáculos y fronteras, la gráfica es sobrepuesta sobre el mapa procesado mediante grafos de visibilidad, de esta manera se permite comparar visualmente las zonas detectadas como obstáculos por los agentes robóticos con el mapa simulado de iguales dimensiones al entorno real.

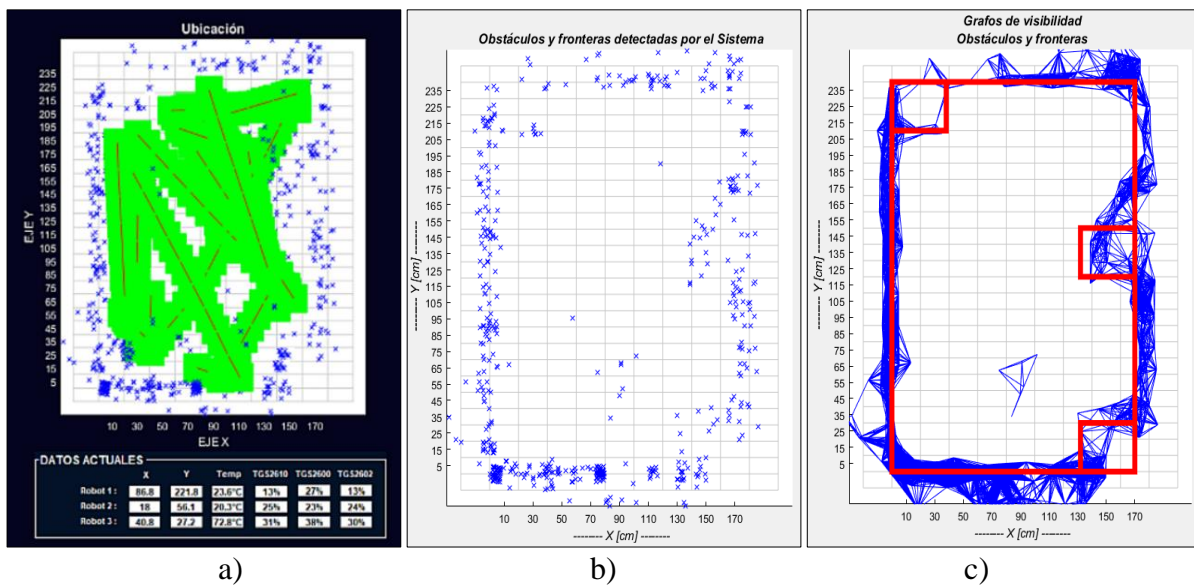


Figura 75. Escenario No. 1: a) Navegación y mapeo de obstáculos en tiempo real, b) Mapeo de obstáculos mediante gráfica de pixeles, c) Mapeo de obstáculos mediante grafos de visibilidad.

En el entorno se ha colocado un ventilador en la posición $P_V(220,20)$ y un calentador en la posición $P_C(20,220)$ para variar los valores correspondientes a los niveles de temperatura en el

ambiente. Mediante la interpolación en 2D y el filtro espacial de media móvil se permitió visualizar la concentración de gases y niveles temperatura con curvas suavizadas que determinen el comportamiento de dispersión de temperatura y gases en el aire. En la Figura 76 se visualiza el mapa de niveles de temperatura en el cual se aprecia que la pluma de dispersión denotada con colores cálidos correspondientes a temperaturas altas, sigue la misma dirección de la salida del calentador. En la prueba del escenario 1 se ha dotado a las muestras con 40 ml de alcohol cada una, en la Figura 77, Figura 78 y Figura 79 se muestra la representación de concentración de alcohol en el ambiente mediante los sensores TGS2610, TGS2600 y TGS2602 respectivamente donde es posible notar que las zonas en rojo correspondientes a niveles altos de concentración de alcohol, coinciden con la posición de la muestras detalladas en la Tabla 21 correspondiente a la configuración del escenario 1.

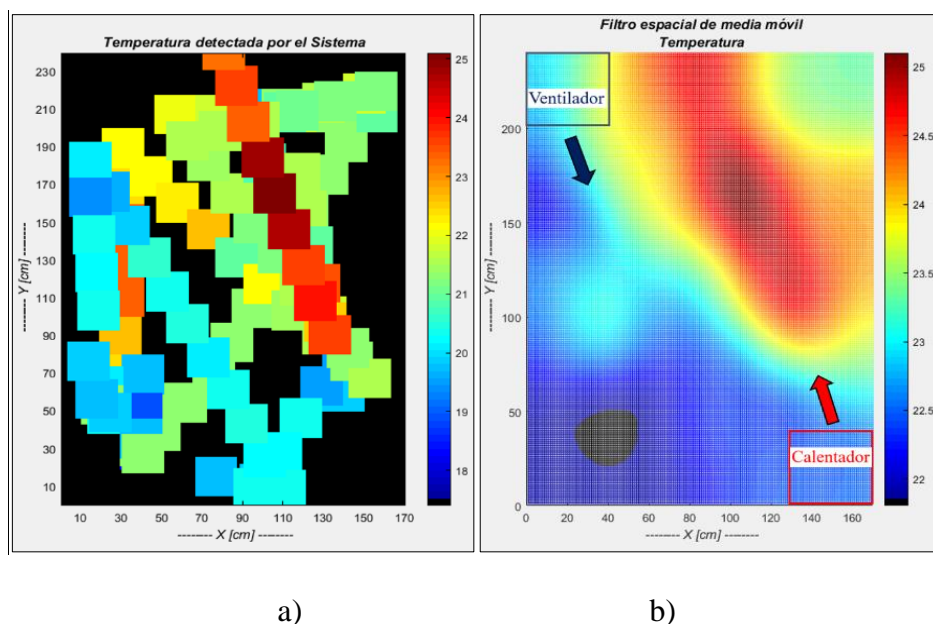


Figura 76. Escenario No. 1: a) Mapeo de temperatura, b) Mapeo 2D de temperatura mediante interpolación 2D y filtro espacial de media móvil.

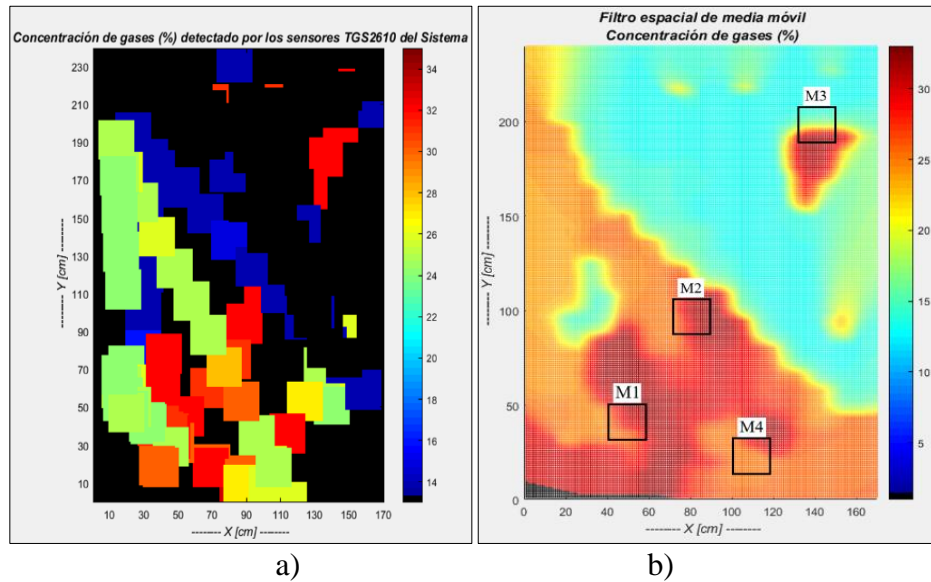


Figura 77. Escenario No. 1, sensor TGS2610: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.

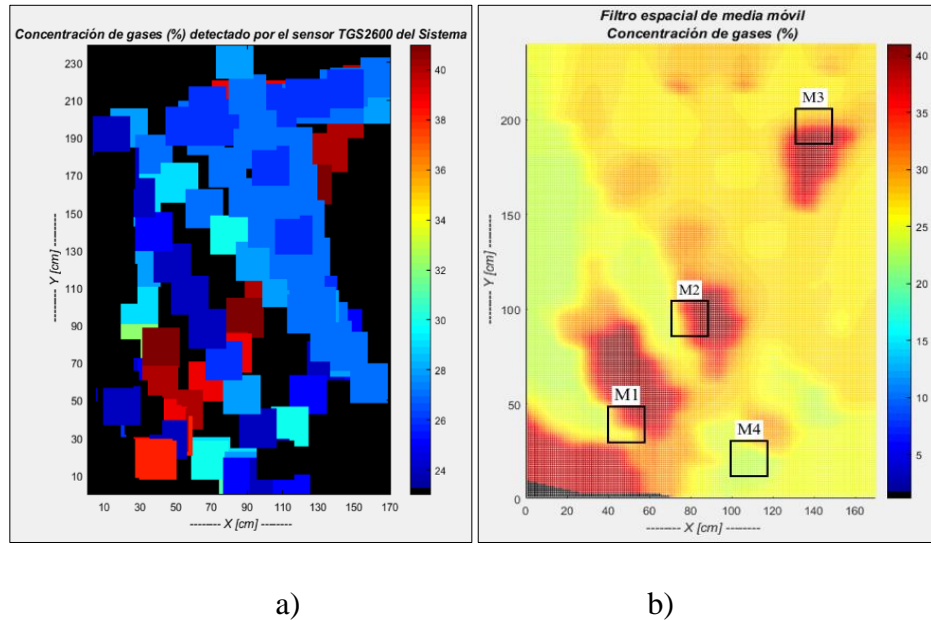


Figura 78. Escenario No. 1, sensor TGS2600: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.

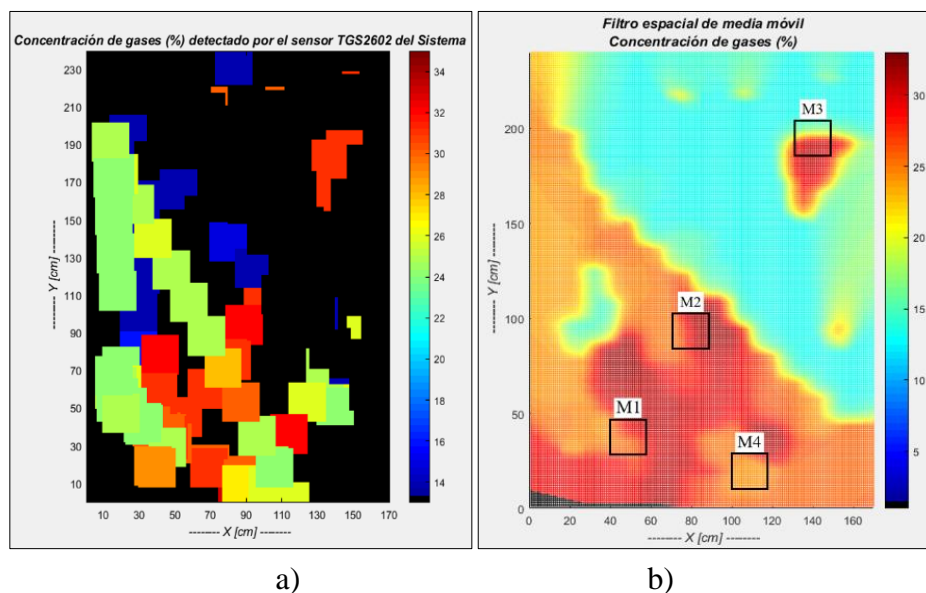


Figura 79. Escenario No. 1, sensor TGS2602: a) Mapeo de concentración de alcohol, b) Mapeo 2D de concentración de alcohol mediante interpolación 2D y filtro espacial de media móvil.

5.7.2. Pruebas del escenario No. 2

A continuación se presentan los resultados de las tres pruebas realizadas en el escenario No.2, el cual se encuentra conformado mediante dos obstáculos ubicados en las siguientes posiciones: $P_{O1}(20, 120)$ y $P_{O2}(150, 220)$, cada prueba ha sido realizada con tres muestras de 40 ml de alcohol, tolueno y acetona. En este escenario se ha colocado un ventilador en la posición $P_V(20, -20)$ y un calentador en la posición $P_C(150, 220)$, estos artefactos se encuentran los más alejados posible de tal manera que los valores sensados correspondientes a los niveles de temperatura se encuentren dispersos para que sea posible notar las zonas del entorno controlado con mayor y menor temperatura en el ambiente.

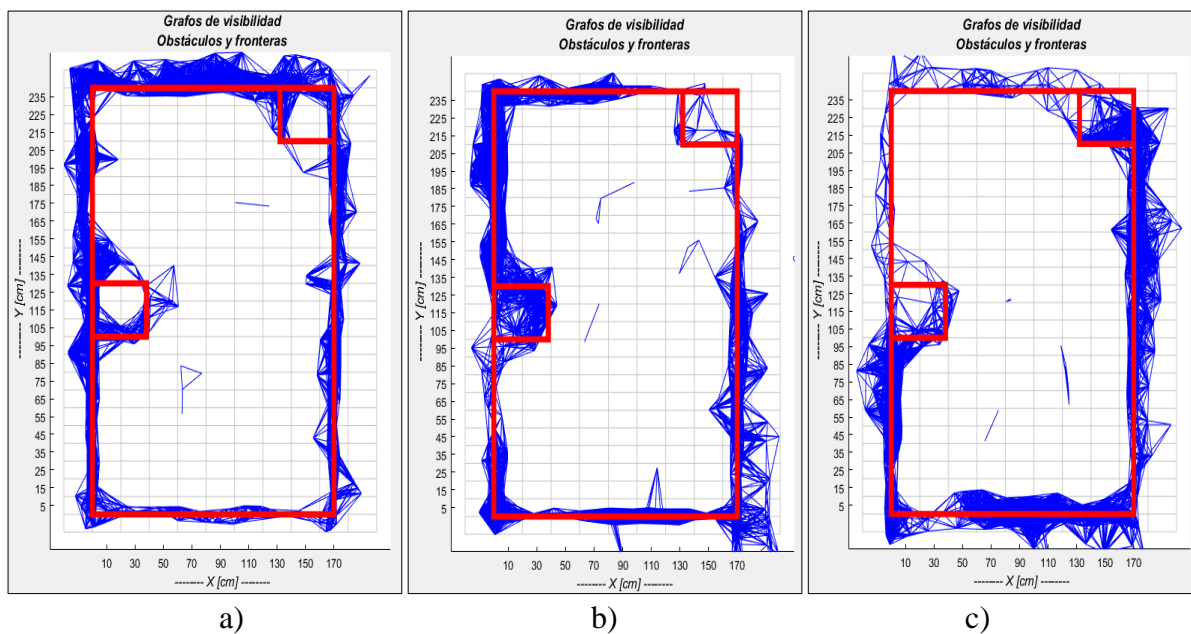


Figura 80. Mapeo de obstáculos en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona

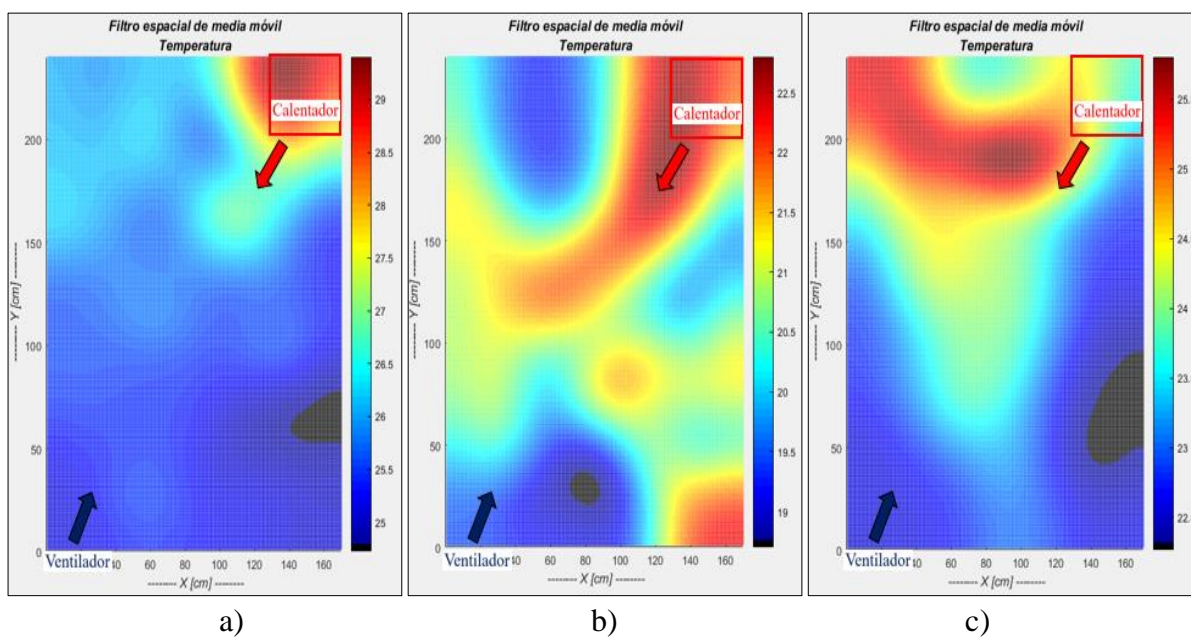


Figura 81. Mapeo de temperatura en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona

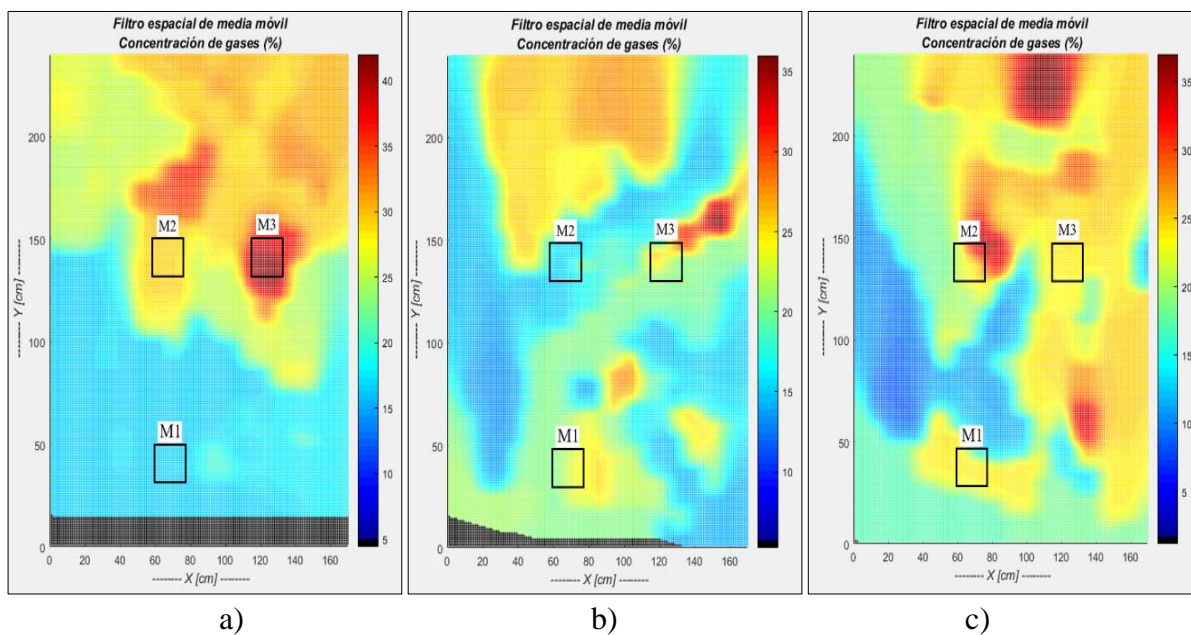


Figura 82. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona

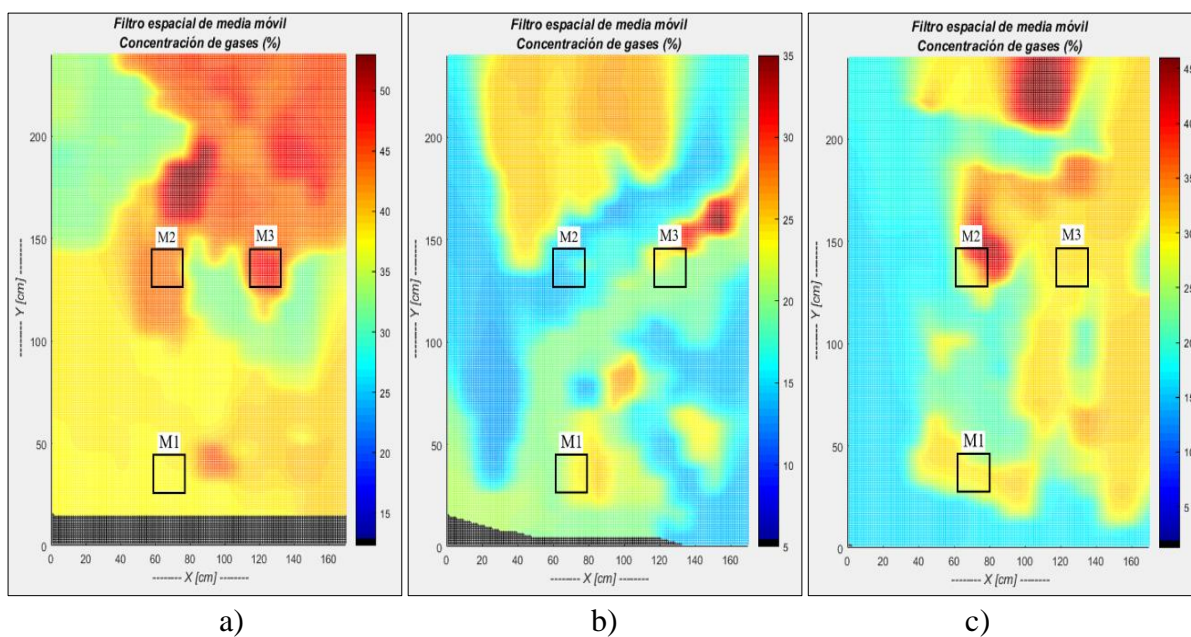


Figura 83. Mapeo de concentración de gases mediante sensor químico TGS2600 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona

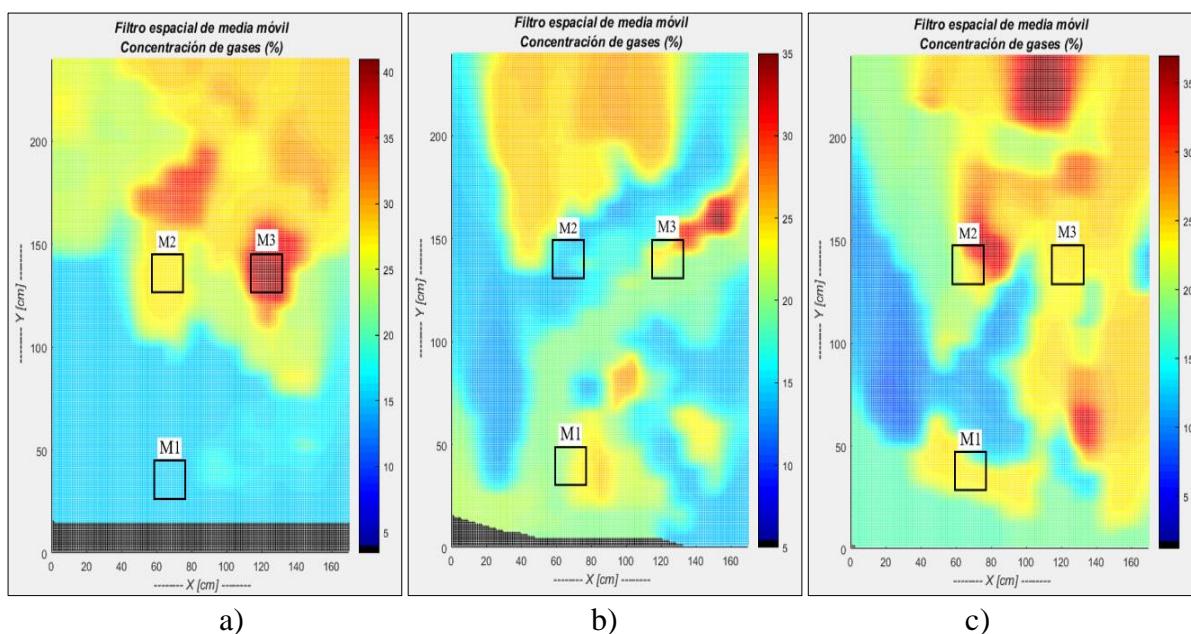


Figura 84. Mapeo de concentración de gases mediante sensor químico TGS2602 en Escenario No. 2: a) Alcohol b) Tolueno, C) Acetona

En la Figura 80 se muestran las 3 pruebas de mapeo de obstáculos realizadas en el escenario No. 2, en las cuales cada prueba se realizó con muestras dotadas de alcohol, tolueno y acetona, en las gráficas se puede apreciar que en las esquinas se produce gráficos distorsionados respecto a la referencia del entorno real, esto ocasiona grafos desfavorables de zonas en las cuales no existen obstáculos; se ha determinado que esta situación se produce debido a una inapropiada resolución en los sensores ultrasónicos, sin embargo, este problema no sucede en las fronteras dado que la estimación de distancia se mide directamente frente a un objeto considerado plano y uniforme.

En la Figura 81 se puede apreciar uniformidad en la dispersión de la temperatura desde los valores más bajos hasta los valores más altos, además se aprecia claramente que la posición del calentador $P_C(150, 220)$ coincide con la zona de mayor temperatura sensada en el ambiente, al igual que la zona de menor temperatura representada mediante colores fríos, sigue la misma dirección de la salida del ventilador ubicado en la posición $P_V(20, -20)$.

En la Figura 82 apartado a) y Figura 84 apartado a), se puede apreciar que no existe dispersión en la zona correspondiente a la muestra del alcohol ubicada en el punto $P_{E1M1}(70,30)$, en esta situación pudieron suceder 2 casos: el primero corresponde a una pérdida en la comunicación inalámbrica en el instante en el cuál un agente robótico X navegó por dicha zona, y por lo tanto el valor correspondiente a la concentración de alcohol en dicho punto se haya perdido, en un segundo caso, pudo suceder una incorrecta lectura del sensor químico TGS2610, en el cual pudo existir la adquisición de un valor negativo, esta condición está programada de tal forma que cuando se detecten valores negativos, los mismos sean reemplazados por el valor medio entre el máximo y el mínimo sensado durante la navegación.

En la Figura 83 se puede apreciar la interacción directa entre la temperatura y la dispersión de la concentración de gases; en la zonas consideradas con temperaturas altas, existe mayor detección de un gas en específico, de hecho en las especificaciones técnicas de los sensores químicos de la familia Figaro TGS, se especifica que los mismos deben ser precalentados para funcionar de manera correcta, por esta razón los robots antes de efectuar todas las pruebas de navegación, deben permanecer encendidos durante un periodo de 3 a 5 minutos.

5.7.3. Pruebas del escenario No. 3

En las muestras correspondientes a cada prueba, se ha dotado de 40 ml de alcohol, tolueno y acetona obteniendo un total de 3 pruebas, una por cada sustancia química. Este escenario está conformado por 3 obstáculos ubicados en las posiciones $P_{O1}(20,220)$, $P_{O2}(90,20)$ y $P_{O3}(150,220)$; al igual que en los dos escenarios anteriores, en el entorno se ha colocado un ventilador y un calentador cuyas posiciones son $P_V(220,20)$ y $P_C(20,220)$ respectivamente.

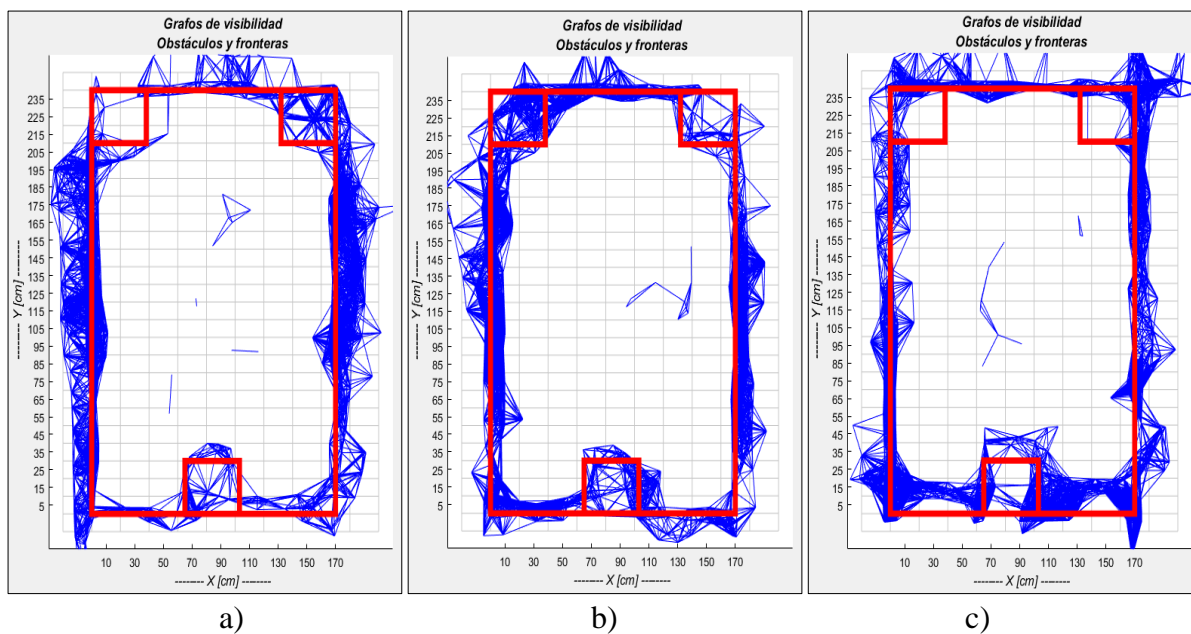


Figura 85. Mapeo de obstáculos en Escenario 3: a) Alcohol b) Tolueno, C) Acetona

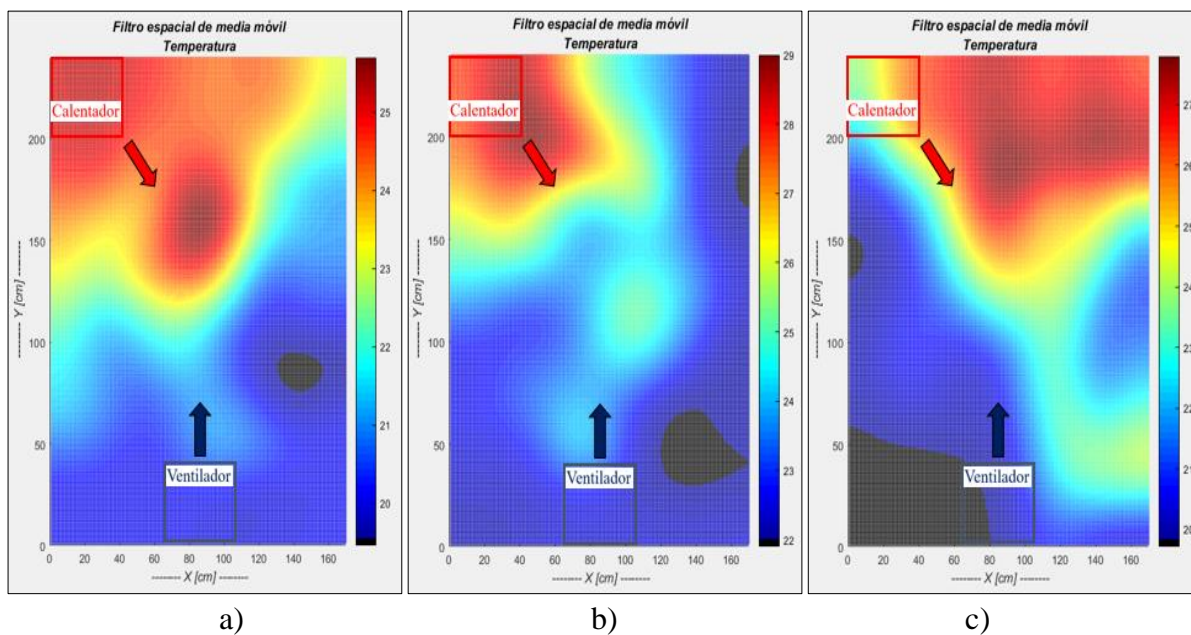


Figura 86. Mapeo de temperatura en Escenario 3: a) Alcohol b) Tolueno, C) Acetona

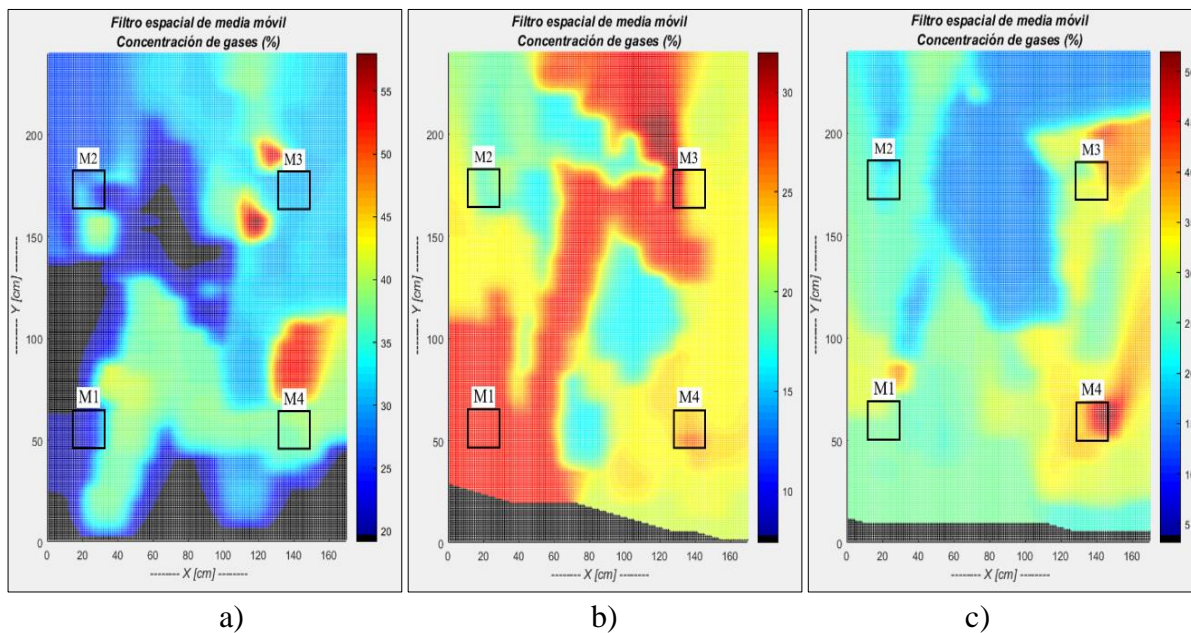


Figura 87. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario 3:
 a) Alcohol b) Tolueno, C) Acetona

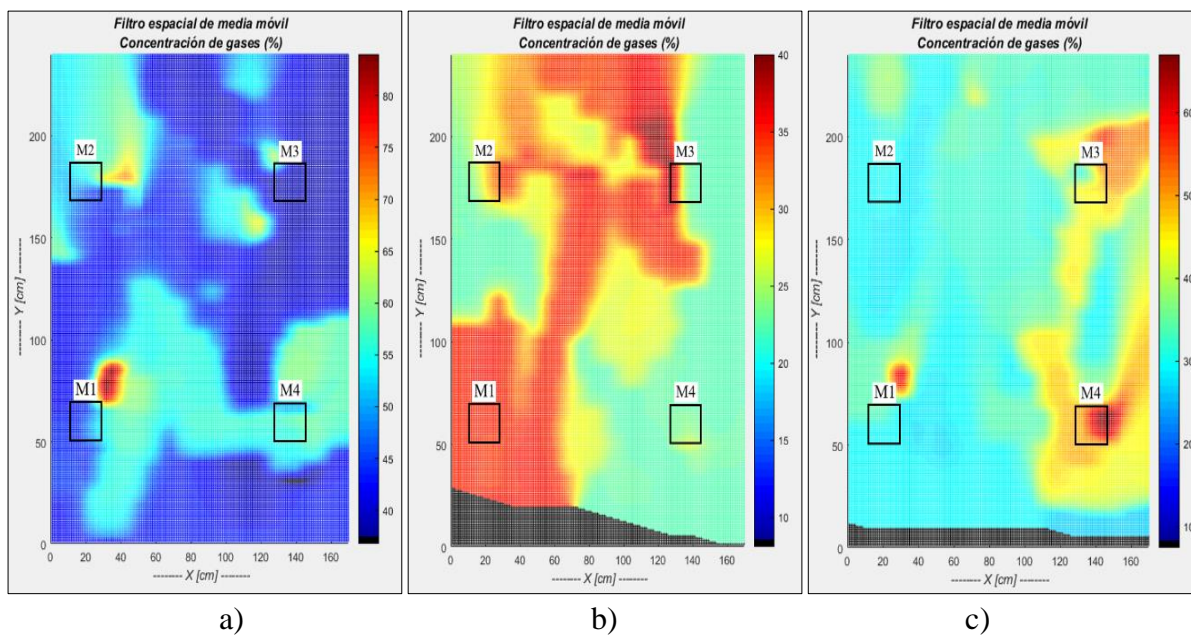


Figura 88. Mapeo de concentración de gases mediante sensor químico TGS2600 en Escenario 3:
 a) Alcohol b) Tolueno, C) Acetona

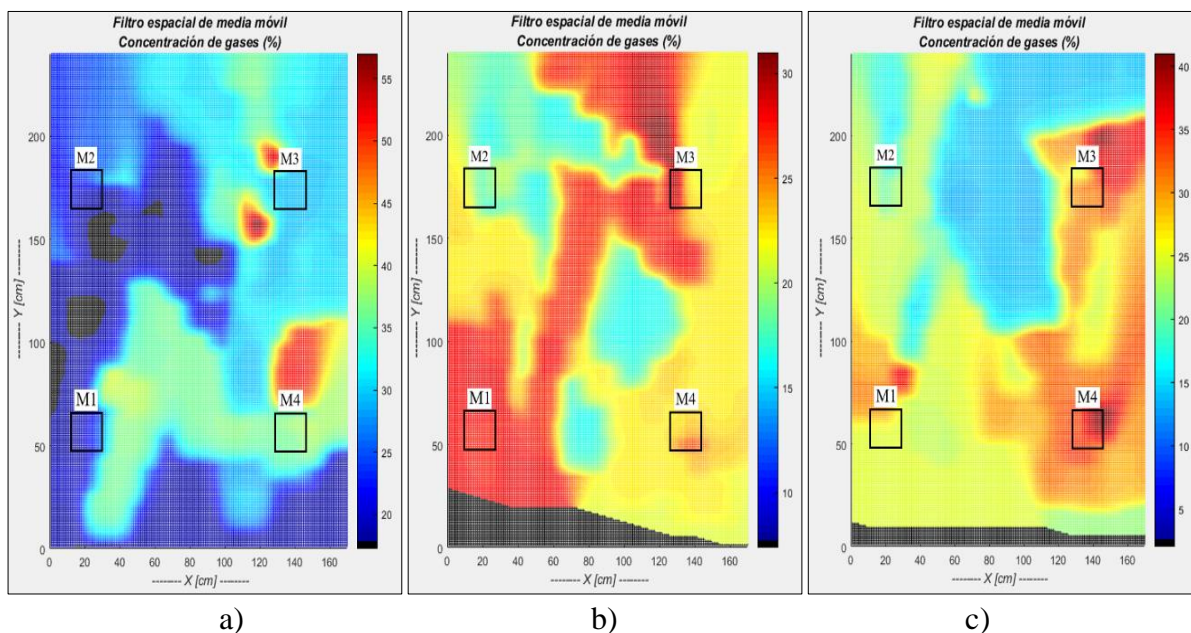


Figura 89. Mapeo de concentración de gases mediante sensor químico TGS2610 en Escenario 3:
a) Alcohol b) Tolueno, C) Acetona

En la Figura 85 apartado c), no se puede apreciar los grafos correspondientes a los obstáculos 1 y 3 ubicados en las posiciones $P_{O1}(20, 220)$ y $P_{O2}(150, 220)$ respectivamente, lo cual sucedió debido a una falsa exploración de un agente robótico X por dichas zonas, esto sucede debido a que el controlador de interfaz gráfica centralizado está programado de tal forma que en las zonas exploradas no se grafiquen los pixeles correspondientes a obstáculos o fronteras detectados por el sistema robótico cooperativo y por ende tampoco se pueda observar los grafos de visibilidad correspondientes a dichos obstáculos. También se puede observar grafos en medio del área de exploración que no corresponden a ningún obstáculo, esto sucede cuándo los robots se detectan entre ellos como obstáculos dinámicos, para solucionar este problema se programó un condición de tal manera que si un obstáculo es detectado en una zona en la cual se encuentra un robot, dicho punto no sea graficado y guardado, sin embargo existen casos en los cuales por apreciación y resolución de los sensores ultrasónicos, el punto correspondiente a un obstáculo se detecte fuera de

la zona correspondiente a la posición actual de un robot móvil, también existe el caso cuando dos sensores ultrasónicos se encuentran frente a frente lo cual causa lecturas incorrectas referentes a las distancias estimadas.

5.7.4. Tiempos de navegación

En las pruebas realizadas se determinó como condición adecuada para detener la navegación de los robots móviles autónomos, un recorrido total de 20 metros cuyo valor se determina mediante la sumatoria de las distancias recorridas por cada uno de los agentes robóticos. Mediante las 9 pruebas realizadas se ha determinado que dicho recorrido se obtiene en un tiempo menor a 14 minutos. En la Tabla 24 se detallan los tiempos en los cuales se ha completado la navegación y el mapeo del entorno controlado.

Tabla 24

Tiempos de navegación

Escenario	Tipo de muestra	Tiempo de navegación
Escenario No. 1	Alcohol	11 min 50 seg.
Escenario No. 2	Alcohol	10 min. 10 seg.
Escenario No. 3	Alcohol	12 min. 10 seg
Escenario No. 1	Tolueno	10 min. 30 seg
Escenario No. 2	Tolueno	13 min. 30 seg.
Escenario No. 3	Tolueno	12 min. 50 seg
Escenario No. 1	Acetona	9 min. 40 seg.
Escenario No. 2	Acetona	11 min. 15 seg.
Escenario No. 3	Acetona	13 min. 20 seg.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Diseño e implementación del hardware

Se ha implementado tres robots móviles capaces de explorar de forma autónoma un entorno controlado de 2.40 m de largo por 1.70 m de ancho; durante la exploración los agentes robóticos realizan el mapeo de obstáculos, temperatura y concentración de gases, para ello fue necesario conocer la odometría del sistema, y cuyos resultados se lograron a partir del cálculo de los ángulos de giro y distancias recorridas.

Para calcular las distancias recorridas, se optó por instalar en cada plataforma robótica 2 encoders en las ruedas posteriores, de esta manera, los pulsos incrementales generados mediante el giro de las ruedas son captados por optointerruptores, y dicha información es traducida en la distancia recorrida de un robot móvil en cualquier instante de tiempo.

En el proyecto de investigación de mapeo y exploración autónoma mediante robótica móvil cooperativa desarrollado por (Moya, 2017), se utilizaron plataformas de configuración diferencial conformadas por dos ruedas y una tercera rueda utilizada como punto de apoyo, la cual ocasionaba problemas para desplazamientos de distancias fijas en línea recta, la plataforma robótica que se escogió para la construcción de los robots móviles correspondientes al presente proyecto de investigación, fue la DAGU DG012-ATV; se decidió continuar usando plataformas de configuración diferencial, debido a su arquitectura ideal para generar giros alrededor de su propio eje y brindar estabilidad en movimientos rectilíneos, además, esta plataforma presenta la ventaja de contar con 4 ruedas, de esta manera al integrar el sistema con una arquitectura más óptima ya

no fue necesario realizar una corrección de error acumulado en desplazamientos rectilíneos de distancias fijas.

Para calcular los ángulos de giro se utilizó un giroscopio integrado en una unidad de medición inercial IMU BNO055, que a diferencia del magnetómetro o brújula electrónica utilizada previamente por (Moya, 2017), la cuál brindaba los ángulos de orientación respecto al norte magnético, la IMU BNO055 presenta la ventaja de contar con su propio microcontrolador encargado de obtener los ángulos de orientación absoluta, tanto del giroscopio como del magnetómetro, los cuales están integrados en el mismo chip, en este caso el giroscopio permite obtener el ángulo de giro (ψ) alrededor del eje Z (Figura 51).

En la investigación previa desarrollada por (Moya, 2017), de las pruebas realizadas, se ha determinado que componentes electrónicos como motores que se encuentran cercanos a una brújula o magnetómetro, alteran los datos sensados por el mismo, de esta manera, en el presente proyecto, la IMU BNO055 fue instalada en el nivel más alto de cada plataforma robótica, con el objetivo de evitar interferencias generadas por la inducción que provocan los motores DC, de esta manera se logró mejorar la lectura de ángulos de giro en comparación a una arquitectura integrada mediante magnetómetros.

Para desarrollar proyectos de prototipos robóticos es muy común el uso de tarjetas de programación de la familia Arduino, sin embargo, en el presente proyecto, para la adquisición de datos se decidió utilizar una tarjeta programable Teensy 3.6, de la cual, una de las mayores ventajas que presenta sobre un Arduino Uno o un Arduino Mega comúnmente usados, es su velocidad de procesamiento; mientras un Arduino Mega corre a una velocidad de 16 MHz que se traduce a 16 millones de operaciones por segundo, la Teensy 3.6 siendo de menor tamaño puede correr aproximadamente a una velocidad 11 veces mayor que llega hasta los 180 MHz.

Además de la Teensy 3.6, se agregó a cada agente robótico una Raspberry Pi 3, considerada como una mini computadora completa, esto permite la separación de la etapa de adquisición y activación de actuadores con la etapa de procesamiento; en la Raspberry Pi 3, aprovechando su velocidad de 700 MHz, la cual es aproximadamente 40 veces mayor a la velocidad de un Arduino, se lleva a cabo todos los cálculos necesarios para obtener la información correspondiente a la posición, orientación y navegación de un robot móvil autónomo, de esta forma, al integrar una Teensy 3.6 y una Raspberry Pi 3 en cada agente robótico se logró una optimización de procesamiento de datos y operaciones de cada sistema considerado de manera independiente.

Para el escaneo de obstáculos, en el proyecto de investigación implementado por (Moya, 2017), se diseñó una torreta acoplada a un servomotor e integrada por dos sensores ultrasónicos, su funcionamiento se basaba en el movimiento de la torreta mediante de seis giros de 30° hasta obtener un escaneo total de 360°, obteniendo un total de seis escaneos; para optimizar el mismo objetivo, en el presente proyecto se diseñó mediante modelamiento CAD una estructura semicircular en la cual se permita la instalación de siete sensores ultrasónicos ubicados cada 45°. A pesar de utilizar más recursos en hardware, de esta manera se ha logrado optimizar el descubrimiento de obstáculos del entorno de navegación a un solo escaneo y por ende se ha reducido el tiempo de adquisición de datos.

Para el sensado químico, cada plataforma robótica está conforma por una nariz electrónica implementada a partir de una caja de aluminio con dimensiones de $10,5 \times 5 \times 6$ cm. Esta caja fue instalada con tres ventiladores que cumplen la función de aspirar el aire a través de un orificio ubicado en la parte inferior de nariz hacia el interior de la caja y de esta manera facilitar el flujo de gases a través de un matriz conformada por 5 sensores químicos.

Mediante pruebas en las cuales el orificio de la caja se colocó a una distancia de 1 *cm* sobre una muestra de alcohol de 30 *ml* y de la cual se tomaron 50 datos (ver Tabla 20), se determinó que los sensores TGS2610, TGS2600 y TGS2602 son los más eficientes respecto sensibilidad y respuesta ante la presencia de las sustancias químicas específicas tales como: alcohol, tolueno y acetona, por tanto, en las pruebas de mapeo se optó por utilizar únicamente los sensores mencionados como más eficientes.

6.2. Desarrollo de software de control, navegación y sensado

Se logró la integración de todos los sistemas de control y comunicación correspondiente al software del sistema robótico cooperativo mediante una interfaz gráfica de control centralizado desarrollada en el entorno gráfico del software Matlab, en la interfaz gráfica se muestra la información principal de la navegación en tiempo real mediante cuadros de texto que presentan las posiciones de los robots móviles durante cada escaneo, los niveles de temperatura y los porcentajes de concentración de gases, además se logró la visualización de los mapas en los cuales se permite al usuario apreciar la ubicación de obstáculos o fronteras e identificar las zonas con mayores y menores niveles de temperatura y concentración de gases mediante un código de colores RGB en el cual los valores más altos se representan mediante colores cálidos y los valores más bajos son representados mediante colores fríos.

Inicialmente, el proyecto fue planeado de tal forma que sea posible la visualización en tiempo real de mapas de obstáculos, temperatura y concentración de gases, sin embargo, surgieron problemas de sobrecarga en el procesamiento gráfico que provocaban el colapso total de Matlab y la computadora en sí; por tanto, se propuso realizar un sistema online para visualización de mapas

de navegación y obstáculos, mientras que los mapas de temperatura y concentración de gases puedan ser visualizados una vez termine la navegación (sistema offline).

El control centralizado y sistema de interfaz gráfica fueron implementados en una computadora portátil de sistema operativo Windows 10, 64 bits, con 8 GB de RAM y un controlador gráfico NVIDIA GT 630M de 1 GB de V-RAM, para determinar el problema de sobrecarga en el procesamiento gráfico, se realizó pruebas del sistema en una computadora portátil con controlador gráfico NVIDIA GTX 1060 de 3GB de V-RAM, en la cual el sistema funcionó adecuadamente sin inconvenientes de sobrecarga, de esta manera se concluye que, para establecer un funcionamiento más óptimo del sistema gráfico centralizado en tiempo real es recomendable el uso de ordenadores con controladores gráficos potentes y con una capacidad de V-RAM mayor a 2 GB.

El algoritmo de navegación implementado en el controlador centralizado permite al sistema lograr una exploración eficaz de un entorno desconocido, mediante el descubrimiento de zonas exploradas y no exploradas por cada agente robótico, de esta manera, el controlador se encarga de ordenar a los robots si deben continuar su navegación en línea recta o si deben girar en dirección a espacios desconocidos.

El sistema de control centralizado integra el procesamiento de la cinemática directa de un robot móvil cuyos cálculos son efectuados en la Raspberry Pi 3, en donde mediante los parámetros de entrada correspondientes a distancias recorridas y ángulos de giro, son procesadas para obtener como salida las coordenadas en (X, Y) de los robots móviles y obstáculos, también se integró el procesamiento de la cinemática inversa efectuada en la Raspberry Pi 3, en la cual como parámetros de entrada se tiene a las coordenadas de las zonas no exploradas cuyos valores son obtenidos en el controlador central, permitiendo obtener como salidas, los ángulos de giro que deben efectuar los agentes robóticos.

En el sistema de control centralizado se utilizó enlaces de comunicación UDP mediante sockets en los cuales se configuró las direcciones IP de los agentes robóticos y sus respectivos puertos de enlace; para evitar problemas de colisión entre tramas se programó la apertura y cierre de dichos enlaces de manera que la comunicación se realice siempre entre 2 puntos, en este caso el ordenador central y el robot móvil a través su respectiva Raspberry Pi 3, esto significa una comunicación síncrona entre el ordenador y cada agente robótico.

Debido a que los tiempos de apertura y cierre de enlaces varían entre *20 ms* y *50 ms*, estas pequeñas variaciones de tiempo entre cada enlace permitieron que la configuración sincrónica sea imperceptible de tal manera que los datos enviados desde los robots sean recibidos al mismo tiempo en el ordenador central, al igual que los comandos enviados desde el ordenador hacia los agentes robóticos se produzcan en el mismo instante de tiempo.

Las pruebas iniciales del sistema robótico cooperativo fueron realizadas con un router TP-Link TLWR840N cuya velocidad de transmisión es de *300 Mbps* en la banda de *2.4 GHz*; en busca de optimizar los tiempos de latencia de datos en la comunicación inalámbrica se optó por utilizar un router Cisco RV130, cuya velocidad de transmisión llega a *800 Mbps*.

Inicialmente se programó enlaces de comunicación UDP mediante sockets en los cuales se utilizó tamaños de trama por defecto de 1024 bytes en la recepción de datos de Matlab y en la recepción de datos de la Raspberry Pi 3, generando un gasto computacional innecesario, por esta razón, se realizó el análisis de los tamaños de trama de comunicación inalámbrica en el cual se determinó que la trama entre la Raspberry Pi 3 y el ordenador central es de 140 caracteres equivalentes a 140 bytes, mientras que la trama de retorno necesita únicamente un tamaño de trama de 16 bytes.

Antes de realizar cada prueba fue necesario verificar el canal en el cual se encuentra la red Wifi correspondiente al sistema robótico, para de esta manera, colocar a la red en el canal con menor congestión y estabilizar las velocidades de transferencia, este cambio junto a la configuración de los tamaños de trama y el cambio de router a uno más eficiente, permitió reducir tiempos de latencia de 10 segundos a tiempos menores a los 5 segundos.

6.3. Pruebas y resultados experimentales

Mediante el trabajo cooperativo de los tres agentes robóticos móviles se permitió realizar la construcción de los mapas correspondientes a tres escenarios con diferentes configuraciones de ubicación de obstáculos, muestras de sustancias químicas y posiciones iniciales de los robots móviles, en cada prueba se obtiene un total de 10 mapas, al haber realizado 9 pruebas (3 por cada escenario propuesto), se obtuvo una cantidad total de 90 mapas divididos entre: mapas de obstáculos y fronteras mediante gráfica de píxeles y grafos de visibilidad, mapas de dispersión de temperatura en 2D mediante aplicación de filtros espaciales de media móvil e interpolación 2D, y mapas de dispersión de gases en 2D mediante aplicación de filtros espaciales de media móvil e interpolación 2D.

Se ha determinado que el método de grafos de visibilidad para la obtención de mapas de obstáculos y fronteras funciona de manera adecuada para la exploración y mapeo de entornos desconocidos, sin embargo, se recomienda para nuevas investigaciones, trabajar en algoritmos que sean capaces de reconocer esquinas, debido a que las mismas se presentan grafos que corresponden a zonas sin obstáculos lo cual es causado por la apreciación y resolución de los sensores ultrasónicos frente a un obstáculo que no es plano ni uniforme.

Para los mapas de dispersión de temperatura, en todas las pruebas se colocó un ventilador y un calentador eléctrico con el objetivo de variar los valores correspondientes a los niveles de temperatura en el ambiente, de esta manera se pudo comprobar que, aquellas zonas que se encuentran cercanas a la posición del calentador eléctrico, presentan los colores más cálidos lo cual representa los niveles más altos de temperatura, al contrario de las zonas cercanas a la posición del ventilador cuyas zonas se interpretan mediante colores fríos que representan los niveles más bajos de temperatura.

En los mapas de concentración de gases se pudo verificar que la mayoría zonas en las cuales se presenta mayor concentración de una determinada sustancia química, sea alcohol, tolueno o acetona, coinciden con la ubicación de las muestras presentadas en la configuración de los escenarios propuestos, también, se concluye que existe una interacción directa entre la temperatura y la dispersión de gases dado que en la zonas consideradas con niveles de temperatura alta, existe mayor detección de un gas en específico.

Cuando un robot móvil logra posicionarse directamente sobre una muestra dotada de una determinada sustancia, los sensores químicos perciben una gran cantidad de partículas correspondientes a dicha sustancia, esto provoca una variación de la resistencia interna de los sensores modificando el voltaje de salida en función de la concentración, para limpiar todas las partículas que se encuentran en la caja se necesita de varios ciclos de ventilación los cuales se realizan durante cada escaneo en el proceso de navegación, por esta razón se puede apreciar que, en varias zonas existe alta concentración de gases, aunque no se hayan ubicado las muestras en dichas posiciones.

6.4. Trabajos futuros

Para poder utilizar los sensores químicos es necesario que los mismos sean precalentados durante un periodo de 3 a 5 minutos, sin embargo para el desarrollo de pruebas de sensado químico más eficientes, se recomienda implementar un sistema de precalentamiento utilizando una fuente de alimentación externa de 5 [V], de tal manera que permita a los sensores un acondicionamiento de su temperatura interna durante periodos mayores a una hora, sin la necesidad de gastar innecesariamente la batería con la cual se alimenta a todos los sensores y actuadores que comprenden al agente robótico.

Los sensores químicos fueron calibrados de tal forma que su respuesta en función del voltaje de salida no supere un umbral de 3.3 [V] para cumplir con los requerimientos de voltaje admitidos en los pines de entrada de la Teensy 3.6, sin embargo para futuras investigaciones, es importante trabajar con las señales de voltaje bajo el umbral de los 5 [V], de esta forma estudiar el comportamiento de los umbrales en los cuales trabaja cada sensor que conforma la matriz ante la presencia de varias muestras de sustancias químicas a la vez, esto permitiría la implementación de un sistema encargado de la discriminación y diferenciación de gases para determinar el tipo de sustancia que fue sensada.

El sistema de comunicación inalámbrica se basa en la creación de enlaces mediante el uso de sockets, sin embargo, es importante plantearse la integración del sistema al Internet de las Cosas, mediante la disposición de un servidor que permita la creación de un base de datos en línea, a su vez esto facilitaría el diseño de una interfaz gráfica en la nube permitiendo el control, monitorización y visualización de datos desde cualquier lugar.

También es necesario el estudio de una configuración que se sea adaptiva a entornos de diferentes dimensiones respecto a su largo y ancho, de esta manera para trabajos futuros se recomienda implementar algoritmos que permitan a más de la configuración de las posiciones iniciales de los robots móviles, la configuración de las dimensiones del espacio de trabajo, permitiendo al sistema un mayor desenvolvimiento para tareas de exploración en entornos controlados.

REFERENCIAS BIBLIOGRÁFICAS

- Ajay, A., & Venkataraman, D. (February de 2013). A survey on sensing methods and feature extraction algorithms for SLAM problem. *International journal of computer science, engineering and applications (IJCSEA)*, 3.
- Alemi, H., & Bridges, T. J. (2010). *Review of the 3-2-1 Euler angles: a yaw-pitch-roll sequence*. University of Surrey, Department of Mathematics.
- Armingol, J. (1997). *Localización geométrica de robots móviles autónomos*. Universidad Carlos III de Madrid, Departamento de Ingeniería Eléctrica, Electrónica y Automática, Leganés.
- Arranz, Á., Baliñas, J., Bronte, S., García, J., González, D., Gutiérrez, J., . . . Sanz, V. (2006). *Aplicaciones de robots móviles*. Madrid.
- Atanasov, N., Ny, J. L., Daniilidis, K., & Pappas, G. J. (26-30 de May de 2015). Decentralized active information acquisition: Theory and application to multi-robot SLAM. *IEEE International conference on robotics and automation (ICRA)*.
- Azcón, A. B. (2003). *Análisis y diseño del control de posición de un robot móvil con tracción diferencial*. Escola Técnica Superior Enginyeria, Departament d'Enginyeria Electrónica Eléctrica i Automática.
- Báez, H. (2012). *Control distribuido aplicado a plataformas robóticas con arquitecturas de multiprocesadores*. Bogotá.
- Baturone, A. O. (2005). *Robotica manipuladores y robots móviles*. Marcombo.
- Bermudez, G. (2002). Robots móviles. teoría, aplicaciones y experiencias. *Tecnura 10*.

- Berná, J., Pérez, F., Gilart, V., Ramos, H., & Ferrándiz, A. (2008). Sistema de control para robots inspirado en el funcionamiento y organización de los sistemas neuroreguladores humanos . 59-75.
- Cabrera, A., & Delgado, G. (2014). *Diseño y construcción de un robot para mapeo y exploración de minas subterráneas*. Universidad del Azuay, Facultad de ciencia y tecnología, Cuenca, Ecuador.
- Carimatto, A. J., Verrastro, C., & Gómez, J. (2013). *Mapeo y navegación por nodos de potencial para robótica móvil*. Instituto nacional de tecnología industrial (INTI), Grupo de inteligencia artificial y robótica, Buenos Aires, Argentina.
- Carrasco, Á. F. (2015). *Estudio de sistemas de comunicaciones para aplicación en robótica cooperativa*. Universidad Autónoma de Madrid, Ingeniería de Telecomunicación, Madrid.
- Cruz, E. B. (2016). *Mapeo y localización basada en odometría usando imágenes como land marks*. Fundación Universitaria los Libertadores, Programa de Ingeniería Electrónica, Bogotá, D. C.
- Electronic Projects Components Available Worldwide. (2018). *www.pjrc.com*. Recuperado el 30 de 08 de 2018, de *www.pjrc.com*: *www.pjrc.com*
- Erazo, M., Matamoros, D., Minchala, J., & Paillacho, D. (2009). *Diseño e implementación de un robot móvil teleoperado en base al reconocimiento de forma y movimiento de objetos*. Escuela Superior Politécnica del Litoral (ESPOL), Facultad de Ingeniería en Electricidad y Computación, Guayaquil, Ecuador.
- Escalante, A. E. (2016). *Análisis del método de odometría lidar LOAM*. Escuela Técnica Superior de Ingeniería, Dep. Ingeniería de Sistemas y Automática, Sevilla.

- Espitia, H. (2011). *Propuesta de un algoritmo para la planeación de trayectorias de robots móviles empleando campos potenciales y enjambres de partículas activas brownianas*. Tesis, Universidad Nacional de Colombia, Facultad de Ingeniería, Bogotá D.C. Recuperado el 28 de 08 de 2018, de <http://bdigital.unal.edu.co/4230/1/02299874.2011.pdf>
- eva.fing. (2013). *eva.fing*. Obtenido de eva.fing: eva.fing.edu.uy
- Fernández, M., Fernández, D., & Valmaseda, C. (2010). *Planificación de trayectorias para un robot móvil*. Madrid.
- FIGARO. (2018). *FIGARO*. Recuperado el 01 de 09 de 2018, de www.figarosensor.com: www.figarosensor.com
- Fischer, T., Nitsche, M. A., & Pedre, S. (2014). *Fusión de encoders de cuadratura, sensores*. Buenos Aires, Argentina.
- Gómez, J. (2011). *Robótica móvil. estudio y caracterización del robot móvil kjunior desarrollo de aplicación de robot laberinto*. Cartagena.
- Holman, M., Edwar, J., & Martínez, F. (2015). Generación de ruta óptima para robots móviles a partir de segmentación de imágenes. *Información tecnológica*.
- Julián, V., & Botti, V. (2000). Agentes inteligentes: el siguiente paso en la inteligencia artificial.
- Linares, L., & Sánchez, L. (2008). *Introducción a los agentes inteligentes*.
- López Pérez, L. H., Mata Herrera, E., & García Pérez, E. (2001). Estado del arte en robótica móvil autónoma distribuida. *Conciencia tecnológica*.
- López, D. (2012). *Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos*. Universidad de Huelva , Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática , Huelva.

- Merino, L., Capitán, J., & Ollero, A. (2009). *Robótica cooperativa e integración con sensores en el ambiente. Aplicaciones en entornos urbanos*. Escuela Politécnica Superior, Universidad Pablo de Olavide, Sevilla.
- Michalczewsky, E., & Fillottrani, P. (2001). Un ambiente para programación y control de robots móviles autónomos.
- Molina, J., & Matellán, V. (1996). Robots autónomos: Arquitectura y control.
- Moreno, R. (2010). *Diseño y simulación de un algoritmo para el control de un robot modular tipo cadena*. Bogotá.
- Moya, C. (2017). *Sistema robótico móvil autónomo cooperativo para mapeo 2D de obstáculos y niveles de temperatura en un ambiente controlado*. Universidad de las Fuerzas Armadas ESPE, Departamento de Eléctrica y Electrónica, Sangolquí.
- Moyolema, D., & Pereira, R. (2014). *Diseño, construcción e implementación de un prototipo de robot móvil para el recorrido de trayectoria definidas por computador para el laboratorio de robótica industrial de DECEM*. Sangolquí.
- Navarro, I. (2005). Robots en el espacio.
- Naylamp Mechatronics. (2018). *Naylamp Mechatronics*. Recuperado el 01 de 09 de 2018, de [naylampmechatronics.com: https://naylampmechatronics.com/robotica-movil/240-encoder-infrarrojo.html](https://naylampmechatronics.com/robotica-movil/240-encoder-infrarrojo.html)
- Pastor, A. I., & Ayesa, C. Z. (2011). *Proyecto arduino UPNA*. Escuela técnica superior de ingenieros industriales y de telecomunicación, Pamplona.
- Pérez, E. (2003). *Implementación de la arquitectura tipo AuRA en mini robots LEGO*. Cholula, Puebla.
- Pineda, I. (2009). *Inteligencia artificial: Estructura de un agente*.

- Ponnu, G., & George, L. (2016). *Real-time Rosberry SLAM*. Cornell University, School of electrical and computer engineering.
- Pons, J. V. (2012). *Localización y generación de mapas del entorno (SLAM) de un robot por medio de una kinect*.
- Posadas, J., Simó, J., Poza, J., Pérez, P., Benet, G., & Blanes, F. (2018). Una arquitectura para el control de robots móviles mediante delegación de código y sistemas multiagente.
- Pretto, A. (2009). *Visual-SLAM for humanoid robots*. Università Di Pavoda, Dipartimento di Ingegneria dell'Informazione.
- robotnik. (2015). *robotnik*. Obtenido de robotnik: www.robotnik.es
- Robots, G. (2018). *Generation Robots*. Recuperado el 29 de 08 de 2018, de www.generationrobots.com: www.generationrobots.com
- Roca, V. M. (2012). *Desarrollo multiplataforma de aplicaciones de control y comunicación para robots móviles*. Valencia.
- Rodrigo, L. D. (2006). *Trabajo cooperativo en robots*. 3º Ingeniería de Telecomunicación.
- Rodríguez, J., Ramos, M., & Peña, M. (2015). Simulación de un robot hexápodo bioinspirado en el tenebrio. *Lámpsakos*, 7.
- Sánchez, G. E. (2013). *Localización y mapeo simultáneos para robot de búsqueda en entornos de desastre*. Universidad Nacional Autónoma de México, Ingeniería Mecánica – Ingeniería Mecatrónica, México, D. F.
- Sánchez, J. H. (2013). *Técnicas modernas en automática* (Vol. 3). Viña del Mar, Chile: Universidad Técnica Federico Santa María. Recuperado el 14 de abril de 2018

- Shoop, B., Johnston, M., Goehring, R., Moneyhun, J., & Skibba, B. (2006). Mobile detection assessment and response systems (MDARS) a force protection, physical security operational success.
- Valencia, D., & González, L. (2014). *Diseño e implementación de un prototipo de robot asistente para personas con discapacidad motriz y adultos mayores, basado en inteligencia artificial*. Universidad Politécnica Salesiana Sede Cuenca, Cuenca, Ecuador.
- Villa, M. A., & Vázquez, E. L. (2014). *Flotilla de robots para trabajos en robótica cooperativa*. Universidad militar Nueva Granada, Facultad de Ingeniería, Bogotá, D.C.
- Wang, X. (2013). *2D mapping solutions for low cost mobile robot*. Stockholm, Sweden: KTH CSC.
- World, S. (2018). *Sharp World*. Recuperado el 31 de 08 de 2018, de <http://www.sharp-world.com>:
http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf
- Zambrano, V. (2015). *Implementación de algoritmos de determinación de rutas para el robotino de FESTO*. Quito.