

Prototipo inalámbrico para inspección de tuberías en redes de alcantarillado en centros urbanos

Jeisson Fabián Téllez Rincón
Mateo Rodriguez Alfonso

Universidad Distrital Francisco José de Caldas
Facultad de ingeniería
Ingeniería Electrónica
Bogotá D.C.
2022

Prototipo inalámbrico para inspección de tuberías en redes de alcantarillado en centros urbanos

Jeisson Fabián Téllez Rincón
20151005071
Mateo Rodriguez Alfonso
20151005061

Trabajo de grado para optar al título de
Ingeniero Electrónico

Director
Leonardo Plazas Nossa

Modalidad
Monografía

Universidad Distrital Francisco José de Caldas
Facultad de ingeniería
Ingeniería Electrónica
Bogotá D.C.
2022

Resumen

En la actualidad la inspección de alcantarillado es un procedimiento complejo y costoso, en Bogotá y en muchas otras ciudades del país no se cuenta con un diagnóstico ni del 10 % de la red total de alcantarillado. Dependiendo de la empresa propietaria de la red de alcantarillado y del tipo de estructura a inspeccionar, existen diferentes normas y requerimientos que deben satisfacer los equipos o dispositivos que se vayan a utilizar para tal fin. Dentro de las características que deben tener los equipos, se destaca un mando que permita controlar el dispositivo a distancia, sistemas de iluminación, sensores y la más importante, permitir la captura de fotografías y videos de la estructura interna de la tubería. Para ello los dispositivos se equipan con una cámara y transmiten imágenes en tiempo real del estado de la tubería, sin embargo en la mayoría de casos la transmisión se realiza utilizando medios alámbricos como cables coaxiales o de fibra óptica lo cual puede resultar poco práctico ya que requiere cargar grandes carretes de cable a cada sitio de inspección. Por ello surge el presente proyecto, que pretende desarrollar un prototipo que sirva como base para un futuro equipo más económico y con las ventajas que trae implementar una comunicación inalámbrica en comparación con los equipos comerciales existentes.

El proyecto parte por explicar la terminología existente respecto a las estructuras de alcantarillado donde se realiza una identificación de los entornos de operación y el análisis de la norma NS-058 de la Empresa de Acueducto y Alcantarillado de Bogotá, la cual dicta los requerimientos para la inspección de la red de alcantarillado, además se realiza un estudio general de las tecnologías disponibles para la comunicación inalámbrica para elegir la más adecuada. A partir de ello se extraen los requerimientos de diseño para el prototipo, se realiza un estudio de equipos comerciales existentes y en base a esto se definen las funcionalidades del prototipo, detallando la selección de componentes, diseño de circuitos y conexiones necesarias. Se explica el procedimiento de construcción del prototipo junto a toda la programación necesaria para cumplir con las funcionalidades planteadas, una vez construido se realizan las pruebas de funcionamiento en un entorno real y una comparación entre el prototipo y los equipos comerciales existentes. Finalmente se dejan varias recomendaciones que se generaron durante el desarrollo del proyecto para trabajos futuros.

Índice

Resumen v

Índice de Figuras viii

Índice de Tablas xi

1. Generalidades **1**
1.1. Introducción . . . 1
1.2. Planteamiento del problema . . . 2
1.3. Objetivos . . . 3
1.3.1. Objetivo general . . . 3
1.3.2. Objetivos específicos . . . 3
1.4. Alcances y limitaciones . . . 4

2. Contextualización y criterios de diseño **5**
2.1. Terminología . . . 5
2.1.1. Alcantarillado . . . 5
2.1.2. Tuberías . . . 5
2.1.3. Estructuras . . . 6
2.2. Entorno de operación . . . 7
2.3. Norma NS-058 de la EAAB . . . 11
2.3.1. Requisitos para ejecutar la inspección . . . 12
2.3.2. Requisitos de los equipos y obtención de datos . . . 14
2.4. Enlaces de comunicación . . . 17
2.4.1. Comunicación inalámbrica . . . 17
2.4.2. Selección de tecnología a implementar . . . 23

3. Diseño del prototipo móvil **24**
3.1. Requerimientos de diseño . . . 24
3.2. Dispositivos y equipos comerciales . . . 24
3.2.1. iPEK ROVION . . . 25
3.2.2. X5-HT Crawler . . . 26
3.2.3. ANATROLLER ARI-10 . . . 27
3.3. Prototipo . . . 29
3.3.1. Movilidad . . . 29
3.3.2. Manipulador . . . 30
3.3.3. Comunicación inalámbrica . . . 32
3.4. Análisis cinemático . . . 35
3.4.1. Navegación del vehículo . . . 35
3.4.2. Manipulador para la cámara . . . 38
3.5. Selección de componentes . . . 42
3.5.1. Estructura . . . 42

3.5.2. Red inalámbrica 51
3.6. Diseño de circuitos y conexiones 53

4. Construcción del prototipo **56**
4.1. Ensamble y distribución de componentes 56
4.1.1. Nivel 1 57
4.1.2. Nivel 2 57
4.1.3. Tapa 58
4.1.4. Prototipo completo 59
4.2. Configuración de la red inalámbrica 60
4.3. Programación del prototipo 64
4.3.1. Aplicación web 64
4.3.2. Página principal 64
4.3.3. Ajustes de la cámara 65
4.3.4. Iluminación 66
4.3.5. Control de movimiento y manipulador para la cámara 67
4.3.6. Grabación de video 70
4.3.7. Toma de fotografías 71
4.3.8. Descarga de archivos 72
4.3.9. Transmisión de video 74
4.4. Diseño de controladores para los motores 75
4.4.1. Caracterización 78
4.4.2. Diseño del control digital 88
4.5. Control digital 99

5. Resultados **100**
5.1. Pruebas de funcionamiento 101

6. Presupuesto **105**

7. Conclusiones y trabajo futuro **106**
7.1. Conclusiones 106
7.2. Trabajo futuro 107

Referencias **111**

A. Código implementado en Arduino para caracterizar los motores. **112**

B. Código implementado en la aplicación web. **113**

C. Código implementado para el movimiento del vehículo. **120**

Índice de Figuras

2.1. Pozo de inspección [6]. 7
2.2. Plano con información de la primer ductería de 16". 8
2.3. Mapa de ubicación de la primer ductería de 16". 9
2.4. Plano con información de la segunda ductería de 16". 9
2.5. Mapa de ubicación de la segunda ductería de 16". 10
2.6. Plano con información de la tercer ductería de 16". 10
2.7. Mapa de ubicación de la tercer ductería de 16". 11
2.8. Esquema de un sistema de transmisión inalámbrico genérico [13]. 18
2.9. Logo de Bluetooth [15]. 19
2.10. Logo de WiFi Alliance [15]. 21
2.11. Modelo cliente-servidor utilizado en el protocolo HTTP [14]. 22
2.12. Estructura de una URL [14]. 22
3.1. Robot de inspección iPEK ROVION [17]. 25
3.2. Robot de inspección X5-HT Crawler [18]. 26
3.3. Robot de inspección ANATROLLER ARI-10 [19]. 28
3.4. Configuraciones para los robots móviles [20]. 29
3.5. Vehículo de carga - Skid Steer [21]. 30
3.6. Estructura de un robot manipulador PUMA 560. 31
3.7. Modelo 3D de un manipulador de codo. 31
3.8. Raspberry Pi 4 Model B [22]. 32
3.9. Regleta de pines GPIO de la Raspberry Pi 4 Model B [22]. 33
3.10. Diagrama de conexión inalámbrica. 33
3.11. Diagramas del vehículo sobre el marco de referencia inercial. 35
3.12. Diagrama con las velocidades lineales de las llantas. 36
3.13. Diagrama del manipulador con los marcos de referencia asociados. 38
3.14. Diagrama del manipulador para el método geométrico. 41
3.15. Motorreductor 625500/C [26]. 42
3.16. Juego de llantas todoterreno [27]. 43
3.17. Módulo puente H L298N. 44
3.18. Módulo Raspberry Pi Camera. 45
3.19. LED de potencia. 45
3.20. Servomotor SG90. 46
3.21. Módulo PCA9685 con l6 canales para señales PWM 47
3.22. Modelos 3D para el chasis y la tapa. 48
3.23. Batería de alimentación para los motores. 50
3.24. Batería de alimentación para la Raspberry Pi. 50
3.25. Router TL-WR840N de la marca TP-Link [32]. 51
3.26. Inversor de voltaje Schumacher XI14. 52

3.27. Batería de plomo 12 V, 5 Ah. 53
3.28. Diagrama de conexión del prototipo. 53
3.29. Diagrama de conexión del router. 54
3.30. Circuito de conexiones hecho en Proteus. 54
4.1. Vista general del chasis. 56
4.2. Vista lateral de los niveles del chasis. 56
4.3. Base en acrílico para el nivel 1. 57
4.4. Modelo 3D con los componentes del nivel 1. 57
4.5. Base en acrílico para el nivel 2. 58
4.6. Modelo 3D con los componentes del nivel 2. 58
4.7. Modelo 3D con el brazo sobre la tapa. 59
4.8. Modelo 3D del prototipo ensamblado. 59
4.9. Panel principal de configuración para el router TL-WR840N. 60
4.10. Panel de configuración inalámbrica para el router TL-WR840N. 61
4.11. Panel de configuración de seguridad inalámbrica para el router TL-WR840N. 61
4.12. Configuración del adaptador de red de la Raspberry Pi 62
4.13. Archivo de configuración para la red WiFi de la Raspberry Pi. 63
4.14. Prueba de conexión con el comando ping desde el computador a la Raspberry pi. 63
4.15. Página principal de la aplicación web. 64
4.16. Página principal con la configuración de la cámara. 66
4.17. Página principal mientras realiza una grabación de video. 70
4.18. Página de multimedia - sección de fotos. 72
4.19. Página de multimedia - sección de videos. 72
4.20. Aproximación lineal ideal 75
4.21. Control PI tiempo continuo. 76
4.22. Sistema de primer orden. 78
4.23. Respuesta del motor 1. 80
4.24. Respuesta del motor 2. 81
4.25. Respuesta del motor 3. 82
4.26. Respuesta del motor 4. 83
4.27. Función aproximada vs Valor Real Motor 1 86
4.28. Función aproximada vs Valor Real Motor 2 86
4.29. Función aproximada vs Valor Real Motor 3 87
4.30. Función aproximada vs Valor Real Motor 4 87
4.31. Sistema con un control discreto. 88
4.32. Lugar geométrico de las raíces Motor 1. 89
4.33. Respuesta al escalón Unitario Motor 1. 90
4.34. Lugar geométrico de las raíces Motor 2. 90
4.35. Respuesta al escalón Unitario Motor 2. 91
4.36. Lugar geométrico de las raíces Motor 3. 91
4.37. Respuesta al escalón Unitario Motor 3. 92
4.38. Lugar geométrico de las raíces Motor 4. 92

4.39. Respuesta al escalón Unitario Motor 4. 93
4.40. Lugar geométrico de las raíces Motor 1. 94
4.41. Respuesta al escalón Unitario Motor 1. 95
4.42. Lugar geométrico de las raíces Motor 2. 95
4.43. Respuesta al escalón Unitario Motor 2. 96
4.44. Lugar geométrico de las raíces Motor 3. 96
4.45. Respuesta al escalón Unitario Motor 3. 97
4.46. Lugar geométrico de las raíces Motor 4. 97
4.47. Respuesta al escalón Unitario Motor 4. 98
4.48. Diagrama control digital motores 99
5.1. Montaje final del prototipo. 100
5.2. Conexiones internas del prototipo. 101
5.3. Apertura del pozo y preparación del prototipo para la inspección. . . . 101
5.4. Primer intento de inspección de una de las tuberías del pozo. 102
5.5. Segundo intento de inspección de la otra tubería del pozo. 103
5.6. Fotos capturadas por el prototipo dentro de la tubería 104

Índice de Tablas

2.1. Materiales y diámetros de las tuberías en instalaciones a zanja abierta [5]. 6
2.2. Materiales y diámetros de las tuberías en instalaciones sin zanja [5]. . . 6
2.3. Criterios de selección de equipos para una videoinspección [1]. 12
2.4. Requisitos para la nomenclatura de los archivos de video [1]. 15
2.5. Requisitos para la nomenclatura de los archivos fotográficos [1]. 16
2.6. Características Bluetooth BR/EDR [15]. 20
2.7. Métodos comunes de una petición en el protocolo HTTP [14]. 23
2.8. Características generales WiFi [15]. 23
3.1. Parámetros D-H para un manipulador de codo. 40
3.2. Materiales de impresión 3D [30]. 48
3.3. Consumos de energía del prototipo. 49
4.1. Relaciones del tiempo establecimiento. 77
4.2. Valores de τ y K para el motor 1. 84
4.3. Valores de τ y K para el motor 2. 84
4.4. Valores de τ y K para el motor 3. 84
4.5. Valores de τ y K para el motor 4. 85
4.6. Valores deseados de diseño 88
4.7. Valores deseados de diseño 94
6.1. Tabla de precios para los componentes utilizados en el prototipo. 105
6.2. Comparación de precios con los distintos robots de inspección. 105
6.3. Comparación de características. 105

1. Generalidades

1.1. Introducción

La verificación del estado de las tuberías en estructuras de redes de alcantarillado necesita de un proceso de reconocimiento visual, donde se utilicen tanto equipos como recursos humanos dependiendo de las características de la tubería. En ocasiones esto presenta un limitante ya que por los tamaños en los que han sido construidas las tuberías, no es posible realizar una inspección visual directa por parte de una persona. Es por ello que se hace necesario contar con diversos equipos o dispositivos que permitan de alguna manera revisar el estado operativo y estructural de las tuberías a las que no es posible acceder de manera sencilla.

Generalmente los sistemas que existen para las inspecciones realizan una transmisión de imágenes en video por medios alámbricos y consisten en un pequeño robot explorador cuyo tamaño depende del diámetro de la tubería. Estos robots llevan abordo una cámara manual de 360° de visión que puede ser controlada por un operario, el cual al mismo tiempo puede manejar la dirección y velocidad del robot, sin embargo estos cuentan con elevados costos lo que los hace poco accesibles.

Dependiendo de la empresa propietaria de la red de alcantarillado, se definen unos criterios para la selección de estos sistemas así como los procedimientos y características que deben cumplir para realizar una inspección. La Empresa de Acueducto y Alcantarillado de Bogotá (EAAB) presenta la norma técnica de servicio NS-058 [1] que define los aspectos técnicos para las labores de inspección de redes y estructuras de alcantarillado, entendiendo como estructuras a los sumideros y pozos de inspección, y a las redes como las tuberías y conductos no circulares de alcantarillado de aguas residuales y pluviales. Esta norma indica las pautas para efectuar la inspección y clasificación de los defectos estructurales y operacionales, así como los requisitos mínimos que debe cumplir el robot o el equipo de inspección, por lo cual es necesario tenerla presente si se piensa llevar a cabo este tipo de procedimientos.

En estos términos y desde la Ingeniería Electrónica es posible proporcionar un sistema o dispositivo que sea capaz de inspeccionar las redes de alcantarillado urbano y que transmita las imágenes en video para analizar el estado en que se encuentran los componentes de la red. Por ello, el propósito del presente proyecto es diseñar y construir, con base en los conocimientos adquiridos en la carrera, un prototipo de un sistema que permita dicha inspección y que transmita de forma inalámbrica el video para su posterior almacenamiento y procesamiento, con el fin de facilitar la identificación de daños para así poder realizar planes de mantenimiento o reparación de ser necesario.

1.2. Planteamiento del problema

Uno de los inconvenientes a la hora de inspeccionar redes de alcantarillado es que en una gran parte de tuberías el tamaño es bastante pequeño para que una persona sea capaz de realizar una inspección directa. Esto genera problemas al momento de determinar con exactitud fallas dentro de la red, que pueden provocar demoras en el mantenimiento o las reparaciones que sean necesarias y dar lugar a posibles accidentes. Con el fin de solucionar dichos inconvenientes, se han creado diferentes tipos de sistemas que deben cumplir ciertas normas a la hora de ser utilizados para realizar una inspección.

La Empresa de Acueducto y Alcantarillado de Bogotá (EAAB) proporciona la norma NS-058 que precisa los aspectos técnicos para la inspección de redes y estructuras de alcantarillado. En esta norma se detalla cómo es el proceso para realizar inspecciones en tuberías de diferentes diámetros. También indica los aspectos generales para realizar la inspección, cómo debe ser el equipo, la pantalla de visualización, la cámara, la iluminación, entre otras indicaciones [1]. Por esto es necesario basarse en esta norma al momento de levantar los requerimientos de diseño para un sistema de inspección de redes de alcantarillado. Estos sistemas van desde cámaras tipo endoscopio, que cuentan con su carrete de fibra óptica y pantalla de visualización, hasta pequeños robots equipados con sensores, motores y cámaras que transmiten por medio de fibra óptica las imágenes para su posterior almacenamiento y análisis.

A lo largo de los últimos años se han venido desarrollando distintos prototipos de robots para inspección que son adaptables a diferentes tipos de tuberías y ambientes. Para tuberías de presión que en general tienen un diámetro menor a 15 cm, se ha diseñado un prototipo de robot con forma de lombriz de tierra capaz de moverse en un espacio tan estrecho y que es capaz de alcanzar distancias de hasta 100 m en tuberías horizontales y verticales. El funcionamiento de este robot se basa en los movimientos musculares de una lombriz, por lo que permite una gran movilidad en espacios pequeños y facilita realizar giros de 90° que suelen ser comunes en las tuberías. El movimiento de este robot se hace utilizando un sistema de carga y descarga de aire por medio de una manguera junto a la cual puede ir un cable de fibra óptica para la transmisión de imágenes [2].

Del mismo modo, para tuberías con diámetros mayores se han creado robots en forma de pequeños vehículos equipados con llantas que son muy comunes para realizar inspecciones de tuberías. Estos robots en algunos casos tienen problemas para desplazarse en las tuberías por causa de inclinaciones u obstáculos que haya en el camino, por lo que se han diseñado prototipos que mejoran la movilidad del robot en la tubería. Uno de esos prototipos consiste en implementar un módulo que permita al robot aumentar su fuerza de tracción cuando encuentre algún obstáculo o se transporte por tuberías de gran inclinación. Del mismo modo la transmisión de imágenes para este tipo de robots se realiza equipando al robot con una cámara y utilizando un cable de fibra óptica [3].

Como este par de ejemplos existen muchos sistemas con diferentes características, pero en una gran mayoría se realiza la transmisión de imágenes por medios alámbricos que pueden provocar inconvenientes en el robot. Por ejemplo si la distancia es muy grande, la fricción del cable con la tubería puede hacer que el robot requiera más potencia para desplazarse porque se genera un efecto de carga, además se limita la movilidad del mismo dentro de la tubería y el hecho de cargar con un carrete de fibra óptica no es muy práctico para los operarios.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar e implementar un sistema robótico inalámbrico de bajo costo capaz de inspeccionar redes de alcantarillado urbano que brinde información relevante de la misma, cumpliendo algunos puntos importantes de la normativa NS-058.

1.3.2. Objetivos específicos

- Identificar los entornos en los cuales va a desplazarse y operar el sistema robótico dentro de la red de alcantarillado.
- Identificar las tecnologías y plataformas que permitan una comunicación inalámbrica de tipo full-duplex.
- Diseñar un robot capaz de moverse de forma óptima y precisa en redes de alcantarillado, y un mando que brinde al operario los elementos necesarios para controlarlo de forma inalámbrica.
- Construir un prototipo funcional capaz de realizar las tareas necesarias para la inspección, y un control que brinde al usuario las herramientas necesarias para controlarlo adecuadamente de forma inalámbrica.
- Diseñar e incluir en el prototipo un sistema para la transmisión de imágenes y vídeo de forma inalámbrica.
- Evaluar el correcto funcionamiento del robot en la inspección de redes de alcantarillado urbano.

1.4. Alcances y limitaciones

El presente proyecto está enfocado en diseñar y construir un prototipo robótico básico que sea funcional en la inspección de redes de alcantarillado con base en la norma NS-058, así que las funcionalidades están enfocadas en satisfacer los requisitos mínimos para los equipos y la manera de obtener los datos que dicta la norma. Una de las principales características con las que deben contar este tipo de sistemas, es garantizar la integridad y perfecto funcionamiento en entornos húmedos y con presencia de aguas, por lo cual los materiales a utilizar como llantas, motores, sensores y la cámara deben ser de una excelente calidad, además de contar con las características adecuadas para dichos entornos como la impermeabilización. Sin embargo, al ser un prototipo los materiales y recursos a utilizar no van a cumplir con esas características, los materiales y componentes a utilizar van a estar enfocados en la funcionalidad con un bajo costo, porque se busca crear un primer modelo que pueda ser perfeccionado a futuro, además al basarse en la norma NS-058 estas limitaciones pueden justificarse en que las inspecciones se deben realizar en tiempo seco.

Por otro lado, al implementar un medio inalámbrico de transmisión de información, las distancias que se pueden alcanzar dentro de la tubería no van a ser las mismas que se logran utilizando un sistema cableado convencional. En un sistema cableado las distancias están directamente relacionadas con la longitud del carrete de cable con el que cuenten, los cuales pueden son tienen alrededor de 100 metros, mientras que para un sistema inalámbrico las distancias dependen de varios factores como la tecnología, los entornos, interferencias, antenas y los mismos equipos utilizados así que no se espera alcanzar esas distancias en este proyecto.

La evaluación del funcionamiento del prototipo no se puede realizar en una red de alcantarillado cualquiera ya que es pertenencia de la Empresa de Acueducto y Alcantarillado de Bogotá, y puede resultar complicado solicitar los permisos necesarios para ello. Por esto y con ayuda del docente director, se realiza la gestión con la Pontificia Universidad Javeriana para acceder a su red de alcantarillado a realizar pruebas, así que el desarrollo del proyecto va a estar enfocado a ese entorno, el cual puede diferir de los entornos en una red de alcantarillado de la Empresa de Acueducto y Alcantarillado de Bogotá. Teniendo esto en cuenta, tampoco se van a aplicar todos los procedimientos que dicta la norma NS-058 para realizar una inspección cuando se realicen las pruebas.

Teniendo todo lo anterior en cuenta el alcance del proyecto es diseñar y construir un primer prototipo funcional básico, que permita realizar las tareas de inspección de alcantarillado transmitiendo y almacenando de forma inalámbrica las imágenes del estado de las tuberías o estructuras inspeccionadas.

2. Contextualización y criterios de diseño

Este capítulo parte por definir la terminología utilizada en los componentes que hacen parte de una red de alcantarillado a nivel general, también se identifican los entornos de operación del prototipo dentro del campus de la Pontificia Universidad Javeriana donde es posible solicitar permisos para acceder a su red de alcantarillado y realizar pruebas, esto con el fin de no solo basarse en la normativa sino utilizar también datos reales del entorno de operación. Se realiza un análisis de la norma NS-058 [1], ya que la idea es cumplir con los requisitos mínimos que allí se mencionan para realizar inspecciones en las redes de alcantarillado. Evidentemente al ser un prototipo no se espera cumplir con la totalidad de los requisitos planteados en la norma, pero si extraer unos requerimientos de diseño para implementar una base funcional que pueda ser mejorada y ajustada a la norma en el futuro. Finalmente se realiza un análisis general sobre algunas tecnologías y plataformas para establecer una comunicación inalámbrica, con el fin de comparar sus características y así mismo elegir la que mejor se ajuste al proyecto.

2.1. Terminología

2.1.1. Alcantarillado

El alcantarillado de determinado lugar es un conjunto de obras para la recolección, conducción y disposición final de las aguas residuales, de infiltración o pluviales. Dentro del alcantarillado se tienen las redes de alcantarillado, que son un conjunto de tuberías, conductos y estructuras que conforman el sistema de evacuación de aguas que pueden ser industriales, residuales o pluviales. Dentro de estas redes se encuentra la denominada red local de alcantarillado donde desembocan las conexiones domiciliarias del alcantarillado de los hogares, hasta llegar a lo que se conoce como red secundaria de alcantarillado. La red secundaria es igualmente un conjunto de canales y tuberías que en su recorrido van acumulando áreas de drenaje, conduciendo lo que proviene de la red local hasta lo que se conoce como red troncal o red principal de alcantarillado, que es el conjunto de intercepciones que definen la estructura básica del drenaje conduciendo los caudales hasta el sitio de vertimiento o tratamiento [4].

2.1.2. Tuberías

Para la construcción y reparación de redes de alcantarillado, se tienen diferentes clases de tuberías con sus respectivas características según los materiales y diámetros definidos en la norma técnica NP-027 [5] titulada "Tuberías para alcantarillado". Dependiendo si la instalación es a zanja abierta (instalación donde se realiza una excavación en el terreno para instalar tuberías subterráneas) o si es una instalación sin zanja (instalación donde se instalan tuberías subterráneas con la mínima excavación y alteración superficial), se tienen diferentes materiales y diámetros de las tuberías, los cuales se observan en las Tablas 2.1 y 2.2.

Tabla 2.1: Materiales y diámetros de las tuberías en instalaciones a zanja abierta [5].

Materiales	**Diámetro (mm)**
Tubería de concreto	
Sin refuerzo	150-500
Reforzada con varillas de acero	600-2750
Reforzada con fibras de acero	200-500
Reforzada y revestida con lámina de polietileno	1000-2750 (mm)
Tubería de PVC	
Con pared interior lisa y exterior corrugada	110-1050
Perfil abierto	160-1500
Perfil cerrado	600-1500
Tubería de Gres	150-900
Tubería de poliéster reforzado con fibra de vidrio (GRP)	200-3600
Tubería metálica corrugada con revestimiento interno en concreto	375-3600
Tubería de Polietileno (PE) de alta densidad	
Pared estructural corrugada	150-800
Doble pared, exterior corrugado e interior liso	150-1500
Doble pared, superficie interna y externa lisa perfilada tipo A2	500-2400

Tabla 2.2: Materiales y diámetros de las tuberías en instalaciones sin zanja [5].

Materiales	Diámetro (mm)
Tubería en concreto tipo Jacking Pipe	900-2750
Tubería en poliéster reforzado con fibra de vidrio (GRP)	250-3600
Tubería PVC enrollado en espiral fabricada en campo SWL	200-4500
Tubería PEAD enrollado en espiral fabricada en campo SWL	750-3000
Tubería Curada en Sitio CIPP	50-2700

Se tiene en cuenta de las tablas anteriores la información sobre los diámetros de las tuberías, con el fin de considerar los escenarios en los cuales va a desplazarse el sistema robótico del proyecto para la etapa de diseño.

2.1.3. Estructuras

Para la ejecución de labores de inspección, además de las tuberías que conforman las redes de alcantarillado, hay dos estructuras a tener en cuenta: los pozos de inspección y los sumideros. Los pozos de inspección son estructuras construidas principalmente de ladrillo y concreto o de materiales plásticos, son diseñados y construidos para la ventilación, el acceso y el mantenimiento de los diferentes colectores de alcantarillado. Cuentan con una tapa removible que cierra la parte superior del pozo de inspección de alcantarillado y cumple la función de controlar el acceso a los pozos, evitar la entrada de elementos extraños y accidentes por la caída de personas y objetos dentro de los pozos, además de permitir la ventilación y un fácil acceso para las labores de mantenimiento

necesarias. También tienen la función de permitir cambios de alineamientos horizontales y verticales dentro del sistema de alcantarillado [4].

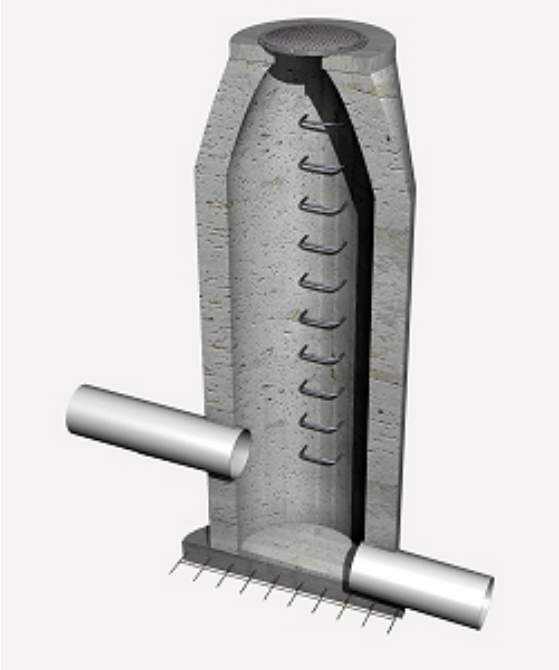


Figura 2.1: Pozo de inspección [6].

Dentro de los pozos de inspección puede tenerse o no lo que se denomina cámaras de caída, que son estructuras de conexión cuya función es dar continuidad al flujo de aguas y evitar la erosión dentro de la misma estructura, además se encargan de disipar la energía mediante el impacto del flujo sobre el fondo del pozo, en la Figura 2.1 se muestra uno de estos pozos de inspección con una cámara de caída [7].

Por otro lado los sumideros son estructuras diseñadas y construidas para cumplir con el propósito de captar las aguas de escorrentía (caudal generado por el agua lluvia que fluye en la superficie de un terreno) que corren por las cunetas de las calzadas de las vías y por las mismas vías, para entregarlas a los pozos de inspección de los alcantarillados de aguas residuales o de aguas lluvias [8]. Los sumideros están compuestos por una estructura de captación y por una caja de sedimentación. La estructura de captación es la parte superior del sumidero que capta las aguas de escorrentía superficial y las conduce hacia la caja de sedimentación.

2.2. Entorno de operación

Gracias a la gestión del docente director Leonardo Plazas, es posible solicitar permisos para acceder a la red de alcantarillado del campus de la Pontificia Universidad Javeriana donde se pueden realizar las pruebas de funcionamiento una vez se tenga

el prototipo, esto es de gran importancia y utilidad ya que puede resultar muy difícil conseguir permisos para realizar pruebas dentro de la red de alcantarillado de la EAAB. Es clave conocer las características de dicho entorno para pensar en el diseño y construcción del prototipo por ello se ha solicitado información sobre las tuberías a las que se puede tener acceso.

Del campus de la Pontificia Universidad Javeriana, se informa que existen tuberías con diámetros de 4", 6", 8", 10" y 16". Con estos datos se estima que lo mejor es solicitar los perfiles e información de las tuberías con mayor diámetro para evitar posibles inconvenientes al momento de realizar las pruebas. Se tienen entonces tres posibles tuberías, en primer lugar como se observa en la Figura 2.2, se cuenta con una tubería de longitud de 19.41 metros, un diámetro de 16", una pendiente en (%) de 17.4 y concreto como material de fabricación.

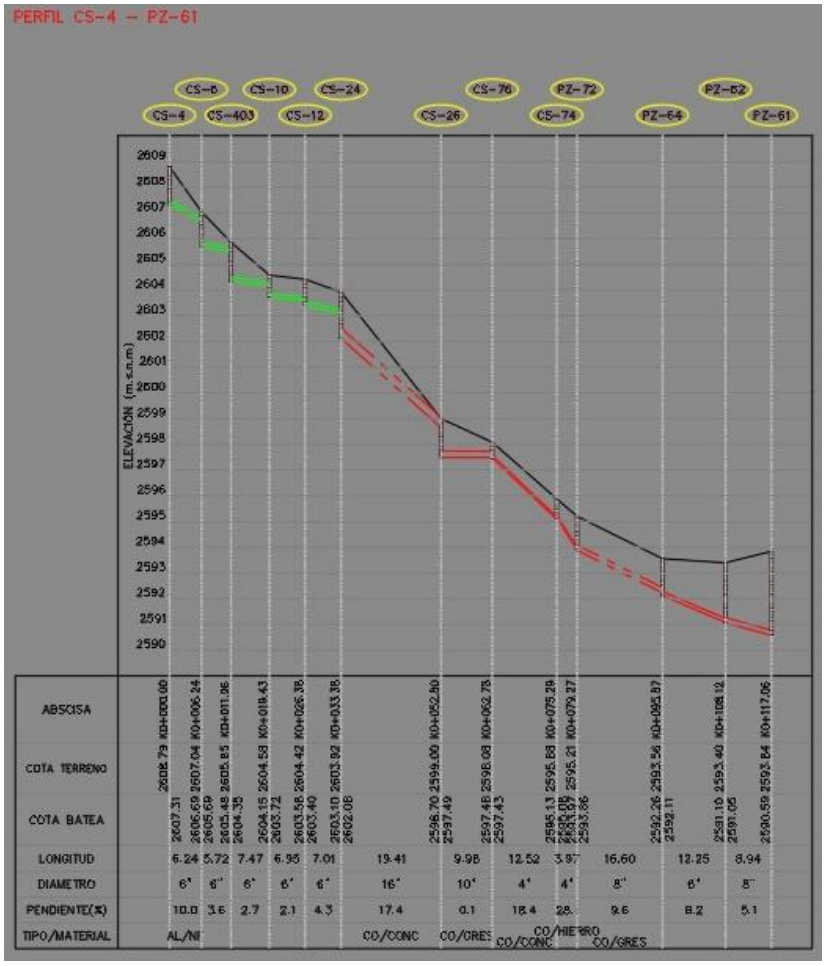


Figura 2.2: Plano con información de la primer ductería de 16".

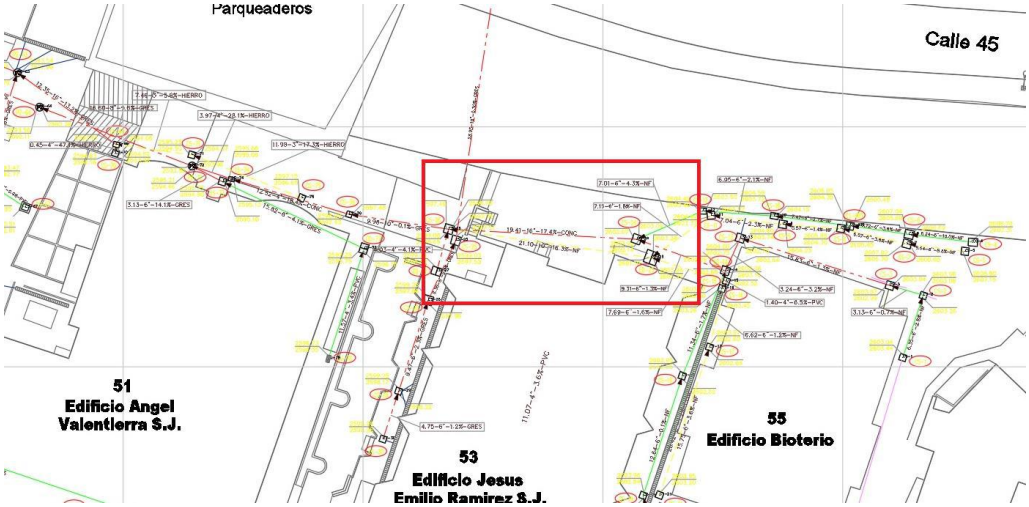


Figura 2.3: Mapa de ubicación de la primer ductería de 16".

Como se observa en la Figura 2.4, se tiene una tubería con una longitud de 12.35 metros, un diámetro de 16", una pendiente en (%) de 13.2, y gres como material de fabricación.

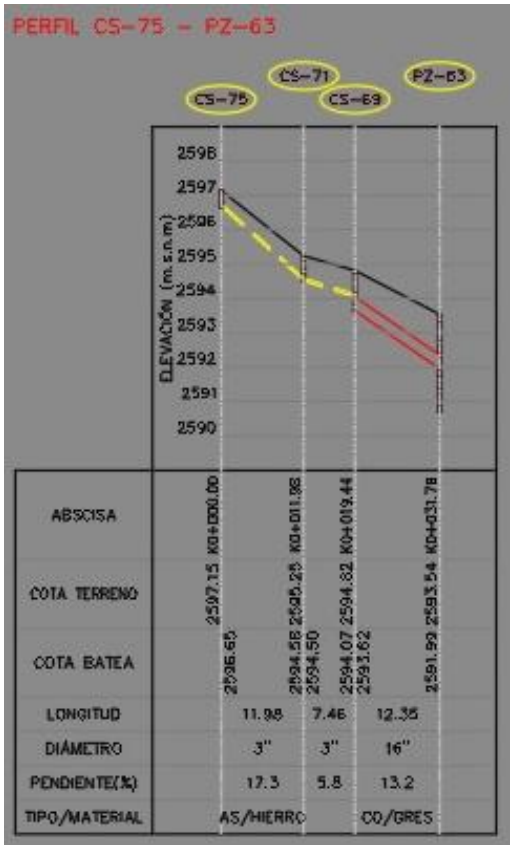


Figura 2.4: Plano con información de la segunda ductería de 16".

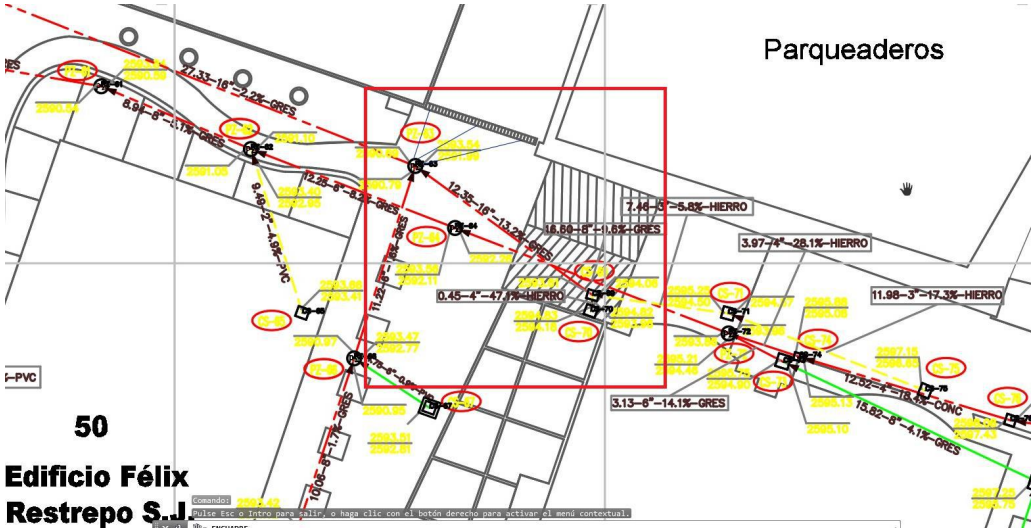


Figura 2.5: Mapa de ubicación de la segunda ductería de 16".

Finalmente como se observa en la Figura 2.6, se cuenta con una tubería de longitud de 27.33 metros, un diámetro de 16", una pendiente en (%) de 2.2, y gres como material de fabricación.

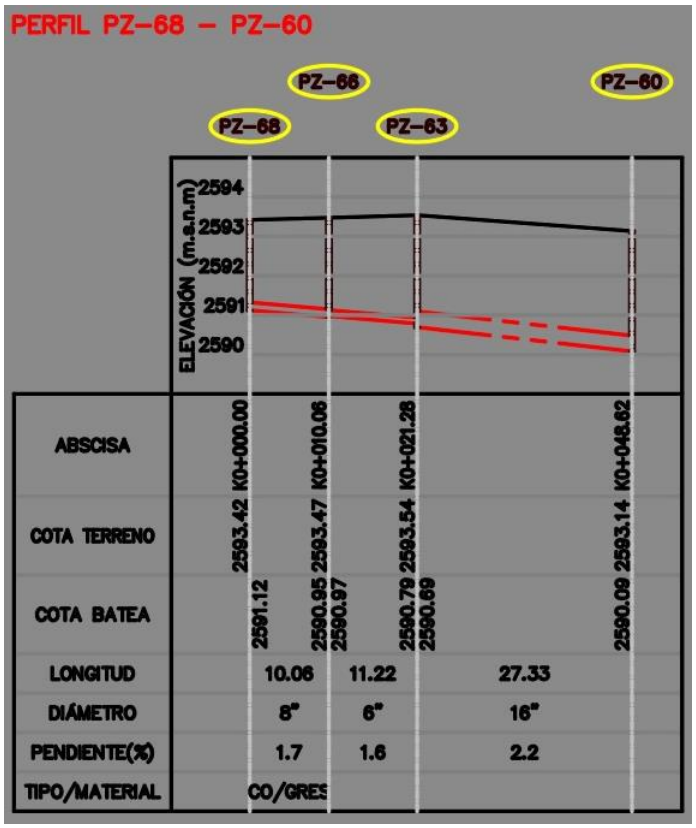


Figura 2.6: Plano con información de la tercer ductería de 16".

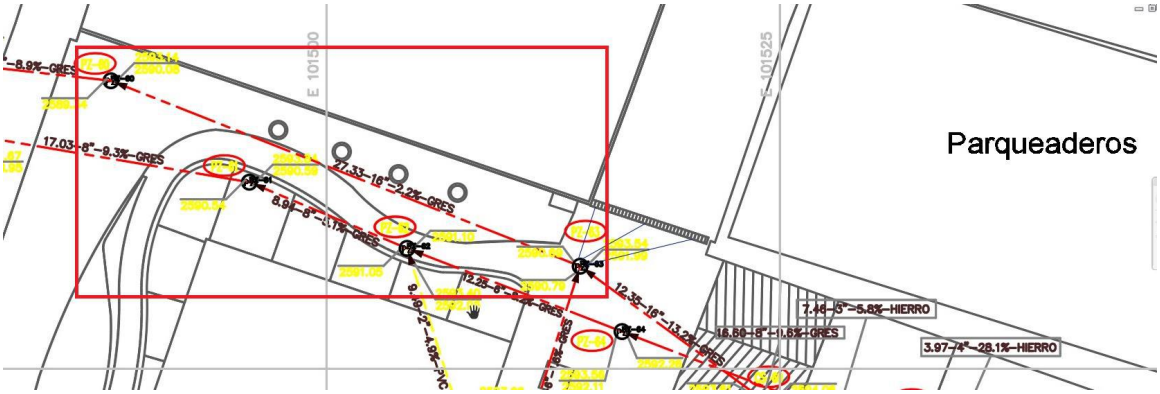


Figura 2.7: Mapa de ubicación de la tercer ductería de 16".

2.3. Norma NS-058 de la EAAB

Para la verificación del estado de las redes y estructuras de alcantarillado, la EAAB define en la norma NS-058 [1], los requisitos mínimos para realizar una inspección de video utilizando equipos de CCTV (circuito cerrado de televisión). El objetivo de realizar este procedimiento es verificar la condición interna de una tubería, para poder identificar entre otros:

- Condiciones actuales de la tubería como problemas de instalación, de operación o a nivel estructural, que permita tener una ubicación aproximada dentro de la red.
- Calidad y alcance de los trabajos de reparación o renovación de la red.
- Conexiones ilegales o erradas en el servicio de alcantarillado.
- Puntos de infiltración en la red de aguas residuales.
- Puntos de exfiltración, escape o vertimientos de aguas residuales.
- Métodos apropiados para realizar mantenimientos, reparaciones o renovaciones.

En general existen dos escenarios donde se realiza una evaluación de este tipo: en tuberías nuevas y en tuberías existentes. Cuando se realiza una inspección a instalaciones nuevas, el objetivo principal es confirmar el estado de la instalación para generar los respectivos recibos de pago. En el caso de las tuberías existentes, el objetivo de la inspección es proveer información que permita realizar una evaluación estructural y operacional de la red de alcantarillado, que permita iniciar con los planes correctivos de la incidencia [1].

2.3.1. Requisitos para ejecutar la inspección

Criterios para la selección de equipos

Inicialmente se requiere que los encargados de realizar el procedimiento tengan conocimientos técnicos y experiencia en la construcción, operación y materiales de las redes de alcantarillado. También hay que considerar previamente que en el momento de iniciar con un trabajo de inspección, pueden ocurrir lluvias que aumenten el caudal dentro de la tubería y arrastrar los equipos que se encuentren en ella, de todos modos es preferible siempre realizar la inspección en tiempo seco. Para todos los casos de inspecciones de video dentro de la red de alcantarillado, se debe utilizar equipos autopropulsados, balsas u otro sistema salvo para tuberías que superen las 60 pulgadas de diámetro, donde el personal puede ingresar a hacer una inspección directa aunque la norma recomienda evitar este tipo de inspecciones [1]. Dependiendo del diámetro y el nivel de flujo dentro de la tubería, la norma define ciertas opciones de tecnologías a utilizar para realizar una inspección. En la Tabla 2.3 se muestran los diferentes casos que pueden existir.

Tabla 2.3: Criterios de selección de equipos para una videoinspección [1].

Diámetro	**Nivel de flujo**	**Opción de tecnología**
4" - 6"	<20 % del diámetro	- Videocámara de empuje
8" - 15"	<20 % del diámetro	- Tractor de orugas o ruedas con CCTV - Equipo de escaneo lateral
	>20 % del diámetro	- Requiere control de flujo o derivación
15" - 36"	<25 % del diámetro	- Tractor de orugas o ruedas con CCTV - Equipo de escaneo lateral
	25 % - 60 % del diámetro	- Balsa CCTV tradicional o HD - Balsa multisensor CCTV/Sonar/Perfil Láser/Gas
36" - 48"	<30 % del diámetro	- Tractor de orugas o ruedas con CCTV
	30 % - 60 % del diámetro	- Balsa CCTV - Balsa multisensor CCTV/Sonar/Perfil Láser/Gas
48" - 72"	<30 % del diámetro	- Tractor de orugas o ruedas con CCTV - Tractor de orugas o ruedas con multisensor
	30 % - 70 % del diámetro	- Balsa multisensor CCTV/Sonar/Perfil Láser/Gas
>72"	<30 % del diámetro	- Tractor de orugas o ruedas con CCTV - Tractor de orugas o ruedas con multisensor - Inspección directa con entrada de personal
	30 % - 70 % del diámetro	- Balsa multisensor CCTV/Sonar/Perfil Láser/Gas

Actividades previas

La norma recomienda realizar una serie de actividades previas a la inspección de una red de alcantarillado. En primer lugar se debe realizar una revisión de los planos o mapas del sector que se va a inspeccionar con el fin de localizar los puntos de acceso que permitan planear la ubicación de todos los equipos que se van a utilizar. También se recomienda realizar una visita técnica a los pozos de inspección para verificar la accesi-

bilidad, nivel de flujo, diámetros, presencia de objetos en la red, gases, entre otros, para determinar si la inspección se puede llevar a cabo o de ser necesario establecer unos tiempos de espera para realizar toda la adecuación necesaria, en los cuales se programen todos los trabajos necesarios como por ejemplo la limpieza de los tramos de tubería a inspeccionar, control de flujo, etc.

Cualquier tramo dentro de la red con diámetros menores a 15" deben tener una programación de trabajos para limpieza en un plazo de al menos 72 horas antes de la inspección. En caso de tener un diámetro mayor, se deben programar trabajos especiales donde empleen los equipos y el personal adecuado. Si existen condiciones especiales como tuberías en mal estado y obstáculos de gran tamaño al interior que impidan la inspección se debe reportar el caso a la EAAB para que determinen el método de limpieza más efectivo [1].

Actividades generales durante la inspección

Se indica también de manera general realizar las actividades que se describen a continuación al momento de realizar una inspección a la red de alcantarillado, cabe aclarar que en la norma se encuentra más detallado el procedimiento [1].

- Ejecutar el plan de manejo de tránsito PMT (si aplica). Este plan se realiza con el fin de determinar el tipo de obra, horarios de intervención, zonas, necesidades, características de infraestructura, impactos sobre el sector, entre otros.
- Realizar la derivación del caudal o un control de flujo (si aplica).
- Realizar un reconocimiento superficial sobre el alineamiento de la tubería, con el fin de detectar el estado y el nivel de las tapas de los pozos, grietas sobre el pavimento, huecos, hundimientos o empozamientos de agua.
- Destapar el pozo aguas arriba y aguas abajo para permitir la ventilación de gases. Hay que aclarar que no se deben dejar los pozos abiertos sin supervisión del personal.
- Realizar una medición de los gases tóxicos que puedan existir en el pozo.
- Asegurar el área cercana a los pozos destapados con elementos de protección para prevenir accidentes.
- Iniciar con la grabación en video realizando las tomas requeridas.
- Bajar el equipo de inspección o ensamblarlo dentro de la tubería en el punto donde se va a iniciar la inspección.
- Desplazar el equipo a través de la tubería entre los dos puntos de acceso y realizar las tomas de video y fotografías que permitan documentar los defectos u observaciones encontradas.

- De manera opcional se pueden realizar grabaciones de audio.
- Una vez se complete la inspección del tramo de la red, recuperar todos los equipos que se hayan insertado.
- Finalmente volver a tapar los pozos.

2.3.2. Requisitos de los equipos y obtención de datos

Medición de distancias

Es un requisito contar con contadores longitudinales calibrados en la inspección, dichos conteos deben hacerse en metros y con un error no mayor a 0.5 % (0.5 m por cada 100m). Para validar la precisión del contador se debe contar con una cinta medidora y desenrollar el carrete de cable que tenga el equipo en caso de ser alámbrico. En caso de que la precisión esté por debajo o por encima del nivel de tolerancia, es necesario calibrar el contador. Es indispensable que las mediciones de distancias sean lo más precisas posible durante la inspección para registrar los datos y tener certeza de en qué punto de la tubería se encuentra el equipo así como indicar de la manera más aproximada los lugares de incidencia si es que se encuentra alguna [1].

Cámara y transportador

Se indican los requisitos mínimos que debe tener el sistema de videocámara y transportador (ruedas u orugas), dichos requisitos se listan a continuación [1].

- El equipo debe ser especialmente diseñado y construido para operaciones en alcantarillados.
- Debe tener la capacidad de operar en medios húmedos hasta el 100 % de humedad relativa.
- Tanto el transportador como la cámara deben tener la capacidad de estar sumergidos en aguas residuales durante periodos prolongados.
- En el caso de la cámara debe permitir realizar paneos y giros para observar el estado estructural interno de la tubería.
- El equipo debe contar con iluminación LED blanca incorporada, que sea ajustable para distintos diámetros y condiciones de oscuridad.
- La cámara debe permitir realizar grabaciones a color con una resolución mínima de 470 líneas horizontales.
- De se posible debe permitir realizar un acercamiento o zoom óptico de 10x y digital de 4x.

- El transportador debe tener la posibilidad de intercambiar ruedas u orugas, y permitir levantar la cámara para inspeccionar distintos diámetros.
- Debe tener control ajustable de velocidad.

Archivos de video

Para los registros visibles de cada tramo inspeccionado se debe entregar a la EAAB los archivos de video con las siguientes especificaciones:

- Formato estándar internacional de codificación de video MPEG-4 con extensión de archivo .mp4 o .m4v, bajo los requisitos de la norma ISO/IEC 14496 [9].
- Resolución mínima de video tipo VGA de 640 x 480 x 30 x 1.5 (resolución vertical x resolución horizontal x cuadros por segundo x bits por pixel/segundo).
- La reproducción del video debe permitir observar una imagen nítida, sin distorsiones, cambios de color, ruido, interferencias o demás imperfecciones.
- Deben permitir su apertura en los diferentes reproductores de video sin importar el sistema operativo.
- Debe reflejar la misma velocidad a la cual se desplaza el equipo al interior del tramo de tubería.
- Cada archivo de video debe corresponder a un segmento de tubería.

De manera adicional se solicita que los archivos de video sean nombrados con la nomenclatura que se muestra en la Tabla 2.4, también se muestra un ejemplo para nombrar un archivo tomado el 1 de febrero del 2019 a las 3 horas 20 minutos de la tarde.

Tabla 2.4: Requisitos para la nomenclatura de los archivos de video [1].

Requisito de nomenclatura	AAAA-MM-DDTHHMM_IDSIG_D.mp4
Ejemplo	2019-02-01T1520_CLT124569_F.mp4

Donde,

- **AAAA-MM-DDTHHMM:** Fecha y hora de la inspección según los requisitos de representación de datos y tiempos con formato extendido de la norma ISO 8601 [10]. Debe contener la fecha en orden descendente en relación con la unidad de tiempo: año, mes, día, hora, minutos, con hora en formato militar. En caso de ser unidades de un solo dígito deben ser precedidas de ceros para cumplir con el formato de la Tabla 2.4.

- **IDSIG:** Código asignado al tramo de la tubería en el Sistema de Información Geográfica (SIGUE).
- **D:** Dirección de desplazamiento del equipo (F - flujo, C - contraflujo).

Fotografías digitales

De manera similar se solicita entregar los archivos fotográficos que deben ser tomados directamente del video cada vez que se registre una condición observada en la tubería. Dichos archivos deben cumplir con los siguientes requerimientos:

- Las fotografías deben presentar claramente todos los requisitos de texto en la pantalla
- Las fotografías deben almacenarse digitalmente utilizando el formato de compresión digital de imágenes JPEG con extensión de archivo .jpeg o .jpg de acuerdo con los requisitos establecidos en la norma ISO/IEC 10918 [11].
- Todas las fotografías deben tener una resolución mínima de 640 x 480 (resolución horizontal x resolución vertical) y debe ser lo suficientemente clara para identificar la incidencia.

Al igual que con los videos se solicita que los archivos de fotografías sean nombrados con la nomenclatura que se muestra en la Tabla 2.5, del mismo modo se presenta un ejemplo para nombrar un archivo que reporta una grieta tomado el 1 de febrero del 2019 a las 3 horas 20 minutos de la tarde.

Tabla 2.5: Requisitos para la nomenclatura de los archivos fotográficos [1].

Requisito de nomenclatura	AAAA-MM-DDTHHMM_IDSIG_CÓDIGO.jpeg
Ejemplo	2019-02-01T1520_CLT124569_CL.jpeg

Donde,

- **AAAA-MM-DDTHHMM:** Fecha y hora del momento en que se captura la fotografía según los requisitos de la norma ISO 8601 [10].
- **IDSIG:** Código asignado al tramo de tubería en el Sistema de Información Geográfica (SIGUE).
- **Código:** Código que describe la fotografía tomada.

La norma NS-058 [1] incluye y detalla más aspectos que para el proyecto no resultan relevantes ya que tratan sobre procedimientos y las diferentes calificaciones para los tramos inspeccionados, que permitan luego generar varios reportes para documentar los resultados de la inspección, sin embargo se recomienda consultarla para aclarar de mejor manera los datos. La información extraída resulta suficiente para establecer unos requerimientos para el prototipo y a partir de ellos seleccionar el tipo de cinemática y componentes para la fases de diseño y construcción.

2.4. Enlaces de comunicación

Consisten en un conjunto de medios que permiten la comunicación a distancia entre equipos y que normalmente se hace con el fin de transmitir datos, audio o video por ondas electromagnéticas utilizando diferentes medios como el aire, cables de cobre, fibra óptica, etc. Existe una clasificación para las redes dependiendo de la tecnología de transmisión y la escala que tienen.

De forma general hay dos tipos de tecnología de transmisión, los enlaces de difusión y los enlaces punto a punto. Estos últimos se encargan de conectar pares individuales de equipos o máquinas, en estos casos simplemente se tiene un emisor y un receptor, y se conoce como unidifusión (unicasting). Al contrario, en las redes de difusión (broadcasting) todos los equipos de la red comparten un mismo canal de comunicaciones por lo que si una máquina envía un paquete, es recibido por todas las demás. Por ello es necesario que se especifique la dirección a la que va dirigida la información dentro del paquete.

Los medios de transmisión que utilizan pueden ser guiados como los medios magnéticos, cables de par trenzado, cables coaxiales o fibra óptica para conexiones alámbricas, o pueden ser inalámbricos utilizando el espectro electromagnético, microondas, transmisión infrarroja, por ondas de luz o por medios satelitales. También se pueden clasificar las redes con base en su escala o su distancia de operación, existen las redes de área personal, redes de área local y redes de área amplia, cada una con una escala mayor que la anterior y con una tecnología diferente. Las redes de área personal llamadas PAN permiten una comunicación entre dispositivos en el rango de una persona y generalmente utilizan tecnologías inalámbricas de corto RFID. Las redes de área local llamadas LAN son redes que permiten la comunicación dentro de un edificio o casa donde es común utilizar los puntos de acceso o AP, que se encargan de transmitir la información entre equipos conectados a la red. Las redes de área amplia o WAN son redes que abarcan una amplia extensión geográfica como ciudades o países las cuales utilizan tecnologías como LoRa empleada para conexiones a grandes distancias y para redes de IoT [12].

2.4.1. Comunicación inalámbrica

Las redes inalámbricas utilizan y estudian la propagación de las ondas de radio en el aire a distintas frecuencias, al ser en escencia ondas son susceptibles a muchos de los fenómenos que se presentan en la naturaleza (dependiendo del entorno por el que viajen) que influyen en su propagación como la reflexión, la abosrción, la difracción, entre otros. El principal inconveniente en este tipo de redes es la atenuación de la señal a medida que viaja, por un lado incluso en la mejor de las situaciones siempre se presenta una atenuación de la onda al viajar en el espacio libre lo cual hace que pierda fuerza limitando la distancia, por el otro las ondas de radio también se ven afectadas por la presencia de objetos que obstruyan o cambien su dirección lo cual las atenúa incluso más al presentarse fenómenos como la reflexión y la absorción. Otro de los inconvenientes que tienen las redes inalámbricas es la posibilidad de interferencias debido

a la existencia de otras redes en el mismo entorno y una desventaja de seguridad ya que casi cualquiera puede interceptar la señal transmitida o conectarse a la red si no se le añade ningún sistema de protección.

Los sistemas de comunicación inalámbricos se pueden separar en distintas categorías dependiendo de las tecnologías que utilicen. Por un lado se tienen los sistemas de localización fija los cuales admiten el uso de antenas estacionarias de alta ganancia y conexiones a la red eléctrica, dentro de ellos se encuentran sistemas punto a punto como por ejemplo las microondas y las comunicaciones ópticas con distancias alrededor de 1 metro. También existen los sistemas de área local los cuales proporcionan una comunicación por proximidad, cuentan con distancias más largas de varias decenas de metros, lo cual permite a los usuarios tener un rango de movilidad en escenarios más flexibles que los sistemas de localización fija, dentro se tienen por ejemplo las comunicaciones Bluetooth y WiFi. Por último se tienen los sistemas móviles de área amplia los cuales proporcionan rutas de transmisión mucho más largas de varios kilómetros, dando a los usuarios mucha más libertad en cuanto a movilidad, con equipos más pequeños y coberturas mayores, como por ejemplo los sistemas para comunicación celular o la popular tecnología de conexión LoRa [13].

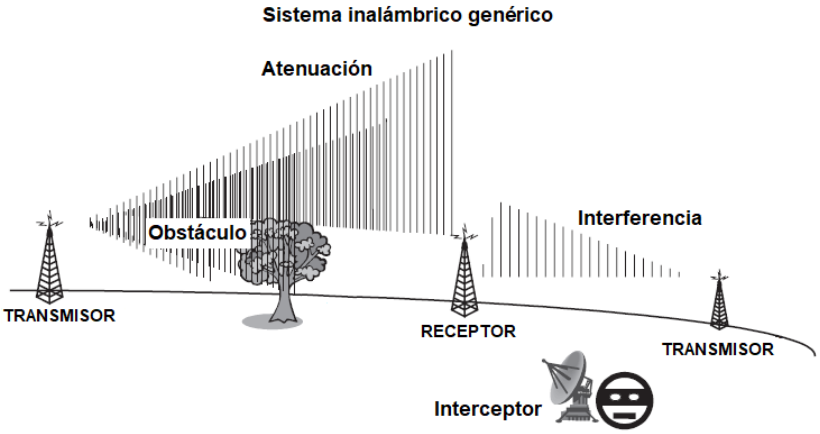


Figura 2.8: Esquema de un sistema de transmisión inalámbrico genérico [13].

Para realizar una elección sobre qué tipo de tecnología inalámbrica utilizar en el proyecto resulta necesario conocer más a profundidad las opciones que se tienen, comparando sus características. Tomando en cuenta las características mencionadas anteriormente, resulta claro que el interés está en las redes de área local debido a las distancias y a las tasas de transmisión. Dentro de una red de alcantarillado normalmente se encuentran tramos de tubería de varias decenas de metros y para una transmisión de video que cumpla con los requisitos mínimos de resolución de acuerdo a la norma, las tasas de transmisión del orden de los Megabits por segundo son bastante apropiadas, lo cual deja dos opciones de tecnología a utilizar en el proyecto: Bluetooth o WiFi.

Bluetooth

Bluetooth es una tecnología de comunicación inalámbrica enfocada en conectar dispositivos como celulares, computadores, cámaras, audífonos, entre otros. Es un tipo de red ad hoc, es decir que no utiliza ningún tipo de infraestructura para crear una red, los mismos dispositivos pueden crearla entre sí. Tiene una gran diversidad de aplicaciones como por ejemplo permitir la conexión de diferentes periféricos a un dispositivo, monitorizar sensores, dispositivos para domótica, o la transferencia de audio.

Por naturaleza este tipo de redes no son muy amplias y utilizan los términos de piconet y scatternet. Una piconet es una red de corto alcance que utiliza Bluetooth, capaz de tener hasta 8 estaciones donde una de ellas es conocida como primaria y las demás son conocidas como secundarias las cuales se sincronizan con la primaria, y en términos de arquitectura de comunicación admiten transferencias de datos de tipo uno a uno o uno a muchos desde la primaria a las secundarias. Por otro lado una scatternet es una combinación de varias piconet donde la estación primaria de una es a su vez una estación secundaria de otra piconet lo cual básicamente permite extender una red de dispositivos Bluetooth [14].

Las redes de Bluetooth implementan un protocolo llamado Personal Area Network (PAN) que define pequeñas redes inalámbricas en un área del tamaño de una o varias habitaciones. Este protocolo está dado por el estándar IEEE 802.15 donde se define como un sistema robusto y sencillo para la comunicación inalámbrica de voz y datos en un corto alcance. Un dispositivo con Bluetooth incorpora un transmisor de radio de corto alcance que opera en la banda de los 2.4 GHz, lo cual da la posibilidad de generar interferencias con redes que utilicen el protocolo IEEE 802.11 conocido también como WiFi. Actualmente existe la Bluetooth SIG, que es una comunidad global de más de 36000 empresas encargadas de crear los estándares de la tecnología Bluetooth, su logo se muestra en la Figura 2.9.



Figura 2.9: Logo de Bluetooth [15].

Dentro de su definición más reciente, cuenta con dos sistemas diferentes denominados Basic Rate (BR), Enhanced Data Rate (EDR) y Low Energy (LE), enfocados en distintos productos o equipos, pero en general es una tecnología de corto alcance que cuenta con un bajo consumo de energía y bajo costo. Independientemente del sistema que se utilice, un dispositivo que cuente con tecnología Bluetooth puede comunicarse con otro dispositivo que también la implemente y actualmente es empleado casi por la totalidad de equipos móviles ya que es una solución simple de conectividad inalámbrica.

En el sistema BR/EDR se tiene una transmisión inalámbrica de radio con baja potencia que es capaz de transmitir datos a través de hasta 79 canales en la banda de frecuencia de los 2.4 GHz. Emplea un transceptor de saltos de frecuencia con el cual se logra una robustez en la comunicación al combatir la interferencia y el desvanecimiento, en cuanto a la tasa de transferencia es capaz de transmitir entre 1 y 3 Mbps. Es compatible con una comunicación de tipo punto a punto y se utiliza principalmente para la transmisión inalámbrica de audio, convirtiéndose en el protocolo de radio estándar que incluyen los diferentes altavoces y audífonos, sin embargo también es útil para la transferencia de datos entre dispositivos [16].

En el caso del sistema LE también opera en la frecuencia de los 2.4 GHz a través de 40 canales y está diseñado para funcionar a muy baja potencia con tasas de transmisión de hasta 2 Mbps, esto brinda la posibilidad de fabricar dispositivos con funciones de conectividad específicas ya que admite también diferentes topologías de comunicación como los sistemas punto a punto y recientemente ha sido empleado en topologías de malla permitiendo la creación de redes de dispositivos a gran escala para controlar, monitorizar y automatizar sistemas con gran cantidad de dispositivos que necesiten comunicarse de forma segura entre ellos, también es utilizado para sistemas de localización por proximidad y posicionamiento [16].

Tomando en cuenta solo el tipo de sistemas BR/EDR ya que es el más común, en la Tabla 2.6 se listan las características de transmisión para un sistema de comunicación que utilice Bluetooth.

Tabla 2.6: Características Bluetooth BR/EDR [15].

Banda de frecuencia	2.4 GHz (2.402 – 2.480 GHz)
Canales	79 canales con espaciado de 1 MHz
Modulación	GFSK, $\pi/4$ DQPSK, 8DPSK
Tasas de transmisión	1, 2 y 3 Mbps
Potencia de transmisión	$\leq$ 100 mW (+20 dBm)
Sensibilidad de recepción	$\leq$ -70 dBm
Distancia de transmisión	10 metros

WiFi

WiFi es una tecnología de comunicación inalámbrica que utiliza el estándar IEEE 802.11 y permite a los usuarios utilizar y conectar una gran variedad de dispositivos generalmente para acceder a internet. En el 2002 surgió lo que se conoce hoy en día como WiFi Alliance cuyo logo se observa en la Figura 2.10, organización mundial conformada por múltiples empresas encargadas del desarrollo de tecnologías, requisitos y estándares para impulsar las tecnologías de red inalámbrica basadas en WiFi. WiFi es actualmente la tecnología más utilizada y el principal medio para el tráfico de internet en el mundo ya que se encuentra en múltiples dispositivos como celulares, computadores, televisores,

entre otros, continuamente se ha ido mejorando y cada vez ofrece mejores velocidades, menores latencias y en general una mejor experiencia para los usuarios que desean acceder a internet desde diferentes lugares con una variedad de dispositivos.



Figura 2.10: Logo de WiFi Alliance [15].

Es un tipo de red LAN compuesta de clientes o equipos móviles y de una infraestructura llamada punto de acceso o estaciones base que se encuentra ubicada en determinado lugar, a través de la cual se lleva a cabo la comunicación. Dentro de su arquitectura, este estándar dicta dos tipos de servicios: el conjunto de servicios básicos BSS y el conjunto de servicios extendidos ESS. El conjunto de servicios básicos BSS se define como el bloque de construcción de una red LAN inalámbrica, y tiene dos tipos: BSS independiente y BSS de infraestructura. En el primer caso la red está compuesta únicamente por estaciones inalámbricas fijas o móviles que se comunican entre ellas, mientras que en el segundo caso se cuenta con una estación base central conocida como punto de acceso o access point (AP). En el caso de que se utilice un access point para BSS se tiene una red independiente desde la cual no es posible enviar datos a otro BSS a menos que hagan parte de un conjunto de servicio extendido ESS. Por otro lado, los conjuntos de servicio extendido ESS son aquellos conformados por dos o más redes de infraestructura BSS que utilizan un punto de acceso. En este caso los BSS son conectados a través de un sistema de distribución utilizando una LAN cableada, que permite la conexión entre los puntos de acceso y los BSS. Algo importante de los conjuntos de servicio extendido es que utilizan dos tipos de estaciones: las móviles y las estacionarias. Las estaciones móviles son las típicas estaciones que hacen parte de un BSS mientras que las estacionarias son las estaciones de punto de acceso que hacen parte de la LAN cableada [14].

Una transmisión con WiFi hace uso del protocolo HTTP (Hypertext Transfer Protocol), el cual permite realizar peticiones de datos o recursos como páginas web en formato HTML. Este protocolo es la base del envío y recepción de información a través de la web y cuenta con un modelo de tipo cliente-servidor. Es un protocolo que pertenece a la capa de aplicación y se transmite sobre el protocolo TCP (Transmission Control Protocol) o sobre TLS (Transport Layer Security), al ser HTTP un protocolo ampliable se puede usar no solo para la transferencia de documentos de hipertexto (páginas HTML), funciona también para la transmisión de imágenes o videos, lo cual expande en muchos aspectos las aplicaciones que tiene.

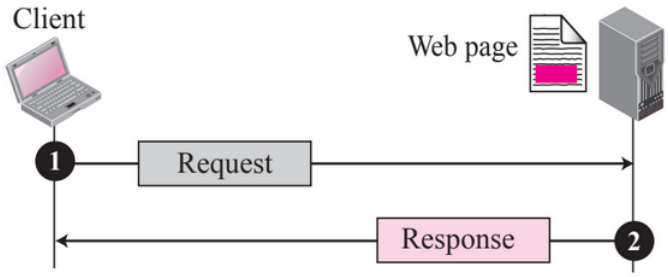


Figura 2.11: Modelo cliente-servidor utilizado en el protocolo HTTP [14].

Como se mencionó anteriormente, este protocolo se encuentra basado en el modelo cliente-servidor, donde un usuario o cliente realiza peticiones a un servidor el cual se encarga de gestionarla y enviar una respuesta como se muestra en la Figura 2.11. Del lado del cliente, por medio de cualquier herramienta o programa (en general se realiza por medio de un navegador web) es que se inician las peticiones y se reciben las respuestas. En el caso de aplicaciones o páginas web, el navegador solicita al servidor un archivo de tipo HTML, al recibirlo es procesado y si el archivo utiliza otros archivos para definir el estilo o la dinámica de la página (como archivos CSS, Javascript, imágenes o videos) realiza más peticiones al servidor para completar la página web con dichos archivos. De acá se tiene que una página web es básicamente un documento HTTP con diferentes enlaces en él, que permiten que el usuario realice más peticiones. En el lado del servidor se tiene un dispositivo capaz de responder con los datos que solicita el cliente.

Un cliente que desee acceder a una aplicación web necesita conocer el nombre del archivo y la dirección. Para facilitar el acceso a los documentos HTML, el protocolo HTTP utiliza localizadores, de allí surge el concepto de URL (Uniform Resource Locators), el cual es un estándar para especificar cualquier tipo de información en internet. Las URL definen cuatro cosas, el protocolo, el host, un puerto y una ruta. Tal como se observa en la Figura 2.12, el host es el nombre del dominio del servidor donde la información se encuentra almacenada, las URL también pueden contener un número de puerto del servidor para especificar qué tipo de servicio está prestando y por último la ruta representa dónde exactamente está la información dentro de la página web, ya que estas normalmente se encuentran estructuradas por medio de rutas [14].



Figura 2.12: Estructura de una URL [14].

Existen diferentes tipos de peticiones que puede hacer un cliente a un servidor para interactuar de una u otra forma con la información contenida en él. Dentro de los más

utilizados se encuentran los métodos GET para solicitar información del servidor y POST para enviar información al servidor, aunque existen algunos más como los que se muestran en la tabla de la Tabla 2.7.

Tabla 2.7: Métodos comunes de una petición en el protocolo HTTP [14].

Método	**Acción**
GET	Solicita un documento al servidor
POST	Envía cierta información del cliente al servidor
PUT	Crea o reemplaza un elemento especificado
DELETE	Elimina el elemento especificado

Por último en general las características de una conexión de WiFi dependen mucho de la generación del estándar IEEE 802.11 y de la fabricación propio dispositivo, pero en general se tienen las características que se listan en la Tabla 2.8 para una transmisión que utilice WiFi.

Tabla 2.8: Características generales WiFi [15].

Banda de frecuencia	2.4 GHz y 5 GHz
Canales	13 canales con espaciado de 22 MHz
Modulación	BPSK, QPSK, y QAM
Tasas de transmisión	Entre 2 Mbps y 2 Gbps
Potencia de transmisión	Entre +20 dBm y + 40 dBm
Sensibilidad de recepción	Entre -100 dBm y -70 dBm
Distancia de transmisión	Entre 10 y 100 metros

2.4.2. Selección de tecnología a implementar

Comparando las características y aplicaciones a las que están destinadas las tecnologías de Bluetooth y WiFi, resulta claro que es más apropiado utilizar una transmisión inalámbrica con WiFi en el proyecto. Esto debido a que principalmente Bluetooth cuenta con tasas de transferencia relativamente bajas que no serían de mucha utilidad para transmitir una gran cantidad de datos (como fotos y videos), además tal como se ha visto esta tecnología está más destinada a redes inalámbricas de corto alcance para un tipo de aplicación específico por lo que en términos de distancia no sería de mucha utilidad para el prototipo, donde es necesario recorrer varias decenas de metros. Del mismo modo al utilizar WiFi se tiene la ventaja de contar con el protocolo HTTP que permite realizar una transferencia de información, con el cual se puede implementar una aplicación de tipo web, para las cuales existe una amplia variedad de frameworks de desarrollo, a través de la cual se realice todo el proceso de transmisión de video, fotografías y en general el control del vehículo de forma inalámbrica sobre una red WiFi.

3. Diseño del prototipo móvil

Dentro de este capítulo se inicia definiendo los requerimientos de diseño con base en la información del capítulo anterior. Luego se identifican algunas plataformas y dispositivos comerciales que existen para realizar inspecciones en las redes de alcantarillado, para con ello tener una idea sobre los posibles tipos de locomoción, cinemáticas y funcionalidades que se pueden implementar en el prototipo que permitan satisfacer los requerimientos planteados. Una vez definida la estructura y funcionalidad del prototipo se realizan los análisis cinemáticos para la movilidad vehículo, se realiza el diseño de circuitos para el funcionamiento y por último una selección de materiales y piezas para la construcción del prototipo.

3.1. Requerimientos de diseño

Partiendo de lo expuesto en el capítulo anterior, se identifican los requerimientos que se desean satisfacer con el proyecto:

- Implementar un pequeño vehículo capaz de desplazarse en el interior de tuberías de máximo 16" (40 cm) de diámetro.
- Equipar el vehículo con una cámara para transmitir vídeo de forma inalámbrica en tiempo real por medio de WiFi.
- Incluir en el vehículo un mecanismo que permita realizar paneos y giros con la cámara para observar el estado interno de la tubería.
- Incluir en el vehículo un sistema de iluminación LED para mejorar la visibilidad en condiciones de oscuridad.
- Implementar un sistema que permita la grabación de video en formato MP4 con resolución mínima de 640 x 480.
- Implementar un sistema que permita la toma de fotografías en formato JPEG con resolución mínima de 640 x 480.
- Incluir un mecanismo para controlar las funcionalidades del vehículo de forma inalámbrica utilizando WiFi.

3.2. Dispositivos y equipos comerciales

En los último años han aparecido empresas fabricantes de diferentes sistemas de inspección de alcantarillado. Estos sistemas existentes cuentan con una gran variedad dependiendo de los materiales, funcionalidades, tamaños y en general las características con las que cuentan. Algunos de esos sistemas se listan a continuación para tener un panorama general detallando las características que tienen.

3.2.1. iPEK ROVION

Fabricado por la empresa PANATEC, el iPEK ROVION es un sistema modular y escalable para inspección robotizada de tuberías de saneamiento, alcantarillado y tuberías industriales. Es un robot tractor multidireccional, con cabrestantes con 200, 300 y 500 m de cable motorizados y automáticos, telecomandados por una unidad de control con pantalla táctil y software de generación de informes Wincan integrado. El sistema iPEK ROVION cuenta con algunas características únicas en este tipo de sistemas como es disponer de hasta dos cámaras de visión trasera, cable umbilical ultrarresistente de 6 hilos, tracción mediante 4 ruedas motrices y unidad de control inteligente, programable y con capacidad de realizar mediciones de nivel, tamaño de fisuras, diámetro de acometidas y multitud de otras funciones digitales de medida [17].



Figura 3.1: Robot de inspección iPEK ROVION [17].

Entre sus características principales se tiene:

- Diseño modular escalable.
- Diferentes configuraciones para cubrir un amplio rango de secciones.
- Cámara oscilogiratoria con zoom (opcional). 10x óptico u 4x digital.
- Medición de distancia e inclinación.
- Sonda transmisora para localización en superficie (512 Hz/33 kHz).
- Construcción en acero inoxidable y aluminio.
- Juego completo de ruedas adaptables a diferentes diámetros de inspección.

- Iluminación mediante diodos LED.
- Sumergible hasta 10 m de columna de agua.

La característica más llamativa de este robot de inspección es que cuenta con una consola de control con pantalla táctil de 10,1" retro iluminada con tratamiento antirreflejos, apta para uso en exteriores y resolución 1280 x 800 px. Memoria interna de 32 GB. Dimensiones: 300 x 267 x 70 mm. Peso: 1,8 kg. Permite el control de movimientos del carro y la cámara. Control de nivel y potencia de iluminación. Selección entre cámara frontal y trasera. Representación en pantalla de valores de presión del carro, cámara, elevador y luces auxiliares. Representación en pantalla de valores de temperatura de funcionamiento del carro tractor y cámara. Indicación permanente de la inclinación transversal del carro, horizonte virtual y alarma antivuelco. Indicación permanente de la pendiente de la conducción y distancia recorrida. Registro de imágenes y vídeos. Medición de defectos y fisuras [17].

3.2.2. X5-HT Crawler

La empresa Easy-Sight cuenta con robots de inspección de CCTV, herramientas efectivas y precisas para acceder a la tubería. Juegan un papel muy importante en la industria de alcantarillado y cuentan con tecnología de punta, los robots de inspección de CCTV Easy-Sight pueden proporcionar una inspección de la mejor calidad con video e imágenes HD. El robot de inspección de CCTV de tuberías X5-HT es un robot de inspección de CCTV de alta definición, que puede ser controlado por una computadora portátil personal o un controlador dedicado. La computadora portátil o el controlador dedicado se conecta a otras partes del sistema por cable, muestra y almacena videos de detección de alta definición en tiempo real [18].



Figura 3.2: Robot de inspección X5-HT Crawler [18].

Dentro de sus características principales tiene:

- Sumergible hasta 10 metros de profundidad.
- Motores DC de 90W con velocidad máxima de 32 m/min.
- Fuente de luz LED delantera y trasera de 10W.
- Fabricado en Latón niquelado y acero inoxidable.
- Incluye sensores de presión, sensor de inclinación, sensor de balanceo y cabeceo.
- Transmisión por cable con carretes de hasta 150 metros.
- Temperatura de trabajo entre -10 y 50° C.
- Peso de 16 Kg.
- Dimensiones de 480x177x182 mm.

Incluye una estructura móvil conducida por motor, con un carrete de cable móvil con una longitud de 150 metros,con cables retráctiles automáticos o manuales resistentes al agua, al desgaste y la corrosión, además cuenta con un contador de distancia con precisión de 1 mm. Para controlar el equipo es necesario instalar un software de computador para el informe y evaluación, con el cual se puede controlar y operar el robot, realizar los paneos, iniciar la grabación y completar los registros de inspección por medio de capturas de pantalla. La cámara cuenta con una resolución de 2592x1520 con zoom óptico x10 y sistema de rotación radial de 360 ° [18].

3.2.3. ANATROLLER ARI-10

Fue hecho para todas las necesidades y desafíos de la limpieza e inspección de tuberías en redes de alcantarillado residenciales. Fabricado por la empresa Robotics Design Inc, la cual realizó amplios estudios en el mercado de la limpieza de conductos para determinar las necesidades del mercado. El ANATROLLER ARI-10 es muy maniobrable, de pequeño tamaño y está diseñado para desplegarse fácilmente de forma rápida y trabajar tanto en conductos verticales como horizontales. Un sistema de movimiento magnético permite al robot descender y ascender en conductos metálicos de forma vertical. Cabe en el interior de los conductos más pequeños y puede atravesar fácilmente espacios reducidos y giros bruscos gracias a sus orugas de gran agarre. Este robot está especializado en la inspección profesional gracias a su avanzado sistema de iluminación y a la selección de cámaras de última generación que envían vídeo en tiempo real a un tablet PC incluido. Sólo pesa 5 kg y puede funcionar con batería, lo que lo hace perfecto para las inspecciones previas y posteriores al trabajo. El robot está conectado con un cable de acero inoxidable a través del cual se realiza la transmisión de video, lo que lo hace resistente al desgaste y fácil de limpiar [19].

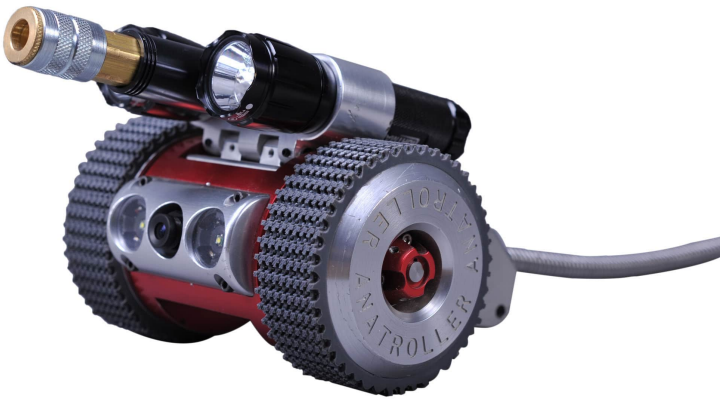


Figura 3.3: Robot de inspección ANATROLLER ARI-10 [19].

Dentro de sus características se tiene:

- Construcción modular que permite adicionar o retirar componentes y accesorios.
- Estando full equipado pesa tan solo 5 Kg.
- Cuenta con carrete de cable de hasta 30 metros.
- Resistente a entornos industriales donde haya aceite, polvo o agua.
- Trabaja en tuberías verticales y horizontales.
- Control fácil de manejar.
- Doble cámara a 1080p full HD con 360°.
- Sistema de iluminación antireflejo.
- Capacidad de incluir sistemas de audio.
- Medidas de 18x18x13cm.
- Velocidad de 30 cm/seg.

Su robusta construcción de aluminio y su cable lavable de acero inoxidable de alta resistencia garantizan que su trabajo continúe sin interrupciones durante muchos años y le permite garantizar la máxima higiene después de finalizar el trabajo. El control del robot es muy fácil de usar, lo que permite a los contratistas que utilicen el ARI-10 limpiar con facilidad sin necesidad de formación especializada. El sistema de control solo tiene un único joystick para el movimiento y perillas para controlar la velocidad y la iluminación del robot [19].

3.3. Prototipo

Resulta necesario discutir y tomar decisión acerca de la locomoción y las funcionalidades a implementar en el prototipo para cumplir con los requerimientos. Para ello se destacan tres secciones: movilidad donde se se decide sobre la locomoción y topología con la que se va a desplazar el prototipo, manipulador donde se define de que forma se apunta la cámara, y comunicación inalámbrica en las cuales se indica la forma de conexión, manejo del prototipo y transmisión de imágenes y video a distancia.

3.3.1. Movilidad

Los robots pueden ser clasificados de acuerdo al mecanismo de interacción y al entorno en el que operan. Dentro del tipo de robots móviles se destacan tres entornos principales ya que requieren de un diseño específico para su desplazamiento: acuático, terrestre y aéreo. Para cada entorno puede haber igualmente diferentes clasificaciones con base en el tipo de locomoción, por ejemplo los robots terrestres pueden tener patas o ruedas, los robots aéreos pueden tener alas fijas o giratorias y los robots acuáticos pueden tener pequeñas turbinas. En los equipos comerciales vistos anteriormente se puede apreciar que son en su mayoría robots móviles de tipo terrestre con ruedas para desplazarse por el entorno.

La locomoción con ruedas es de las más utilizadas en los robots terrestres ya que logran una buena eficiencia en el movimiento y su implementación mecánica es sencilla. Existen varios tipos de ruedas (a parte de las convencionales) utilizadas en los robots móviles como las ruedas de castor, las ruedas suecas o las ruedas esféricas, la elección depende del tipo de arreglo y el movimiento que se quiera lograr. En la Figura 3.4 se observan algunas de las configuraciones que se tienen para los robots móviles.

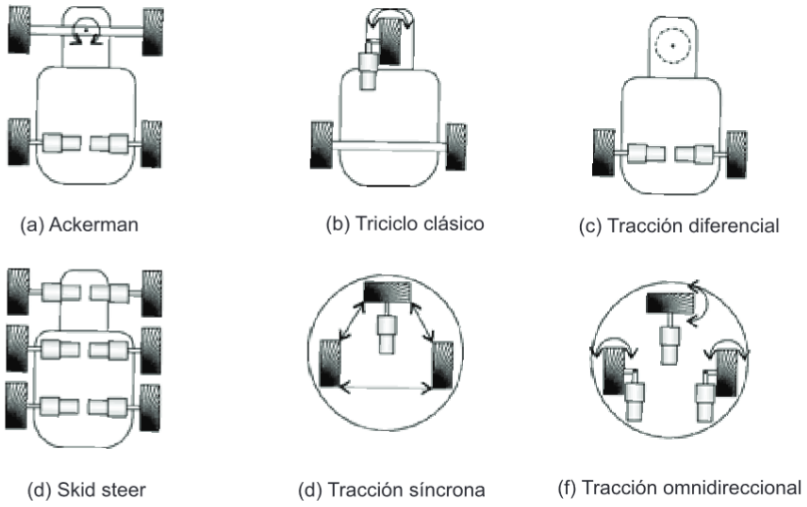


Figura 3.4: Configuraciones para los robots móviles [20].

Para los dispositivos comerciales encargados de la inspección de tuberías, se observa que tienen arreglos de 4 y 2 ruedas sin dirección, es decir que las ruedas no giran para orientar al vehículo hacia determinado lugar a medida que avanzan. Este tipo de vehículos cuentan con lo que se conoce como cinemática diferencial, donde las ruedas de cada lado giran de forma simultánea y dependiendo de la velocidad de giro es posible controlar la dirección del vehículo. Esto es de gran ayuda porque permite realizar giros sobre el propio eje del vehículo, lo cual en lugares estrechos como las tuberías resulta de mucha utilidad para maniobrar con el vehículo.

El número de ruedas también es un aspecto importante en la estabilidad del vehículo, entre más ruedas se tengan se tienen más puntos de apoyo o contacto con el terreno lo que ayuda en el balance para que el vehículo no vuelque. Por estos motivos se plantea entonces utilizar una topología con 4 ruedas sin dirección que utilice la cinemática diferencial para controlar la dirección, esta configuración es conocida como Skid Steer debido a que es implementada en los vehículos de carga utilizados en las construcciones como el que se observa en la Figura 3.5.



Figura 3.5: Vehículo de carga - Skid Steer [21].

3.3.2. Manipulador

Existen otro tipo de robots conocidos como robots fijos, la mayoría son máquinas de tipo industrial que trabajan en entornos adaptados para ellas y que realizan tareas de forma repetitiva como soldaduras, pinturas o ensamblado de partes para ciertos productos. Los brazos articulados, también conocidos como manipuladores, están conformados por enlaces y articulaciones donde cada articulación se encarga de mover un enlace. Las articulaciones van desde la base del robot, la cual es fija, hasta el extremo final conocido como efector. Asumiendo que cada articulación tiene un único grado de libertad, se tienen dos movimientos para las articulaciones los cuales pueden ser rotacionales o translacionales. Una articulación rotacional permite una rotación relativa entre dos articulaciones mientras que una articulación translacional, también conocida como prismática, permite un desplazamiento relativo lineal entre dos enlaces. En la Figura 3.6 se muestran las partes de un robot manipulador de la marca PUMA.

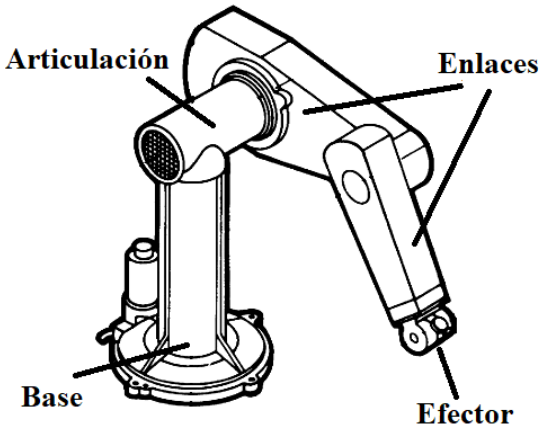


Figura 3.6: Estructura de un robot manipulador PUMA 560.

Dentro de los dispositivos comerciales se observa que utilizan estructuras para ajustar la altura de la cámara utilizando una articulación rotacional, mientras que la cámara (ubicada en el efector) es capaz de rotar en varias direcciones utilizando una configuración conocida como muñeca de rotación esférica. Para la movilidad de la cámara y poder realizar los paneos que permitan visualizar puntos específicos dentro de la tubería, se plantea entonces utilizar un brazo articulado en configuración de codo con tres grados de libertad en cuyo efector se ubica la cámara. La configuración de codo cuenta con articulaciones rotacionales y guarda una gran similitud con las extremidades superiores humanas (por ello se denomina de codo), esto permite apuntar la cámara a cualquier punto de un espacio tridimensional dentro del rango de acción del brazo que en este caso se ría una esfera alrededor del brazo para el caso general donde las articulaciones giren 360 grados. Se elige utilizar un robot manipulador en configuración de codo como el que se observa en la Figura 3.7, ya que es fácil de implementar con materiales de bajo costo.

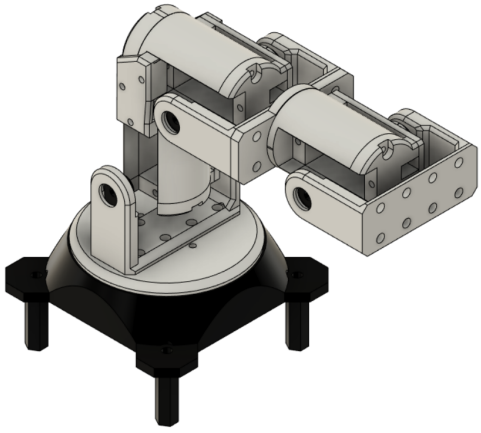


Figura 3.7: Modelo 3D de un manipulador de codo.

3.3.3. Comunicación inalámbrica

Para la comunicación inalámbrica que permita realizar la transmisión de video, así como controlar al vehículo, se opta por utilizar WiFi ya que es de las más utilizadas y permite realizar una comunicación de tipo full dúplex, además es fácil de implementar brindando un alcance medio y tasas de transferencia relativamente altas lo cual resulta apropiado para validar el funcionamiento principal del prototipo. Dado que la inspección se realiza in situ, no es necesario realizar una configuración que permita tener acceso a internet, basta con crear una red WLAN por medio de un router, al cual se van a conectar todos los dispositivos y sobre esta red se realiza el envío y la recepción de información entre el prototipo y un equipo de cómputo que permite controlarlo. Ya que se puede utilizar el protocolo HTTP para la transferencia de información sobre la red WiFi, la idea es implementar una aplicación web que se aloje en el prototipo y a la cual se puede acceder desde el equipo de cómputo conectado a la red. Esto quiere decir que el prototipo va a actuar como un servidor, ya que va a estar capturando y almacenando los datos, mientras que el equipo de cómputo va a ser un cliente que realiza solicitudes al servidor para tener acceso a los datos capturados.

El prototipo debe estar basado en un dispositivo que permita montar un servidor para la aplicación web, además de permitir implementar todas las demás funcionalidades como la cámara, generar las señales de control para los motores y conexión WiFi. Para ello se elige utilizar una Raspberry Pi, la cual es un computador funcional con un formato pequeño y de bajo costo, que funciona tanto para navegar en internet, jugar o para escribir programas que permitan crear circuitos y dispositivos físicos.

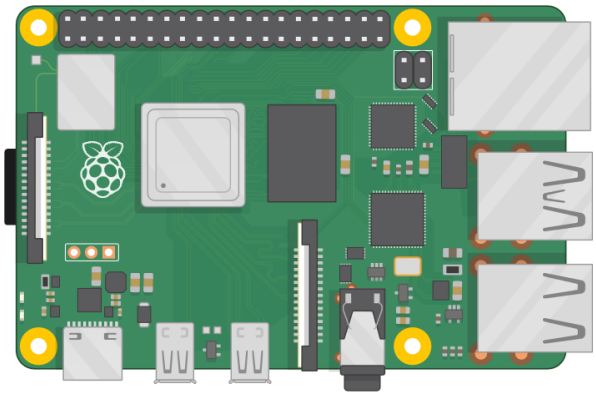


Figura 3.8: Raspberry Pi 4 Model B [22].

Debido a su estructura se le conoce como un computador de una sola placa, ya que es capaz de prestar los mismos servicios que un computador normal pero está construido todo sobre una única placa de circuito impreso. Existen diferentes modelos de Raspberry Pi, cada uno con características específicas para un uso particular, pero una de las versiones más populares y más potentes es la denominada Raspberry Pi

4 Model B que se muestra en la Figura 3.8. Dentro de los componentes que tiene la Raspberry Pi se tiene en primer lugar el SoC (System on a Chip), el cual es un circuito integrado que incluye la unidad central de procesamiento CPU y la unidad de procesamiento gráfico GPU, también al igual que los computadores convencionales cuenta con una memoria RAM y como disco duro utiliza una memoria microSD. Cuenta también con un módulo de radio que permite a la Raspberry Pi comunicarse de forma inalámbrica a través de WiFi o Bluetooth y a nivel de puertos tiene 4 puertos USB (dos 2.0 y dos 3.0), un puerto Ethernet para conexión a red cableada, un puerto AV de 3.5 mm para audio y video, dos puertos micro HDMI para conexiones multimedia de alta definición, un puerto de alimentación USB tipo C, un puerto o conector de pantalla DSI diseñado específicamente para conectar una pantalla de Raspberry Pi táctil y por último un puerto o conector de interfaz serie para cámara CSI.

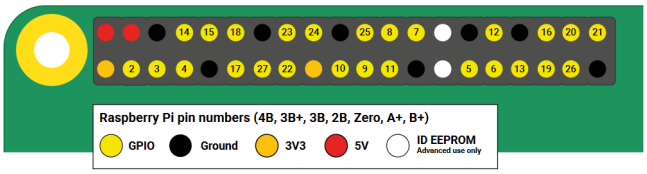


Figura 3.9: Regleta de pines GPIO de la Raspberry Pi 4 Model B [22].

Uno de los principales conectores que trae, es una regleta de 40 pines metálicos divididos en dos filas de 20 pines cada una los cuales conforman el sistema GPIO (General Purpose Input/Output) como se observa en la Figura 3.9, estos se pueden utilizar para comunicarse con hardware o componentes electrónicos como LEDs, pulsadores, sensores, etc. Esto brinda las mismas posibilidades de uso que un microcontrolador o sistema embebido normal como Arduino, ESP32 o PSoC pero con todas las ventajas que ofrece un computador ya que permite implementar programas más complejos que hagan uso de los GPIO, como por ejemplo servidores que alojen una aplicación web que interactúe con componentes físicos. Con esto la red WiFi queda planteada tal como se muestra en la Figura 3.10.

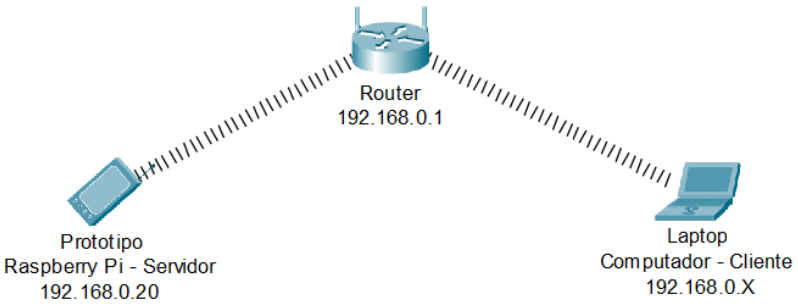


Figura 3.10: Diagrama de conexión inalámbrica.

Ya que el cliente debe saber siempre a dónde debe conectarse para acceder a la aplicación web, la Raspberry Pi debe tener una dirección IPv4 estática que le sea asignada cada vez que se conecte al router, por ello es necesario realizar dicha configuración la cual se detalla en el siguiente capítulo.

Para el desarrollo de la aplicación web existen diversos frameworks de desarrollo basados en distintos lenguajes de programación, en este caso se elige uno desarrollado en Python conocido como Django, ya que es fácil de implementar sobre la Raspberry Pi la cual tiene un sistema operativo basado en Linux que trae Python instalado por defecto. Django es un framework web de alto nivel escrito en Python, el cual se basa en MVC (modelo-vista-controlador) que es un patrón de diseño de software cuyo objetivo es separar la lógica del programa y la visualización del mismo. Permite desarrollar sitios web seguros y mantenibles de forma rápida ya que se encarga de gran parte de las molestias que se pueden encontrar dentro del desarrollo web, por lo que los desarrolladores pueden centrarse en escribir el código de la aplicación. Por medio de la aplicación web es posible utilizar el protocolo HTTP para realizar la transferencia de información entre el cliente y el servidor por medio de peticiones HTTP. Una aplicación web desarrollada a partir de Django, al basarse en el modelo MVC, se divide en varios archivos encargados de una parte específica:

- Views: Son las encargadas de encapsular la lógica responsable de recibir las peticiones HTTP que realiza un cliente y generar respuestas de tipo HTTP. Las vistas pueden acceder a los datos necesarios contenidos en una base de datos (la cual es opcional, dependiendo del objetivo de la aplicación) por medio de los modelos y definir la forma en la que se muestran en los templates.

- Models: Son objetos de Python que permiten definir y gestionar la estructura de datos de la aplicación, por lo cual permiten crear, leer, actualizar y eliminar registros de la base de datos que se utilice.

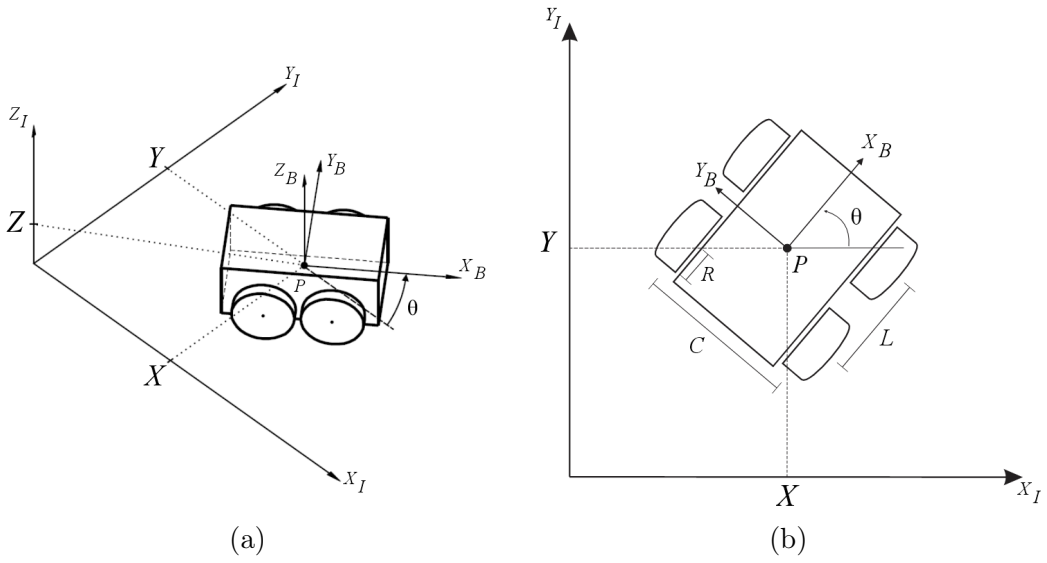
- Templates: Son archivos basados en HTML que definen la estructura visual de la aplicación, es sobre ellos que se renderiza la información entregada por las vistas. Al ser archivos HTML admiten el uso de CSS y Javascript, lo cual permite agregar estilos y hacer la página más dinámica.

Todo esto funciona en conjunto gracias a un mapeador de URLs encargado de redirigir las peticiones HTTP a la vista apropiada en base a la URL a la que se realice la petición, específicamente en este archivo se registran las rutas con la respectiva lógica de la aplicación web. Django además incluye muchas más herramientas que facilitan el desarrollo y que pueden ser consultadas y profundizadas en su documentación oficial [23]. En el siguiente capítulo se detalla el desarrollo y funcionalidad de la aplicación web implementada en el proyecto.

3.4. Análisis cinemático

3.4.1. Navegación del vehículo

Para la navegación del vehículo en el espacio es necesario utilizar marcos de referencia que permitan obtener información del movimiento del vehículo como su posición y orientación respecto a un punto fijo en el espacio. Se define entonces un marco de coordenadas inercial (X_I, Y_I, Z_I) y a partir de su origen es que se obtiene la información del movimiento del vehículo. Dentro del marco de coordenadas inercial se ubica el vehículo sobre el cual se define un marco de coordenadas locales (X_B, Y_B, Z_B) cuyo origen se ubica en el centro de gravedad del vehículo como se observa en la Figura 3.11a.



(a) (b)

Figura 3.11: Diagramas del vehículo sobre el marco de referencia inercial.

Como el tipo de locomoción del vehículo es terrestre se considera que el punto Z es constante dentro del marco de referencia inercial y el análisis de la cinemática del vehículo se ve reducido a los ejes X, Y tanto del marco inercial como del marco local como se muestra en la Figura 3.11b.

Se supone que el vehículo se mueve en el plano con una velocidad lineal definida dentro del marco de referencia local como $v = [\ v_x\ 0\ 0\ ]^T$ (únicamente tiene componente en x ya que el vehículo se mueve solo sobre el eje X_B) y rota con una velocidad angular definida como $\omega = [\ 0\ 0\ \omega_z\ ]^T$. Si se define también el vector $q = [\ X\ Y\ \theta\ ]^T$ que describe las coordenadas del vehículo sobre el marco de referencia inercial, donde la posición es X, Y y la orientación es θ, entonces se obtiene el vector de velocidades sobre el marco de referencia inercial definido como $\dot{q} = [\ \dot{X}\ \dot{Y}\ \dot{\theta}\ ]^T$ [24].

Como el interés está en obtener la información de las velocidades del vehículo sobre el marco de referencia inercial, es decir obtener $\dot{q}$, se utiliza una matriz de rotación

que permita pasar las velocidades del marco de referencia local al marco de referencia inercial. En el caso de las velocidades lineales v se tiene

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} v_x \\ 0 \end{bmatrix}, \tag{3.1}$$

para el caso de la velocidad angular ω, la equivalencia es directa y se tiene que

$$\dot{\theta} = \omega_z, \tag{3.2}$$

por lo tanto la expresión resultante para las velocidades en el marco de referencia inercial a partir de las velocidades en el marco de referencia local es

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ 0 \\ \omega_z \end{bmatrix} \tag{3.3}$$

Ahora tomando en cuenta que el vehículo es de 4 ruedas con motores independientes para cada una, es necesario encontrar una relación que permita obtener las velocidades lineal v_x y angular ω_z del vehículo a partir de las velocidades angulares de cada rueda, las cuales son físicamente medibles utilizando encoders, para así completar el modelo cinemático del vehículo.

Con los encoders se obtiene una medida de la velocidad angular a la que gira la rueda definida como ω_i donde $i = 1, 2, 3, 4$, de acuerdo a esto se tiene la expresión

$$v_{ix} = R\omega_i, \tag{3.4}$$

donde v_{ix} es la velocidad lineal de la rueda y R es el radio de la rueda.

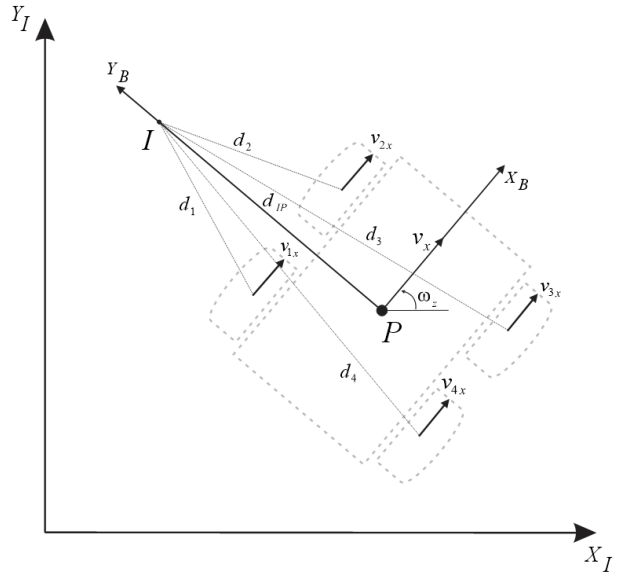


Figura 3.12: Diagrama con las velocidades lineales de las llantas.

En la Figura 3.12 se muestran los vectores de radio de giro $d_i = [\ d_{ix}\ d_{iy}\ ]^T$ y los vectores de velocidad lineal v_{ix} para cada rueda, también se tiene el vector de radio de giro del vehículo $d_{IP} = [\ 0\ d_{IPy}\ ]^T$ respecto al punto I que representa un centro instantáneo de rotación. De la geometría planteada se puede deducir la siguiente expresión

$$\omega_z = \frac{v_x}{d_{IP}} = \frac{v_{ix}}{d_{iy}}, \tag{3.5}$$

del mismo modo se obtienen las siguientes relaciones para las componentes d_{iy}

$$\begin{aligned} d_{1y} = d_{2y} = d_{IP} - C/2 \\ d_{3y} = d_{4y} = d_{IP} + C/2 \end{aligned}, \tag{3.6}$$

donde C es la distancia entre las ruedas mostrada en la Figura 3.11b. Combinando (3.5) y (3.6) se obtienen las siguientes relaciones entre las velocidades de las ruedas

$$\begin{aligned} v_L = v_{1x} = v_{2x} \\ v_R = v_{3x} = v_{4x} \end{aligned}, \tag{3.7}$$

donde v_L y v_R representan las velocidades lineales de las ruedas izquierdas y derechas respectivamente. Reemplazando los valores de (3.5) con los valores obtenidos en (3.6) y (3.7) se obtienen las siguientes expresiones

$$\begin{aligned} \frac{v_x}{d_{IP}} = \frac{v_L}{d_{IP} - C/2} \\ \frac{v_x}{d_{IP}} = \frac{v_R}{d_{IP} + C/2} \end{aligned}, \tag{3.8}$$

despejando v_L y v_R

$$\begin{aligned} v_L = \frac{v_x d_{IP}}{d_{IP}} - \frac{v_x C/2}{d_{IP}} \\ v_R = \frac{v_x d_{IP}}{d_{IP}} + \frac{v_x C/2}{d_{IP}} \end{aligned}, \tag{3.9}$$

y tomando en cuenta que $\omega_z = v_x/d_{IP}$, se obtiene la siguiente transformación que describe la relación entre las velocidades de las ruedas y las velocidades del vehículo

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1 & -C/2 \\ 1 & C/2 \end{bmatrix} \begin{bmatrix} v_x \\ \omega_z \end{bmatrix}, \tag{3.10}$$

ahora de acuerdo a (3.4), las velocidades angulares de las ruedas ω_L y ω_R son

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{R} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \tag{3.11}$$

finalmente con (3.10) y (3.11) se obtiene la relación aproximada entre las velocidades angulares de las ruedas y las velocidades del vehículo como

$$\begin{bmatrix} v_x \\ \omega_z \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/C & -R/C \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{3.12}$$

esta respuesta es la misma que se tiene con un vehículo que implemente una cinemática diferencial, por lo que la cinemática inversa está dada por

$$\begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} 1/R & C/2R \\ 1/C & -C/2R \end{bmatrix} \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} \tag{3.13}$$

Con las expresiones (3.3) y (3.12) es posible obtener las velocidades a las que se desplaza el vehículo sobre el marco de referencia inercial, a partir de las velocidades angulares a las que giren las ruedas, las cuales se pueden controlar. Algo interesante es que a partir de esas velocidades sobre el marco inercial, es posible obtener información sobre la posición del vehículo en dicho marco, si se realiza la integral de la velocidad.

3.4.2. Manipulador para la cámara

Los manipuladores están conformados por n enlaces y n articulaciones, de acuerdo a la configuración de codo elegida en la sección anterior se tiene el diagrama de la Figura 3.13, el cual tiene 3 enlaces cuyas longitudes son L_1, L_2 y L_3, 3 articulaciones ubicadas en los puntos o_0, o_1, o_2 y el efector ubicado en el punto o_3.

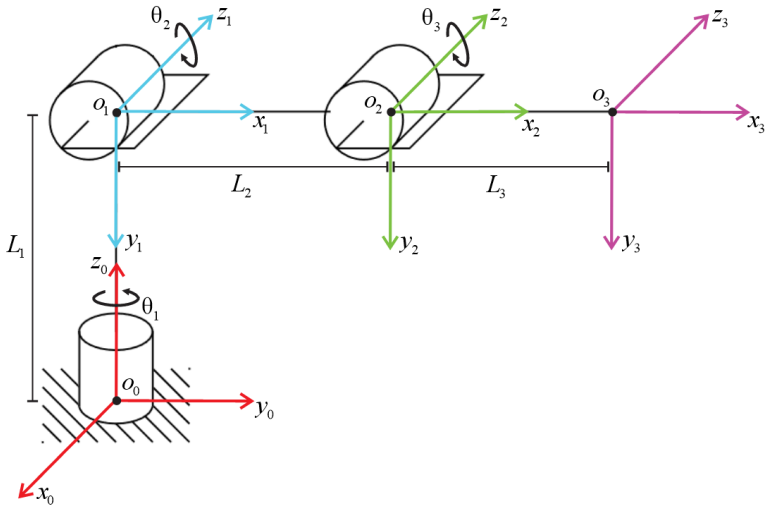


Figura 3.13: Diagrama del manipulador con los marcos de referencia asociados.

Para el análisis cinemático se asocian marcos de referencia sobre cada articulación que une los enlaces partiendo desde la base y utilizando transformaciones homogéneas se establece una relación entre ellos. La asignación de los marcos de referencia no es arbitraria, para ello se utiliza la convención de Denavit y Hartenberg que proporciona una forma sistemática de escoger dichos marcos, los cuales quedan ubicados como se observa en la Figura 3.13. Cada articulación tiene asociada una variable articular definida como q_i

$$q_i = \begin{cases} \theta_i \Rightarrow \text{si es rotacional} \\ d_i \Rightarrow \text{si es prismática} \end{cases}, \tag{3.14}$$

por lo que para el caso solo se tienen variables rotacionales que detallan los sentidos de giro de cada articulación, definidas como θ_1, θ_2 y θ_3. En esta convención cada transformación homogénea denominada A_i es representada como el producto de 4 transformaciones básicas de rotación y translación (3.15), esta matriz expresa la posición y orientación del marco de referencia i con respecto al marco de referencia $i-1$ [25].

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}, \tag{3.15}$$

donde

$$Rot_{z,\theta_i} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 & 0 \\ sin(\theta_i) & cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.16}$$

$$Trans_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.17}$$

$$Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.18}$$

$$Rot_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha_i) & -sin(\alpha_i) & 0 \\ 0 & sin(\alpha_i) & cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.19}$$

entonces la matriz de transformación homogénea, conformada por una matriz de rotación R y un vector de translación d, queda definida como

$$A_i = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

De acuerdo a la configuración de codo, en la Tabla 3.1 se muestran los parámetros obtenidos con la convención de Denavit y Hartenberg que se obtienen para a_i, α_i, d_i y θ_i, los cuales se definen como longitud del enlace, torsión del enlace, offset del enlace y ángulo de la articulación respectivamente.

Tabla 3.1: Parámetros D-H para un manipulador de codo.

Enlace	a_i	α_i	d_i	θ_i
1	0	-90	L_1	θ_1^*
2	L_2	0	0	θ_2^*
3	L_3	0	0	θ_3^*

Con estos parámetros obtenidos a partir del modelo de la Figura 3.13, es posible construir las matrices homogéneas A_i y con ello la matriz de transformación que permite obtener las variables de la posición y orientación del marco de referencia del efector respecto al marco de referencia de la base.

$$A_1 = \begin{bmatrix} cos(\theta_1) & 0 & -sin(\theta_1) & 0 \\ sin(\theta_1) & 0 & cos(\theta_1) & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

$$A_2 = \begin{bmatrix} cos(\theta_2) & -sin(\theta_2) & 0 & L_2cos(\theta_2) \\ sin(\theta_2) & cos(\theta_2) & 0 & L_2sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

$$A_3 = \begin{bmatrix} cos(\theta_3) & -sin(\theta_3) & 0 & L_3cos(\theta_3) \\ sin(\theta_3) & cos(\theta_3) & 0 & L_3sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

La matriz de transformación, definida como

$$T = A_1A_2A_3, \tag{3.24}$$

resulta de la forma

$$T = \begin{bmatrix} c_{\theta_2+\theta_3}c_{\theta_1} & -s_{\theta_2+\theta_3}c_{\theta_1} & -s_{\theta_1} & c_{\theta_1}(L_3c_{\theta_2+\theta_3} + L_2c_{\theta_2}) \\ c_{\theta_2+\theta_3}s_{\theta_1} & -s_{\theta_2+\theta_3}s_{\theta_1} & c_{\theta_1} & s_{\theta_1}(L_3c_{\theta_2+\theta_3} + L_2c_{\theta_2}) \\ -s_{\theta_2+\theta_3} & -c_{\theta_2+\theta_3} & 0 & L_1 - L_3s_{\theta_2+\theta_3} - L_2s_{\theta_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

De esta matriz se tiene que la cinemática directa para un manipulador de codo, que hace referencia a la posición del efector en el marco de referencia de la base a partir de los ángulos de rotación y las longitudes de los enlaces, es

$$\begin{aligned} x &= c_{\theta_1}(L_3 c_{\theta_2+\theta_3} + L_2 c_{\theta_2}) \\ y &= s_{\theta_1}(L_3 c_{\theta_2+\theta_3} + L_2 c_{\theta_2}) \\ z &= L_1 - L_3 s_{\theta_2+\theta_3} - L_2 s_{\theta_2} \end{aligned} \tag{3.26}$$

Obtener la cinemática inversa es un poco más complejo y para ello existen diferentes métodos, se puede realizar directamente manipulando la matriz de transformación obtenida en la cinemática directa a partir de la convención de Denavit y Hartenberg o realizando una aproximación geométrica utilizando relaciones trigonométricas. El método más sencillo para manipuladores con pocos grados de libertad es el geométrico donde se utilizan relaciones geométricas para obtener la posición del efector en función de las variables de cada articulación. Iniciando con el punto deseado del efector como (x, y, z) y tomando en cuenta el gráfico de la Figura 3.14.

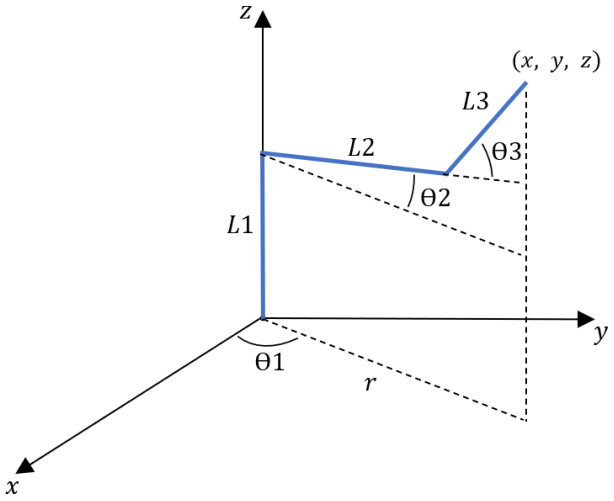


Figura 3.14: Diagrama del manipulador para el método geométrico.

Realizando las respectivas proyecciones y empleando las diferentes relaciones geométricas, se obtienen las siguientes expresiones para los ángulos de cada articulación

$$\begin{aligned} \theta_1 &= tan^{-1}\left(\frac{y}{x}\right) \\ \theta_2 &= tan^{-1}\left(\frac{z}{\sqrt{x^2+y^2}}\right) - tan^{-1}\left(\frac{L_3 sin(\theta_3)}{L_2 + L_3 cos(\theta_3)}\right) \\ \theta_3 &= tan^{-1}\left(\frac{-\sqrt{1-cos^2(\theta_3)}}{cos(\theta_3)}\right) \end{aligned} \tag{3.27}$$

donde

$$cos(\theta_3) = \frac{x^2 + y^2 + (z - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3}$$
$$sin(\theta_3) = \sqrt{1 - cos(\theta_3)} \tag{3.28}$$

En el cálculo de $sin(\theta_3)$ se pueden obtener dos soluciones válidas a partir de la raíz cuadrada, para el caso se toma la solución negativa que correspondería a una configuración de codo arriba. Algo importante a mencionar es que esta solución aplica para el caso general donde todas las articulaciones tienen un ángulo de rotación de 360 grados, en caso de que las articulaciones tengan un rango limitado es necesario hacer el análisis respectivo.

3.5. Selección de componentes

De acuerdo a las características definidas y a los requerimientos de diseño identificados, se detalla la selección de componentes para la construcción del prototipo explicando las características de los motores, llantas, cámaras y demás componentes necesarios para implementar las funcionalidades. También se detallan los equipos necesarios para crear la red inalámbrica sobre la cual se realiza la conexión entre el prototipo y un computador.

3.5.1. Estructura

Motores

Como se planteó anteriormente la topología seleccionada es de 4 ruedas sin dirección tipo skid steer similar a algunos de los equipos comerciales. Normalmente para la tracción de estos dispositivos se emplean únicamente dos motores uno para las ruedas de cada lado las cuales están conectadas por medio de engranajes, sin embargo por simplicidad se elige utilizar un motor para cada rueda y por medio de controladores digitales se asegura que ambas ruedas de cada lado giren a la misma velocidad. Dado que es preferible fuerza y no velocidad para mover el vehículo, se opta por utilizar motorreductores.

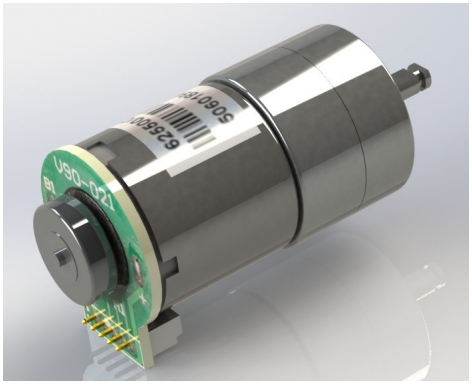


Figura 3.15: Motorreductor 625500/C [26].

Los motorreductores DC de referencia 625500/C, como el que se observa en la Figura 3.15 cumplen con los requerimientos mencionados anteriormente, además traen encoders magéticos incluidos con los que se puede medir la velocidad a la que giran y cuentan con las siguientes especificaciones:

- Corriente sin carga: 50 mA
- Corriente torque detenido: 1 A
- Velocidad sin carga: 64 RPM
- Voltaje de trabajo: 6-12 V (32-64 rpm)
- Relación de reducción: 1:110
- Peso : 74 g
- Salida: 220 pulsos cada vez que el eje de salida del motor da una vuelta

Llantas

Para poder poner llantas en los motores se utiliza un adaptador que de un lado va atornillado al eje del motor y del otro tiene una punta hexagonal de 12 mm de diámetro a la cual se pueden atornillar las llantas. Gracias a este adaptador es posible colocar llantas con diferentes diámetros y tamaños a los motores, si tienen la misma entrada hexagonal la cual es bastante común en llantas para vehículos a control remoto. Se elige el juego de llantas mostrado en la Figura 3.16.



Figura 3.16: Juego de llantas todoterreno [27].

Estas llantas son fabricadas en plástico con neumáticos de goma los cuales tienen un inserto de espuma, son resistentes al desgaste y ofrecen una gran tracción y agarre, lo que las hace perfectas para cualquier tipo de entorno. Dentro de las medidas tienen un diámetro exterior de 125 mm, un diámetro interior de 80 mm, ancho de 67 mm y una entrada hexagonal de 12 mm, adecuada para el adaptador que va a los motores.

Módulo puente H

Para poder controlar la velocidad y el sentido de giro de los motores, se decide utilizar el módulo puente H de referencia L298N que se muestra en la Figura 3.17, ya que es de bajo costo y fácil de utilizar.

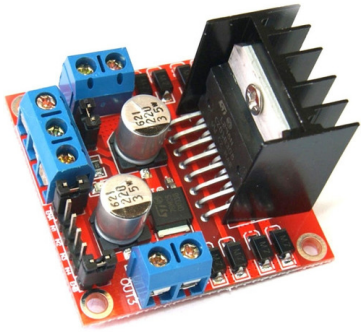


Figura 3.17: Módulo puente H L298N.

Cada módulo permite conectar dos motores DC, cuenta con conectores para enviar señales digitales como la modulación por anchos de pulso PWM, con las cuales se puede controlar tanto la velocidad y también cuenta con entradas digitales para controlar el sentido de giro de cada motor. Este módulo cuenta con las siguientes especificaciones:

- Voltaje de Operación DC: 5 V a 46 V
- Corriente máxima de operación por canal: 2 A
- Puede controlar 2 motores DC de manera independiente ó 1 motor de 4 hilos paso a paso bipolar
- Dimensiones: 43 mm x 23,9 mm x 43 mm

Cámara

Como se ha mencionado anteriormente el prototipo va a estar basado en una Raspberry Pi 4 Model B la cual trae un conector CSI para cámaras, por ello para realizar la captura de imágenes y vídeo se decide utilizar el módulo Raspberry Pi Camera mostrado en la Figura 3.18, que cuenta con un sensor de 5 megapixeles capaz de grabar video y tomar fotos a una resolución máxima de 1080p que puede ser ajustada a diferentes resoluciones menores [28].

Utiliza el cable CSI para conectarse a la Raspberry Pi y cuenta con una biblioteca escrita en Python para controlar sus funcionalidades, esto es de gran utilidad ya que hace compatible el uso de la cámara dentro del framework de desarrollo web Django elegido para el proyecto. Además de su bajo costo existen diferentes tipos de cámaras

que utilizan el mismo conector, lo que permite cambiarla fácilmente en caso de daño por otra igual o por otra con mejores especificaciones.



Figura 3.18: Módulo Raspberry Pi Camera.

Dentro de las especificaciones generales de este modelo se tiene:

- Tamaño: 25 mm x 24 mm x 9 mm
- Peso: 3 g
- Resolución: 5 Megapixeles
- Modos de video: 1080p30, 720p60 and 640 × 480p60/90
- Sensor: OmniVision OV5647

Iluminación

Para el sistema de iluminación que va junto a la cámara lo más apropiado es utilizar un LED de color blanco, dado que se necesita tener una buena iluminación de los interiores de la tubería se decide utilizar un LED de potencia como el que se muestra en la Figura 3.19. Este tipo de LED es ampliamente utilizado en dispositivos que requieren gran iluminación como linternas o faroles, además debido a la potencia que consume cuenta con un disipador de calor y requiere de un circuito de alimentación.



Figura 3.19: LED de potencia.

Las especificaciones de este LED se muestran acontinuación:

- Voltaje típico de operación: 5 V
- Corriente: 600 mA
- Potencia: 3 W
- Disipador en estrella

Servomotores

Para mover las articulaciones del brazo robótico sobre el cual se va a colocar la cámara, se utilizan servomotores tipo SG90 como el que se muestra en la Figura 3.20. Los servomotores permiten un control preciso en términos de la posición angular a diferencia de los motores normales, para ajustar la posición se utilizan señales PWM donde dependiendo del ciclo útil se obtiene un giro determinado.



Figura 3.20: Servomotor SG90.

Un aspecto importante de estos servomotores, es que únicamente permiten giros en un rango de 180 grados, lo cual va a limitar la movilidad de las articulaciones del brazo. En general las especificaciones de los servomotores SG90 son:

- Tamaño: 22 mm x 11.5 mm x 27 mm
- Peso: 9 g
- Torque: 1.2 Kg/cm
- Voltaje de operación: 4 a 7 V
- Corriente de operación: 100 a 300 mA
- Ángulo de rotación: 180 grados

Módulo de señales PWM

Para generar las señales PWM que permitan controlar los servomotores del brazo robótico se decide utilizar el módulo PCA9685 que se observa en la Figura 3.21, este módulo cuenta con la ventaja de pemitir generar hasta 16 señales PWM únicamente utilizando dos pines de la Raspberry Pi ya que emplea el protocolo I2C.

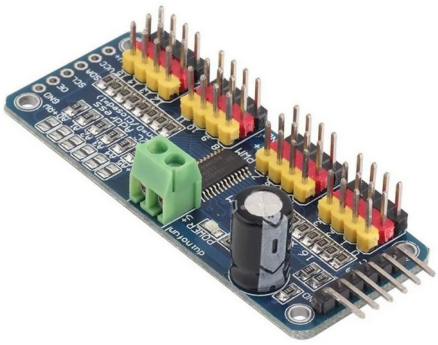


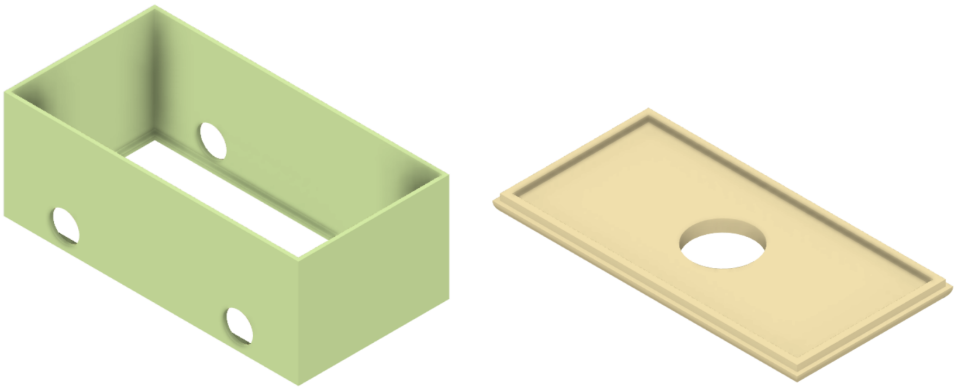
Figura 3.21: Módulo PCA9685 con l6 canales para señales PWM

Sus principales características son:

- Voltaje de funcionamiento: 3 a 5 V
- Voltaje máximo: 6 V
- Número de salidas: 16 canales
- Resolución: 12 bits
- Frecuencia de operación 40 a 1000 Hz
- Dimensiones: 60 mm x 25 mm

Chasis

Para el chasis que va a contener todos los componentes del vehículo, se decide utilizar un modelo de caja con tapa a presión para evitar filtraciones en entornos con agua como las tuberías, el chasis cuenta con dos piezas principales: la caja con orificios para los motores y la tapa sobre la cual va el brazo robótico, los modelos 3D diseñados para ambas piezas se muestran en la Figura 3.22.



(a) (b)

Figura 3.22: Modelos 3D para el chasis y la tapa.

Brazo robótico

Como se mencionó anteriormente, se decide utilizar un robot manipulador en configuración de codo con tres articulaciones como el que se observa en el modelo 3D de la Figura 3.7, el cual es una adaptación del diseño original denominado Snapper Mini [29] que cuenta con 5 grados de libertad. Este brazo utiliza servomotores tipo SG90 para mover las articulaciones y en su efector se fija la cámara y el sistema de iluminación, cuenta con un diseño robusto y es fácil de armar. Todas las piezas necesarias se pueden descargar gratis desde el perfil de Thingiverse del autor.

Materiales de impresión

Ambos modelos, el del chasis y el del brazo, se imprimen en 3D. Existen diferentes materiales para realizar la impresión, dentro de los más comunes se tiene el ABS, el PLA y sus diferentes mezclas. Dependiendo de la impresora se pueden usar otro tipo de materiales especializados que ofrecen diferentes propiedades, en la Tabla 3.2 se muestran los tres tipos de materiales más comunes para realizar impresiones en 3D.

Tabla 3.2: Materiales de impresión 3D [30].

Material	**Características**	**Usos**
ABS (acrilonitrilo butadieno estireno)	Tenaz y duradero, resistente a impactos y al calor	Propotipos funcionales
PLA (ácido poliláctico)	Rígido pero frágil ante impactos, menos resistente al calor y a los productos químicos, biodegradables	Modelos conceptuales y prototipos estéticos
PETG (tereftalato de polietileno glicolizado)	Resistente a la humedad y a los productos químicos, alta transparencia, puede ser apto para la alimentación	Aplicaciones impermeables y componentes de fijación

Para el material de impresión se decide utilizar **PLA**, ya que puede ser utilizado para prototipos y no es demasiado costoso, sin embargo para este tipo de aplicación lo más recomendable es utilizar **PETG** por las características que ofrece.

Alimentación

Uno de los componentes más importantes son las baterías ya que a partir de ellas se van a alimentar los demás componentes del prototipo, además dependiendo de su capacidad se obtiene determinado tiempo de autonomía. Para poder seleccionar la batería apropiada resulta necesario conocer los consumos que va a tener el prototipo para su funcionamiento, esto hace referencia a los consumos de: la Raspberry Pi, los motores, el LED de iluminación, servomotores, entre otros. Para facilitar el cálculo de la capacidad mínima necesaria, se va a asumir que cada componente requiere su valor máximo de corriente.

Partiendo por la Raspberry Pi 4 el consumo depende de los procesos que esté ejecutando y de los periféricos que tenga conectados, según la documentación se tiene un consumo aproximado de 1.25 A bajo un estrés elevado, es decir mientras ejecuta ciertos procesos que requieren un uso intensivo de los recursos de la placa. Obviamente el consumo se ve aumentado si se conectan periféricos a la Raspberry Pi, para este caso únicamente se conecta la cámara la cual va a grabar videos a baja resolución y según la documentación, presenta un consumo aproximado de 500 mA grabando a 1080p [31].

Para el caso de los demás componentes, de las características listadas anteriormente se tiene que cada motor tiene un consumo máximo de 1 A con el torque detenido, cada servomotor tiene un consumo máximo de 300 mA y el LED de iluminación tiene un consumo máximo de 600 mA. Con estos valores, en la Tabla 3.3 se muestra el consumo total aproximado que tiene el vehículo.

Tabla 3.3: Consumos de energía del prototipo.

Componente	**Corriente**	**Cantidad**
Raspberry Pi 4	1.25 A	1
Cámara	500 mA	1
Motorreductor	1 A	4
Servomotor	300 mA	3
LED	600 mA	1
Total	7.25 A	

Con esto se tiene que el vehículo consume máximo aproximadamente 7.25 A durante su funcionamiento. Asumiendo que el tiempo mínimo para realizar una inspección sea de 30 minutos, se hace necesario contar con una batería cuya capacidad sea de aproximadamente 3700 mAh a 12 V (tomando en cuenta que es el voltaje máximo de todos los componentes). Se plantea el uso de dos baterias en el prototipo, debido a

que al desarrollar el prototipo se trabajaba en paralelo con la parte móvil relacionada con los motores y con la parte funcional relacionada a la Raspberry Pi que controla las funcionalidades, sin embargo es recomendable emplear una única batería con las especificaciones obtenidas anteriormente.



Figura 3.23: Batería de alimentación para los motores.

Tal como se observa en las características de los motores tienen un consumo de 1 A con el torque detenido, lo cual evidentemente no es deseable, por lo que realizando la medición de corriente en cada motor girando en el suelo con las llantas acopladas y todos los componentes encima, se observa que con el voltaje nominal tienen un consumo de aproximadamente 500 mA. Es por ello que para la alimentación de los motores, se realizó un acople de 3 baterías de ion de litio como las que se observan en la Figura 3.23, donde cada una entrega 3.7 V, las cuales al estar en serie entregan un voltaje de 11.1 V y cuentan con una capacidad de 2000 mAh en total, con ellas se alimentan los módulos puente H a los que se conectan los motores y se obtiene una autonomía de aproximadamente 1 hora para los motores.

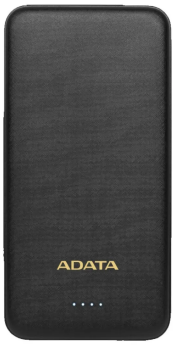


Figura 3.24: Batería de alimentación para la Raspberry Pi.

Finalmente para alimentar la Raspberry Pi, el módulo PCA9685 que requiere de alimentación externa para los servomotores y el LED, se utiliza una Power Bank que

entrega 5 V y tiene una capacidad de 10000 mAh como la que se muestra en la Figura 3.24. Entre todos estos componentes se tiene un consumo máximo de 3.25 mA, claramente es menor el consumo con el prototipo en funcionamiento por lo que con esta batería se puede lograr una autonomía de más de 3 horas para la parte de control. Una de las razones de utilizar dos baterías, una para la parte de control y otra para la parte de potencia, es reducir cualquier tipo de ruido. Sin embargo es deseable que solo se utilice una batería para alimentar todos los componentes por temas de espacio y facilidad de conexión dentro del chasis. Se recomienda entonces mejorar la parte de la alimentación en trabajos futuros.

3.5.2. Red inalámbrica

Router

Para crear la red inalámbrica WiFi que permita la comunicación entre el prototipo y un equipo de cómputo se hace necesario utilizar un router. Se decide utilizar el router de referencia TL-WR840N de la marca TP-Link que se observa en la Figura 3.25, ya que es liviano, de bajo costo y sus características son apropiadas para la comunicación inalámbrica que se plantea. La idea es ubicarlo en la entrada de la tubería ya que permite un mayor alcance de conexión con el vehículo al llegar la señal de forma directa.



Figura 3.25: Router TL-WR840N de la marca TP-Link [32].

Dentro de las características de este router se tiene:

- Velocidad de transmisión inalámbrica de 300 Mbps ideal tanto para tareas sensibles al ancho de banda como para trabajos básicos.
- Admite el modo de punto de acceso para crear un nuevo punto de acceso WiFi.
- Admite el modo Range Extender para aumentar la cobertura inalámbrica existente en su habitación.

- Compatible con IPv6 (Protocolo de Internet versión 6).
- Guest Network proporciona acceso independiente para invitados mientras protege la red doméstica.

Inversor

Lo deseable para el proyecto es que el router cuente con una batería integrada ya que en los sitios de inspección puede ser complicado acceder a una toma de corriente para alimentar el router, sin embargo este tipo de dispositivos son bastante más costosos que el router que se elige, así que se opta por utilizar un inversor de voltaje que permita obtener 120 V AC a partir de un voltaje DC dado. El inversor a utilizar es de referencia XI14 de la marca Schumacher como el que se observa en la Figura 3.26.



Figura 3.26: Inversor de voltaje Schumacher XI14.

Sus características se listan a continuación:

- Corriente de entrada máxima: 13 A
- Voltaje de entrada: 10 a 15 V DC
- Frecuencia de salida: 60 Hz
- Voltaje de salida: 120 V AC
- Energía continua: 140 W
- Sobretensión: 280 W

Batería

Para alimentar el inversor y poder obtener los 120 V AC en el lugar de la inspección se plantea utilizar una batería de plomo 12 V DC como la que se muestra en la Figura 3.27.



Figura 3.27: Batería de plomo 12 V, 5 Ah.

Este tipo de baterías son utilizadas para diferentes aplicaciones como sistemas de alarmas de seguridad, CCTV, sistemas de iluminación de emergencia, alimentación de equipos eléctricos OEM, arranque para pequeños vehículos como karts, entre otros. Es una batería recargable y cuenta con un voltaje nominal de 12 V DC y una capacidad de 5 Ah, lo cual permite un funcionamiento prolongado del router.

3.6. Diseño de circuitos y conexiones

A partir de los componentes seleccionados se plantea el diagrama de conexiones que se muestra en la Figura 3.28.

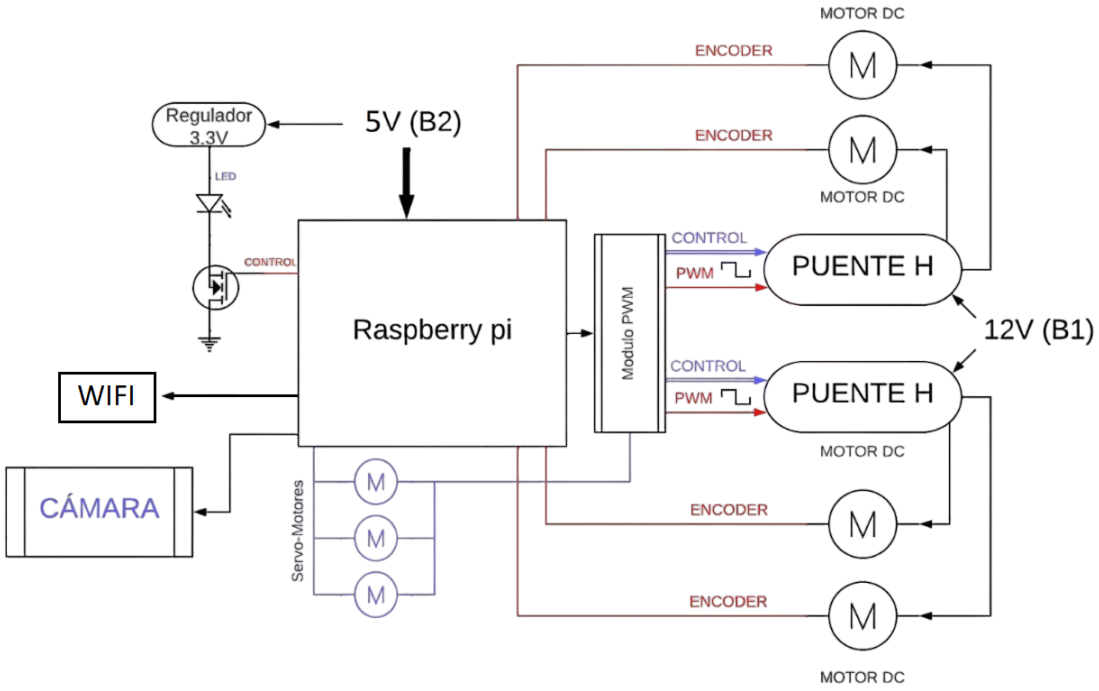


Figura 3.28: Diagrama de conexión del prototipo.

Además de las conexiones dentro del prototipo, se tendrá energizado un router que será el medio de comunicación entre el prototipo y el computador cliente, para esto se contará con una batería de 12 V, la cual con ayuda de un inversor de 12 V a 120 VAC energizará el router.

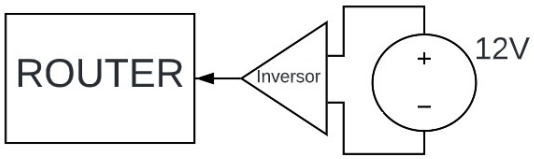


Figura 3.29: Diagrama de conexión del router.

Con estos diagramas se realiza el circuito final en el software Proteus donde se obtiene lo que se muestra en la Figura 3.30.

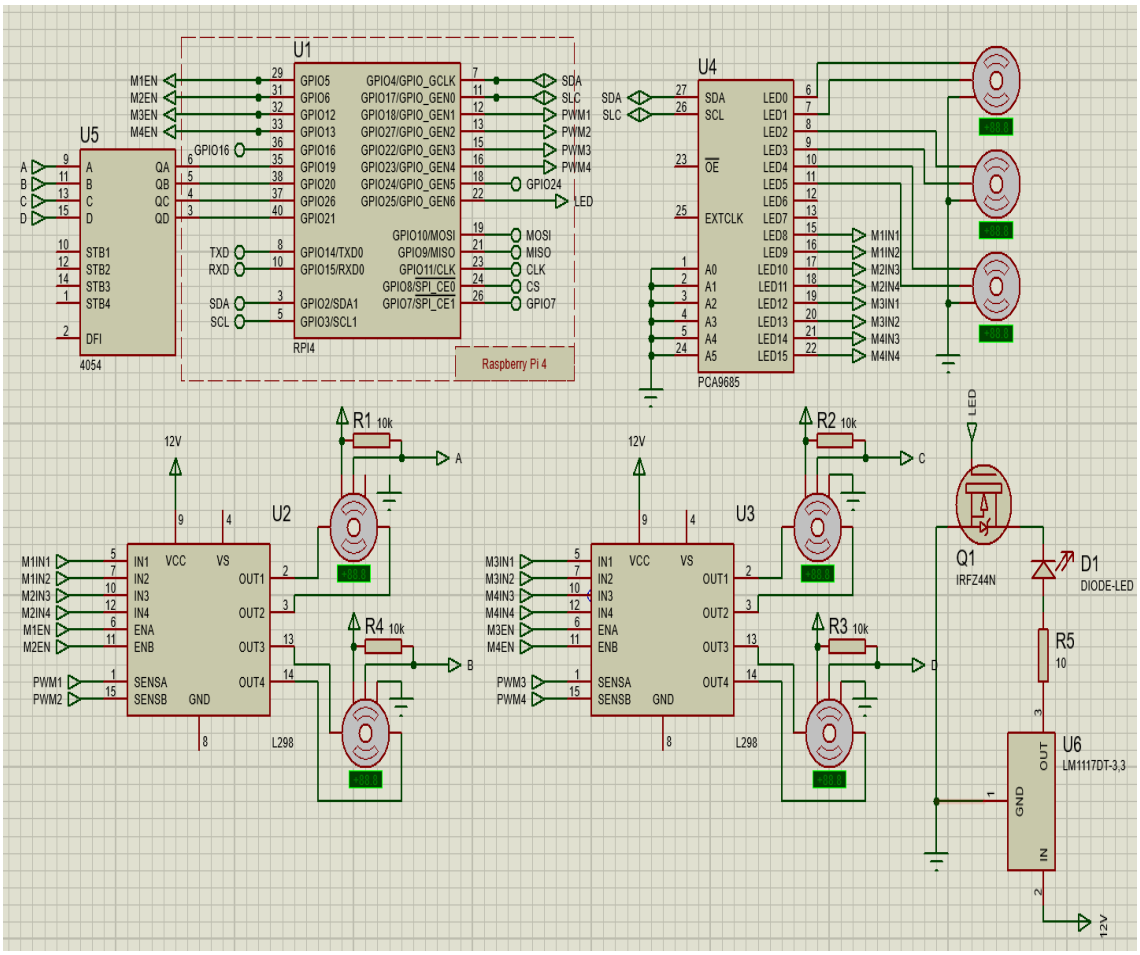


Figura 3.30: Circuito de conexiones hecho en Proteus.

Del circuito que se observa en la Figura 3.30 podemos resaltar 4 secciones diferentes. En la parte superior izquierda se observa la Raspberry Pi 4 (U1) que será usada como controlador general, para la lectura de la velocidad de los motores se utilizan interrupciones que permitan contar y calcular las RPM a las que gira cada motor y con ello implementar los controladores digitales.

En la parte superior derecha se observa un Módulo de PWM que nos permitirá generar múltiples señales PWM (U4) para el control de los servomotores y la velocidad de giro de los motorreductores. En la parte inferior izquierda se observan 2 módulos Puente H (U2-U3) que con ayuda de dos señales PWM, nos permitirán controlar la velocidad de los motorreductores, junto a estos podemos observar los 4 motorreductores que serán usados para el movimiento del prototipo. En la parte inferior derecha se observa un pequeño circuito conformado por un MOSFET y una resistencia, el cual ayudará a controlar el encendido y apagado de un LED de potencia para iluminar.

En general este circuito es una aproximación al modelo que se desea implementar donde faltan aún piezas y conexiones que no se contemplan en este diseño. Las cuales serían la conexión con las baterías y la cámara.

4. Construcción del prototipo

En este capitulo se documenta el ensamble del prototipo donde se incluye toda la distribución de componentes sobre en el chasis para lo cual se utiliza como apoyo un modelo 3D realizado en Autodesk Inventor. También se detalla la configuración de la red sobre la que se realiza la comunicación inalámbrica con el prototipo, toda la programación de lógica necesaria para la aplicación web que incluye la obtención de videos y fotografías, así como las funcionalidades planteadas por los requerimientos de diseño. Finalmente respecto a los controladores de velocidad para los motores, se realiza la caracterización y diseño para el control digital.

4.1. Ensamble y distribución de componentes

Empezando con el diseño del chasis, al momento de acomodar todos los componentes dentro de este, se adaptan 2 piezas de acrílico de 12 cm x 24 cm para tener una buena distribución de los elementos tal como se muestra en la Figura 4.1.

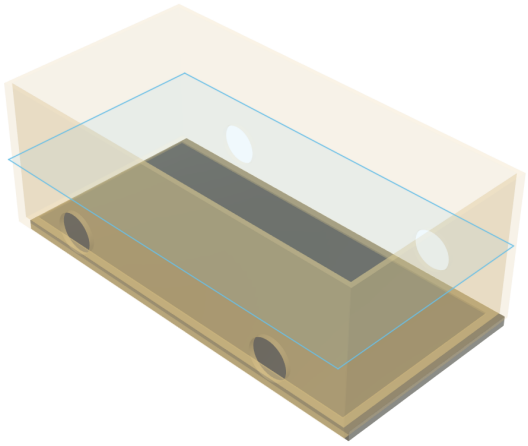


Figura 4.1: Vista general del chasis.

La pieza naranja es el chasis diseñado previamente, y las otras dos piezas, las de color azul y negro, representan dos niveles sobre los que irán los componentes del prototipo móvil. En la Figura 4.2 se muestra una vista lateral de la ubicación de los niveles.

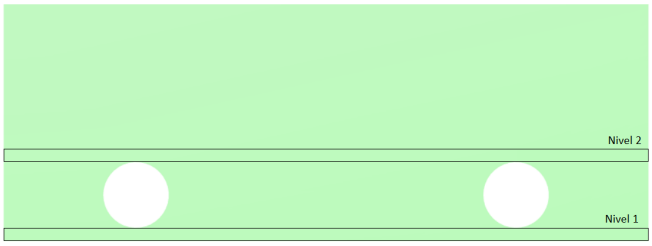


Figura 4.2: Vista lateral de los niveles del chasis.

4.1.1. Nivel 1

Para el nivel uno se cuenta con una tabla de acrílico con 8 agujeros (color gris oscuro), los cuales nos permitirán el paso de unos tornillos para asegurar que los motores se mantengan en su lugar. En este mismo nivel sobre el centro se ubican los dos módulos puente H tal como se muestra en la Figura 4.3.

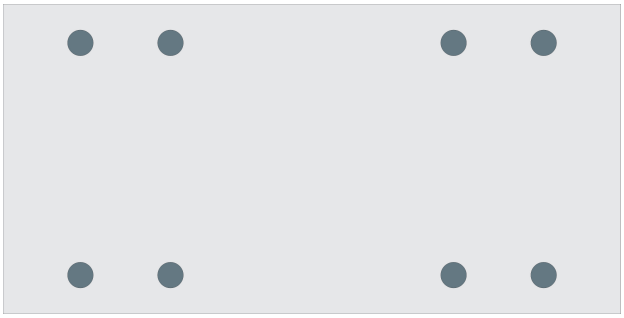


Figura 4.3: Base en acrílico para el nivel 1.

Igualmente para observar la distribución de los componentes sobre este nivel se muestra un modelo 3D en la Figura 4.4, los 4 motores y en el centro de cada par los dos módulos puente H.

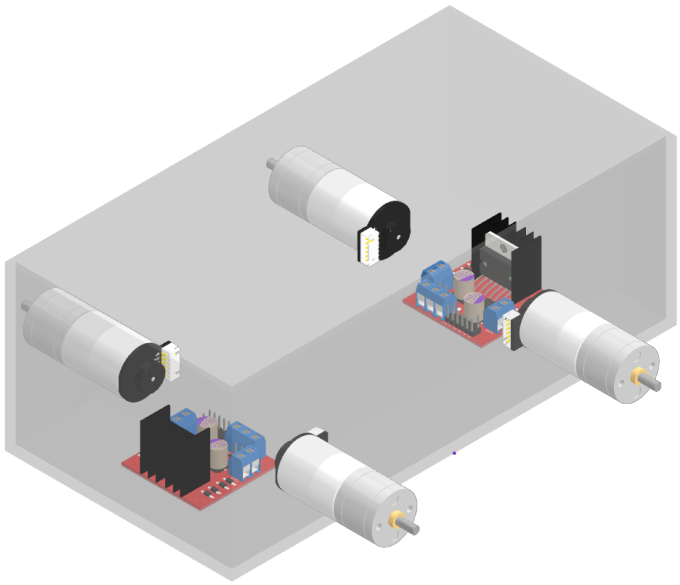


Figura 4.4: Modelo 3D con los componentes del nivel 1.

4.1.2. Nivel 2

En el nivel 2 se tiene una tabla de acrílico con los mismos 8 agujeros que la tabla del nivel uno, la idea es colocar esta tabla sobre los motores y ajustar ambos niveles con

8 tornillos. Esta tabla cuenta con dos agujeros rectangulares a los lados (color azul), a través de los cuales es posible pasar el cableado de los motores y los módulos puente H para conectarlos a las baterías y a la Raspberry Pi.

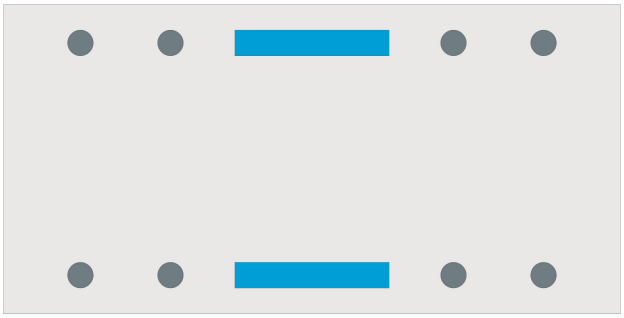


Figura 4.5: Base en acrílico para el nivel 2.

Para ilustrar de mejor manera la distribución de componentes en el nivel 2, en la Figura 4.6 se muestra el modelo 3D con los componentes necesarios. La caja negra representa la batería de 5 V que alimenta la Raspberrry Pi, sobre ella se encuentra la placa de la Raspberry Pi y una caja azul que representa la batería de 12 V que alimenta los motores. En frente se encuentra el módulo para generar los PWM y por último se encuentra una protoboard para el circuito de iluminación.

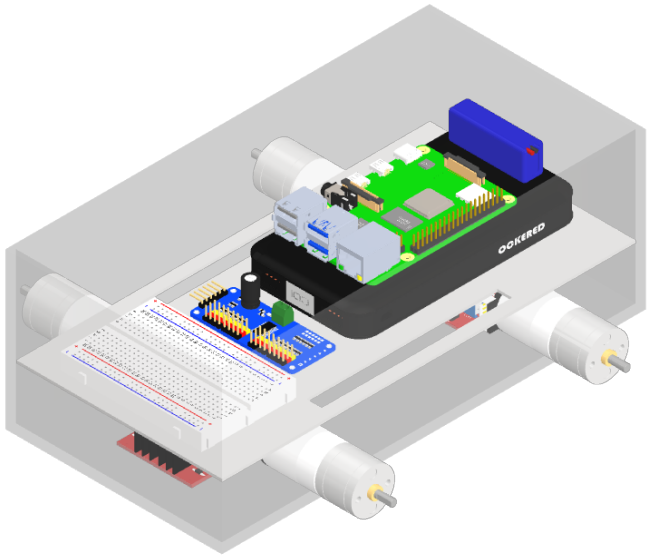


Figura 4.6: Modelo 3D con los componentes del nivel 2.

4.1.3. Tapa

Por último se tiene la tapa del chasis sobre la cual va fijado el brazo robótico que sostiene la cámara y el LED de iluminación, tal como se muestra en el modelo 3D de

la Figura 4.7.

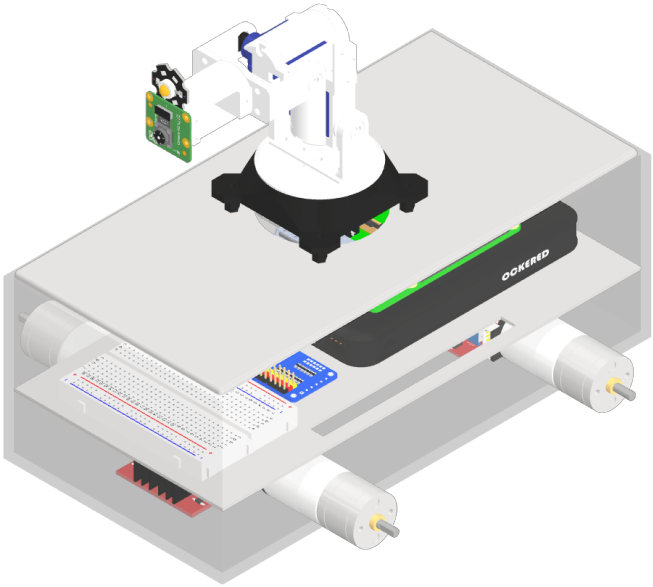


Figura 4.7: Modelo 3D con el brazo sobre la tapa.

4.1.4. Prototipo completo

Finalmente si se adicionan las llantas a los motores, se obtiene el modelo 3D del prototipo completo que se muestra en la Figura 4.8.

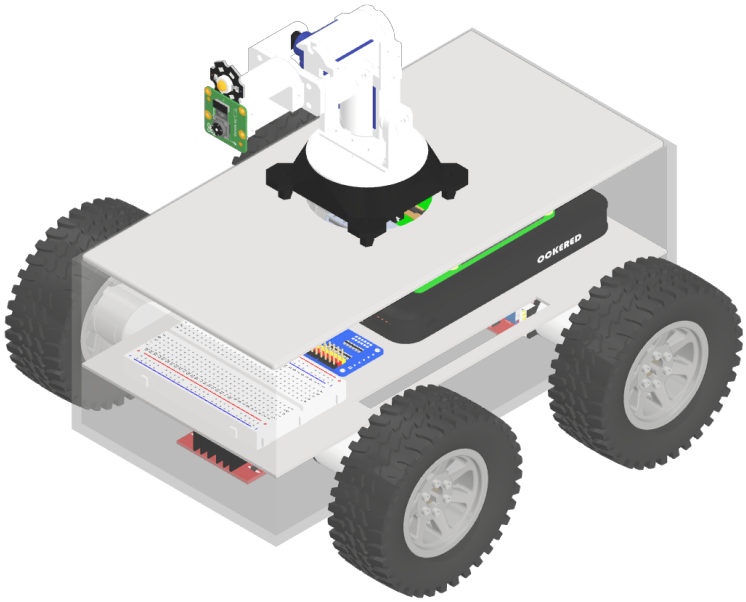


Figura 4.8: Modelo 3D del prototipo ensamblado.

4.2. Configuración de la red inalámbrica

Como se mencionó para la comunicación inalámbrica se utiliza WiFi y dado que la inspección se realiza in situ, no es necesario realizar una configuración que permita tener acceso a internet, basta con configurar el router para crear una red WLAN a la cual se van a conectar todos los dispositivos y sobre la cual se realiza el envío y la recepción de información entre el prototipo y un equipo de cómputo. Con el router se crea una red que permita verificar el funcionamiento del prototipo, claro está que si se quieren alcances y tasas de transferencia mayores se pueden utilizar routers con características mejores. Para poder encender este router, dado que en el sitio de inspección no se cuenta con tomas de fácil acceso, se utiliza el inversor de voltaje. Basta con conseguir el conector hembra con forma de cigarrera que se conecte a la batería de 12 V y a dicho conector se conecta el inversor que cuenta con una toma de 120 VAC incorporada en la parte superior donde se puede conectar el adaptador del router. También para evitar esta conexión, se puede utilizar un router más moderno que cuente con una batería integrada, teniendo en cuenta la capacidad de la misma para estimar los tiempos de inspección. Es importante que tanto el router como las conexiones eléctricas se encuentren impermeabilizadas ya que la idea es colocar el router al mismo nivel de la tubería que se quiere inspeccionar.

Ahora es necesario crear la red WLAN, para ello en primer lugar se configura la puerta de enlace del router, el router utilizado trae por defecto la dirección 192.168.0.1 así que no es necesario realizar dicha configuración pero en caso de no traerlo de fabrica es necesario definirla. Luego de ello se procede a configurar dentro del router la red WiFi, para lo cual se conecta primero un computador al router y desde cualquier navegador se accede a la dirección de la puerta de enlace 192.168.0.1, esto abre el panel de configuración del router mostrado en la Figura 4.9.

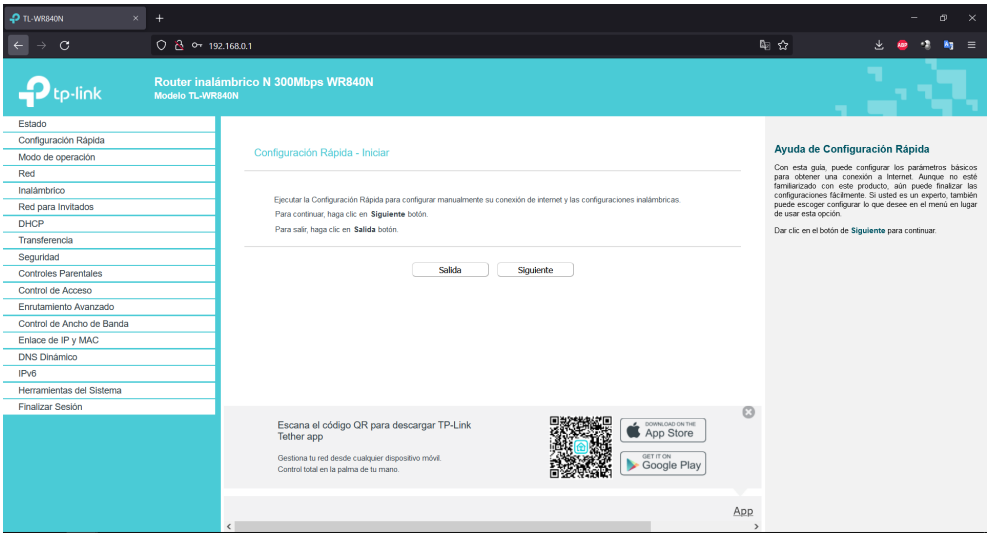


Figura 4.9: Panel principal de configuración para el router TL-WR840N.

Estando en el panel principal, se selecciona del menú de la izquierda la opción Inalámbrico lo cual despliega un nuevo panel como el que se muestra en la Figura 4.10. De este panel lo más importante es definir un SSID o nombre de red y el canal sobre el cual va a estar la red, esto último con el fin de prevenir o disminuir posibles incrementos en la latencia provocados por la interferencia con redes WiFi que existan en la zona. Para ello se selecciona un canal o frecuencia específica de los 14 canales existentes dentro de la banda de 2.4 GHz que sea poco utilizado en el lugar.

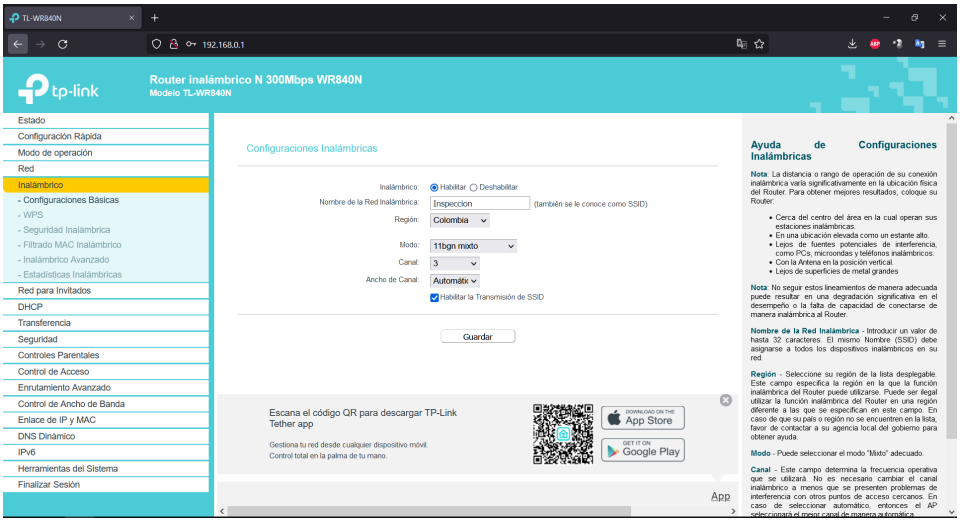


Figura 4.10: Panel de configuración inalámbrica para el router TL-WR840N.

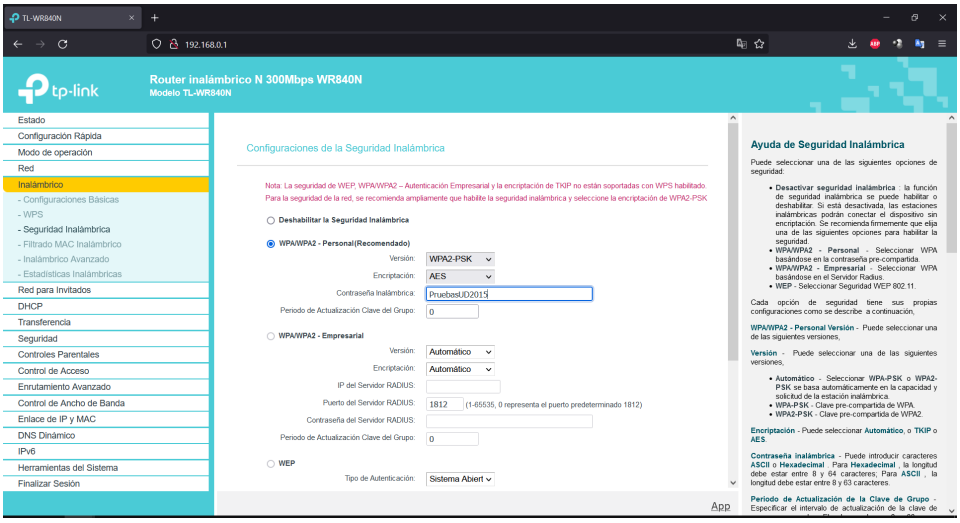


Figura 4.11: Panel de configuración de seguridad inalámbrica para el router TL-WR840N.

Luego de esto se da clic en el botón Guardar y el router se reinicia para aplicar los cambios, ahora lo único que falta es configurar una contraseña para la red inalámbrica

para lo cual hay que conectarse nuevamente a la red desde el computador, acceder a la dirección Ipv4 192.168.0.1 desde un navegador y dentro del menú Inalámbrico seleccionar la opción Seguridad Inalámbrica. Esto despliega el panel que se observa en la Figura 4.11, en el cual se define el cifrado inalámbrico de la red. Siguiendo las recomendaciones del fabricante se selecciona la encriptación WPA2-PSK y se ingresa la contraseña deseada, nuevamente se da clic en Guardar y el router se reinicia una vez más. Con esto la red WLAN ya queda configurada y el paso siguiente es conectar los dispositivos que van a comunicarse a través de ella: el prototipo (servidor) y el computador (cliente).

Ahora para el caso del prototipo, dentro de las configuraciones del adaptador inalámbrico integrado se define una dirección IPv4 estática para que cada vez que se conecte al router le sea asignada dicha dirección, para el caso se selecciona la IPv4 192.168.0.20 y se ingresa también la dirección IP de la puerta de enlace como se aprecia en el menú de la Figura 4.12.

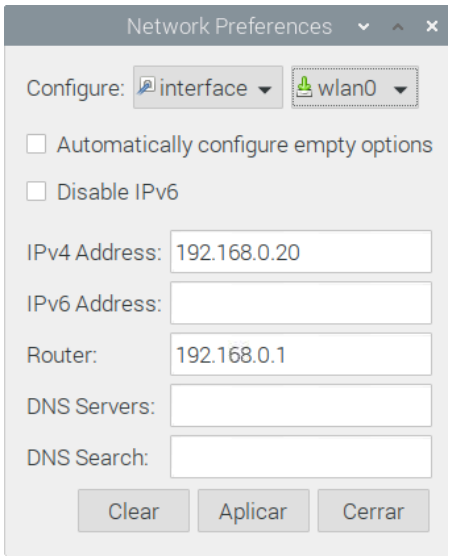


Figura 4.12: Configuración del adaptador de red de la Raspberry Pi

Finalmente se conecta y registra en el prototipo la red WLAN creada, este registro puede hacerse de dos formas: la primera y la más simple es, si el sistema operativo cuenta con una interfaz gráfica, conectar una pantalla a la Raspberry Pi y realizar la conexión como si fuese un computador normal. Para la segunda, la Raspberry Pi cuenta con un archivo llamado *wpa_supplicant.conf* dentro del cual se registran el identificador de red SSID y la contraseña como se observa en la Figura 4.13, la ventaja de realizar la configuración de forma manual por medio de este archivo es que cada vez que la Raspberry Pi se encienda se conectará automáticamente a dicha red si la encuentra. Finalmente se conecta el computador sobre el cual se va a ejecutar la aplicación web a la misma red WLAN y por medio del comando ping se verifica la conexión entre ambos dispositivos como se observa en la Figura 4.14.

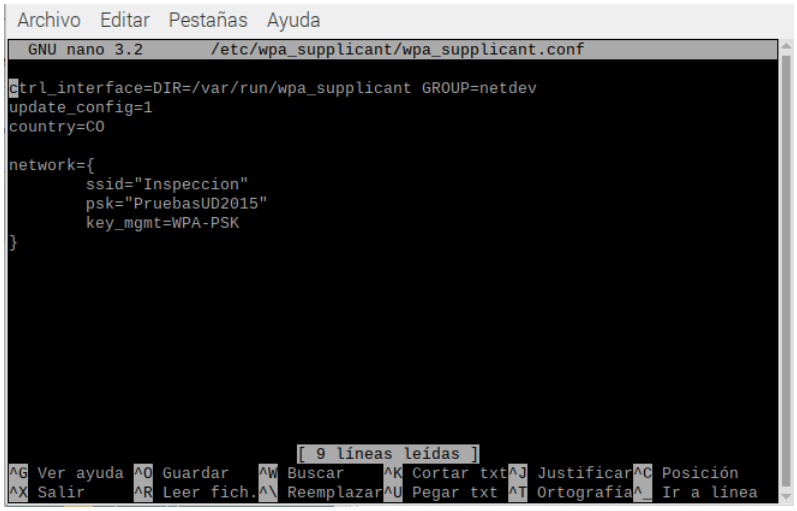


Figura 4.13: Archivo de configuración para la red WiFi de la Raspberry Pi.

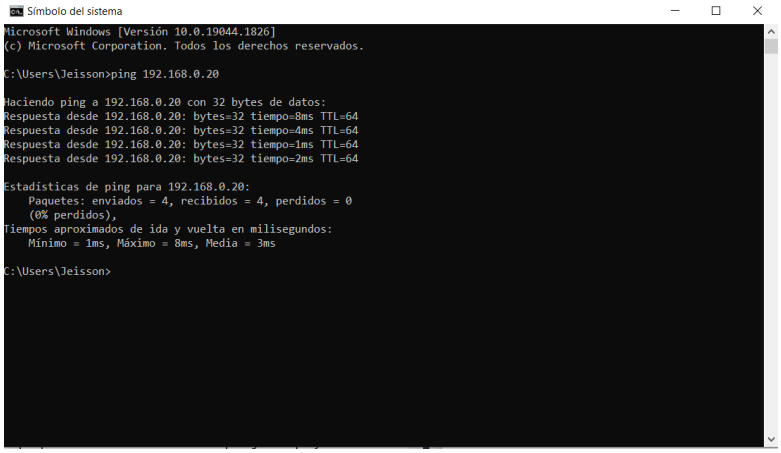


Figura 4.14: Prueba de conexión con el comando ping desde el computador a la Raspberry pi.

Ahora para que el cliente pueda acceder a la aplicación web (explicada en la siguiente sección), se inicia un servidor web de desarrollo ligero en la Raspberry Pi, se opta por utilizar este tipo de servidor ya que es suficiente para verificar el funcionamiento del prototipo aunque es recomendable que para una versión final se despliegue el proyecto en otro tipo de servidores como Nginx o Apache que se ejecuten sobre la Raspberry Pi. Simplemente resta que el cliente ingrese desde un navegador web la URL de la aplicación web, esta URL está conformada por la dirección IPv4 de la Raspberry Pi junto al puerto (si es que se define dentro del servidor) y una ruta que define la página principal de la aplicación, para el caso es http://192.168.0.20:9001/home, o si se prefiere y el servidor lo tiene definido, se puede configurar un hostname para esa URL en el equipo cliente y acceder directamente desde él.

4.3. Programación del prototipo

4.3.1. Aplicación web

La parte de la aplicación web está únicamente centrada en proporcionar las herramientas que permitan realizar el proceso de inspección dentro de una red de alcantarillado, sin embargo para una versión más avanzada es necesario ampliar e incluir nuevas funcionalidades como el proceso que se realiza a nivel de documentación, bases de datos, autenticación, reportes y todo el desarrollo que complementa la toma de fotografías y videos de una inspección según la norma NS-058. Es por ello que la aplicación web cuenta con una única app que brinda algunas de las funcionalidades básicas para desarrollar una inspección.

4.3.2. Página principal

Desde la plantilla principal que se muestra en la Figura 4.15, se puede observar en el menú del lateral izquierdo botones para navegar entre las diferentes páginas, se tiene un botón de Inicio y uno de Multimedia desde el cual se puede acceder a una nueva plantilla para descargar los archivos de foto y video capturados, en este menú se incluirían más páginas que brinden diferentes funcionalidades como por ejemplo una sección para generar reportes en formato pdf con las incidencias encontradas, tal como lo solicita la norma NS-058.

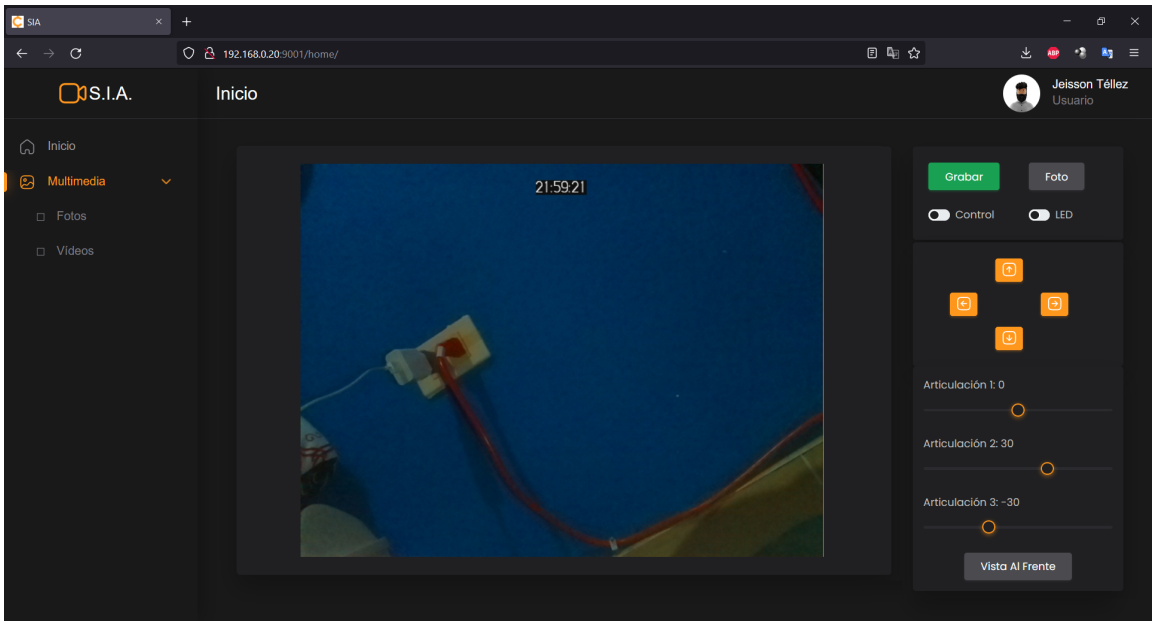


Figura 4.15: Página principal de la aplicación web.

En la mitad se tiene un recuadro donde se plasman las imágenes transmitidas desde la cámara, en la parte superior de la imagen se ubica un overlay de texto que indica la hora, pero tal como lo menciona la norma hace falta añadir más información sobre la

inspección. En el lateral derecho superior se plasma el nombre del usuario que utiliza la aplicación, más abajo se tienen dos botones: uno para grabar y otro para tomar fotos, se tienen dos switches uno para cambiar entre el control remoto y la configuración de la cámara, y otro para encender y apagar el LED de iluminación.

Al estar basado en el framework Django, la lógica que recibe las peticiones hechas desde la plantilla, está contenida en diferentes vistas basadas en clases las cuales se encargan de generar respuestas o realizar ciertas acciones. La primera de ellas se llama Home y se encarga de renderizar la página de inicio con los valores iniciales de la configuración de la cámara como el brillo, contraste, saturación e ISO, también con los valores iniciales de los ángulos para cada articulación como se observa en la siguiente sección de código.

```
class Home(TemplateView):
    """Renderiza la pagina de inicio"""
    template_name = 'home.html'

    def get_context_data(self, *args, **kwargs):
        brillo = camara.brightness
        contraste = camara.contrast
        iso = camara.iso
        saturacion = camara.saturation

        global art1
        global art2
        global art3
        context = {
            "brillo": brillo,
            "contraste": contraste,
            "saturacion": saturacion,
            "iso": iso,
            "articulacion1": art1,
            "articulacion2": art2,
            "articulacion3": art3,
        }
        return context
```

4.3.3. Ajustes de la cámara

Los ajustes de la cámara se muestran sobre la misma página principal al accionar el switch para cambiar entre el control y la configuración. Como se mencionó la vista de la página principal se encarga de cargar los valores por defecto que tiene la cámara para el brillo, contraste, saturación e ISO, utilizando los sliders respectivos se pueden cambiar dichos valores y presionando sobre el botón Aplicar se confirman los cambios que son inmediatamente reflejados en el recuadro que transmite las imágenes desde la cámara. La página que muestra la sección de configuración se observa en la Figura 4.16.

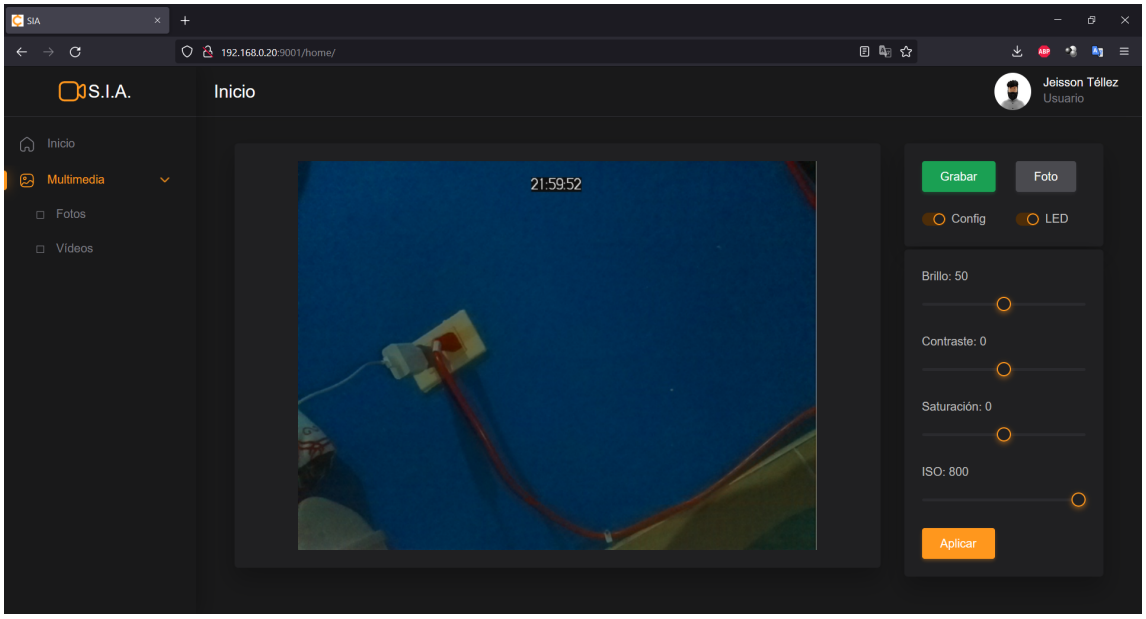


Figura 4.16: Página principal con la configuración de la cámara.

Para procesar la petición del cliente y aplicar la nueva configuración a la cámara se define una nueva función dentro de la vista Home, la cual toma los datos de brillo, contraste, saturación e ISO de la petición y los aplica sobre cada uno de los atributos de la cámara. En la siguiente sección de código se muestra dicha función.

```
class Home(TemplateView):
    """Renderiza la pagina de inicio"""
    template_name = 'home.html'

    def post(self, request, *args, **kwargs):
        post_data = json.loads(request.body.decode("utf-8"))
        camara.brightness = int(post_data["brillo"])
        camara.contrast = int(post_data["contraste"])
        camara.saturation = int(post_data["saturacion"])
        camara.iso = int(post_data["iso"])

        return JsonResponse({"Recibido": True})
```

4.3.4. Iluminación

En el caso de la iluminación se tiene una vista llamada Iluminación, la cual prende o apaga el LED dependiendo de la posición del switch que está en la página principal. Cada vez que cambia de posición el switch se realiza una petición para encender o apagar el LED, la cual es procesada por la vista. Acá lo único que se hace es generar un alto o un bajo en un pin 17 de la Raspberry Pi, señal que actúa como entrada del circuito de iluminación. El código de esta vista se muestra a continuación.

```
class Iluminacion(View):
    """Con esta vista se prende o apaga el LED de iluminacion"""
    def get(self, request, *args, **kwargs):
        estado_led = request.GET.get('estado_led')
        if estado_led == 'on':
            print("LED ENCENDIDO")
            GPIO.output(17, GPIO.HIGH)
        else:
            print("LED APAGADO")
            GPIO.output(17, GPIO.LOW)

        return JsonResponse({"Recibido": True})
```

4.3.5. Control de movimiento y manipulador para la cámara

Motores

La vista Motores se encarga del procesamiento de las peticiones realizadas con los botones que indican el movimiento del vehículo desde la página principal, se encarga de abrir subprocesos que ejecutan un script dependiendo del botón que se pulse, el cual contiene todo el código de los controladores para accionar los motores. Cuando se suelta el botón se envía una petición para detener dicho script y el movimiento como tal.

```
class Motores(View):
    """Se encarga de abrir subprocesos que corren un script para el
   movimiento de los motores"""
    def get(self, request, *args, **kwargs):
        sentido = request.GET.get('sentido')
        global p, s

        if s:
            s.kill()
        if sentido == 'up':
            p = subprocess.Popen(['python','arriba.py'])
        if sentido == 'left':
            p = subprocess.Popen(['python','izquierda.py'])
        if sentido == 'right':
            p = subprocess.Popen(['python','derecha.py'])
        if sentido == 'down':
            p = subprocess.Popen(['python','abajo.py'])
        if sentido == 'stop':
            if p:
                p.kill()
                s = subprocess.Popen(['python', 'parar.py'])

        return JsonResponse({"Recibido": True})
```

En el caso de los scripts denominados **derecha.py** e **izquierda.py**, lo que se hace es mover los motores de cada lado en diferentes sentidos para que el vehículo gire sobre su propio eje y poder orientarlo hacia un lado u otro, evidentemente lo mejor para orientar un vehículo de este tipo es emplear la cinemática del vehículo que permita determinar la velocidad a la que deben girar los motores para apuntar a un ángulo deseado, pero esta implementación se deja como trabajo futuro. Los otros dos scripts denominados **arriba.py** y **abajo.py** contienen todo el código de implementación para la medición de la velocidad con los enconders y los controladores de los motores que permitan al vehículo moverse a una determinada velocidad lo más recto posible mientras se desplaza al interior de la tubería. Estos cuatro códigos se pueden observar en el apéndice C.

Manipulador

Para controlar el ángulo de las articulaciones, se envía desde la página principal el valor que se seleccione en los slider y un número que indica qué articulación se quiere mover. Dicho valor es recibido en la vista llamada Brazo, y al estar entre -90 y 90 grados es mapeado a un valor entre 0 y 180 los cuales corresponden al ángulo de entrada para cada servomotor. Cada ángulo se encarga de generar la señal PWM desde el canal respectivo del módulo PCA9685 con el ciclo útil requerido. En la siguiente sección de código se detalla la lógica empleada en esta vista.

```
class Brazo(View):
    """Se define el angulo de cada servo dependiendo del valor recibido
    """
    def get(self, request, *args, **kwargs):
        articulacion = request.GET.get('articulacion')
        angulo = request.GET.get('angulo')

        global art1
        global art2
        global art3

        if articulacion == '1':
            art1 = int(angulo)
            pwm1 = mapeo(art1,-90,90,0,180)
            servo1.angle = pwm1
        if articulacion == '2':
            art2 = int(angulo)
            pwm2 = mapeo(art2,-90,90,180,0)
            servo2.angle = pwm2
        if articulacion == '3':
            art3 = int(angulo)
            pwm3 = mapeo(art3,-90,90,180,0)
            servo3.angle = pwm3
        return JsonResponse({"A":1})
```

De forma adicional en la plantilla de inicio, justo en la parte de abajo de los sliders para controlar los ángulos, se tiene un botón encargado de ubicar la cámara hacia el frente del vehículo, las coordenadas de dicho punto se calculan utilizando la cinemática directa y por medio de la cinemática inversa se ubica la cámara. Para el presente proyecto no se realiza una implementación más compleja utilizando la cinemática del manipulador ya que los ángulos de rotación de los servomotores están limitados a un rango entre 0 y 180 grados haciendo imposible alcanzar ciertos puntos. Sin embargo el análisis deja la posibilidad de implementar trayectorias para la cámara utilizando las expresiones obtenidas, por ejemplo una trayectoria que permita recorrer la circunferencia interna de la tubería para grabar videos si se emplean servomotores con ángulos de rotación mayores (por lo menos en la base). En las siguientes dos secciones se muestra el código empleado para la vista y la función que implementa la cinemática inversa del manipulador.

```
class PuntoInicial(View):
    """Utiliza la cinematica inversa para apuntar la camara al punto inicial"""
    def get(self, request, *args, **kwargs):
        global art1, art2, art3

        px = 6.8301
        py = 0.0
        pz = 5.5

        angulos = cinematica_inversa(px, py, pz)
        art1 = 90 - angulos['angulo1']
        art2 = 180 - angulos['angulo2']
        art3 = 90 - angulos['angulo3']
        context = {
            "angulo1" : art1,
            "angulo2" : art2,
            "angulo3" : art3
        }
        return JsonResponse(context)
```

```
def cinematica_inversa(px, py, pz):
    global l1, l2, l3

    d = ((px**2)+(py**2)+((pz-l1)**2)-l2**2-l3**2)/(2*l2*l3)
    theta1 = math.atan2(py,px)
    theta3 = math.atan2(-math.sqrt(1-d**2),d)
    theta2 = math.atan2(pz-l1,math.sqrt(px**2+py**2)) - math.atan2(l3*math.sin(theta3),l2+l3*math.cos(theta3))

    angulo1 = mapeo(math.degrees(theta1),-90,90,0,180)
    angulo2 = mapeo(math.degrees(theta2),0,180,180,0)
    angulo3 = mapeo(math.degrees(theta3),-90,90,180,0)
```

```
    servo1.angle = angulo1
    servo2.angle = angulo2
    servo3.angle = angulo3

    context = {
        "angulo1" : angulo1,
        "angulo2" : angulo2,
        "angulo3" : angulo3
    }
    return context
```

4.3.6. Grabación de video

Para iniciar una grabación de video basta con dar clic sobre el botón Grabar, una vez iniciada la grabación cambia de color y de funcionalidad como se observa en la Figura 4.17 siendo ahora un botón para detener la grabación, esto sirve de indicador para que el operario sepa cuando está o no realizando una grabación de video.

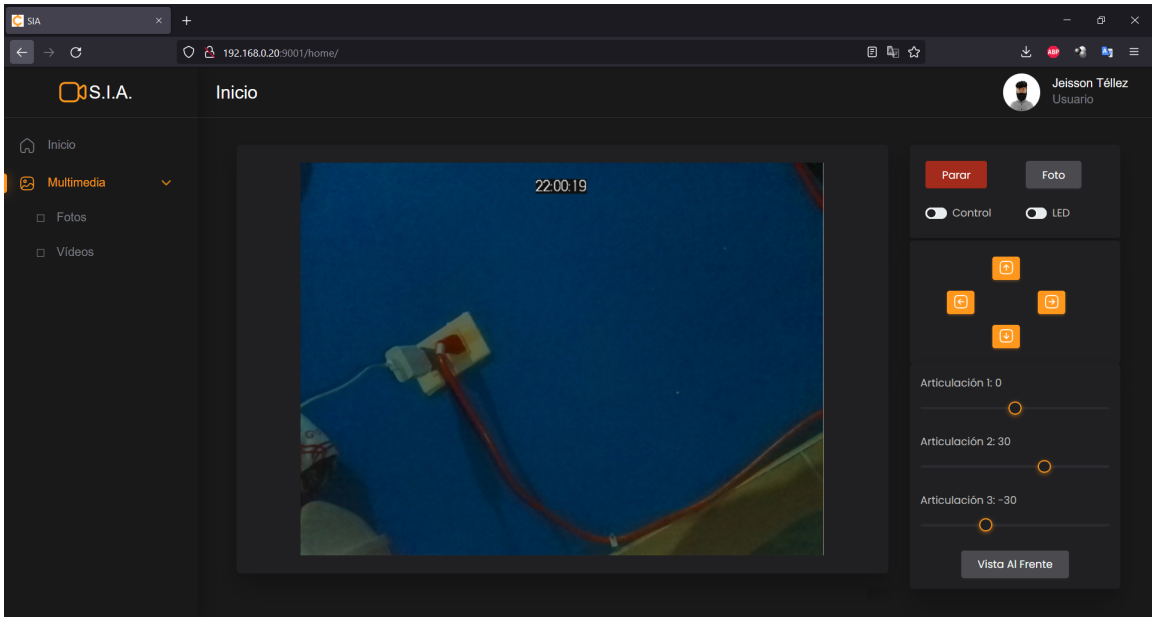


Figura 4.17: Página principal mientras realiza una grabación de video.

Respecto al procesamiento de las peticiones del cliente, se tiene una vista llamada Grabar, esta vista se encarga de recibir la petición y dependiendo de la opción se inicia o se detiene la grabación de video. Al iniciar la grabación primero se obtiene la fecha y la hora actual para colocarle el nombre al archivo, y lo guarda en la dirección respectiva. Cuando se detiene la grabación se convierte el archivo a MP4, esto es debido a que la cámara genera los videos en formato H264 el cual no es compatible con algunos sistemas operativos como Windows y de acuerdo a la norma, es requerido que los archivos de

video tengan el formato MP4. A continuación se muestra la sección de código empleada para esta vista.

```
class Grabar(View):
    """Inicia o detiene la grabacion de video"""
    def get(self, request, *args, **kwargs):
        estado = request.GET.get("estado")
        print("ESTADO", estado)
        global nombre
        path = '/home/pi/inspeccion/inspeccion/files/videos/'
        if estado == "grabar":
            nombre = time.strftime("%Y%m%d-%H%M%S.h264")
            vid = os.path.join(path, nombre)
            camara.start_recording(vid, quality=20)
            camara.annotate_text = 'REC'
            respuesta = "Grabando"
        elif estado == "parar":
            camara.annotate_text = ''
            camara.stop_recording()
            nombre2 = os.path.splitext(nombre)[0]
            vid = os.path.join(path, nombre)
            vid2 = os.path.join(path, nombre2)
            command = "MP4Box -add {} {}.mp4".format(vid, vid2)
            call([command], shell=True)
            os.remove(vid)
            respuesta = "Detenido"
        context = {"Respuesta": respuesta}
        return JsonResponse(context)
```

4.3.7. Toma de fotografías

Justo al lado del botón de Grabar, se tiene un botón llamado Foto el cual envía una petición para tomar una fotografía. Del mismo modo se tiene una vista llamada Fotos la cual hace prácticamente el mismo procedimiento que la vista anterior pero solo captura una foto. El código de esta vista se muestra a continuación.

```
class Foto(View):
    """Toma una foto para registrar la incidencia"""
    def get(self, request, *args, **kwargs):
        nombre = time.strftime("%Y%m%d-%H%M%S")
        pic = open('/home/pi/inspeccion/inspeccion/files/fotos/'+nombre+'
    .jpeg','wb')
        camara.capture(pic)
        respuesta = "Tomada"
        context = {"Respuesta": respuesta}
        return JsonResponse(context)
```

4.3.8. Descarga de archivos

Si el operario quiere acceder a los archivos mientras sigue con el proceso de inspección, puede acceder a la página de multimedia que cuenta con las secciones de foto y video como se muestra en las Figuras 4.18 y 4.19, desde allí puede ver el nombre con el que se guarda cada archivo, su peso en MB y un botón de descarga. Para el caso del prototipo se optó por poner como nombre del archivo la fecha y la hora en la que fue generado.

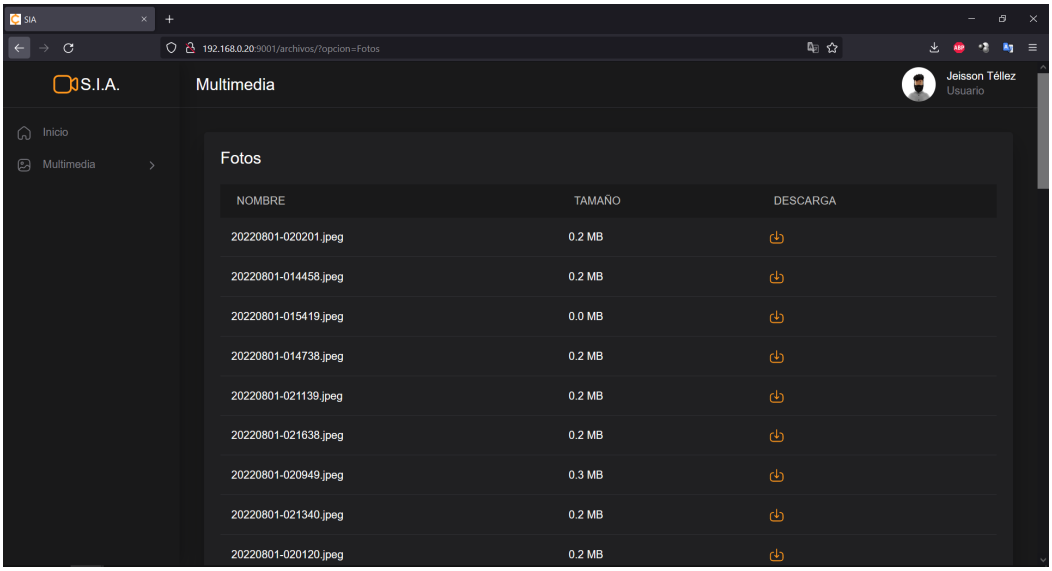


Figura 4.18: Página de multimedia - sección de fotos.

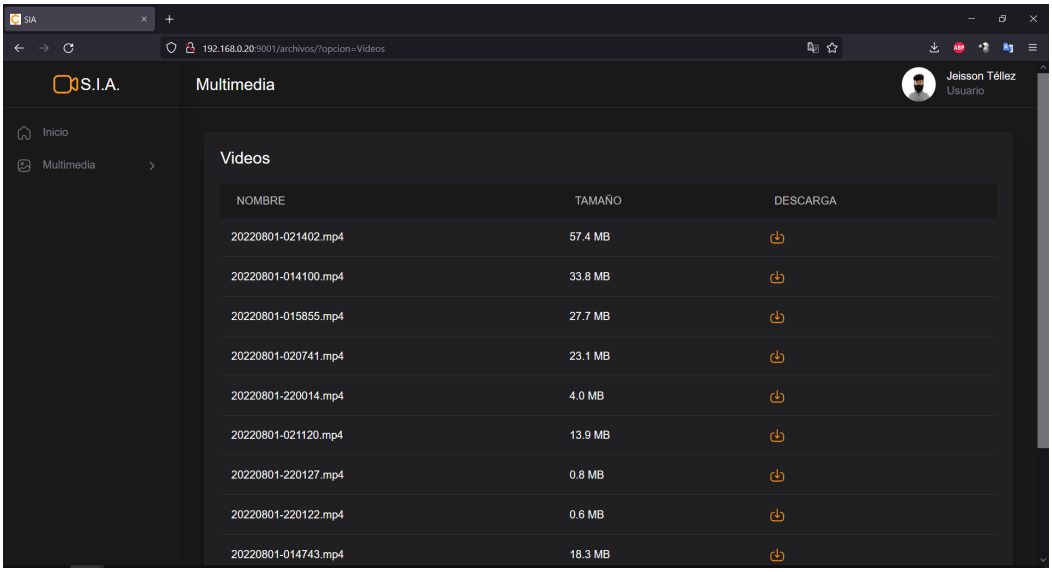


Figura 4.19: Página de multimedia - sección de videos.

Para esto se tiene una vista llamada Archivos, la cual se encarga de listar los archivos dependiendo de si se quieren las fotos o los videos. Lo que hace la vista es ir a la dirección donde se están almacenando los archivos y listar todos los nombres junto al peso y retornarlo a la plantilla para mostrarlos visualmente junto a un botón que permite realizar la descarga del archivo. El código empleado para esta vista se muestra a continuación.

```
class Archivos(View):
    """Entrega la lista de archivos almacenados (fotos o videos)"""
    def get(self, request, *args, **kwargs):
        template_name = 'multimedia.html'
        opcion = request.GET.get("opcion")
        ruta = '/home/pi/inspeccion/inspeccion/files/'
        archivos = []
        tamano = []
        if opcion == 'Fotos':
            contenido = os.listdir(ruta+'fotos')
            for c in contenido:
                size = os.path.getsize(ruta+'fotos/'+c)
                tamano.append(round(size/10**6,1))
        else:
            contenido = os.listdir(ruta+'videos')
            for c in contenido:
                size = os.path.getsize(ruta+'videos/'+c)
                tamano.append(round(size/10**6,1))
        for i in range(0,len(contenido)):
            archivos.append(
                (contenido[i], tamano[i]),
            )
        context = {
            "archivos": archivos,
            "opcion": opcion
        }
        return render(request, template_name, context)
```

Del mismo modo se tiene una vista llamada Descarga encargada de despachar los archivos cuando se pulsa el botón de descarga, para ello recibe el nombre del archivo y dependiendo de si es una foto o video se recupera de la respectiva ruta para luego responder con él. A continuación se muestra el código de esta vista.

```
class Descarga(View):
    """Retorna el archivo seleccionado"""
    def get(self, request, *args, **kwargs):
        nombre_archivo = request.GET.get("nombre")
        tipo = request.GET.get("tipo")
```

```
        if tipo == "Fotos":
            ruta = '/home/pi/inspeccion/inspeccion/files/fotos/'
        else:
            ruta = '/home/pi/inspeccion/inspeccion/files/videos/'

        file = open(ruta+nombre_archivo,'rb')
        response = FileResponse(file)
        response['Content-Type']='application/octet-stream'
        response['Content-Disposition']='attachment; filename="'+
    nombre_archivo+'"'

        return response
```

4.3.9. Transmisión de video

A pesar de que el streaming de vídeo sobre la web es complicado, por medio del protocolo HTTP es posible realizar una transmisión de vídeo en tiempo real sobre una página web. Para ello lo que se hace es crear una clase encargada de almacenar cada frame captado por la cámara dentro de un buffer el cual, al completarse con todos los datos del frame, notifica al la vista encargada de despachar la imagen que está disponible y se transmite para se renderizado sobre la página web. Con esto se logra un efecto de transmisión de vídeo en tiempo real aunque pueden buscarse métodos para una transmisión en tiempo real verdadera que tenga menos latencias, pero para el prototipo y los objetivos del proyecto es más que suficiente. A continuación se muestra el código empleado para la transmisión de video.

```
class StreamingOutput():
    """Se encarga de guardar cada frame del video en un buffer para ser
    transmitido"""
    def __init__(self):
        self.frame=None
        self.buffer=io.BytesIO()
        self.condition=Condition()

    def write(self,buf):
        if buf.startswith(b'\xff\xd8'):
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)


class Stream(View):
    """Vista para realizar la transmision de video al template"""
    def get(self,request, *args, **kwargs):
        def cam():
```

```
            output=StreamingOutput()
23          #camara.rotation=180
            camara.start_recording(output, splitter_port=2, format='mjpeg
    ')
25          try:
                while True:
27                  with output.condition:
                        output.condition.wait()
29                      frame = output.frame
                        h = time.strftime("%H:%M:%S")
31                      camara.annotate_text = h
                    yield(b'--FRAME\r\n'b'Content-Type: image/jpeg\r\n\r\
    n' + frame + b'\r\n')
33          finally:
                camara.stop_recording(splitter_port=2)
35      resp = StreamingHttpResponse(cam(), status=200, content_type='
    multipart/x-mixed-replace; boundary=FRAME')
        return resp
```

4.4. Diseño de controladores para los motores

Para el diseño de los controladores de los motores, partimos de las características de este, sabiendo que tiene un voltaje de trabajo de 6V a 12V, donde corresponde a 32 RPM a 64 RPM respectivamente, hacemos una aproximación lineal, aún sabiendo que **NO** existe un motor lineal real, sin embargo, por motivos de aproximación, se hace esta consideración.

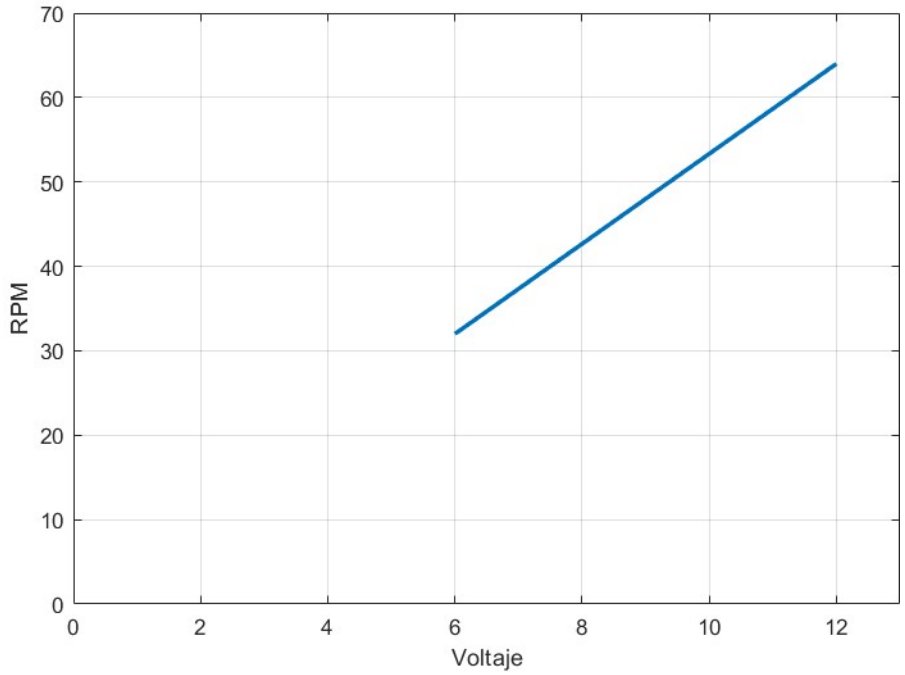


Figura 4.20: Aproximación lineal ideal

El primer paso a seguir es la caracterización de los motores, para ello, con ayuda del arduino que permita encontrar esta caracterización. Se realiza el control de los motores DC utilizando un control digital, con ayuda de un regulador y el uso de dos puente H, los cuales ayudarán al sentido de giro con señales digitales y será la etapa de potencia del sistema.

En cuanto al diseño de control, se optó por un controlador proporcional-integral [33], debido a que el error en estado estable de un sistema de control puede ser mejorado directamente, colocando un polo en el origen en el camino de transferencia directa [34], y con ayuda de la ganancia proporcional, podremos asegurar el LGR del sistema. El control digital requiere algunas mediciones para poder realizar un diseño adecuado, por ello, y porque cada motor cuenta con una función de transferencia diferente, es conveniente numerarlos para tener un control especifico, que brinde un comportamiento similar entre los motores en lazo cerrado. Para ello especifica cada uno de los valores que son utilizados para la caracterización de los sistemas.

Un control de lazo cerrado aplicado sobre un sistema (en este caso un motor) es de la forma que se observa en la Figura 4.21.

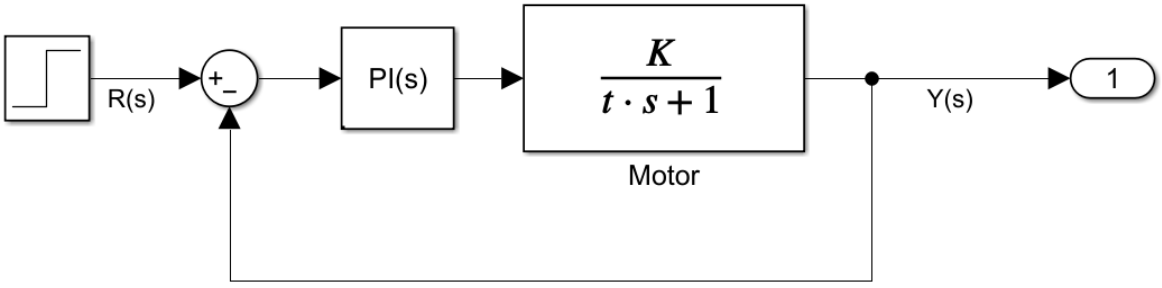


Figura 4.21: Control PI tiempo continuo.

En donde se conoce la función de transferencia de un sistema de primer orden, caracterizada de la siguiente manera

$$\frac{H(s)}{\alpha(s)} = \frac{K}{\tau s + 1} e^{-\theta s} \tag{4.1}$$

Donde:

- H(S) = Salida del sistema (velocidad medida gracias a encoders)
- α(s) = Entrada del sistema (señal generada por el controlador)
- K = Ganancia estática del sistema de primer orden
- τ = La constante de tiempo del sistema

- θ = Retardo de tiempo del sistema

Se supondrá un retardo de $\theta = 0$, por lo que queda de la forma

$$\frac{H(s)}{\alpha(s)} = \frac{K}{\tau s + 1} \tag{4.2}$$

La ganancia estática del sistema, se obtiene de la siguiente manera

$$K = \frac{H_{max} - H_{min}}{\alpha_{max} - \alpha_{min}} \tag{4.3}$$

En donde las salidas del sistema y entradas son medidas en velocidad lineal. Teniendo en cuenta que en un sistema creciente, la constante de tiempo es el periodo que tarda la respuesta al escalón del sistema en alcanzar $1 - \frac{1}{e} \approx 63,2\,\%$ de su valor final (asintótico) (ya que se caracterizará a partir de un aumento escalonado). Por ello τ es el tiempo que le toma al sistema en llegar al 63.2 % del estado estable, el cual para su medición se obtiene de la siguiente manera

$$\tau = A \cdot K \cdot 0{,}632 \tag{4.4}$$

Obteniendo el τ sabemos la relación que tiene con el tiempo de establecimiento y estado estacionario, la cual se observa en la Tabla 4.1.

Tabla 4.1: Relaciones del tiempo establecimiento.

t	$h(t)$
0	0
τ	$AK(1 - e^{-1}) = 0{,}632AK$
2τ	$AK(1 - e^{-2}) = 0{,}865AK$
3τ	$AK(1 - e^{-3}) = 0{,}95AK$
4τ	$AK(1 - e^{-4}) = 0{,}982AK$
5τ	$AK(1 - e^{-4}) = 0{,}993AK$
∞	$AK(1 - e^{\infty}) = AK$

Con ello, podemos decir que A · K será igual al valor del estado estable del sistema, el cual tiene una función que se comporta de la forma que se muestra en la Figura 4.22.

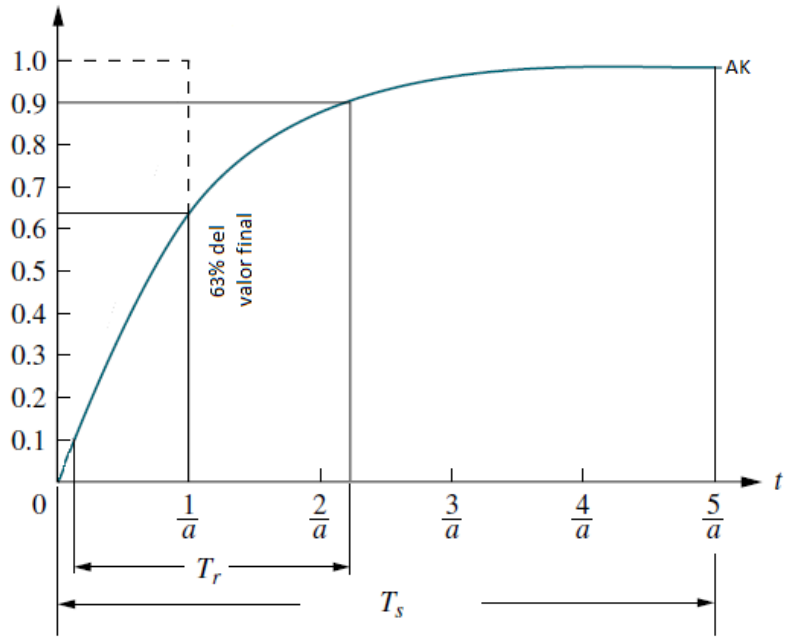


Figura 4.22: Sistema de primer orden.

4.4.1. Caracterización

Se deben encontrar primero las constantes de cada una de las funciones de transferencia del sistema, para esto, se implementa un código en Arduino que nos permite medir la salida de este sistema y poder encontrar las constantes que lo caracterizan; Midiendo los tiempos entre los pulsos que genera el encoder magnético de cada motor, es posible calcular la frecuencia angular a la que está girando el motor, con esa velocidad y conociendo el diámetro de las llantas es posible obtener también la velocidad lineal a la cual se desplaza cada llanta. Esta velocidad se envía por comunicación serial a Matlab para luego ser graficadas como se observa en las figuras. El código utilizado en Arduino se encuentra en el Anexo A: Código implementado en Arduino para caracterizar los motores.

Para realizar el diseño, primero se plantean unos valores deseados los cuales resultan en un diseño inicial, que si no cumple con las especificaciones de diseño, se irá ajustando para así obtener una respuesta más cercana a la deseada. El tiempo de establecimiento se usará como $ts = 5\tau$ ya que es una buena aproximación al sistema. Partimos definiendo un tiempo de establecimiento y un sobrepaso máximo deseados

$$ts_d = 0{,}75 * 5\tau \tag{4.5}$$

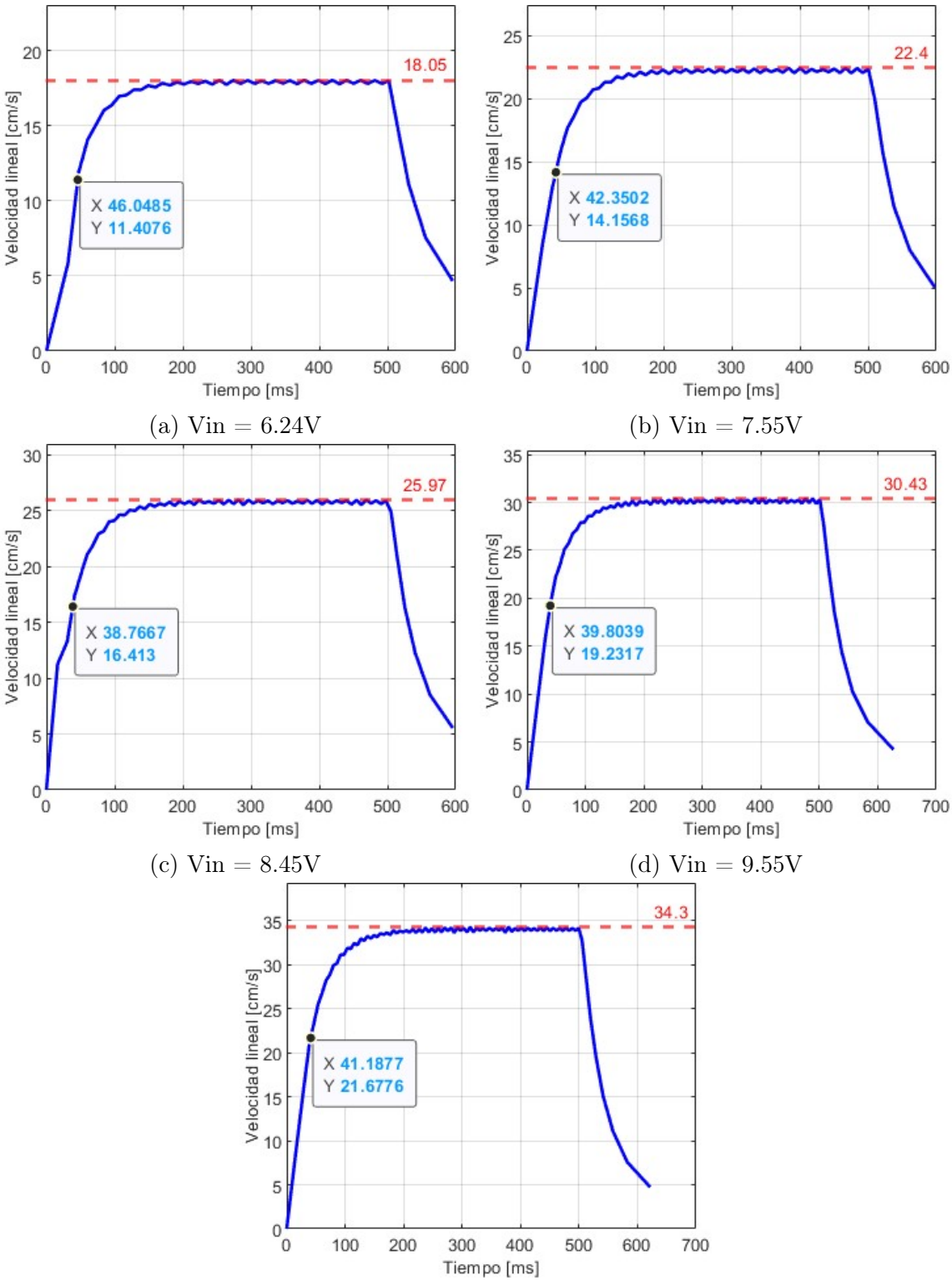
$$Mp_d = 5\,\% \tag{4.6}$$

$$\frac{tr}{12} < T < \frac{tr}{8} \tag{4.7}$$

Donde:

- ts_d = tiempo de establecimiento deseado en lazo cerrado
- ts = tiempo de establecimiento
- Mp_d = sobrepaso máximo
- td = tiempo de levantamiento
- T = tiempo de muestreo

Teniendo claro la manera que se establece el tiempo de muestreo, y la caracterización del sistema, se procede a graficar la velocidad lineal vs el tiempo los resultados obtenidos con ayuda del Arduino.



(a) Vin = 6.24V

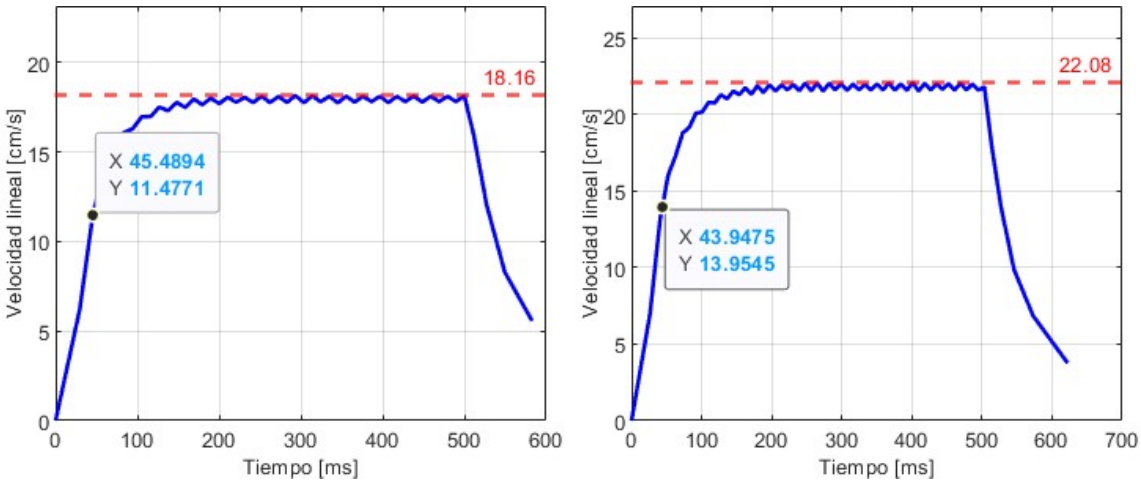
(b) Vin = 7.55V

(c) Vin = 8.45V

(d) Vin = 9.55V

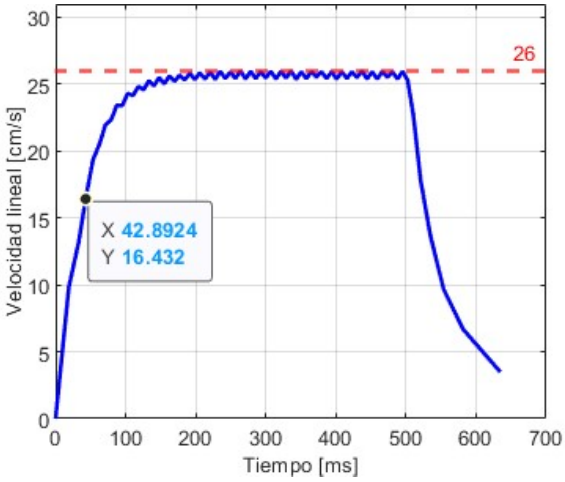
(e) Vin = 10.58V

Figura 4.23: Respuesta del motor 1.

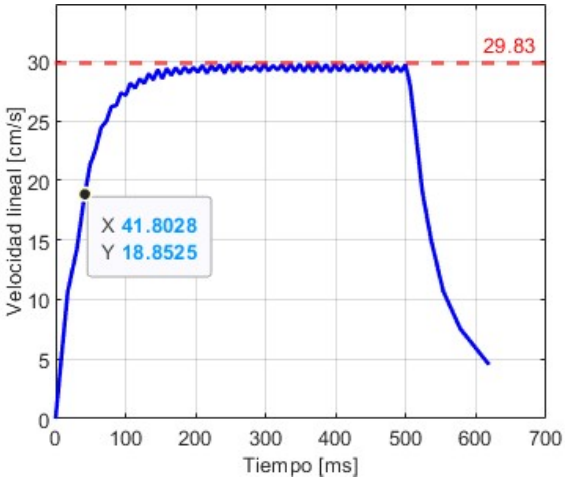


(a) Vin = 6.48V

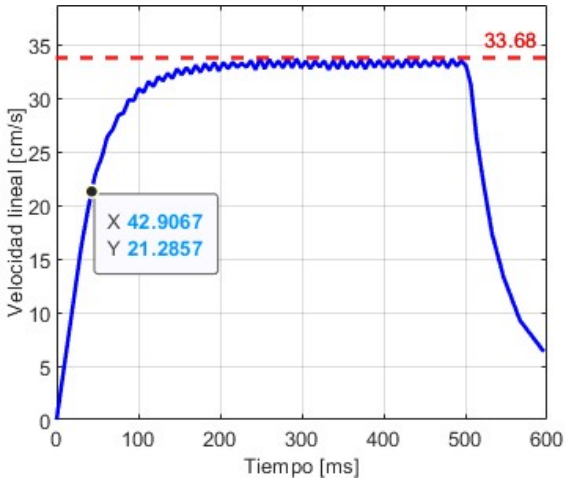
(b) Vin = 7.48V



(c) Vin = 8.50V

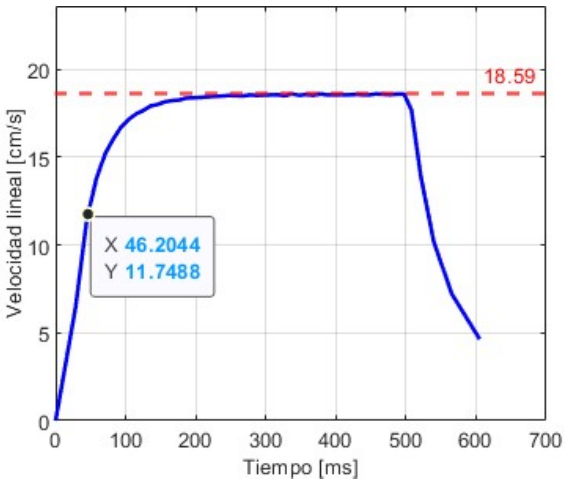


(d) Vin = 9.48V

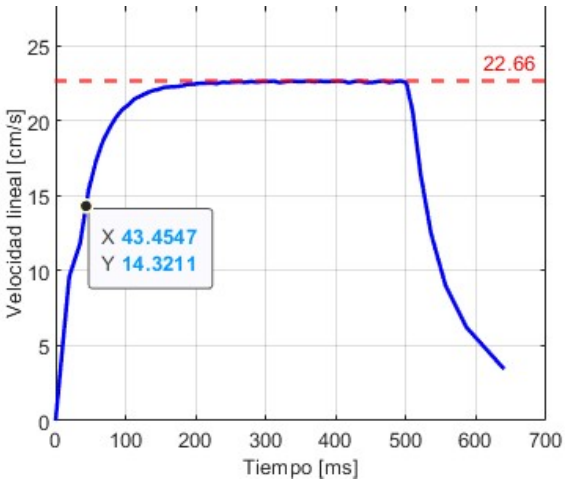


(e) Vin = 10.59V

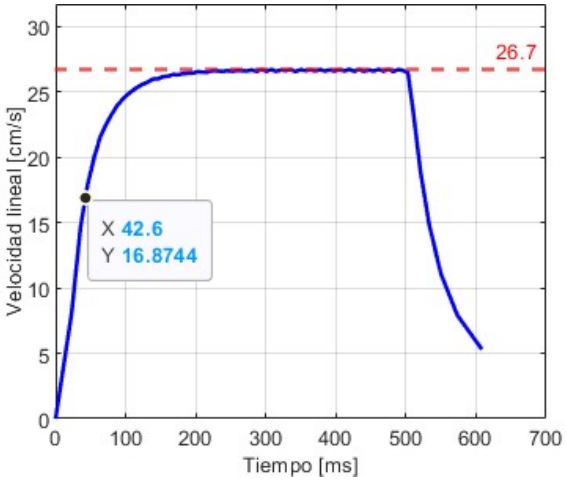
Figura 4.24: Respuesta del motor 2.



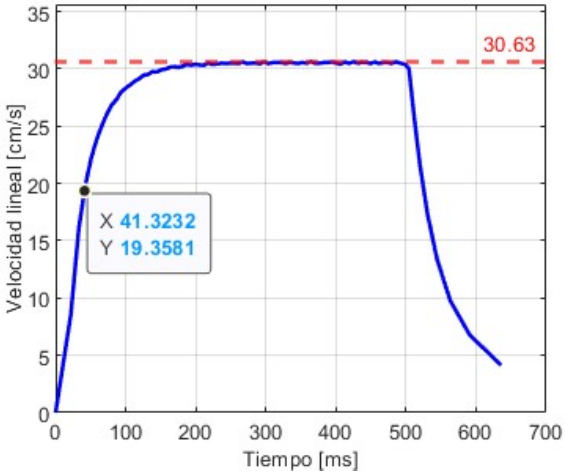
(a) Vin = 6.47V



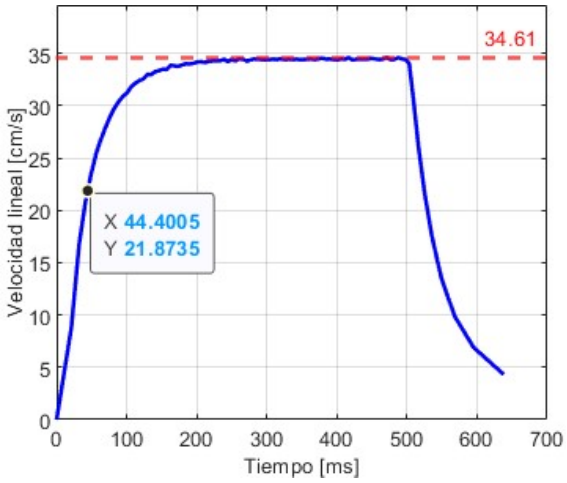
(b) Vin = 7.50V



(c) Vin = 8.49V

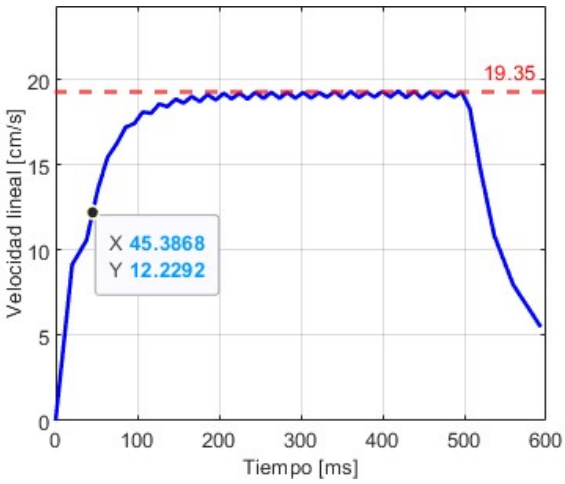


(d) Vin = 9.48V

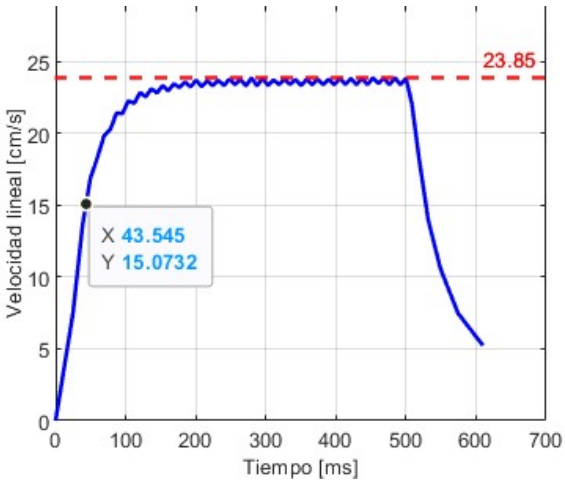


(e) Vin = 10.51V

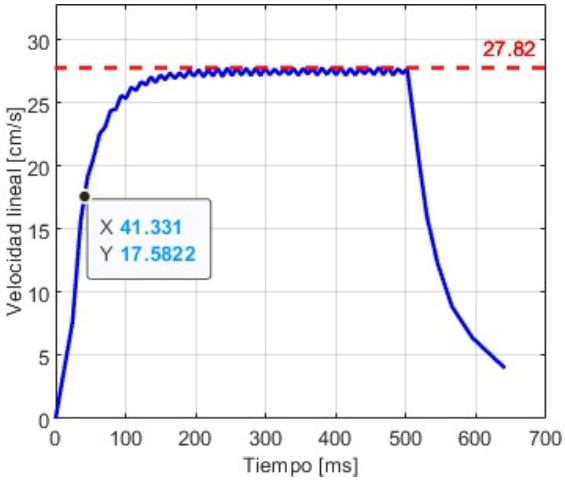
Figura 4.25: Respuesta del motor 3.



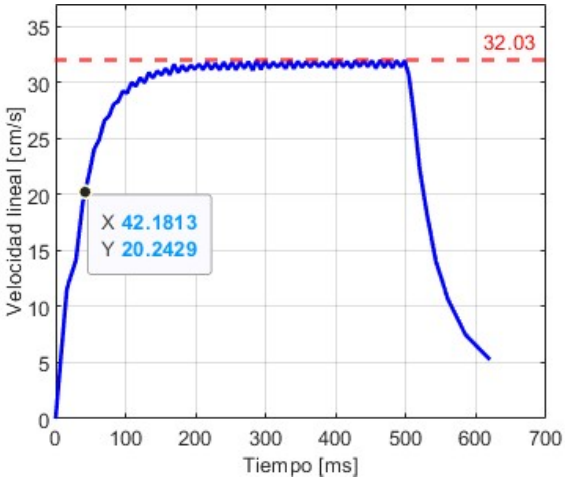
(a) Vin = 6.45V



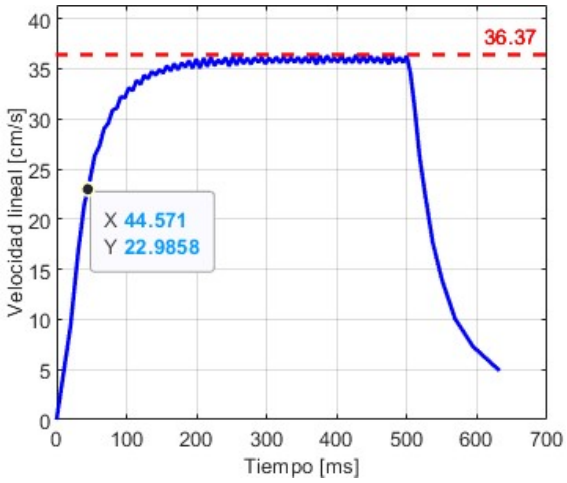
(b) Vin = 7.52V



(c) Vin = 8.49V



(d) Vin = 9.55V



(e) Vin = 10.65V

Figura 4.26: Respuesta del motor 4.

Como se observa en las Figuras 4.23, 4.24, 5.6 y 4.26, las entradas al sistema se variaron para obtener diferentes valores de τ y de K. Hay que tener en cuenta que el τ a considerar, está en los valores que son más estables, que en general es entre 7 V y 10 V. Con todos estos valores y con ayuda de las Ecuaciones 4.3 y 4.4, se obtienen los datos registrados en las siguientes tablas para caracterizar el sistema.

Motor 1	τ	K
6.24V	0.0460485	0.769
7.55V	0.0423502	0.817
8.45V	0.0387667	0.846
9.55V	0.0398039	0.877
10.58V	0.0411877	0.886
Promedio	0.0405271	0.8373

Tabla 4.2: Valores de τ y K para el motor 1.

Motor 2	τ	K
6.48V	0.0454894	0.772
7.48V	0.0439475	0.813
8.50V	0.0428924	0.842
9.48V	0.0418028	0.866
10.59V	0.0429067	0.876
Promedio	0.0428873	0.8323

Tabla 4.3: Valores de τ y K para el motor 2.

Motor 3	τ	K
6.47V	0.0462044	0.791
7.50V	0.0434547	0.835
8.49V	0.0426	0.866
9.48V	0.0413232	0.890
10.51V	0.0444005	0.907
Promedio	0.0429446	0.8578

Tabla 4.4: Valores de τ y K para el motor 3.

Motor 4	τ	K
6.45V	0.0453868	0.826
7.52V	0.043545	0.873
8.49V	0.041331	0.901
9.55V	0.0421813	0.923
10.65V	0.044571	0.940
Promedio	0.0429070	0.8926

Tabla 4.5: Valores de τ y K para el motor 4.

Gracias a la ecuación 4.2, se halla la ecuación del sistema de primer orden con ayuda de la respuesta de un escalón unitario, por lo que la función, al utilizar otra entrada (e) se multiplicará esta ganancia.

$$\frac{H(s)}{\alpha(s)} = \frac{K * e}{\tau s + 1} \tag{4.8}$$

Ahora bien, nuestra entrada debe corresponder a las medidas tomadas previamente, por ello es necesario convertir la entrada de Voltaje a Velocidad lineal, para ello se utiliza la siguiente ecuación.

$$e = V_{in} * W * (D/13) \tag{4.9}$$

$$W = (\frac{RPM_{min-max}}{Vtm_{min-max}}) * \frac{2 * \pi}{60} \tag{4.10}$$

Donde:

- e = Velocidad lineal del motor
- V_{in} = Voltaje de entrada
- $RPM_{min-max}$ = Revoluciones por minuto (mínima ó máxima)
- $Vtm_{min-max}$ = Voltaje trabajo del motor (mínima ó máxima)
- D = Diametro de la rueda

Sabiendo esto, despejamos de la 4.9 y queda de la forma

$$e = V_{in} * 3{,}6303 \tag{4.11}$$

Partiendo de las tablas 4.2, 4.3, 4.4 y 4.5, se procede a graficar y comparar con las figuras de las respuestas de los motores, para poder determinar una buena caracterización.

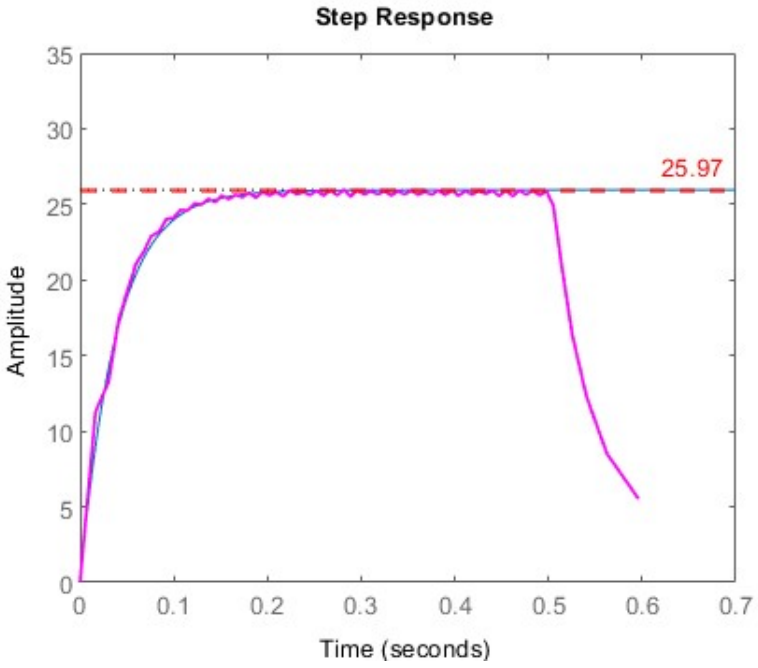


Figura 4.27: Función aproximada vs Valor Real Motor 1

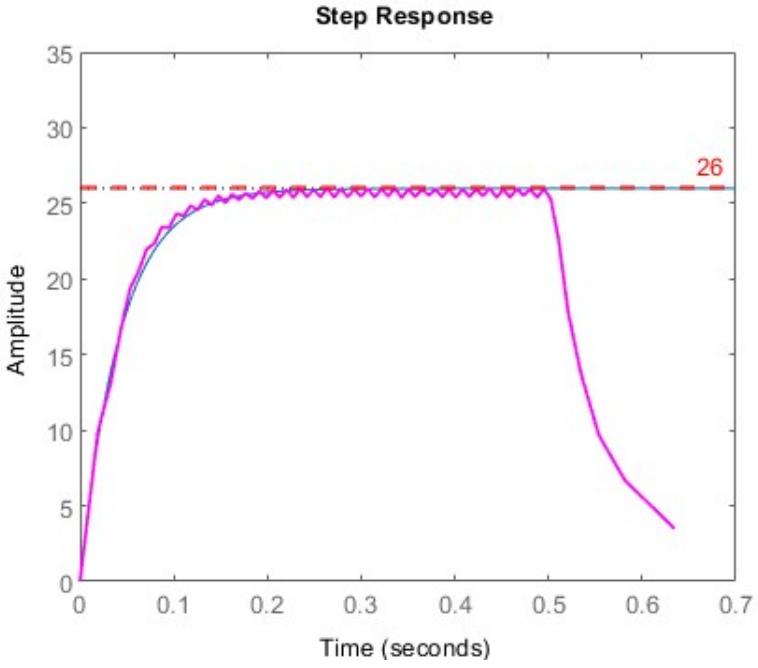


Figura 4.28: Función aproximada vs Valor Real Motor 2

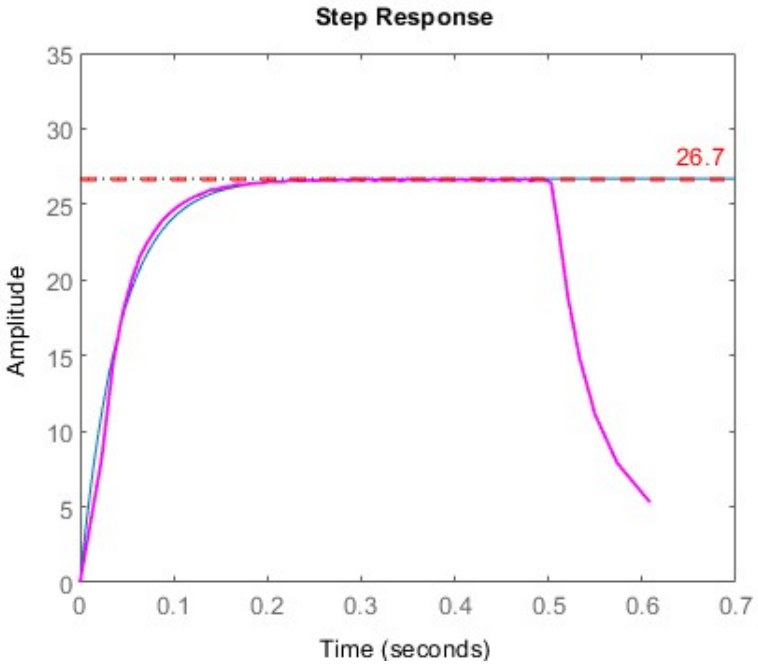


Figura 4.29: Función aproximada vs Valor Real Motor 3

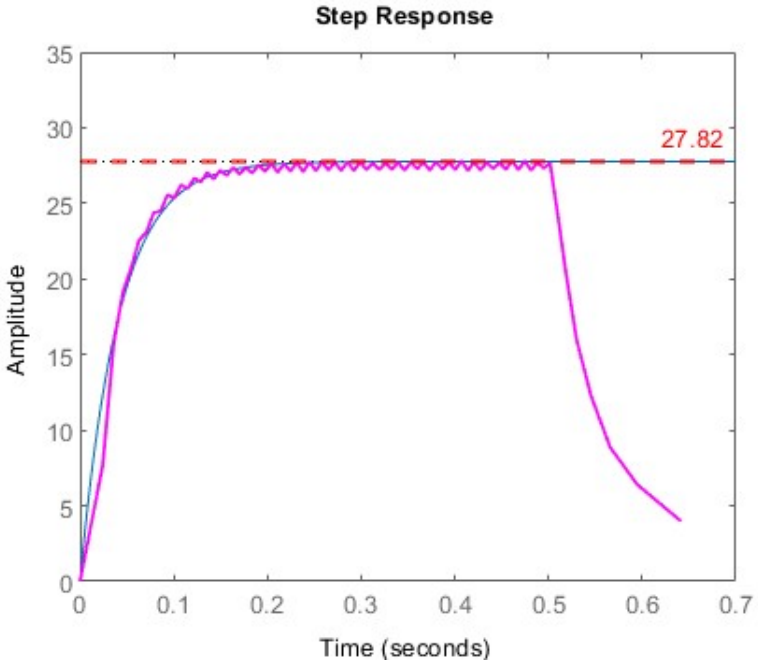


Figura 4.30: Función aproximada vs Valor Real Motor 4

4.4.2. Diseño del control digital

Al discretizar se tiene el sistema mostrado en la Figura 4.31.

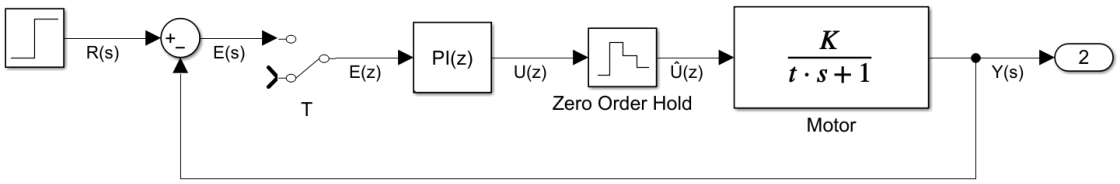


Figura 4.31: Sistema con un control discreto.

En la cual podemos deducir que:

$$\frac{U(z)}{E(z)} = \frac{k(1 - a \cdot z^{-1})}{1 - z^{-1}} \tag{4.12}$$

Al aplicar transformada Z inversa a la ecuación (8) se obtiene

$$u(k) - u(k-1) = k \cdot e(k) - a \cdot k \cdot e(k-1) \tag{4.13}$$

$$u(k) = u(k-1) + k \cdot e(k) - a \cdot k \cdot e(k-1) \tag{4.14}$$

en donde

- u(k)= salida del sistema
- e(k)= error del sistema
- u(k-1)= salida anterior (se inicializa en 0)
- e(k-1)= error anterior (se inicializa en 0)

Para el diseño de los controladores se utiliza la herramienta Matlab, que facilita los cálculos y permite hacer simulaciones de los diseños de control que se realizan. Se parte de una tiempo de establecimiento y un sobrepaso máximo deseados como valores de diseño, se observa que el periodo de muestreo en general es $T = 0{,}0035$.

Para esto, se hallan los ts_d y los polos deseados discretizados de los sistemas con el diseño elegido, obteniendo de esta forma la Tabla 4.6.

	ts_d	z_d
Motor 1	0.1520	0.8504+0.1346i
Motor 2	0.1608	0.8586+0.1283i
Motor 3	0.1610	0.8587+0.1282i
Motor 4	0.1609	0.8586+0.1283i

Tabla 4.6: Valores deseados de diseño

Con los valores deseados de los polos y los tiempos de establecimiento, se obtienen las expresiones para los controladores en tiempo discreto:

$$Motor1 \rightarrow PI(z) = \frac{3{,}1252(z - 0{,}8129)}{z - 1} \tag{4.15}$$

$$Motor2 \rightarrow PI(z) = \frac{3{,}135(z - 0{,}8217)}{z - 1} \tag{4.16}$$

$$Motor3 \rightarrow PI(z) = \frac{3{,}0417(z - 0{,}8219)}{z - 1} \tag{4.17}$$

$$Motor4 \rightarrow PI(z) = \frac{2{,}9242(z - 0{,}8218)}{z - 1} \tag{4.18}$$

Simulando los controladores obtenidos se tienen los resultados para cada motor.

Motor 1

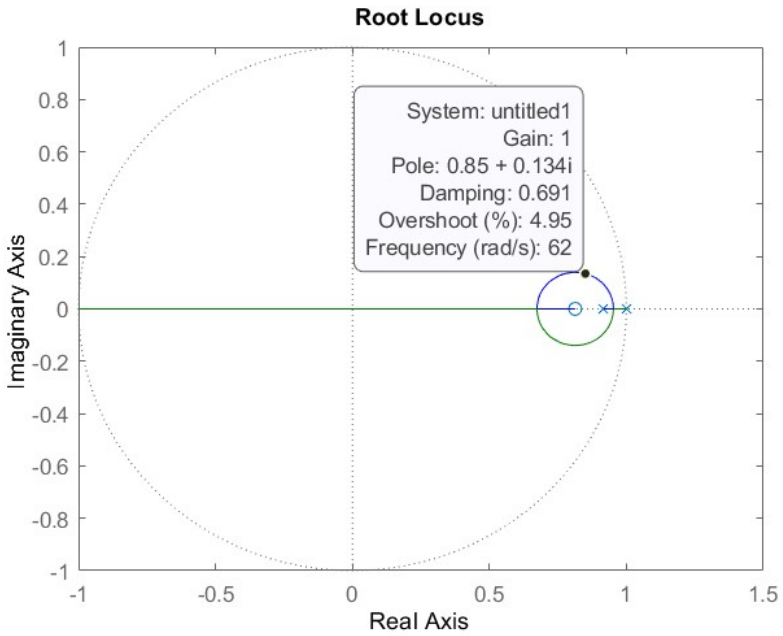


Figura 4.32: Lugar geométrico de las raíces Motor 1.

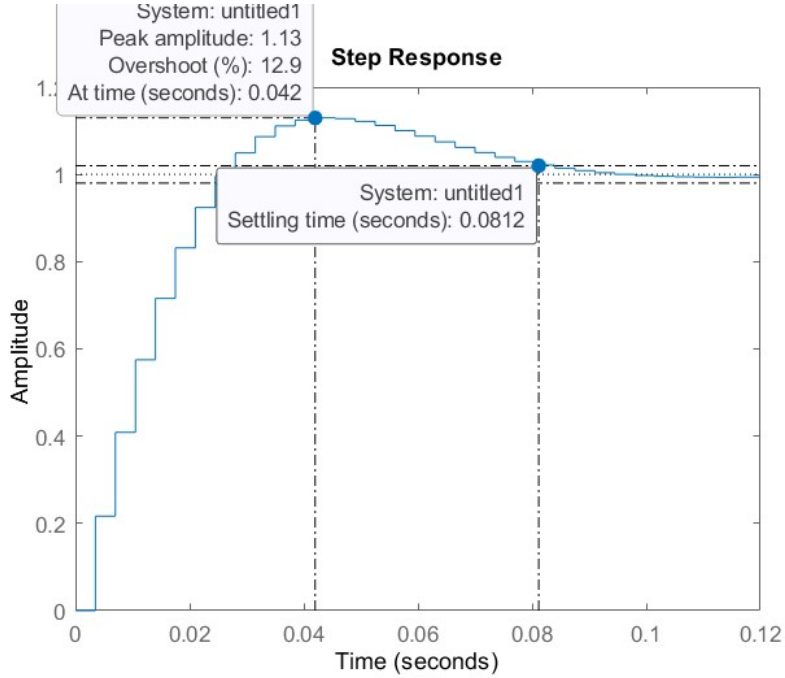


Figura 4.33: Respuesta al escalón Unitario Motor 1.

Motor 2

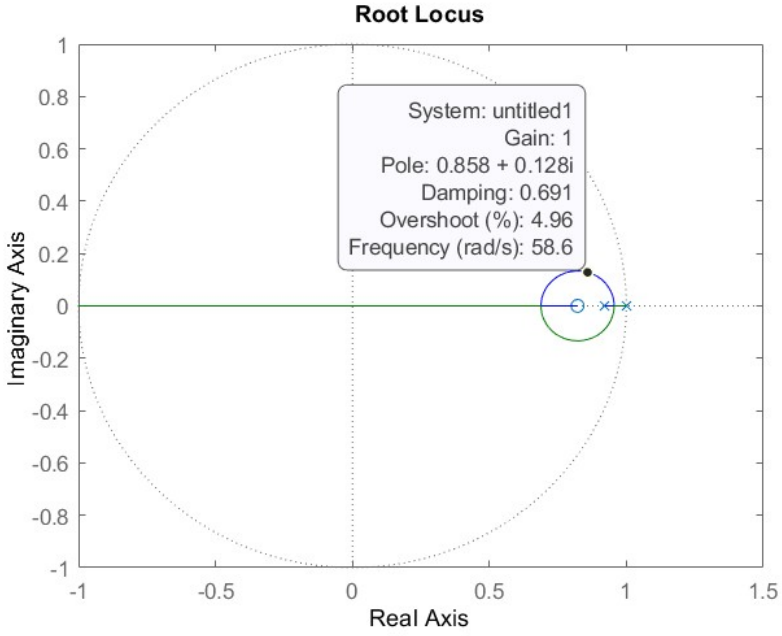


Figura 4.34: Lugar geométrico de las raíces Motor 2.

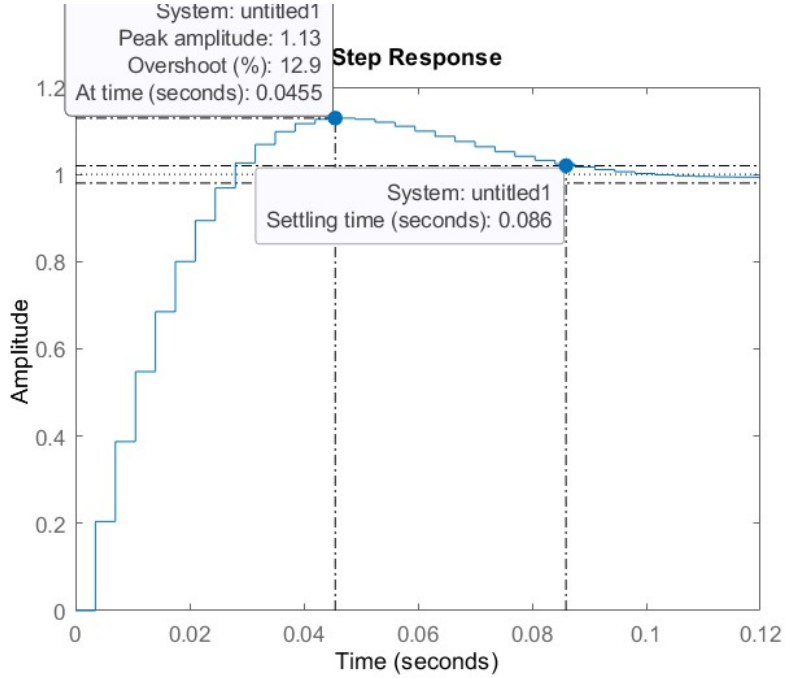


Figura 4.35: Respuesta al escalón Unitario Motor 2.

Motor 3

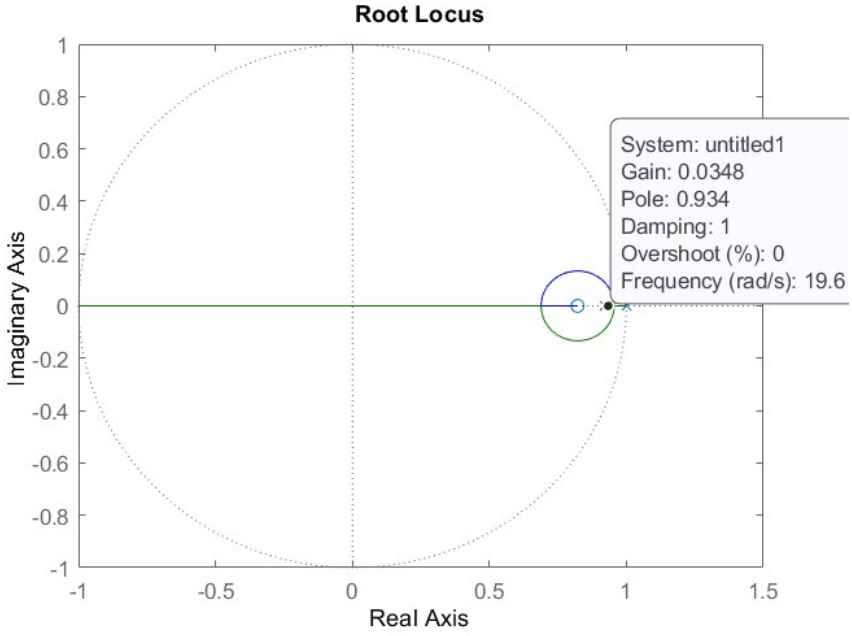


Figura 4.36: Lugar geométrico de las raíces Motor 3.

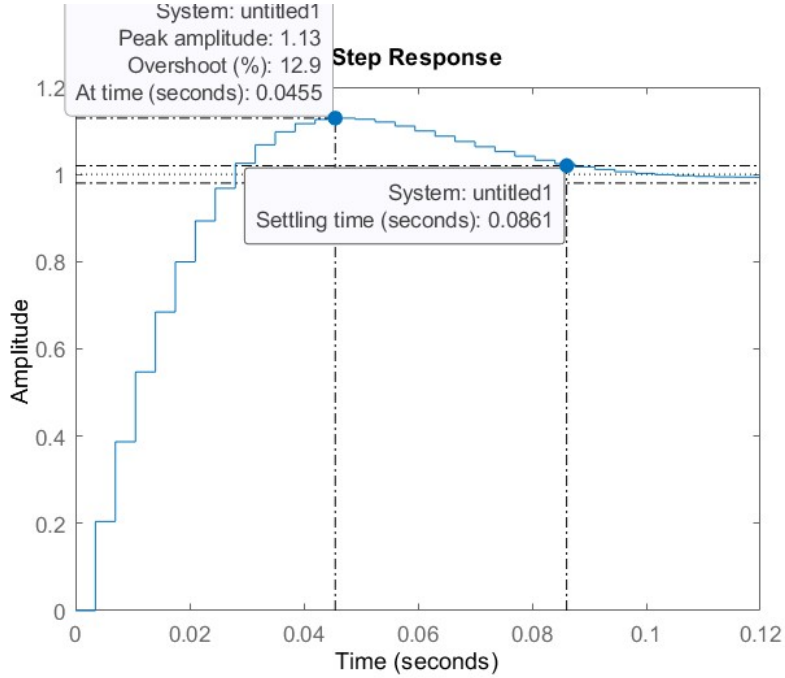


Figura 4.37: Respuesta al escalón Unitario Motor 3.

Motor 4

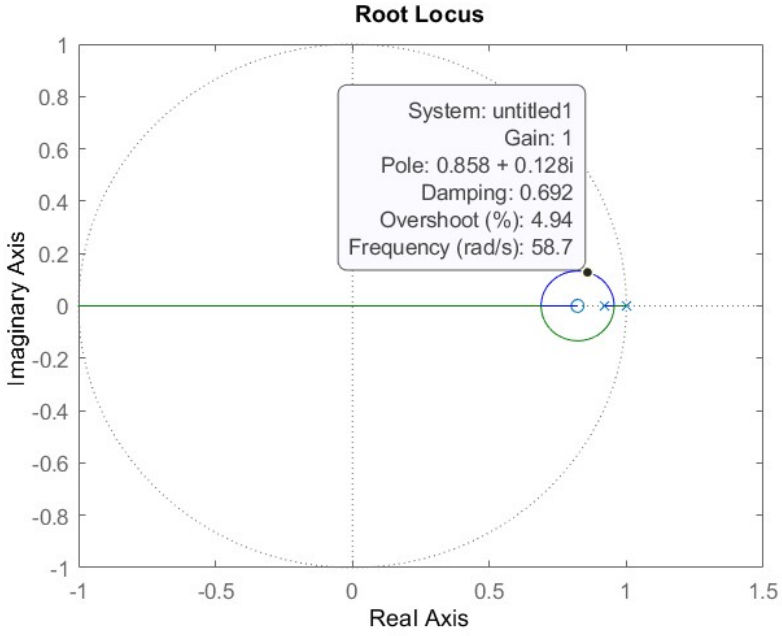


Figura 4.38: Lugar geométrico de las raíces Motor 4.

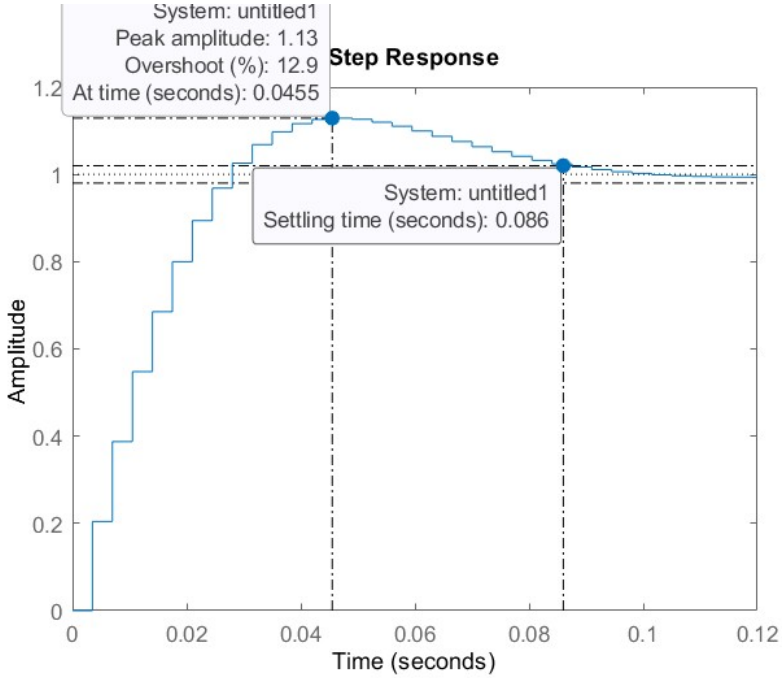


Figura 4.39: Respuesta al escalón Unitario Motor 4.

Como se observan en las Figuras 4.32, 4.34, 4.36 y 4.38, se observa el LGR, se aprecia que se implementó de manera correcta el controlador PI, ya que cumple con los Polos deseados, y con una ganancia igual a 1.

En relación a las Figuras 4.33, 4.35, 4.37 y 4.39, se observa la respuesta a una entrada escalón, de la cual se aprecia que en ninguna cumple con un sobrepaso máximo del 5 %, pero se cumple con el tiempo de establecimiento, teniendo uno mucho menor al deseado. Por ello aunque el LGR esté en el lugar adecuado, no se satisface al ingresar un escalón al sistema, por esta razón, y para poder satisfacer nuestras consideraciones iniciales, se reajustarán los valores de diseño de la siguiente manera

$$ts_d = 5\tau \tag{4.19}$$

$$Mp_d = 1\,\% \tag{4.20}$$

Al realizar este nuevo diseño, se encuentra un nuevo LGR y un periodo de muestreo $T = 0{,}007$, con este valor se rediseña los controladores, ya que cambiarán nuestros polos deseados.

	ts_d	z_d
Motor 1	0.2026	0.7895+0.1219i
Motor 2	0.2144	0.8004+0.1167i
Motor 3	0.2147	0.8006+0.1165i
Motor 4	0.2145	0.8004+0.1166

Tabla 4.7: Valores deseados de diseño

Con los valores deseados de los polos y los tiempos de establecimiento, se obtienen las nuevas expresiones para los controladores en tiempo discreto:

$$Motor1 \rightarrow PI(z) = \frac{1{,}9755(z - 0{,}7745)}{z - 1} \tag{4.21}$$

$$Motor2 \rightarrow PI(z) = \frac{1{,}9847(z - 0{,}785)}{z - 1} \tag{4.22}$$

$$Motor3 \rightarrow PI(z) = \frac{1{,}9256(z - 0{,}7853)}{z - 1} \tag{4.23}$$

$$Motor4 \rightarrow PI(z) = \frac{1{,}8498(z - 0{,}7851)}{z - 1} \tag{4.24}$$

Simulando los controladores obtenidos se tienen los resultados para cada motor.

Motor 1

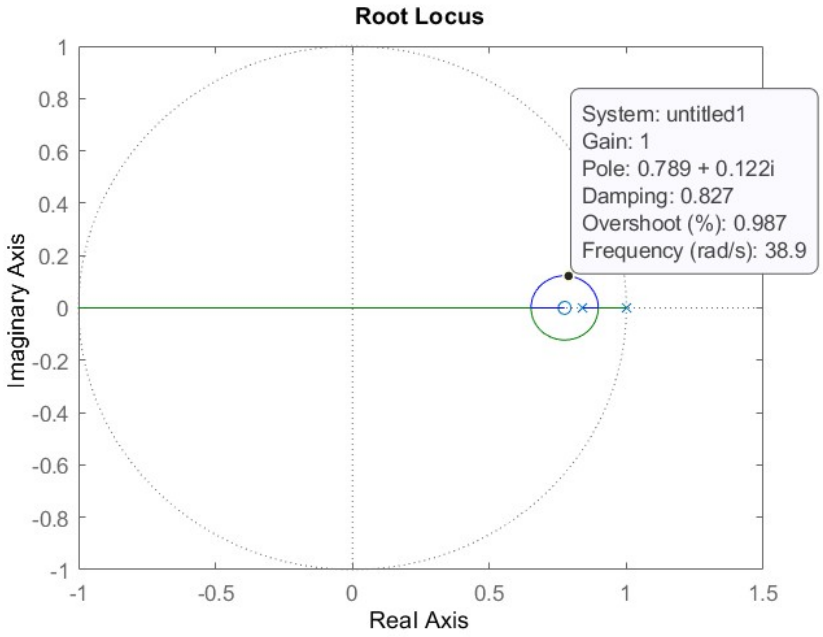


Figura 4.40: Lugar geométrico de las raíces Motor 1.

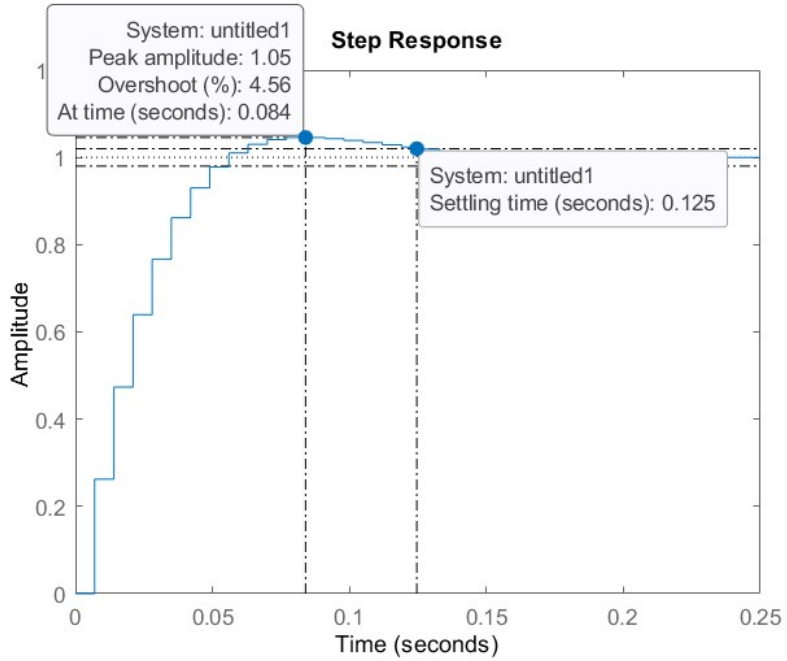


Figura 4.41: Respuesta al escalón Unitario Motor 1.

Motor 2

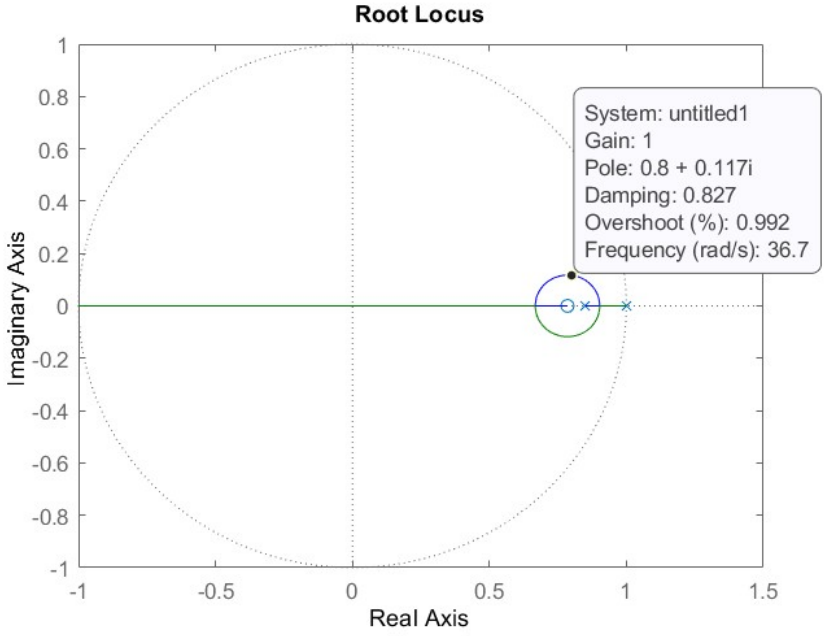


Figura 4.42: Lugar geométrico de las raíces Motor 2.

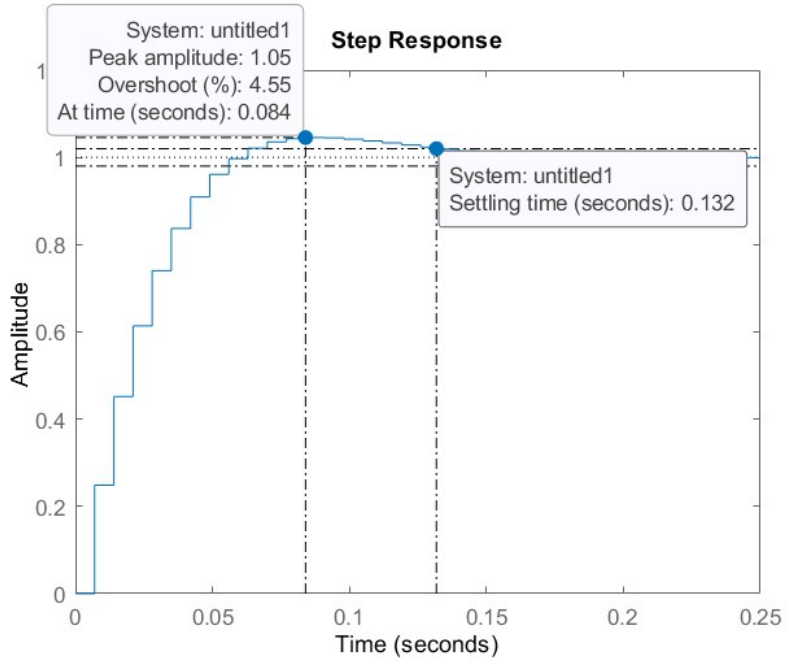


Figura 4.43: Respuesta al escalón Unitario Motor 2.

Motor 3

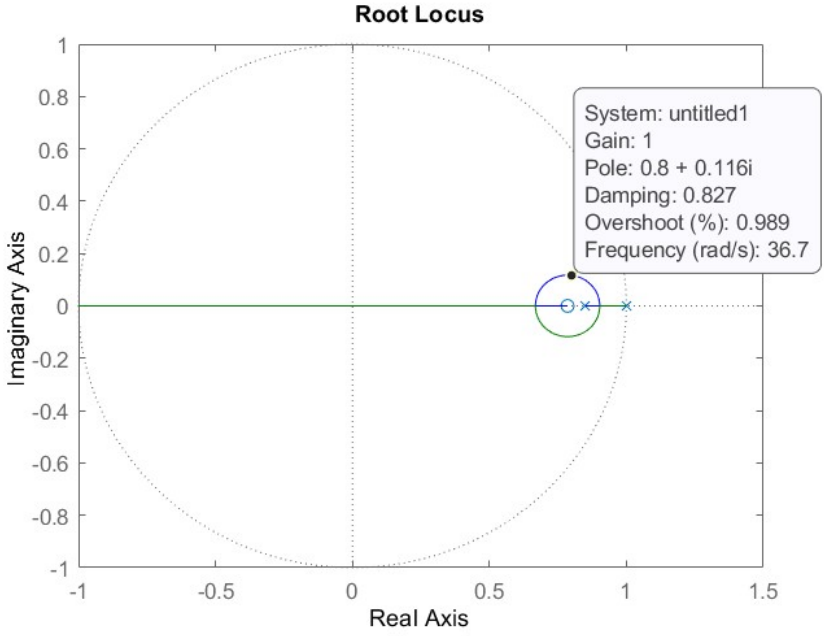


Figura 4.44: Lugar geométrico de las raíces Motor 3.

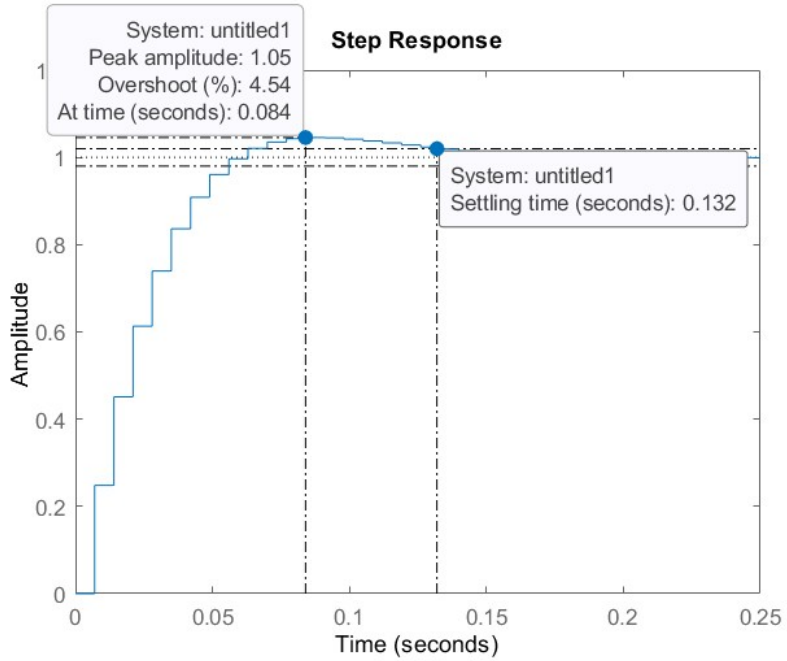


Figura 4.45: Respuesta al escalón Unitario Motor 3.

Motor 4

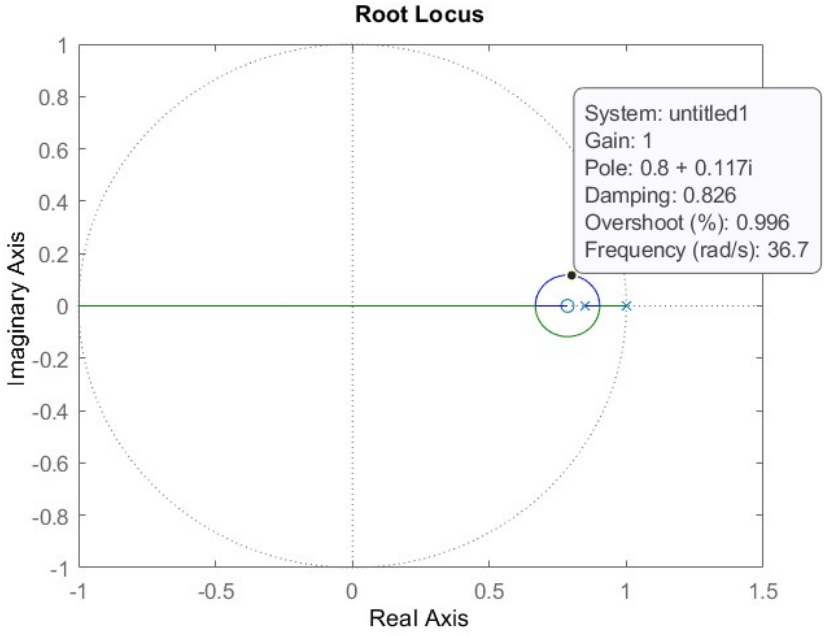


Figura 4.46: Lugar geométrico de las raíces Motor 4.

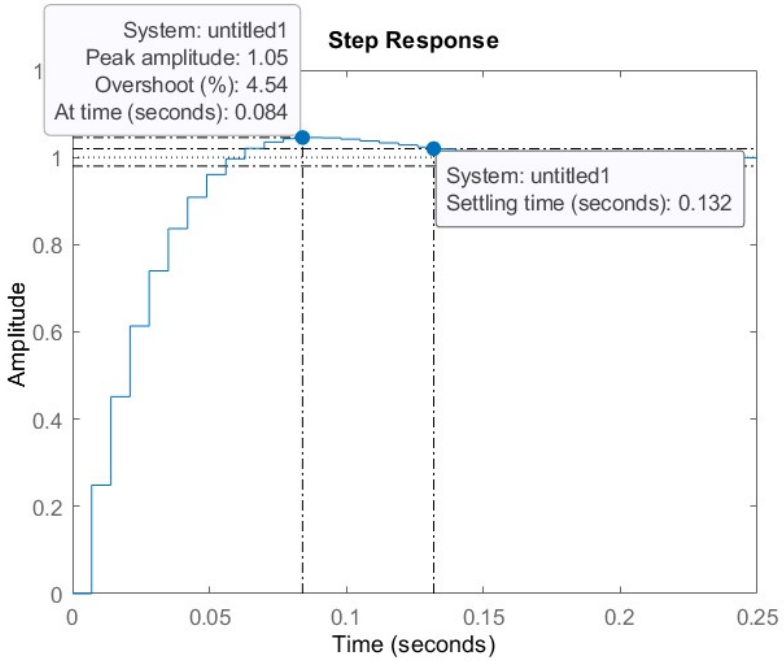


Figura 4.47: Respuesta al escalón Unitario Motor 4.

4.5. Control digital

Como se observa en las Figuras 4.41. 4.43, 4.45 y 4.47 podemos ver que la respuesta escalón cumple con el sobrepaso máximo deseado que se planeó desde un inicio, acá se evidencia que el rediseño fue exitoso y se cumple con los valores deseados. Una vez obtenidos los valores de control, se procede a realizar su implementación en el microcontrolador, introduciendo los valores en la Ecuación 4.14. El diagrama de la Figura 4.48 muestra el procedimiento que sigue el microcontrolador para obtener una señal de control con base en el error que se obtiene.

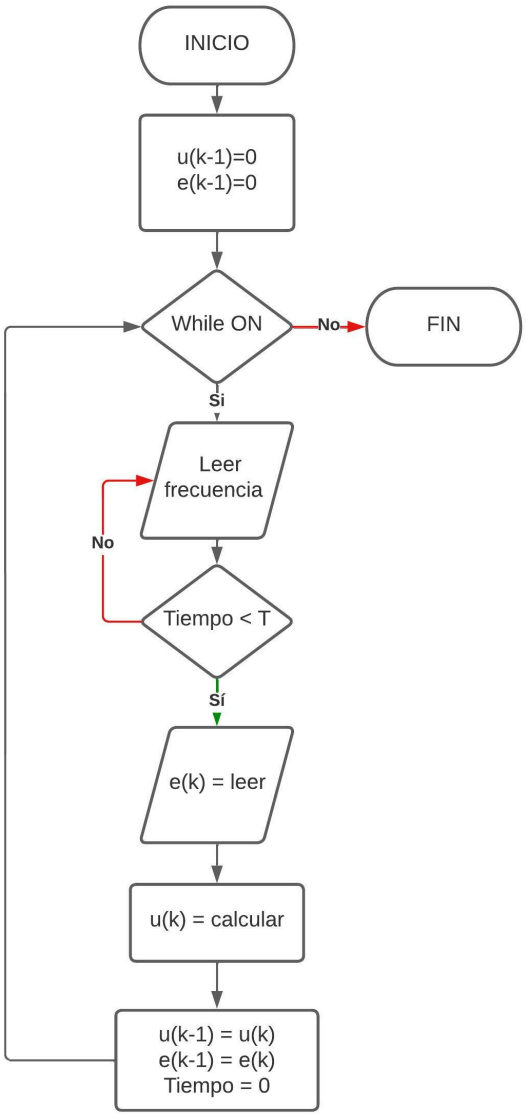


Figura 4.48: Diagrama control digital motores

La primera parte del código empleado para implementar los controladores en la Raspberry Pi se observa en el apéndice C, comienza básicamente con una función que se ejecuta cuando ocurre una interrupción la cual se activa con cada pulso que genera el encoder. Dentro de esta función se calculan las RPM de cada motor, por lo cual se tienen cuatro funciones idénticas con las respectivas variables de los cuatro motores.

Una vez se obtienen las RPM de cada motor se realiza el procedimiento mostrado en la Figura 4.48. En primer lugar se obtiene la señal de error con base en un valor de RPM de referencia y en las RPM medidas de cada motor, posterior a ello se aplica la Ecuación 10 para cada uno de los controladores obtenidos en las Ecuaciones 4.15, 4.16, 4.17 y 4.18. Una vez se obtiene la señal de control, se procede a actualizar los valores del error y la señal de control en el instante anterior. Es necesario limitar la señal de control entre dos valores, ya que es importante es tener en cuenta los límites que va a tener físicamente la señal de control. Para el caso se escoge un rango entre 0 y 500, y dicho valor se convierte a un valor de ciclo útil para la señal de PWM, en este caso es una señal PWM que va entre 0 y 4095.

5. Resultados

En la Figura 5.1 se observa el prototipo montado y funcionando como resultado de todo el procedimiento detallado en las secciones anteriores.

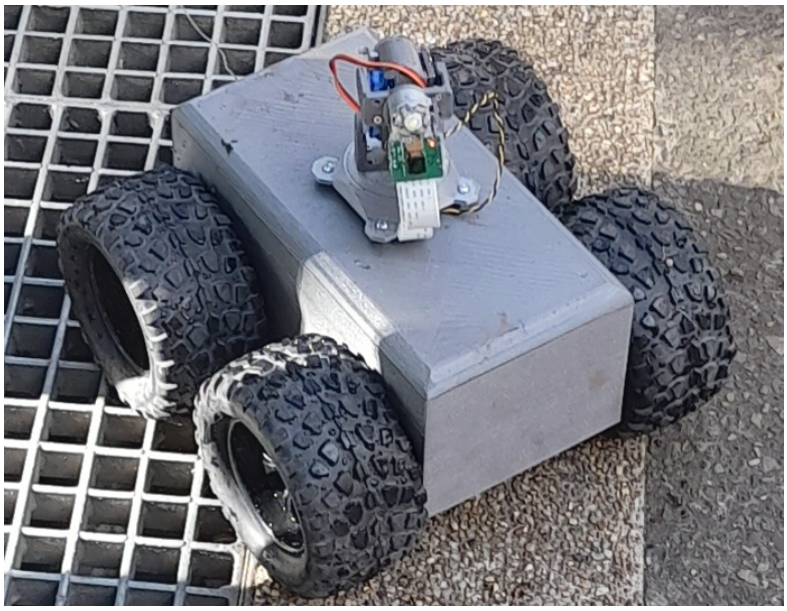


Figura 5.1: Montaje final del prototipo.

Del mismo modo se muestra un vistazo a las conexiones y componentes internos que se tienen, esto se observa en la Figura 5.2.

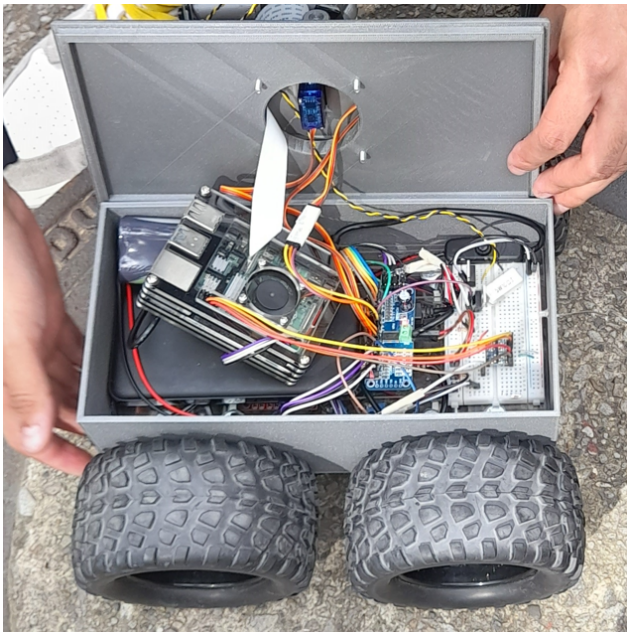


Figura 5.2: Conexiones internas del prototipo.

5.1. Pruebas de funcionamiento

Para las pruebas de campo, gracias a la gestión y ayuda del Ingeniero Civil Andrés Eduardo Torres Abella, se logró realizar una visita a la Pontificia Universidad Javeriana para utilizar la red de alcantarillado del campus y poder probar el funcionamiento del prototipo. El día 1 de Agosto del 2022, sobre las 2:00 pm (GMT-5), se accede a las instalaciones de la Pontificia Universidad Javeriana para realizar las pruebas de campo.



Figura 5.3: Apertura del pozo y preparación del prototipo para la inspección.

Se empezó realizando un análisis de las diferentes opciones de ducterías que se brindaron para seleccionar uno de fácil acceso, de los tres perfiles diferentes, se encontró que en el perfil (PZ-60 - PZ-63) Figura 2.4, contaba con un pozo de 3 metros, el cual a una altura de 1.5 metros encima de él tenía un desagüe perfil (PZ-63 - PZ-69) Figura 2.6, por lo cual se dirige hacia el lugar de la tapa para abrir y observar el estado. En la Figura 5.3 se observa el pozo abierto y la conexión realizada del router. Con ayuda de hilo de Nylon, se amarra el prototipo para tener una línea de salva y poder recuperar el vehículo en caso de que se pierda conexión o suceda algo inesperado. Con ese mismo hilo se baja al prototipo hasta el fondo del pozo para que ingrese a la tubería del fondo tal como se observa en la Figura 5.4. Sin embargo no se contaba con que justo a la entrada de la tubería existía un daño en el suelo y el vehículo no podía ingresar a la tubería.



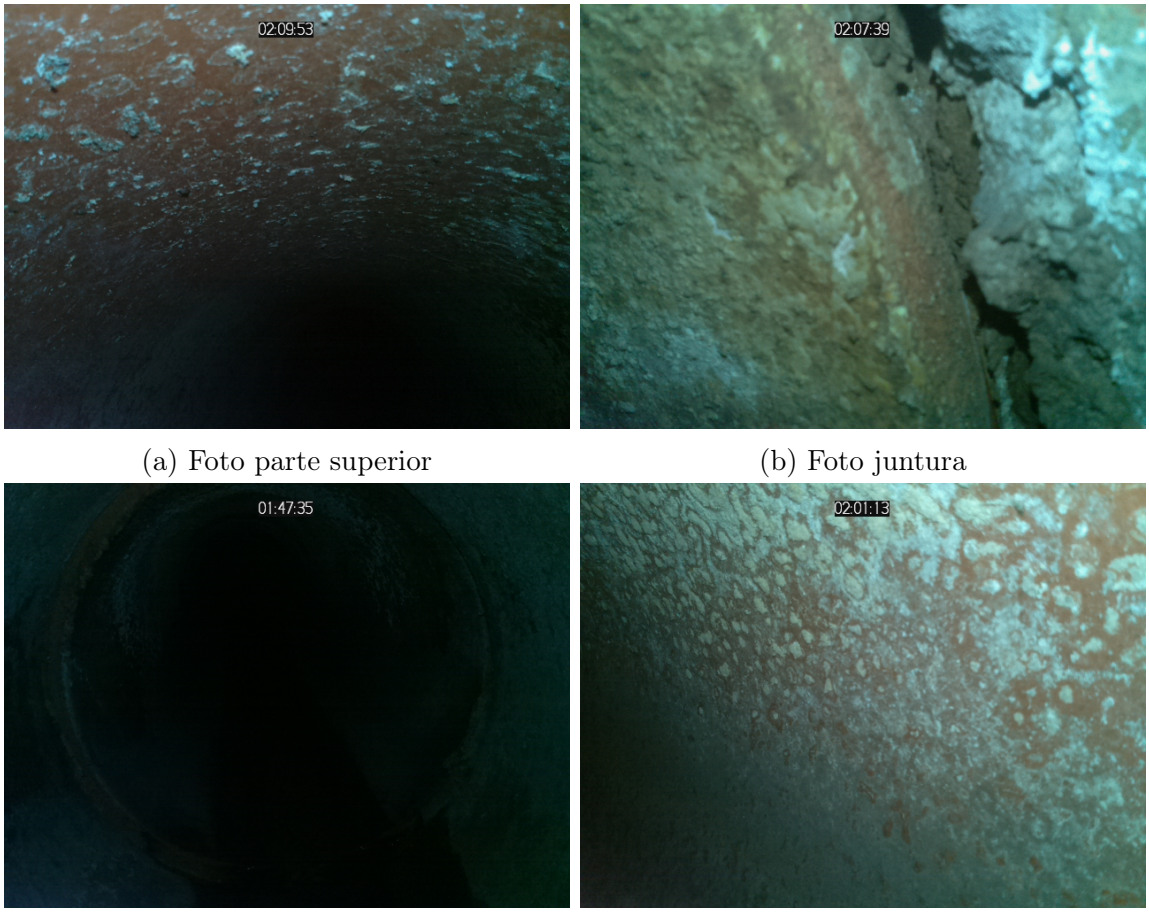
Figura 5.4: Primer intento de inspección de una de las tuberías del pozo.

Sin embargo todavía se tenía la opción de la ductería encontrada a 1.5 metros encima dentro del mismo pozo. Se encontró que durante el primer intento de inspección, cayó agua sobre el lente y no se podía observar nada por lo que hay que tener en cuenta este detalle para trabajos futuros. Una vez limpiado el lente, con ayuda del docente Leonardo Plazas se posiciona el prototipo dentro de la segunda ductería como se observa en la Figura 5.5, allí dentro se inicia con la inspección.



Figura 5.5: Segundo intento de inspección de la otra tubería del pozo.

Una vez dentro de la ductería se observa que, como fue planteado, se iba a perder la señal si el router no se colocaba al mismo nivel, por lo cual con ayuda de una cuerda se cuelga el router justo en la entrada como se muestra en la Figura 5.5. Esto mejor considerablemente la señal con el prototipo y permite realizar la inspección con la toma de videos y fotografías.



(a) Foto parte superior

(b) Foto juntura

(c) Foto plano general

(d) **Foto paredes**

Figura 5.6: Fotos capturadas por el prototipo dentro de la tubería

6. Presupuesto

Para este proyecto se hizo un aproximado de los gastos realizados a lo largo de su elaboración, el cual se registra en la siguiente tabla 6.1

Tabla 6.1: Tabla de precios para los componentes utilizados en el prototipo.

Recurso	*Precio (Pesos COP)*
Raspberri Pi 4b	500.000
Impresiones 3d	300.000
Baterías	120.000
Motores DC	160.000
Llantas + Repuestos	200.000
Servomotores	40.000
Puente H	30.000
Módulo PWM	20.000
Demás componentes electrónicos	50.000
Total	1'420.000

Ya teniendo este valor se procede a comparar con los que se encuentran en el mercado, los cuales se registran en la Tabla 6.2

Tabla 6.2: Comparación de precios con los distintos robots de inspección.

Producto	Precio (Dolares)
Prototipo	350
X5-HT CCTV Robot de inspección[35]	35.000
Mini Robot de inspección de tuberías de alcantarillado HZ-R408[36]	10.000
Robot de inspección de tuberías FB20X[37]	18.000
W3-C100 WITSON inspección de tuberías[38]	20.000

Tabla 6.3: Comparación de características.

Producto	Resolución	Formato vídeo	Peso	Clase Protección
Prototipo	640*480 +	H264	4Kg	-
X5-HT CCTV [35]	1280*720	H264	-	IP68
HZ-R408 [36]	720*480	H265	5.7Kg	IP68
FB20X [37]	1920*1080	-	20Kg	IP68
W3-C100 WITSON[38]	752 x 582	-	7Kg	IP68

7. Conclusiones y trabajo futuro

7.1. Conclusiones

Si se compara este prototipo con todas las opciones que existen en el mercado a nivel de precios, se tiene que es más económico una implementación de este tipo ya que dichos equipos pueden llegar a costar más de 10 veces, sin embargo hay que considerar que al ser un prototipo se deben realizar mejoras tanto a nivel de hardware con mejores materiales, como a nivel de software con más funcionalidades que faciliten realizar una inspección a una red de alcantarillado. Realizar mejoras en el hardware puede llevar a disminuir los tamaños del prototipo para poder utilizarlo en inspecciones de tuberías que tengan diámetros menores, estas mejoras pueden incluir cambios en los sistemas de desplazamiento o en los mismo componentes que se pensaron para este primer prototipo.

Como se observa en la tabla 6.3, hay características que se pueden mejorar, en cuanto a la resolución las características de nuestro prototipo son aumentables, sin embargo lo limitamos para cumplir la norma, en cuanto al formato de vídeo observamos que usamos un formato que es usado habitualmente, también podemos observar que el peso es una característica a tener en cuenta, ya que esto ayudará en cuanto a la recuperación del sistema, por el contrarío observamos que hace falta la implementación de una protección de humedad más exhaustiva.

Un sistema inalámbrico para el envío de información tiene limitaciones en cuanto al alcance que tiene y a que es susceptible de interferencias, sin embargo se pudo observar que si el router se coloca sobre el mismo nivel de la tubería por la que se mueve el vehículo, la señal mejora considerablemente disminuyendo la latencia en la transmisión de video y en el control remoto del prototipo. A pesar de no lograr inspeccionar la totalidad de la tubería en las pruebas y así verificar el alcance real del prototipo, está claro que puede alcanzar distancias mayores de hasta más de 30 metros de distancia ya que depende de la señal del router inalámbrico, del mismo modo si se utilizan routers más potentes con mucho más alcance esta distancia se puede aumentar, únicamente si la tubería es recta, ya que un giro representaría un corte en la señal entre el router y el prototipo.

Para la inspección se implementó un sistema de iluminación en el prototipo aunque como se observa en las fotografías y vídeos capturados es necesario mejorar o aumentar la potencia de la misma, ya que el camino por el que se desplaza el vehículo no se podía observar muy bien. Por esto incluir un sistema que permita variar la intensidad de la iluminación facilitaría mucho más la detección de obstáculos y defectos al interior de la tubería.

El prototipo cumple con el objetivo general del proyecto al brindar un sistema para inspeccionar las redes de alcantarillado con un bajo costo en comparación con

los sistemas comerciales brindados por diferentes empresas y con la ventaja de ser inalámbrico, del mismo modo se comprueba que es posible cumplir muchos de los puntos de la norma NS-058 para llevar a cabo una inspección con este tipo de herramientas aunque requieren desarrollos más avanzados que no hacían parte del prototipo como tal. Las pruebas de campo permitieron conocer mejor como se comporta el sistema en un ambiente real de inspección, para así tomar consideraciones ante trabajos futuros con los errores o dificultades observados, los cuales pueden ser corregidos fácilmente sentando una base funcional para trabajos futuros.

7.2. Trabajo futuro

A lo largo del desarrollo del prototipo, y en el desarrollo de las pruebas de campo se fueron generando sugerencias que se deben tener en cuenta para trabajos futuros

- Es recomendable utilizar para el chasis material de impresión para zonas húmedas PETG es mejor que el material utilizado (PLA), al tener una mejor impermeabilidad para proteger mejor los componentes internos ante filtraciones de agua.
- Las dimensiones del chasis pueden disminuirse para lograr que el sistema de inspección se desplace en tuberías con diámetros menores, ya sea con una mejor distribución interna o variando el mecanismo de desplazamiento.
- Para el control de la cámara podría implementarse un control por ángulo, que permita ajustar de una forma más precisa el ángulo a donde apunta la cámara. También podría implementarse una interfaz dentro de la misma aplicación web, que permita a los operarios mover y conocer la dirección o ángulo al cual está apuntando el efector, es decir la cámara, simplemente ingresando un dato como una coordenada deseada sobre el espacio tridimensional. Para esto sería oportuno realizar todo el análisis de la cinemática directa e inversa del brazo.
- Las grabaciones de video y fotografías deben incluir más características en pantalla a modo de overlay sobre la inspección como ubicación, puntos de inspección, fecha, hora, distancia recorrida, entre otros, según la norma NS-058.
- A manera de mejorar la toma de datos, sería recomendable incluir más sensores para obtener y medir más variables tanto en el ambiente al interior de la tubería como variables propias del prototipo, como por ejemplo sensores que permitan detectar gases, temperatura, humedad o unidades de medición inercial, entre otros.
- En el caso de la distancia recorrida, si se quiere obtener medidas más precisas se recomendaría incluir más sensores a parte de los encoder como una IMU que permitan medir dicha variable y por medio de un filtro de Kalman ajustar y obtener mejores medidas.
- Implementar una impermeabilización total del prototipo, lo cual permita trabajar en cualquier tipo de ductería.

- Unificar toda la parte electrónica en un circuito (PCB), el cual facilitaría su replica, así como permitir una organización mejor de los circuitos para que ocupen menos espacio y cuenten con conexiones fáciles de realizar.
- Mejora en el diseño del brazo mecánico e implementarlo, ya que este no estaba diseñado para salvaguardar la circuitería de la humedad.
- Contar con un sistema que permita acomodar el carro de inspección dentro de la tubería deseada, sin importar su altura, o su conexión inicial.
- Implementar sensores de giroscópico para saber los ángulos de inclinación del vehículo al interior de la tubería, ya que únicamente se cuenta con las imágenes visuales del entorno más no del estado a bordo del propio vehículo, para poder corregirlo o ajustarlo y evitar que choque o vuelque.
- Implementar impermeabilización en el LED y en la cámara, y de ser necesario un mecanismo que limpie el lente de la cámara, ya que como se observó es susceptible a pequeñas gotas que impiden una correcta visualización y que obligan a retirar el vehículo de la tubería, limpiarlo y volverlo a introducir, lo cual alarga los tiempos de inspección.
- Implementar un pequeño servidor sobre la propia Raspberry Pi basado en Apache o Nginx que aloje la aplicación web creada y la ponga operativa simplemente al encenderla, ya que actualmente solo se estaba utilizando un servidor de pruebas y esto también afecta el rendimiento general al momento de recibir y atender peticiones realizadas por los clientes.
- Para la reducción del tamaño se plantea el problema de implementar una batería que alimente todo el sistema, por ello es necesario implementar un sistema de disipación para la placa programable, debido a que se sabe que sus características se ven afectadas con el aumento de la temperatura.

Referencias

[1] Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), *Aspectos técnicos para la investigación y calificación de redes de alcantarillado con equipos de CCTV*. Norma Técnica de Servicio Código NS-058, 2020.

[2] R. Ishikawa, T. Tomita, Y. Yamada and T. Nakamura, *Development of a peristaltic crawling robot for long-distance complex line sewer pipe inspections*. IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016.

[3] A. Hadi, M. Abdollahi, K. Alipour and B. Tarvirdizadeh, *Design and Prototyping a New Add-on Module to Increase Traction Force of a Wheeled Sewer Inspection Robot*. RSI International Conference on Robotics y Mechatronics (ICRoM), 2017.

[4] Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), *Terminología de alcantarillado*. Norma Técnica de Servicio Código NT-003, 2003.

[5] Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), *Tuberías para alcantarillado*. Norma Técnica de Servicio Código NP-027, 2022.

[6] CYPE, *Modelo de un pozo de inspección*. dirección: `http://www.colombia.generadordeprecios.info/espacios_urbanos/Instalaciones/Urbanas/Redes_de_saneamiento/IUS052_Pozo_de_registro_de_concreto_simple.html`.

[7] Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), *Pozos de inspección*. Norma Técnica de Servicio Código NS-029, 2006.

[8] Empresa de Acueducto y Alcantarillado de Bogotá (EAAB), *Sumideros*. Norma Técnica de Servicio Código NS-047, 2020.

[9] ISO/IEC, *Coding of audio-visual objects*. Information technology ISO/IEC 14496, 2021.

[10] ISO, *Data elements and interchange formats*. Information interchange ISO 8601, 2019.

[11] ISO/IEC, *Digital compression and coding of continuous-tone still images: Requirements and guidelines*. Information technology ISO/IEC 10918, 2005.

[12] Andrew S. Tanenbaum and David J. Wetherall, *Redes de computadoras*. Pearson, Quinta edición, 2012.

[13] Bruce Fette, Roberto Aiello, Praphul Chandra, Daniel M. Dobkin, Alan Bensky, Douglas Miron, David A. Lide, Farid Dowla and Ron Olexa, *RF Wireless Technologies*. Elservier Inc, 2008.

[14] Behrouz A. Forouzan, *TCP/IP Protocol Suite*. McGraw-Hill, Cuarta edición, 2010.

[15] Bluetooth Special Interest Group, *Bluetooth*. 2022. dirección: `https://www.bluetooth.com/`.

[16] Bluetooth Special Interest Group, *Bluetooth Core Specification*. Bluetooth SIG Proprietary, 2021. dirección: https://www.bluetooth.com/.

[17] PANATEC, *Robot de Inspección Visual - iPEK ROVION*. 2022. dirección: http://www.panatec-agua.com/robot-inspeccion-ipek.php.

[18] Wuhan Easy-Sight Technology Co, *X5-HT Pipeline CCTV Inspection Robot*. 2022. dirección: https://www.pipedetect.com/products/x5-ht-pipe-camera-inspection/.

[19] Robotics Design Inc., *ANATROLLER ARI-10*. 2022. dirección: https://www.directindustry.es/prod/robotics-design-inc/product-57161-930191.html.

[20] V. Barrientos, J. García y R. Silva, *Robots Móviles: Evolución y Estado del Arte*. Redalyc, 2007.

[21] Michigan CAT, *Skid Steer Loader*. dirección: https://www.michigancat.com/rental-equipment/machines/skid-steer-loaders/226-skid-steer-loader/.

[22] Gareth Halfacree, *La guía oficial de Raspberry Pi para principiantes*. dirección: https://magpi.raspberrypi.com/books/beginners-guide-4th-ed-es.

[23] Django, *Django documentation*. dirección: https://docs.djangoproject.com/en/4.0/.

[24] K. Kozlowski and D. Pazderski, *Modeling and control of a 4-wheel skid-steering mobile robot*. 2004.

[25] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*. 2004.

[26] China Automated Store, *Motor de engranaje de metal con codificador*. AliExpress. dirección: https://es.aliexpress.com/item/4000281595585.html.

[27] JIUWU, *Llantas todoterreno para vehículos a RC*. Amazon. dirección: https://www.amazon.com/gp/product/B00ZZ9LTBU/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1.

[28] Raspberry Pi Documentation, *Camera Modules*. Raspberry Pi. dirección: https://www.raspberrypi.com/documentation/accessories/camera.html.

[29] Robotgeek, *Snapper Mini*. Thingiverse. dirección: https://www.thingiverse.com/thing:2373939.

[30] Formlabs, *Guía de materiales de impresión 3D: Tipos, aplicaciones y propiedades*. 2022. dirección: https://formlabs.com/latam/blog/materiales-impresion-3d/.

[31] Raspberry Pi, *Raspberry Pi Documentation*. Raspberry Pi. dirección: https://www.raspberrypi.com/documentation/.

[32] TP-Link, *Router Inalámbrico TL-WR840N V2*. TP-Link. dirección: https://www.tp-link.com/co/home-networking/wifi-router/tl-wr840n/.

[33] Katsuhiko Ogata, *Modern Control Engieneering - 5th edition*. Pearson, 2010.

[34] Katsuhiko Ogata, *Sistema de control en tiempo discreto - 2nd edition*. Prentice Hall Hispanoamericana, 2013.

[35] *X5-HT CCTV Robot de inspección*. dirección: https://spanish.alibaba.com/product-detail/X5-HT-Undeground-Robotic-CCTV-survey-1600249978233.html.

[36] *Mini Robot de inspección de tuberías de alcantarillado HZ-R408*. dirección: https://spanish.alibaba.com/product-detail/HZ-R408-Under-Vehicle-Inspection-Robot-1600080087670.html.

[37] *Robot de inspección de tuberías FB20X*. dirección: https://spanish.alibaba.com/p-detail/pipe-1600273801656.html?spm=a2700.7724857.0.0.5da63ed5aN1e5q.

[38] *W3-C100 WITSON inspección de tuberías*. dirección: https://spanish.alibaba.com/product-detail/WITSON-W3-C100-Pipe-Inspection-Crawler-60786241942.html.

A. Código implementado en Arduino para caracterizar los motores.

```
// Variables para definicion del tiempo de interrupci n y calculo de la
   velocidad motor derecho
double ActualI = 0       // Tiempo de la interrupci n actual
double AnteriorI = 0     // Tiempo de la interrupci n anterior
double Delta = 0         // Delta de Tiempo de las interrupciones
int encoder = 3          // pin de conexi n del encoder
double frecuenciaR = 0   // frecuencia de interrupci n
double Wr = 0;           // Velocidad angular
double Vl = 0;           // velocidad Lineal
int N = 220;             // n ero de ranuras del encoder
float diametro = 13;     // diametro de la llanta cm
int DirA = 13, DirB = 12;  // Pin de direcci n del motor

void setup() {           // funci n setup
  attachInterrupt(digitalPinToInterrupt(encoder),IntEnc,RISING);  //
    linea para a adir una interrupci n a un PIN
  Serial.begin(250000);     // Comunicaci n del arduino con el PC
  pinMode(DirA, OUTPUT);    // Declara pines de salida
  pinMode(DirB, OUTPUT);    // Declara pines de salida
  digitalWrite(DirB, LOW); // Declara un pin como bajo para controlarlo
    desde el otro    }

void IntEnc() {                          // funci n de interrupci n del
   enconder llanta
    Delta = ActualI -  AnteriorI;        // diferencia tiempos de
   interruciones de ticks del motor
    AnteriorI = ActualI;                 // se actualiza el tiempo de
   interrupci n anterior
    frecuenciaR = (1000000)/Delta;       // frecuencia de interrupci n
    Wr = ((2*3.141516)/N)*frecuenciaR; // frecuencia angular Rad/s
    Vl = Wr*(diametro/2);                // velocidad lineal cm/s
    Serial.print(Vl);
    Serial.print(",");
    Serial.println(ActualI/1000);     }

void loop() {           // funci n loop
  ActualI = micros();  // se asigna el tiempo de ejecuci n a el muestreo
    actual
  if ((unsigned long)(ActualI ) < 500000){// Tiempo actual menor a 500ms
    digitalWrite(DirA, HIGH);            // Activaci n de direcci n del
   motor                                   }
  if ((unsigned long)(ActualI) >= 500000){// Tiempo actual mayor a 500ms
    digitalWrite(DirA, LOW);             // Desactivaci n de direcci n
   del motor } }
```

B. Código implementado en la aplicación web.

```python
# Archivo de configuracion para las urls de la aplicacion
from django.urls import path, include
from .views import *

urlpatterns = [
    path('home/', Home.as_view(), name='home'),
    path('home/stream/', Stream.as_view(), name='stream'),
    path('home/grabar', Grabar.as_view(), name='grabar'),
    path('home/foto', Foto.as_view(), name='foto'),
    path('configuracion/stream/', Stream.as_view(), name='stream'),
    path('archivos/', Archivos.as_view(), name='archivos'),
    path('descarga/', Descarga.as_view(), name='descarga'),
    path('brazo/', Brazo.as_view(), name='brazo'),
    path('motores/', Motores.as_view(), name='motores'),
    path('iluminacion/', Iluminacion.as_view(), name='iluminacion'),
    path('camara_inicio/', PuntoInicial.as_view(), name='camara-inicio')
]
```

```python
# Archivo de vistas con la logica de la aplicacion
from django.shortcuts import render, redirect
from django.views.generic import *
from django.http import StreamingHttpResponse, FileResponse, JsonResponse
import os
import io
import picamera

import time
import busio
import math
from board import SCL, SDA
from adafruit_motor import servo
from adafruit_pca9685 import PCA9685

import json
import Adafruit_PCA9685
import subprocess
import RPi.GPIO as GPIO
from subprocess import call
from threading import Condition

# Se crea el objeto de la clase picamera con sus configuraciones
    iniciales.
camara = picamera.PiCamera()
camara.resolution = (640,480)
camara.meter_mode = 'matrix'
camara.iso = 800
```

```
camara.framerate = 30
camara.annotate_foreground = picamera.Color('white')
camara.annotate_background = picamera.Color('black')
camara.annotate_text_size = 18

nombre = ''

# Variables globales sobre las que se crean los subprocesos para mover
    los motores
p = None
s = None

# Longitud de los enlaces del brazo
l1 = 3
l2 = 5
l3 = 2.5

# Objeto i2c para comunicarse con el modulo PCA9685
i2c = busio.I2C(SCL,SDA)

pwm = PCA9685(i2c)
pwm.frequency = 50

servo1 = servo.Servo(pwm.channels[9], min_pulse=500, max_pulse=2500)
servo2 = servo.Servo(pwm.channels[8], min_pulse=500, max_pulse=2500)
servo3 = servo.Servo(pwm.channels[7], min_pulse=500, max_pulse=2500)

servo1.angle = 90
servo2.angle = 150
servo3.angle = 120

# Variables globales para el context de las articulaciones
art1 = 0
art2 = 30
art3 = -30

# GPIO iluminaci n
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)

# Vistas y clases.
class StreamingOutput():
    """Se encarga de guardar cada frame del video en un buffer para ser
    transmitido, creo :v"""
    def __init__(self):
        self.frame=None
        self.buffer=io.BytesIO()
        self.condition=Condition()

    def write(self,buf):
        if buf.startswith(b'\xff\xd8'):
            self.buffer.truncate()
```

```
            with self.condition:
79              self.frame = self.buffer.getvalue()
                self.condition.notify_all()
81          self.buffer.seek(0)
        return self.buffer.write(buf)
83
class Stream(View):
85      """Vista para realizar la transmisi n de v deo al template"""
        def get(self,request, *args, **kwargs):
87          def cam():
                output=StreamingOutput()
89              #camara.rotation=180
                camara.start_recording(output, splitter_port=2, format='mjpeg
    ')
91              try:
                    while True:
93                      with output.condition:
                            output.condition.wait()
95                          frame = output.frame
                            h = time.strftime("%H:%M:%S")
97                          camara.annotate_text = h
                        yield(b'--FRAME\r\n'b'Content-Type: image/jpeg\r\n\r\
    n' + frame + b'\r\n')
99              finally:
                    camara.stop_recording(splitter_port=2)
101         resp = StreamingHttpResponse(cam(), status=200, content_type='
    multipart/x-mixed-replace; boundary=FRAME')
            return resp
103
class Home(TemplateView):
105     """Renderiza la p gina de inicio"""
        template_name = 'home.html'
107
        def get_context_data(self, *args, **kwargs):
109         brillo = camara.brightness
            contraste = camara.contrast
111         iso = camara.iso
            saturacion = camara.saturation
113
            global art1
115         global art2
            global art3
117
            context = {
119             "brillo": brillo,
                "contraste": contraste,
121             "iso": iso,
                "saturacion": saturacion,
123             "articulacion1": art1,
                "articulacion2": art2,
125             "articulacion3": art3,
            }
```

```
127         return context
129     def post(self, request, *args, **kwargs):
            post_data = json.loads(request.body.decode("utf-8"))
131         camara.brightness = int(post_data["brillo"])
            camara.contrast = int(post_data["contraste"])
133         camara.saturation = int(post_data["saturacion"])
            camara.iso = int(post_data["iso"])
135
            return JsonResponse({"Recibido": True})
137
class Grabar(View):
139     """Inicia o detiene la grabaci n de v deo"""
        def get(self, request, *args, **kwargs):
141         estado = request.GET.get("estado")
            print("ESTADO", estado)
143         global nombre
            path = '/home/pi/inspeccion/inspeccion/files/videos/'
145
            if estado == "grabar":
147             nombre = time.strftime("%Y%m%d-%H%M%S.h264")
                vid = os.path.join(path, nombre)
149             camara.start_recording(vid, quality=20)
                camara.annotate_text = 'REC'
151             respuesta = "Grabando"
            elif estado == "parar":
153             camara.annotate_text = ''
                camara.stop_recording()
155             nombre2 = os.path.splitext(nombre)[0]
                vid = os.path.join(path, nombre)
157             vid2 = os.path.join(path, nombre2)
                command = "MP4Box -add {} {}.mp4".format(vid, vid2)
159             call([command], shell=True)
                os.remove(vid)
161             respuesta = "Detenido"

163         context = {"Respuesta": respuesta}
            return JsonResponse(context)
165
class Foto(View):
167     """Toma una foto y renderiza un formulario para registrar la
    incidencia"""
        def get(self, request, *args, **kwargs):
169         nombre = time.strftime("%Y%m%d-%H%M%S")
            pic = open('/home/pi/inspeccion/inspeccion/files/fotos/'+nombre+'
    .jpeg','wb')
171         camara.capture(pic)
            respuesta = "Tomada"
173
            context = {"Respuesta": respuesta}
175         return JsonResponse(context)
```

```
177 class Archivos(View):
    """Entrega la lista de archivos almacenados (fotos o videos)"""
179     def get(self, request, *args, **kwargs):
            template_name = 'multimedia.html'
181         opcion = request.GET.get("opcion")
            ruta = '/home/pi/inspeccion/inspeccion/files/'
183         archivos = []
            tama o = []
185
            if opcion == 'Fotos':
187             print("FOTOS")
                contenido = os.listdir(ruta+'fotos')
189             for c in contenido:
                    size = os.path.getsize(ruta+'fotos/'+c)
191                 tama o.append(round(size/10**6,1))
            else:
193             print("VIDEOS")
                contenido = os.listdir(ruta+'videos')
195             for c in contenido:
                    size = os.path.getsize(ruta+'videos/'+c)
197                 tama o.append(round(size/10**6,1))

199         for i in range(0,len(contenido)):
                archivos.append(
201                 (contenido[i], tama o[i]),
                )
203
            context = {
205             "archivos": archivos,
                "opcion": opcion
207         }
            return render(request, template_name, context)
209
    class Descarga(View):
211     """Retorna el archivo seleccionado"""
        def get(self, request, *args, **kwargs):
213         nombre_archivo = request.GET.get("nombre")
            tipo = request.GET.get("tipo")
215
            if tipo == "Fotos":
217             ruta = '/home/pi/inspeccion/inspeccion/files/fotos/'
            else:
219             ruta = '/home/pi/inspeccion/inspeccion/files/videos/'

221         file = open(ruta+nombre_archivo,'rb')

223         response = FileResponse(file)
            response['Content-Type']='application/octet-stream'
225         response['Content-Disposition']='attachment; filename="'+
        nombre_archivo+'"'

227         return response
```

```
229 class Brazo(View):
        """Se define el  ngulo  de cada servo dependiendo del valor recibido
    """
231     def get(self, request, *args, **kwargs):
            articulacion = request.GET.get('articulacion')
233         angulo = request.GET.get('angulo')
            print("Articulacion", articulacion, "Angulo", angulo)
235         global art1
            global art2
237         global art3
            if articulacion == '1':
239             art1 = int(angulo)
                pwm1 = mapeo(art1,-90,90,0,180)
241             servo1.angle = pwm1
            if articulacion == '2':
243             art2 = int(angulo)
                pwm2 = mapeo(art2,-90,90,180,0)
245             servo2.angle = pwm2
            if articulacion == '3':
247             art3 = int(angulo)
                pwm3 = mapeo(art3,-90,90,180,0)
249             servo3.angle = pwm3
            return JsonResponse({"A":1})
251
    class Iluminacion(View):
253     """Con esta vista se prende o apaga el LED de iluminaci n"""
        def get(self, request, *args, **kwargs):
255         estado_led = request.GET.get('estado_led')
            if estado_led == 'on':
257             GPIO.output(17, GPIO.HIGH)
            else:
259             GPIO.output(17, GPIO.LOW)
            return JsonResponse({"Recibido": True})
261
    class Motores(View):
263     """Se encarga de abrir subprocesos que corren un script para el
       movimiento de los motores"""
        def get(self, request, *args, **kwargs):
265         sentido = request.GET.get('sentido')
            print(sentido)
267         global p
            global s
269         if s:
                s.kill()
271         if sentido == 'up':
                p = subprocess.Popen(['python','arriba.py'])
273         if sentido == 'left':
                p = subprocess.Popen(['python','izquierda.py'])
275         if sentido == 'right':
                p = subprocess.Popen(['python','derecha.py'])
277         if sentido == 'down':
```

```
            p = subprocess.Popen(['python','abajo.py'])
        if sentido == 'stop':
            if p:
                p.kill()
                s = subprocess.Popen(['python', 'parar.py'])

        return JsonResponse({"Recibido": True})

class PuntoInicial(View):
    """Utiliza la cinem tica inversa para apuntar la c mara al punto inicial"""
    def get(self, request, *args, **kwargs):
        global art1
        global art2
        global art3

        px = 6.8301
        py = 0.0
        pz = 5.5

        angulos = cinematica_inversa(px, py, pz)

        art1 = 90 - angulos['angulo1']
        art2 = 180 - angulos['angulo2']
        art3 = 90 - angulos['angulo3']

        context = {
            "angulo1" : art1,
            "angulo2" : art2,
            "angulo3" : art3
        }
        return JsonResponse(context)

# Funciones
def mapeo(x, i_min, i_max, o_min, o_max):
    return int((x-i_min)*(o_max-o_min)/(i_max-i_min)+o_min)

def cinematica_inversa(px, py, pz):
    global l1
    global l2
    global l3

    d = ((px**2)+(py**2)+((pz-l1)**2)-l2**2-l3**2)/(2*l2*l3)
    theta1 = math.atan2(py,px)
    theta3 = math.atan2(-math.sqrt(1-d**2),d)
    theta2 = math.atan2(pz-l1,math.sqrt(px**2+py**2)) - math.atan2(l3*math.sin(theta3),l2+l3*math.cos(theta3))

    angulo1 = mapeo(math.degrees(theta1),-90,90,0,180)
    angulo2 = mapeo(math.degrees(theta2),0,180,180,0)
    angulo3 = mapeo(math.degrees(theta3),-90,90,180,0)

```

```
    servo1.angle = angulo1
329 servo2.angle = angulo2
    servo3.angle = angulo3
331
    context = {
333     "angulo1" : angulo1,
        "angulo2" : angulo2,
335     "angulo3" : angulo3
    }
337 return context
```

C. Código implementado para el movimiento del vehículo.

```
1 # Script para mover el carro hacia adelante o hacia atras (ambos son
    iguales solo cambia la direccion de giro)
  import RPi.GPIO as GPIO
3 import Adafruit_PCA9685
  import time
5 import math

7 # Objeto PWM
  pwm = Adafruit_PCA9685.PCA9685()
9 pwm.set_pwm_freq(60)

11 # Definici n de pines para la lectura de los encoders
  Pin_encoder1 = 21
13 Pin_encoder2 = 16
  Pin_encoder3 = 20
15 Pin_encoder4 = 26

17 # Definici n de variables generales
  n = 220
19 diametro = 13
  ref = 35 #rpm
21
  # Variables para cada motor
23 t_actual1 = 0
  t_anterior1 = 0.0
25 delta1 = 0.0
  frecuencia1 = 0
27 w1 = 0
  rpm1 = 0
29 u1 = [0, 0]
  e1 = [0, 0]
31 pwm1 = 0
```

```
t_actual2 = 0
t_anterior2 = 0.0
delta2 = 0.0
frecuencia2 = 0
w2 = 0
v2 = 0
rpm2 = 0
u2 = [0, 0]
e2 = [0, 0]
pwm2 = 0

t_actual3 = 0
t_anterior3 = 0.0
delta3 = 0.0
frecuencia3 = 0
w3 = 0
v3 = 0
rpm3 = 0
u3 = [0, 0]
e3 = [0, 0]
pwm3 = 0

t_actual4 = 0
t_anterior4 = 0.0
delta4 = 0.0
frecuencia4 = 0
w4 = 0
v4 = 0
rpm4 = 0
u4 = [0, 0]
e4 = [0, 0]
pwm4 = 0

# Configuraci n de los GPIO
GPIO.setmode(GPIO.BCM)

GPIO.setup(Pin_encoder1, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(Pin_encoder2, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(Pin_encoder3, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(Pin_encoder4, GPIO.IN, pull_up_down=GPIO.PUD_UP)

def int_motor1(channel):
  global t_actual1, t_anterior1, delta1, frecuencia1, w1, v1, diametro, n
    , rpm1
  t_actual1 = time.clock_gettime(1)
  delta1 = t_actual1 - t_anterior1
  t_anterior1 = t_actual1
  frecuencia1 = 1/delta1
  w1 = ((2*math.pi)/n)*frecuencia1
  rpm1 = w1/((2*math.pi)/60)#    v1 = w1*(diametro/2)

def int_motor2(channel):
```

```
    global t_actual2, t_anterior2, delta2, frecuencia2, w2, v2, diametro, n
      , rpm2
    t_actual2 = time.clock_gettime(1)
    delta2 = t_actual2 - t_anterior2
    t_anterior2 = t_actual2
    frecuencia2 = 1/delta2
    w2 = ((2*math.pi)/n)*frecuencia2
    rpm2 = w2/((2*math.pi)/60)#    v2 = w2*(diametro/2)

def int_motor3(channel):
    global t_actual3, t_anterior3, delta3, frecuencia3, w3, v3, diametro, n
      , rpm3
    t_actual3 = time.clock_gettime(1)
    delta3 = t_actual3 - t_anterior3
    t_anterior3 = t_actual3
    frecuencia3 = 1/delta3
    w3 = ((2*math.pi)/n)*frecuencia3
    rpm3 = w3/((2*math.pi)/60)#    v3 = w3*(diametro/2)

def int_motor4(channel):
    global t_actual4, t_anterior4, delta4, frecuencia4, w4, v4, diametro, n
      , rpm4
    t_actual4 = time.clock_gettime(1)
    delta4 = t_actual4 - t_anterior4
    t_anterior4 = t_actual4
    frecuencia4 = 1/delta4
    w4 = ((2*math.pi)/n)*frecuencia4
    rpm4 = w4/((2*math.pi)/60)#    v4 = w4*(diametro/2)

GPIO.add_event_detect(Pin_encoder1, GPIO.FALLING, callback=int_motor1)
GPIO.add_event_detect(Pin_encoder2, GPIO.FALLING, callback=int_motor2)
GPIO.add_event_detect(Pin_encoder3, GPIO.FALLING, callback=int_motor3)
GPIO.add_event_detect(Pin_encoder4, GPIO.FALLING, callback=int_motor4)

# Definen el sentido de giro de los motores, toc  con el m dulo de pwm
# porque el L298n necesita 5V en los pines y los pines de la Raspberry
# solo entregan 3.3V

# Primer L298n
pwm.set_pwm(3, 0, 0) # IN1
pwm.set_pwm(2, 0, 4095) #IN2
pwm.set_pwm(1, 0, 0) #IN3
pwm.set_pwm(0, 0, 4095) #IN4

# Segundo L298n
pwm.set_pwm(12, 0, 0) #IN4
pwm.set_pwm(13, 0, 4095) #IN3
pwm.set_pwm(14, 0, 0) #IN2
pwm.set_pwm(15, 0, 4095) #IN1

def control1():
    e1[0] = ref - rpm1
```

```
    u1[0] = u1[1] + (1.9755*e1[0]) - (1.53002475*e1[1])
    if u1[0]>100:
        u1[0]=100
    if u1[0]<0:
        u1[0]=0
    pwm1 = u1[0]*(4095/100)
    pwm.set_pwm(5, 0, round(pwm1)) #PWMA
    u1[1] = u1[0]
    e1[1] = e1[0]

def control2():
    e2[0] = ref - rpm2
    u2[0] = u2[1] + (1.9847*e2[0]) - (1.5579895*e2[1])
    if u2[0]>100:
        u2[0]=100
    if u2[0]<0:
        u2[0]=0
    pwm2 = u2[0]*(4095/100)
    pwm.set_pwm(4, 0, round(pwm2)) #PWMB
    u2[1] = u2[0]
    e2[1] = e2[0]

def control3():
    e3[0] = ref - rpm3
    u3[0] = u3[1] + (1.9256*e3[0]) - (1.51217368*e3[1])
    if u3[0]>100:
        u3[0]=100
    if u3[0]<0:
        u3[0]=0
    pwm3 = u3[0]*(4095/100)
    pwm.set_pwm(10, 0, round(pwm3)) #PWMA
    u2[1] = u2[0]
    e2[1] = e2[0]

def control4():
    e4[0] = ref - rpm4
    u4[0] = u4[1] + (1.8498*e4[0]) - (1.45227798*e4[1])
    if u4[0]>100:
        u4[0]=100
    if u4[0]<0:
        u4[0]=0
    pwm4 = u4[0]*(4095/100)
    pwm.set_pwm(11, 0, round(pwm4)) #PWMB
    u4[1] = u4[0]
    e4[1] = e4[0]

while True:
    control1()
    control2()
    control3()
    control4()
    print("Adelante",rpm1,rpm2,rpm3,rpm4)
```

```
# Script para mover el carro hacia la derecha o la izquierda (ambos son
    iguales solo cambia la direccion de giro de los motores)
import RPi.GPIO as GPIO
import Adafruit_PCA9685
from time import sleep

pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(60)

# Primer L298n
pwm.set_pwm(5, 0, 4095) #PWMA
pwm.set_pwm(4, 0, 4095) #PWMB
pwm.set_pwm(3, 0, 4095) # IN1
pwm.set_pwm(2, 0, 0) #IN2
pwm.set_pwm(1, 0, 0) #IN3
pwm.set_pwm(0, 0, 4095) #IN4

# Segundo L298n
pwm.set_pwm(10, 0, 4095) #PWMA
pwm.set_pwm(11, 0, 4095) #PWMB
pwm.set_pwm(12, 0, 4095) #IN4
pwm.set_pwm(13, 0, 0) #IN3
pwm.set_pwm(14, 0, 0) #IN2
pwm.set_pwm(15, 0, 4095) #IN1
```