

**SEGUIMIENTO DE TRAYECTORIA DE VEHÍCULOS ELÉCTRICOS A ESCALA BAJO UN
ENTORNO CONTROLADO**

SERGIO DANIEL LAVERDE BONILLA

VALENTINA PILONIETA CAMARGO

DIRECTOR

ING. DIEGO ALEJANDRO PATIÑO GUEVARA Ph.D.



PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

BOGOTÁ D.C

JUNIO DE 2017

Índice general

1. Introducción	4
2. Descripción general	6
3. Planeación de rutas y procesamiento de imágenes	8
3.1. Definición del entorno	8
3.2. Algoritmo de enrutamiento A*	9
3.3. Implementación del algoritmo	11
3.4. Calibración de la cámara	12
3.5. Detección	13
4. Control de bajo nivel	16
4.1. Especificaciones de los vehículos	16
4.2. Protocolo de comunicación	17
4.3. Caracterización de los motores	17
4.4. Control de velocidad	18
4.5. Control de seguimiento de trayectoria	20
4.5.1. Controlador PI	22
4.5.2. Controlador ON/OFF	23
5. Implementación	25
5.1. Implementación con un vehículo	25
5.2. Implementación con dos vehículos	29
6. Resultados y protocolo de pruebas	32
6.1. Línea Recta	32
6.2. Giro	34
6.3. Rutas con un vehículo	37
6.4. Rutas con dos vehículos	38
7. Conclusiones y recomendaciones	40
Bibliografía	40

Índice de figuras

2.1. Diagrama de bloques general	6
3.1. Sector base para diseño del entorno	8
3.2. Grafo definitivo diseñado	9
3.3. Diagrama de flujo algoritmo A*	10
3.4. Diagrama de bloques algoritmo	11
3.5. Imágenes adquiridas mediante la técnica de detección	14
4.1. Diagrama de bloques general de control	16
4.2. Vehículos implementados	17
4.3. Respuesta en lazo abierto GmI y GmD	18
4.4. Respuesta en lazo cerrado GmI y GmD	19
4.5. Esquemático sensor CNY70	20
4.6. Lazo de control de seguimiento de trayectoria	20
4.7. Modelo cinemático del vehículo	21
4.8. Simulación control PI	22
4.9. Simulación control On/Off	23
5.1. Entorno construido	25
5.2. Modelo implementado para los vehículos	26
5.3. Trayectoria obtenida de la ruta # 1	28
5.4. Colisión de intersección	29
5.5. Trayectoria obtenida de la ruta conjunta # 1	30
6.1. Resultados prueba de línea recta 1	32
6.2. Resultados prueba de línea recta 2	33
6.3. Resultados prueba de línea recta 3	33
6.4. Resultados prueba de línea recta 4	34
6.5. Rutas evaluadas con un vehículo	37
6.6. Trayectoria obtenida de la ruta conjunta # 2	39

Índice de cuadros

3.1. Análisis de distorsión	15
4.1. Parámetros de desempeño controlador PI y On/Off	24
5.1. Punto de inicio y paraderos deseados	26
5.2. Ruta y lista de instrucciones ruta #1	27
5.3. Errores de desplazamiento y mediciones estadísticas	28
5.4. Puntos de inicio y paraderos deseados ruta conjunta 1	30
5.5. Ruta y lista de instrucciones ruta conjunta #1	30
5.6. Errores de desplazamiento y mediciones estadísticas ruta conjunta	31
6.1. Error y condiciones de desempeño pruebas de línea recta	34
6.2. Valores obtenidos pruebas de giro	35
6.3. Error en giro	36
6.4. Datos estadísticos rutas individuales	38
6.5. Puntos de inicio y paraderos deseados ruta conjunta 2	38
6.6. Datos estadísticos ruta conjunta	39
1. Punto de inicio y paraderos deseados	43
2. Ruta y lista de instrucciones ruta #1	43
3. Punto de inicio y paraderos deseados	44
4. Ruta y lista de instrucciones ruta #2	44
5. Punto de inicio y paraderos deseados	45
6. Ruta y lista de instrucciones ruta #3	45
7. Punto de inicio y paraderos deseados	46
8. Ruta y lista de instrucciones ruta #4	46
9. Errores de desplazamiento y mediciones estadísticas Ruta 1.1	47
10. Errores de desplazamiento y mediciones estadísticas Ruta 1.2	48
11. Errores de desplazamiento y mediciones estadísticas Ruta 2.1	48
12. Errores de desplazamiento y mediciones estadísticas Ruta 2.2	49
13. Errores de desplazamiento y mediciones estadísticas Ruta 3.1	49
14. Errores de desplazamiento y mediciones estadísticas Ruta 3.2	50
15. Errores de desplazamiento y mediciones estadísticas Ruta 4.1	50
16. Errores de desplazamiento y mediciones estadísticas Ruta 4.2	51

Capítulo 1

Introducción

En la actualidad la expansión urbana y el crecimiento poblacional que ocurre en las grandes ciudades como Bogotá, genera mayores necesidades en términos de transporte, infraestructura y planeación. Los sistemas de transporte integrado se presentan como una solución a los problemas de distancia y tiempo de recorridos, cobertura y frecuencia de operación de los buses.

En un estudio realizado por la secretaria distrital de planeación de Bogotá se analizan todos los componentes sociales, económicos, geográficos y urbanos involucrados en el diseño de un sistema de transporte público para la ciudad. Así, se concluye que el crecimiento económico de la ciudad y el modelo de sistema de transporte generan un impacto recíproco, en el cual el crecimiento de uno promueve nuevas necesidades y progresos en el otro [1]. El diseño de las rutas para estos sistemas de transporte integrado es fundamental debido a la necesidad de una amplia cobertura para todas las zonas de la ciudad, afectando directamente la calidad del servicio. En un documento emitido por la subgerencia técnica y de servicio de la empresa Transmilenio S.A. se realiza una caracterización del sistema SITP implementado en la ciudad, haciendo énfasis en la necesidad de ejecutar una división zonal debido a la alta demanda de usuarios del sistema, todo esto enfocado en mejorar la cobertura y frecuencia de operación del sistema [2].

Por otro lado, las altas emisiones de carbono generadas por los vehículos convencionales de combustión interna son un directo agravante de la problemática ambiental, en comparación, las nulas emisiones de carbono de los vehículos eléctricos en el momento de su operación los convierte en una solución eficiente para la reducción de estas emisiones y por lo tanto mitigar el impacto ambiental que tienen los medios de transporte, por esta razón, estos vehículos son comúnmente usados en los nuevos sistemas de transporte público de las ciudades y se han convertido en un pilar en el desarrollo de las administraciones locales y nacionales. En Bogotá para el año 2014 empezaron a circular dentro del sistema de transporte integrado, buses híbridos (eléctricos y diésel) fabricados por la empresa Volvo [3].

A pesar de estos avances en la implementación de esta tecnología, aún no se tienen buses completamente eléctricos operando en el sistema de transporte de Bogotá, debido a que el desempeño de este tipo de vehículos se ve comúnmente afectado por la duración de la batería generando la necesidad de realizar constantes recargas en el momento de ejecutar recorridos largos. Para evitar problemas como estos, es necesario que las rutas que cubren este tipo de vehículos sean eficientes en términos de distancia del recorrido, tiempo de cumplimiento de ruta y cobertura.

Este trabajo de grado pretenderá implementar un sistema de control de seguimiento de ruta en vehículos eléctricos a escala para que recorran un entorno controlado, para lo cual se definirán los siguientes objetivos específicos:

- Establecer una ruta eficiente para cada vehículo mediante un algoritmo de enrutamiento conocido (Dijkstra, A*, Floyd-Warshall).
- Implementar un sistema de reconocimiento de imágenes para seguir la trayectoria de cada vehículo.

- Diseñar el sistema de control de navegación de los vehículos que cumpla los parámetros de desempeño establecidos (tiempo de establecimiento menor a un segundo, error en estado estable menor a 10% y sobrepico menor o igual al 20%).
- Realizar las pruebas correspondientes en el entorno, con el fin de verificar el comportamiento del sistema de control.

Mediante el cumplimiento de estos objetivos se pretende desarrollar cada una de las partes necesarias para el alcanzar el objetivo general planteado para el proyecto.

Capítulo 2

Descripción general

La metodología seguida para el desarrollo del presente trabajo de grado tiene como base el diagrama de bloques de la figura 2.1 en este se definen todas las partes necesarias para el desarrollo del proyecto.

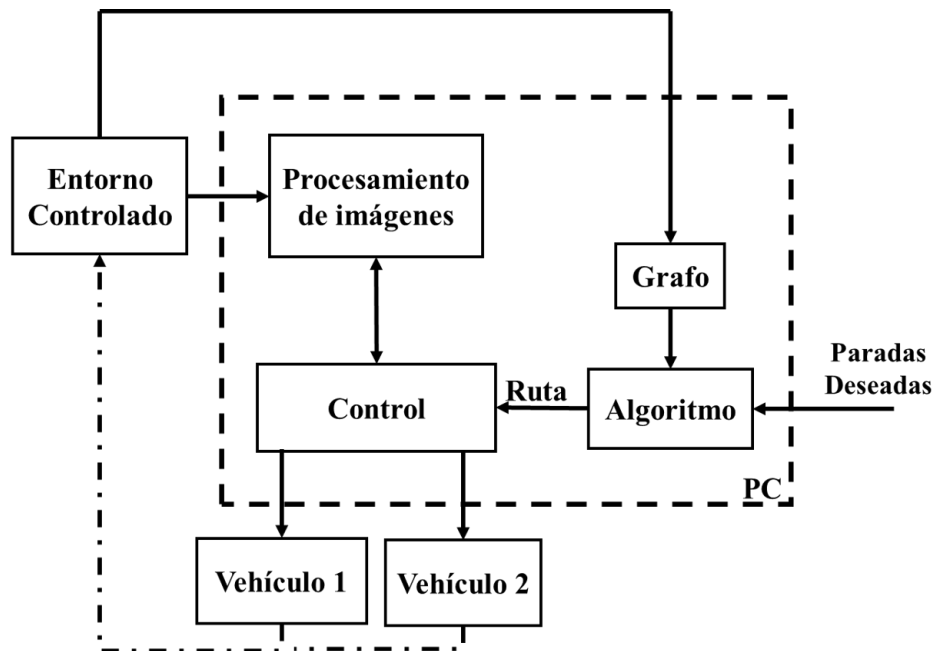


Figura 2.1: Diagrama de bloques general

Inicialmente, partiendo de un entorno seleccionado se obtuvo un grafo equivalente de costos para implementar el algoritmo de enrutamiento A*, el cual, a partir de paradas designadas, puntos de inicio y fin, genera dos rutas que serán recorridas por cada uno de los vehículos de forma simultánea con el fin de garantizar la cobertura total del entorno propuesto, lo cual corresponde, al desarrollo teórico y la implementación de los bloques Grafo y Algoritmo que se presentan en el capítulo 2 de este documento. Posteriormente con el objetivo de realimentar el lazo de control y mediante el uso de una cámara se implementó el bloque Procesamiento de imágenes en el cual, mediante una técnica de segmentación por color obtiene la posición actual de los vehículos en el entorno, en el capítulo 3 se presentan las bases teóricas y los procedimientos necesarios de calibración para el correcto funcionamiento de esta parte del sistema.

Finalmente en el bloque Control, se desarrolla la parte central del trabajo de grado integrando todas las estrategias de control necesarias para el correcto funcionamiento del sistema, esto incluye controladores clásicos de velocidad, trayectoria y adicionalmente una estrategia de prevención de colisiones debido al movimiento simultaneo de los vehículos dentro del entorno, el diseño de los controladores de velocidad y

trayectoria junto con las especificaciones técnicas de los vehículos a utilizar se encuentran en el capítulo 4. Todo el sistema de control y de procesamiento de imágenes se llevó a cabo en un computador de forma centralizada con el fin de garantizar alta capacidad de procesamiento y reducir el tiempo de respuesta. El sistema de navegación de los vehículos se ejecutó a través de la tarjeta de desarrollo integrada en cada uno, esta tarjeta se encargó de realizar la comunicación con el computador a través de un módulo Bluetooth, recibir instrucciones y ejecutarlas mediante los actuadores necesarios.

La estrategia de implementación conjunta del sistema se expone en el capítulo 5, en este se describe la división del problema en dos tipos de movimiento, la traducción de la ruta entregada por el algoritmo a instrucciones enviadas a cada vehículo y la forma de recorrer el entorno por ambos vehículos simultáneamente. Finalmente, se ejecutaron las pruebas necesarias para documentar y analizar el comportamiento del control, realizar un análisis estadístico y definir el cumplimiento de los parámetros de desempeño definidos para el sistema.

Capítulo 3

Planeación de rutas y procesamiento de imágenes

3.1. Definición del entorno

El entorno propuesto para el desarrollo de este trabajo de grado presentado en la figura 3.1 se compone de una red de seis calles y cuatro carreras principales de un sector de la ciudad de Bogotá, el sector comprende desde la calle 34 hasta la calle 57 incluyendo las calles 39, 45, 49 y 53, en cuanto a las carreras comprende desde la carrera séptima hasta la 17 incluyendo la carrera 13 y la Av Caracas.

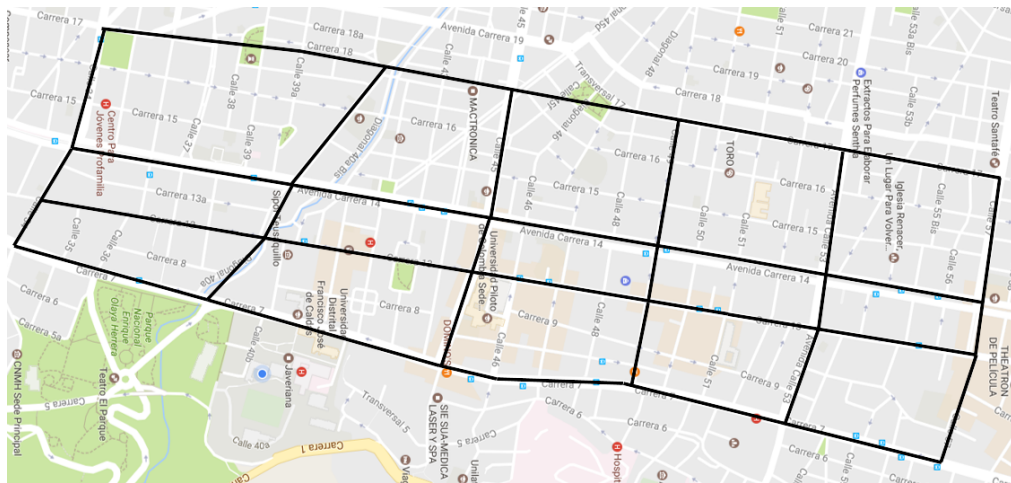


Figura 3.1: Sector base para diseño del entorno

A partir de este entorno propuesto se procedió a construir el grafo equivalente, las dos variables tenidas en cuenta para el desarrollo de este grafo fueron la distancia entre cada una de las intersecciones y límites de velocidad asignados a algunos tramos, para las mediciones aproximadas de distancia de cada uno de los tramos se utilizó la herramienta de medición de Google Maps, los límites de velocidad fueron asignados arbitrariamente teniendo en cuenta el flujo de vehículos y el tamaño de las vías. Adicionalmente, con el objetivo de tener una mayor densidad de posibles paraderos y rutas, se agregaron nodos en los puntos intermedios de cada tramo obteniendo así el grafo de la figura 3.2.

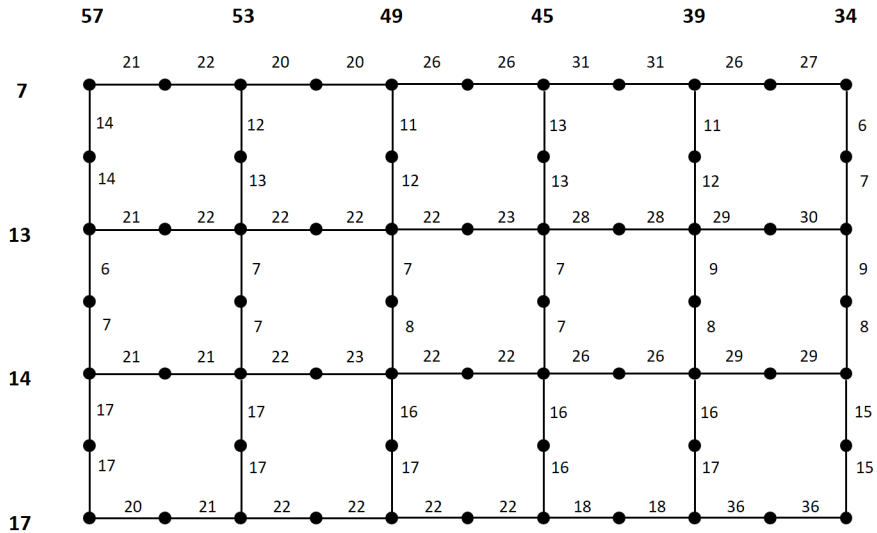


Figura 3.2: Grafo definitivo diseñado

En cuanto al límite de velocidad los tramos seleccionados fueron: calle 39 y 49 en su totalidad, calles 34 y 57 en los tramos entre la carrera séptima y la Av Caracas y las calles 45 y 53 entre la carrera 13 y la Av caracas, los costos de los tramos con límite de velocidad fueron multiplicados por un factor de dos en los costos del grafo.

Debido a consideraciones de espacio en el momento de la implementación física, el entorno diseñado inicialmente fue acotado a las carreras 7,13 y Av caracas con las calles 39, 45, 49 y 53, sin embargo el diseño del algoritmo de enrutamiento A* se aplicó para ambas condiciones de tamaño del entorno, los límites de velocidad definidos previamente fueron mantenidos de igual forma que los pesos definidos mediante las mediciones de distancia. Como consecuencia de esta reducción del entorno la cantidad de paraderos disminuyó a 29 posibles puntos, con los cuales fue posible diseñar las dos rutas con al menos 10 paradas en total.

3.2. Algoritmo de enrutamiento A*

Un algoritmo “shortest path” o algoritmo de enrutamiento es utilizado para definir la distancia más corta entre dos puntos en un grafo cualquiera. En el presente trabajo de grado se busca obtener la ruta más corta desde un punto inicial hasta un punto objetivo con paradas obligatorias establecidas durante el recorrido, para ello se analizaron dos tipos de algoritmo, Dijkstra y A*, con el fin de poder definir cuál de los dos era el apropiado para cumplir el objetivo. Se eligió el algoritmo A* ya que utiliza la función heurística además de la función de costo para determinar la mejor ruta con el menor costo, lo que hace que la búsqueda óptima sea más rápida y eficiente con respecto al Dijkstra que genera una ruta óptima desde el nodo de origen hasta cada uno de los demás nodos existentes aumentando el tiempo de ejecución del algoritmo. El algoritmo A* es un algoritmo de optimización que pretende minimizar el costo de navegar un grafo de un punto inicial a un punto final deseado, es utilizado por ejemplo, en el diseño de vídeo juegos basados en grandes espacios y mapas, para encontrar rutas eficientes de desplazamiento entre dos puntos, adicionalmente este algoritmo es usado para solucionar problemas de VRP (Vehicle Routing Problem) enfocados en situaciones reales como distribución de paquetes en mensajería, transporte público e incluso redes de transmisión de datos.

En general, el algoritmo se define mediante el siguiente diagrama de flujo [4].

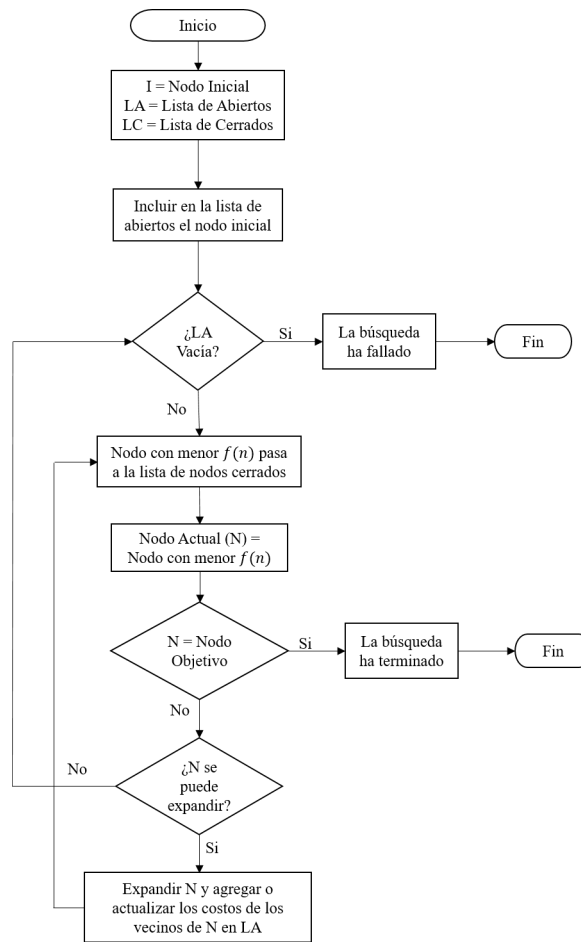


Figura 3.3: Diagrama de flujo algoritmo A*

El algoritmo A* basa su funcionamiento en encontrar el mínimo costo de la función objetivo de la ecuación 3.1, donde $g(n)$ es la función de costos reales del grafo a optimizar, $h(n)$ es la función heurística definida para el problema y $f(n)$ es la función de costo total para cada nodo del grafo [5].

$$f(n) = g(n) + h(n) \quad (3.1)$$

La función heurística $h(n)$ indica un estimado del costo existente entre cada nodo perteneciente al grafo y el nodo objetivo, esta función debe ser construida adecuadamente para el problema y de sus características depende la eficiencia del algoritmo en el momento de su implementación [6]. El comportamiento del algoritmo en función de la heurística tiene las siguientes características:

- Si $h(n)$ es 0, solo $g(n)$ afecta la función total y por lo tanto el algoritmo se comporta como un algoritmo Dijkstra.
- Si $h(n)$ es siempre menor o igual al costo real de llegar al nodo objetivo, el algoritmo garantiza encontrar la ruta más corta a éste.
- Si $h(n)$ es exactamente igual en todo momento al costo real de moverse al nodo objetivo, el algoritmo solo seguirá la ruta más corta sin expandirse.
- Si $h(n)$ es mayor que el costo real en algunos nodos, el algoritmo no necesariamente encontrará la ruta más corta pero será más rápido.

- Si $h(n)$ es bastante mayor que el costo real, el algoritmo se comporta como un algoritmo de búsqueda del mejor nodo inmediato.

Este algoritmo a partir de las funciones $g(n)$ y $h(n)$ expande el nodo inicial en el grafo evaluando el menor costo existente en cada uno de los nodos vecinos al nodo actual, de esta forma decide que nodo será visitado a continuación para realizar el mismo procedimiento, moviendo los nodos evaluados y visitados entre dos listas de nodos abiertos y cerrados, se repite este procedimiento siempre eligiendo el nodo con menor $f(n)$ hasta encontrar el nodo objetivo.

3.3. Implementación del algoritmo

El programa realizado para implementar el algoritmo funciona a partir de un punto de inicio, un punto de fin y las paradas deseadas para cada ruta, generando la ruta óptima para recorrer el entorno con las condiciones ingresadas. La implementación en código del algoritmo se representa mediante el diagrama de bloques de la figura 3.4.

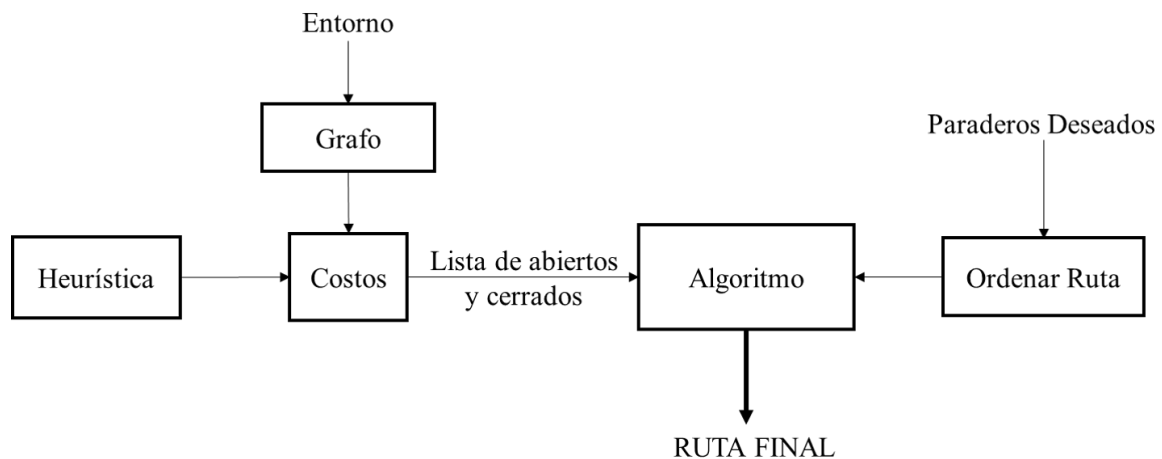


Figura 3.4: Diagrama de bloques algoritmo

En primer lugar fue necesario obtener de una manera ordenada los costos de cada tramo del entorno, para ello se decidió generar una matriz con todos los costos tanto de carreras y calles que fuera fácil de consultar en el momento de implementar el algoritmo. Adicionalmente, se vió la necesidad de dar un orden a las paradas ingresadas, para esto se utilizó la siguiente estrategia:

- Definir una matriz heurística de todos los puntos hasta el punto de inicio.
- Definir una matriz heurística de todos los puntos hasta el punto de fin.
- Definir el punto de inicio ingresado como el primer nodo de la ruta.
- Encontrar los nodos de parada con el mayor valor de heurística respecto al punto de fin y el menor al punto de inicio.
- Si el mismo punto cumple ser máximo y mínimo, se agrega como la siguiente parada en la ruta.
- Si máximo y mínimo no son el mismo punto se da prioridad al punto mínimo y se agrega como la siguiente parada a la ruta.
- Repetir el proceso para las paradas restantes.

- Agregar el punto de fin como la última parada de la ruta.

Debido a que el algoritmo A* no se puede aplicar para problemas que incluyen puntos obligatorios dentro de la ruta a optimizar, se dividió el problema en varias rutas de un paradero deseado a otro, para de esta manera ejecutar el algoritmo repetidas veces hasta completar la ruta desde el punto de inicio hasta el punto final objetivo.

Para el algoritmo, primero se definió una estructura de datos que permitiera almacenar la información necesaria de los nodos visitados y actualizarla con cada iteración de la siguiente forma:

- Las coordenadas actuales.
- Las coordenadas del nodo antecesor.
- El costo actual $g(n)$.
- El valor de la función heurística $h(n)$.
- El costo acumulado $f(n)$.
- Indicador de si el nodo se encuentra abierto o cerrado.

Para el manejo de esta lista de datos, se desarrolló una estrategia que permitiera obtener los costos desde el punto actual hasta todos sus puntos vecinos disponibles para desplazarse, organizando la información de forma matricial con el fin de actualizar la lista en cada nodo visitado. En el momento de agregar los nuevos nodos disponibles a la lista se verifica si estos ya han sido accedidos desde alguna ruta diferente, si esto se cumple se evalúa la ruta para acceder a este nodo con menor $f(n)$ para actualizar la información de éste y así garantizar que los costos en la lista siempre sean los menores.

Posteriormente con la lista de nodos actual se procede a evaluar los nodos que se encuentran abiertos y determinar cuál de estos tiene el menor costo total $f(n)$, esto con el fin de determinar cuál será el siguiente nodo a visitar y de esta forma siempre expandir la ruta con menor costo, finalmente el nodo que ya fue evaluado pasa a ser un nodo cerrado. Este proceso se realiza recursivamente hasta que el nodo actual se convierte en el nodo objetivo.

Con el nodo objetivo alcanzado se procede a organizar la ruta encontrada por medio de la lista con las coordenadas de nodos antecesores, es decir, a partir del nodo final se recorre la lista evaluando los nodos antecesores almacenándolos hasta encontrar nuevamente el punto inicial.

3.4. Calibración de la cámara

Como se indicó en la descripción general del sistema, la realimentación de la trayectoria de los vehículos fue realizada mediante la implementación de una cámara, con esta se realiza todo el proceso de detección (3.5) y se entregan las coordenadas de los vehículos al sistema, permitiendo el funcionamiento del controlador de trayectoria y de la técnica de prevención de colisiones. Debido a esto la selección y calibración de la cámara influyen directamente en el desempeño del sistema.

Para la elección de la cámara a utilizar se tuvieron en cuenta las necesidades principales para la detección, con el fin de obtener una visualización amplia y completa del entorno, la cámara debía estar ubicada directamente sobre este por lo cual se necesitaba que ésta fuera liviana y pequeña, por otro lado la resolución no fue un parámetro crítico para su elección. Se seleccionó la cámara Logitech HD Pro Webcam C920, la cual cuenta con una resolución Full HD 1080p (hasta 1920 X 1080 píxeles), peso de 441.3 g, largo de 19.1 cm, ancho 7.2 cm y profundidad de 22.7 cm adicionalmente cuenta con corrección automática de iluminación escasa y grabaciones de video Full HD [7].

Para lograr la conexión de la cámara con Matlab fue necesario utilizar el paquete de soporte para Web-Cams, el cual permite identificar las cámaras conectadas al computador, crear un objeto tipo cámara y a partir de éste tomar capturas de imágenes para procesarlas.

Una cámara puede ser representada de forma simple mediante el modelo de cámara pinhole [8], el cual caracteriza la cámara como un pequeño agujero por el cual la luz ingresa y proyecta una imagen invertida del mundo real. A partir de esto se obtiene una representación en dos dimensiones de un espacio 3D, por lo cual es necesario establecer una conexión entre estos dos espacios. Esta relación está definida como:

$$W \cdot \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \cdot P \quad (3.2)$$

Donde P es conocida como la matriz de cámara la cual está compuesta por sus parámetros intrínsecos K y extrínsecos R y t , los parámetros intrínsecos son el punto principal y la distancia focal y dependen directamente de las propiedades físicas de la cámara mientras que los extrínsecos dependen de su ubicación espacial, estos parámetros extrínsecos son conocidos como matriz de rotación y vector de traslación.

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K \quad (3.3)$$

Para realizar la calibración de la cámara se utilizó la herramienta de Matlab Camera Calibration, mediante el uso de esta herramienta se tomaron 20 imágenes de una plantilla de cuadros blancos y negros equidistantes, en estas imágenes se identifican el punto de origen y las intersecciones de cada uno de los cuadros con el fin de obtener los parámetros intrínsecos y extrínsecos de la cámara, como imagen final se toma la plantilla apoyada sobre el suelo en el espacio donde será ubicado el entorno, esto con el fin de obtener los parámetros de la cámara a nivel del suelo y aplicarlos a las coordenadas obtenidas con la cámara y así conocer la distancia real de los puntos en el espacio.

La matriz de rotación y el vector de traslación obtenidos mediante la calibración de la cámara se presentan en las ecuaciones 3.4 y 3.5.

$$R = \begin{bmatrix} 0,9968 & 0,0486 & -0,0642 \\ -0,0484 & 0,9988 & 0,0044 \\ 0,0643 & -0,0013 & 0,9979 \end{bmatrix} \quad (3.4)$$

$$t = \begin{bmatrix} -261,952 & -411,704 & 2291,216 \end{bmatrix} \quad (3.5)$$

La matriz de parámetros intrínsecos obtenida en la calibración se presenta en la ecuación 3.6.

$$K = \begin{bmatrix} 623,6563 & 0 & 0 \\ 0 & 619,3293 & 0 \\ 308,4118 & 233,7732 & 1 \end{bmatrix} \quad (3.6)$$

Adicionalmente con el fin de poder realizar mediciones de distancia real en el entorno, fue necesario calcular un factor de escala para aplicar en el momento de convertir puntos de la imagen en puntos reales en el espacio, el factor de escala encontrado para la resolución en la que se trabajó la cámara se presenta en la ecuación 3.7.

$$FE = 232,2532 \quad (3.7)$$

3.5. Detección

La técnica de detección se basó en el modelo aditivo de Color RGB, el cual permite expresar cualquier color como la suma de tres colores básicos (Rojo, Verde, Azul), se basa en la teoría de tricromatismo del ojo humano en el cual se identifican tres conos receptores de luz que funcionan a diferentes frecuencias y

permiten lograr una visión en color, este modelo permite descomponer una imagen a color en sus tres componentes R, G y B y tratar sus componentes de color por separado. Con el objetivo de realizar la detección de las posiciones y direcciones de los vehículos y las paradas en el entorno se seleccionaron cuatro colores, rojo, azul, fucsia y amarillo. Para todos los colores con excepción del amarillo el procedimiento de detección fue idéntico.

Primero, de la imagen obtenida del entorno se substrajo el componente deseado (R,G,B), posteriormente se aplicó un filtro a la imagen en el cual el valor de cada pixel es reemplazado por el valor de la mediana de sus pixeles adyacentes, esto con el fin de eliminar ruido de la imagen, la cual, después de esto fue binarizada con umbrales distintos para cada uno de los colores que se deseaban detectar, para este punto, la imagen obtenida presenta figuras blancas en las áreas donde se detectó el color deseado y negro en las demás zonas (figura 3.5), finalmente con el fin de remover pequeños objetos que pudieran ser detectados sin ser deseados se implementó una función que remueve áreas con una cantidad de pixeles menor a la especificada. A partir de esta imagen se obtuvieron los centroides de cada una de estas áreas para obtener un punto X,Y y asociarlo al marcador detectado con este color.

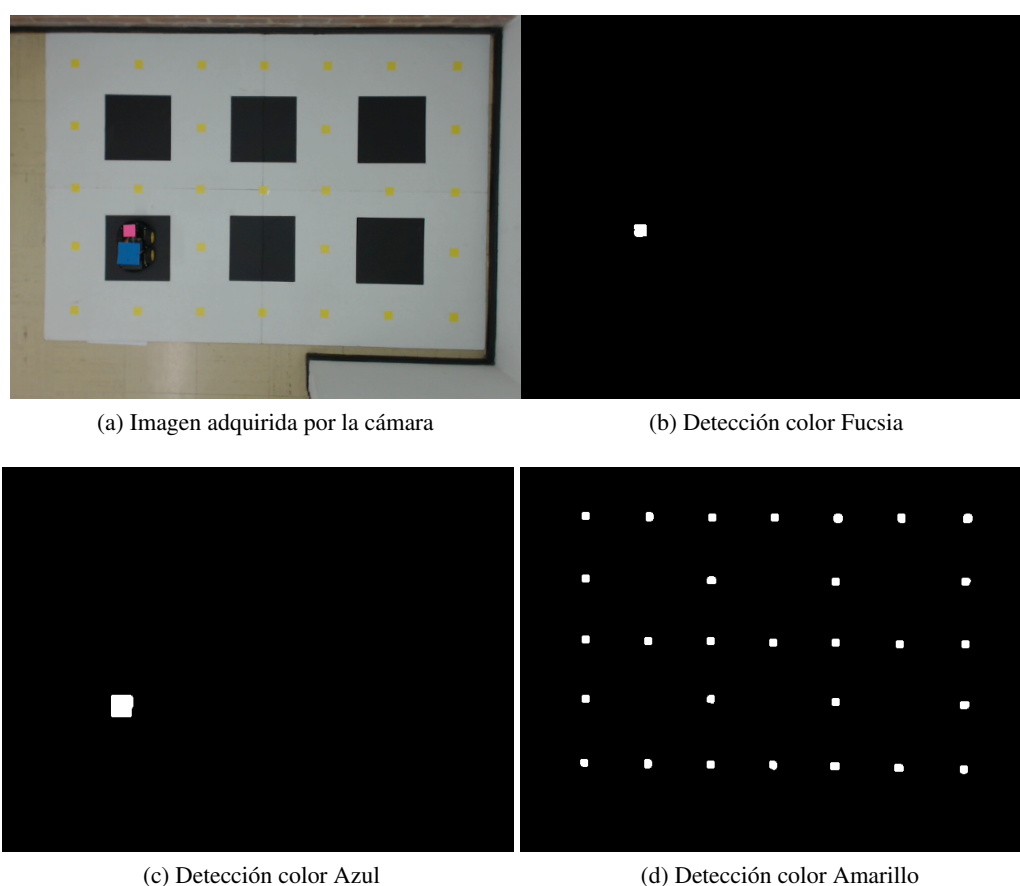


Figura 3.5: Imágenes adquiridas mediante la técnica de detección

Para la detección del color amarillo fue necesario aplicar una transformación a la imagen original aplicando el complemento, realizado esto el color amarillo es fácilmente detectable en el canal B de la imagen en complemento con la misma técnica de detección usada en los otros colores.

Con los centroides obtenidos en la detección y a partir de los parámetros encontrados en la calibración de la cámara se procedió a eliminar la distorsión presente en la medición, este procedimiento se realizó con el fin de garantizar una mayor fidelidad de las medidas, en el cuadro 3.1 se presenta una comparación entre las coordenadas medidas (X,Y) y las coordenadas sin distorsión (X',Y') obteniendo el valor del error entre ellas y su porcentaje equivalente, en los valores de porcentaje de error se observa que la mayor corrección

de distorsión se efectúa en los valores de X más cercanos a los bordes de la imagen. Finalmente los valores obtenidos se redondean y se entregan al sistema

X	Y	X'	Y'	Error X	Error Y	% Error X	% Error Y
81,663	367,663	85,369	365,475	3,706	2,189	4,539	0,595
83,500	61,500	87,481	64,549	3,981	3,049	4,768	4,958
83,430	138,110	86,782	139,535	3,352	1,425	4,018	1,032
83,500	215,500	86,499	215,744	2,999	0,244	3,592	0,113
83,356	288,317	86,470	287,563	3,114	0,755	3,736	0,262
161,500	217,500	162,486	217,609	0,986	0,109	0,611	0,050
161,500	369,500	163,144	367,981	1,644	1,519	1,018	0,411
163,115	61,740	165,058	64,040	1,943	2,301	1,191	3,726
240,852	141,500	241,139	141,892	0,287	0,392	0,119	0,277
239,500	217,500	239,616	217,527	0,116	0,027	0,048	0,013
239,663	289,663	239,843	289,517	0,180	0,146	0,075	0,050
239,500	370,852	239,999	369,860	0,499	0,992	0,208	0,267
241,222	62,500	241,883	64,186	0,661	1,686	0,274	2,697
316,890	371,430	316,839	370,603	0,051	0,827	0,016	0,223
317,656	219,115	317,655	219,116	0,001	0,002	0,000	0,001
319,500	63,500	319,404	64,982	0,096	1,482	0,030	2,333
394,500	373,500	393,801	372,365	0,699	1,135	0,177	0,304
395,500	143,500	395,062	143,954	0,438	0,454	0,111	0,316
395,500	219,500	395,275	219,537	0,225	0,037	0,057	0,017
395,430	292,890	395,115	292,676	0,315	0,214	0,080	0,073
398,670	63,634	397,714	65,434	0,955	1,801	0,240	2,830
474,139	375,713	472,024	373,902	2,114	1,811	0,446	0,482
475,500	220,500	474,106	220,611	1,394	0,111	0,293	0,050
477,942	63,577	475,466	66,063	2,476	2,486	0,518	3,910
556,250	296,815	552,345	295,821	3,905	0,993	0,702	0,335
555,500	377,500	551,092	374,936	4,408	2,564	0,794	0,679
558,330	143,366	554,206	144,858	4,124	1,492	0,739	1,041
557,500	221,500	553,698	221,687	3,802	0,187	0,682	0,085
560,600	64,333	555,854	67,522	4,746	3,188	0,847	4,956

Cuadro 3.1: Análisis de distorsión

Capítulo 4

Control de bajo nivel

En este capítulo se presentará el desarrollo teórico y en simulación de los controladores necesarios para la implementación física del sistema, el diagrama de bloques de la figura 4.1 presenta el lazo de control en cascada definido como solución al problema de la implementación.

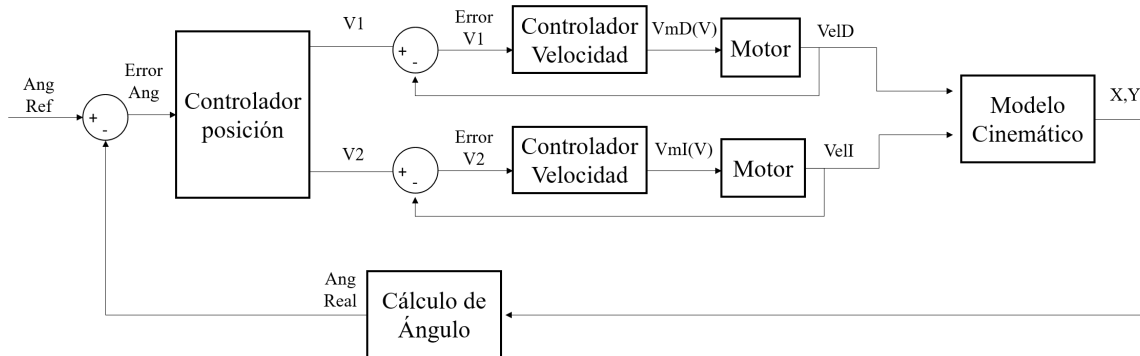


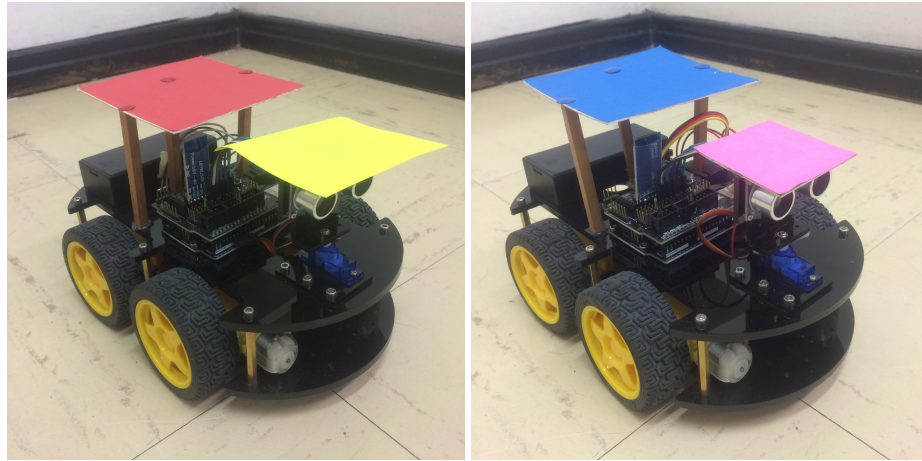
Figura 4.1: Diagrama de bloques general de control

4.1. Especificaciones de los vehículos

Los vehículos elegidos para el desarrollo de este trabajo de grado fueron los Smart Robot car kit producidos por la empresa Elegoo industries, las especificaciones de los vehículos son las siguientes:

- Tarjeta de desarrollo Elegoo Uno R3 inspirada en Arduino Uno R3.
- Tarjeta de extensión de sensores V5.0.
- Driver para los motores L298N.
- Sensor de ultrasonido HC-SR04.
- Módulo de comunicación Bluetooth HC-06.
- 4 motores DC.
- 2 Baterías Ultrafire 18650 de 3.7V 4200 mAh
- Elementos básicos para montaje y ensamblaje.

Estos vehículos fueron seleccionados debido a su fácil ensamblado, adicionalmente incluyen el driver en una placa especialmente diseñada para la alimentación de los motores, la tarjeta de desarrollo puede ser programada en el IDE de Arduino y además incluyen los sensores necesarios para la implementación del sistema de control [9].



(a) Vehículo Rojo

(b) Vehículo Azul

Figura 4.2: Vehículos implementados

4.2. Protocolo de comunicación

Bluetooth es una herramienta que permite la comunicación de diferentes dispositivos mediante una red inalámbrica [10], en la cual es posible conectar diversos dispositivos permitiendo transmisión de datos de forma bidireccional, opera en la banda de 2.4GHz ISM (Industrial, Scientific, Medical) y se basa en un protocolo de tipo serial, debido a esto es de fácil implementación en tarjetas de desarrollo como las integradas en los vehículos y en herramientas computacionales como Matlab. En la implementación del sistema se utilizó un módulo Bluetooth HC-06, el cual permite conectarse a otros dispositivos como un dispositivo esclavo y capaz de recibir y transmitir datos, adicionalmente es de fácil conexión con la tarjeta de desarrollo integrada en los vehículos vía puerto serial.

Para poder implementar la comunicación Bluetooth entre el computador y los vehículos fue necesario utilizar el Instrument Control Toolbox de Matlab [11], éste permite establecer comunicación entre Matlab y diferentes instrumentos mediante conexiones Bluetooth, TCP/IP, UDP entre otras.

4.3. Caracterización de los motores

Con el objetivo de implementar un control de velocidad para los vehículos fue necesario caracterizarlos y encontrar una función de transferencia asociada para estos [12], esta caracterización se realizó como una función de primer orden de ganancia y tiempo muerto (4.1). Se realizó esta caracterización para dos motores, lado derecho e izquierdo del vehículo considerando los dos motores de cada lado idénticos.

$$G(s) = \frac{K}{T_s \cdot s + 1} \quad (4.1)$$

Para caracterizar la ganancia se aplicaron diferentes valores de voltaje a los motores y se midió la velocidad del motor en RPM (Revoluciones por minuto) para cada uno de estos valores, esta ganancia se definió como el valor final de velocidad del motor sobre el valor de voltaje aplicado. Las ganancias obtenidas para cada motor se presentan en las ecuaciones 4.2 y 4.3.

$$K_{motorI} = 64,172_{[RPM/V]} \quad (4.2)$$

$$K_{motorD} = 56,074_{[RPM/V]} \quad (4.3)$$

La medición de tiempo muerto se realizó aplicando una señal paso de voltaje al motor y midiendo la señal de corriente de arranque de éste, se definió como tiempo muerto el tiempo que tarda la corriente en llegar al 63% de su valor final. Este tiempo muerto fue definido igual para ambos motores y se presenta en la ecuación 4.4.

$$Ts = 6,72ms \quad (4.4)$$

Finalmente las funciones de transferencia asociadas a cada motor se presentan en las ecuaciones 4.5 y 4.6.

$$G_{mI}(s) = \frac{64,172}{6,72 \cdot s + 1} \quad (4.5)$$

$$G_{mD}(s) = \frac{56,074}{6,72 \cdot s + 1} \quad (4.6)$$

4.4. Control de velocidad

Debido a la diferencia encontrada en la caracterización de los motores derecho e izquierdo del vehículo, aplicando un voltaje de entrada igual para cada motor se obtendrían valores de velocidad distintos como salida, generando desviaciones no deseadas en el momento de mover el vehículo en línea recta, adicionalmente para obtener un mejor desempeño del controlador de seguimiento de trayectoria la velocidad de desplazamiento de cada lado del vehículo debería mantenerse constante, por lo cual, se hizo necesario diseñar e implementar un control de velocidad para estos.

Inicialmente se obtuvo la respuesta en lazo abierto de las funciones de transferencia de los motores a partir de la caracterización presentada en la sección 5.2.

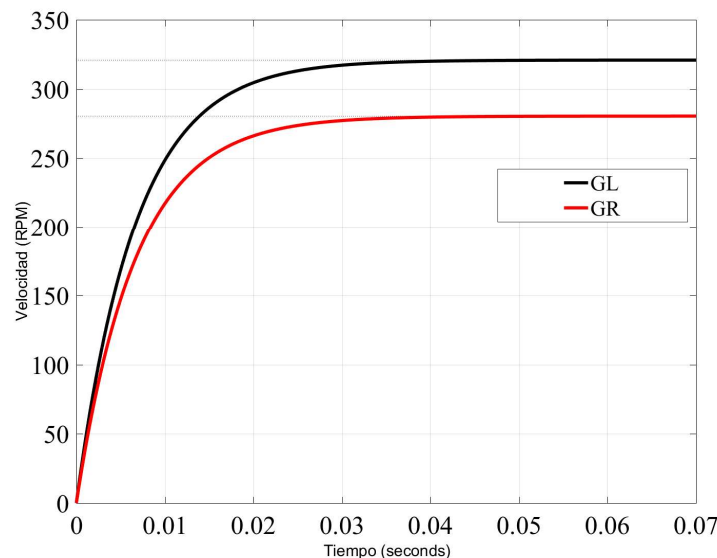


Figura 4.3: Respuesta en lazo abierto GmI y GmD

Posteriormente se realizó el diseño de un controlador PI para la velocidad de los motores, los parámetros de desempeño deseados para éste fueron $T_s = 50ms$ y $PO = 10\%$. Las funciones de transferencia de los controladores se presentan en las ecuaciones 4.7 y 4.8.

$$C_{mI}(s) = \frac{0,00117 \cdot s + 2,3238}{s} \quad (4.7)$$

$$C_{mD}(s) = \frac{0,00134 \cdot s + 2,6594}{s} \quad (4.8)$$

Con el fin de evaluar el comportamiento del controlador diseñado en simulación en la figura se presenta la respuesta paso en lazo cerrado del control de velocidad para cada una de las funciones de transferencia de los motores.

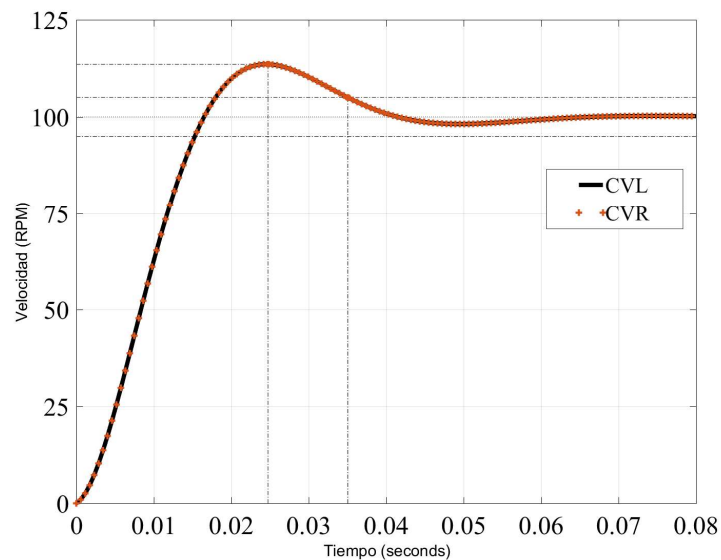


Figura 4.4: Respuesta en lazo cerrado GmI y GmD

Para realizar la implementación del control de velocidad fue necesario diseñar una estrategia de medición de la velocidad de los motores, esto debido a que el vehículo no contaba con un encoder integrado, el diseño del encoder se centró en un sensor infrarrojo CNY70, el cual contiene un led emisor de luz infrarroja con un foto transistor en un solo componente de cuatro pines. El funcionamiento del sensor se basa en la cantidad de luz emitida por el led infrarrojo y recibida por el foto transistor, este nivel varía dependiendo de la distancia y la reflexión del objeto frente al sensor, partiendo de esto para medir la velocidad de los motores se pintaron sobre la parte interior de las ruedas ocho líneas blancas espaciadas igualmente, en este caso los colores blanco y negro reflejan la luz infrarroja en niveles diferentes generando en la salida del sensor valores cercanos a 0V y 5V que pueden ser tomados como 0 ó 1 lógico en la tarjeta de desarrollo del vehículo. El esquemático del circuito implementado se presenta en la figura 4.5.

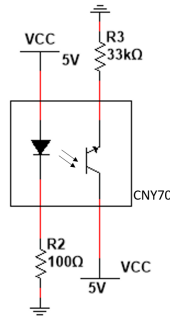


Figura 4.5: Esquemático sensor CNY70

Este sensor se instaló en el chasis del vehículo ubicado de tal forma que identificara los sectores blanco y negro de la rueda con el fin de leer una señal digital, a partir de ésta medir el tiempo de rotación de la rueda y así calcular su velocidad en RPM. El control de velocidad diseñado fue implementado directamente en la tarjeta de desarrollo del vehículo, para esto se utilizó la librería Arduino PID [13], la cual, permite mediante funciones incluidas declarar un controlador con los valores de K_p , K_i y K_d , definir un valor de referencia y calcular el valor de la señal de control a partir de la señal de salida. En la implementación del sistema conjunto, el control de velocidad presentaba un tiempo de respuesta mayor al control de seguimiento de trayectoria, lo cual incrementaba el error en el seguimiento sobre todo en tramos cortos como los implementados dentro del entorno, por esta razón el control de velocidad no fue tenido en cuenta en el momento de la implementación final.

4.5. Control de seguimiento de trayectoria

Para el diseño del controlador como primera instancia es necesario modelar el vehículo, para este caso se estableció un modelo cinemático en el cual el vehículo se representa como una partícula sobre la cual actúan dos vectores ortogonales de velocidad, mediante este modelo el ángulo de desplazamiento del vehículo depende directamente de la magnitud de estos vectores, los cuales son asociados en el momento de la implementación a las velocidades derecha e izquierda del vehículo.

A partir de este modelo se definió el lazo de control de la figura 4.6, en el cual, la posición del vehículo se evalúa punto a punto con el fin de calcular el ángulo de su trayectoria y el error respecto a la trayectoria deseada, ajustar las velocidades del vehículo y así corregir la trayectoria.

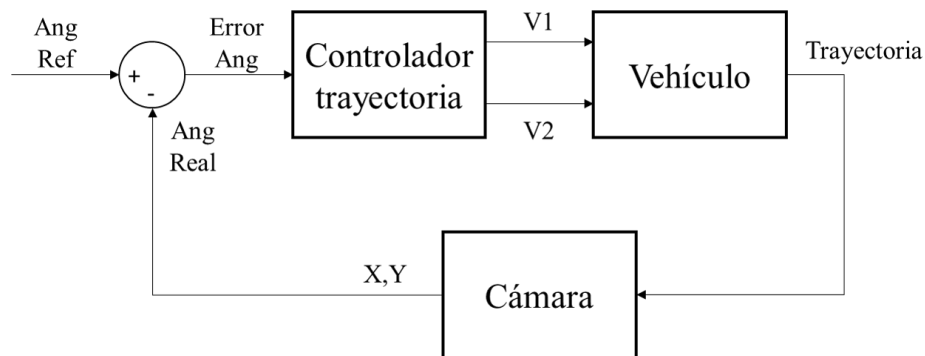


Figura 4.6: Lazo de control de seguimiento de trayectoria

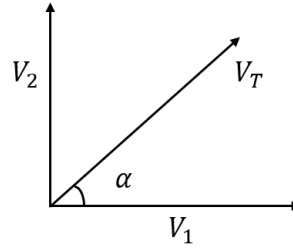


Figura 4.7: Modelo cinemático del vehículo

Las ecuaciones cinemáticas que describen el modelo(4.7) se expresan en términos de las velocidades derecha (V_1) e izquierda (V_2) que determinan la magnitud de la velocidad del vehículo. A partir de la velocidad es posible obtener la distancia y los puntos X,Y del movimiento y así mismo controlar el seguimiento de trayectoria.

$$V_1 = V_T \cdot \cos \alpha \quad (4.9)$$

$$V_2 = V_T \cdot \sin \alpha \quad (4.10)$$

$$V_{Mag} = \sqrt{V_1^2 + V_2^2} \quad (4.11)$$

A partir de la magnitud del vector velocidad es posible obtener el valor de la magnitud del vector posición con la ecuación (4.12), con esta magnitud se pueden hallar los puntos X, Y actuales de la trayectoria como se muestra en las ecuaciones (4.13) y (4.14) donde a medida que el vehículo avanza se debe sumar el punto P_x y P_y anterior, además de restar el punto inicial para poder obtener los puntos siempre respecto a la misma referencia.

$$distancia = velocidad \cdot tiempo \quad (4.12)$$

$$P_x = P_x + P_{Mag} \cdot \cos(V_{Ang}) - x_i \quad (4.13)$$

$$P_y = P_y + P_{Mag} \cdot \sin(V_{Ang}) - y_i \quad (4.14)$$

Con los puntos X, Y obtenidos es posible calcular el ángulo α como se muestra en la ecuación (4.15) para poder determinar si el vehículo se encuentra en la dirección adecuada, es decir que a partir de este ángulo es posible determinar el seguimiento de la referencia del sistema de control.

$$\alpha = \arctan\left(\frac{P_y}{P_x}\right) \quad (4.15)$$

A partir de este modelo se diseñó la estrategia de control para el seguimiento de trayectoria, con un punto de inicio y fin definidos en coordenadas cartesianas se define un vector desplazamiento, la magnitud y dirección de este vector representan la distancia y ángulo asociados al recorrido. Inicialmente las velocidades derecha e izquierda del vehículo son idénticas para iniciar un movimiento en línea recta, ya iniciado el desplazamiento, mediante la detección de imágenes se asocia un punto X,Y al vehículo con el fin de definir un nuevo vector para el desplazamiento real de éste, el sistema de control a partir del ángulo de este nuevo vector genera una señal de error entre éste y el ángulo de referencia, el controlador genera las variaciones necesarias en las velocidades derecha e izquierda con el fin de llevar este error a cero y alinear el vehículo con el vector desplazamiento deseado, el sistema da por terminado el desplazamiento en el momento en que las magnitudes de ambos vectores son iguales, alcanzando el punto objetivo.

4.5.1. Controlador PI

Como estrategia inicial se diseñó un controlador PI en el dominio del tiempo, esto con el fin de garantizar las condiciones de desempeño necesarias para el sistema, en este caso la señal de control definida fue el ángulo del vector velocidad, a partir del cual y mediante el uso de las ecuaciones 4.9 y 4.10 se calculaban los valores de velocidad enviados al vehículo, mediante la aplicación de las ecuaciones cinemáticas del modelo se calculó la posición del vehículo y sus coordenadas X,Y para así obtener el ángulo de éste respecto al punto inicial y lograr realizar la corrección respecto a la referencia definida con los puntos de inicio y fin. En la figura 4.8 se presentan las gráficas de simulación de este controlador.

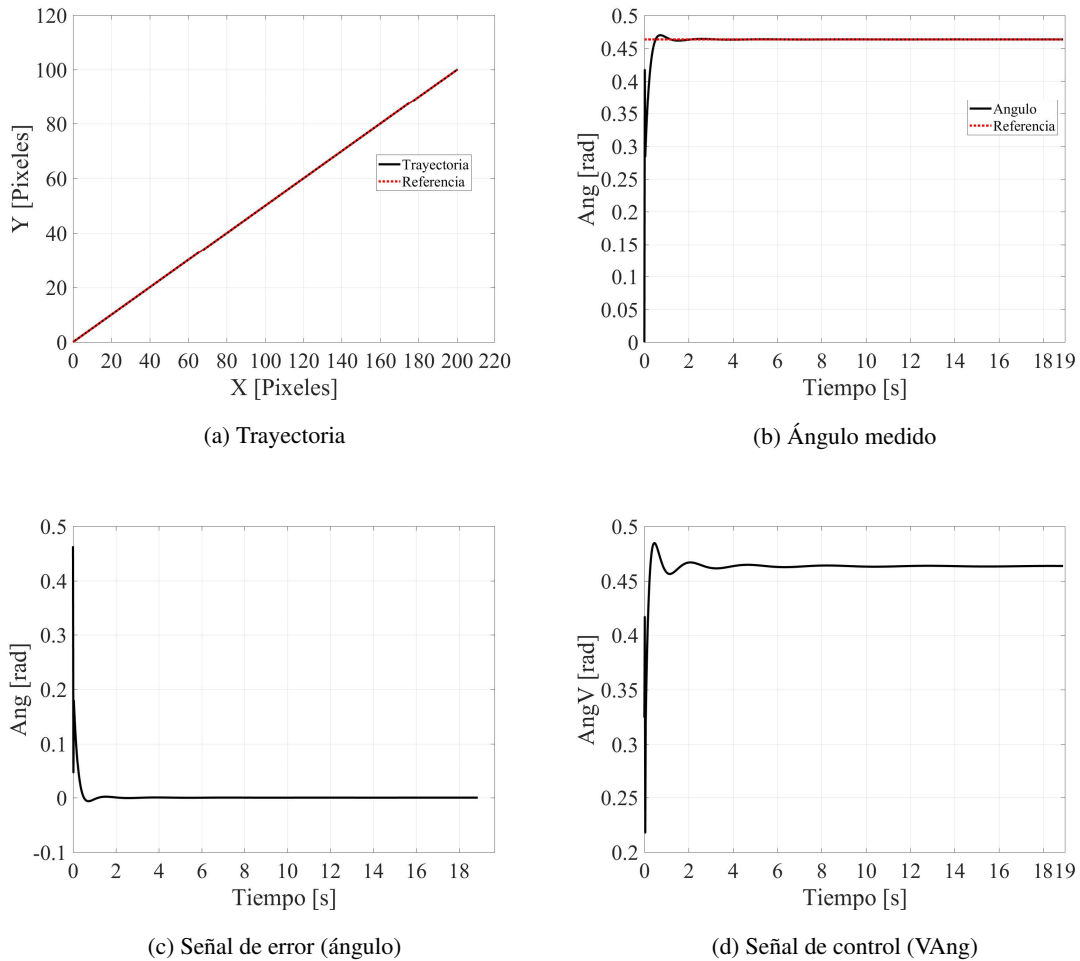


Figura 4.8: Simulación control PI

En la etapa de implementación este controlador presentó inconvenientes, principalmente debido a la dependencia del tiempo de la señal de control, esto sumado al tiempo de ejecución del sistema y a la dificultad del manejo de la señal de control en el momento de enviarla a la tarjeta de desarrollo del vehículo, no permitieron lograr una implementación adecuada, por este motivo el controlador implementado fue el On/Off de la sección 4.5.2. Las constantes del controlador para la simulación fueron $K_p = 0,5$ y $K_i = 10$, estas constantes fueron ajustadas manualmente buscando las condiciones de desempeño esperadas, las condiciones de desempeño obtenidas para el controlador PI simulado se presentan en el cuadro 4.1.

4.5.2. Controlador ON/OFF

Se diseñó una ley de control a partir del modelo cinemático del vehículo, la cual como señal de control maneja las velocidades izquierda y derecha apagando o encendiendo los motores según el valor del error de ángulo calculado mediante la ecuación 4.15, el movimiento del vehículo se simuló con las ecuaciones cinemáticas 4.13 y 4.14, finalmente los valores de referencia de magnitud y ángulo fueron definidos mediante un punto de inicio y fin arbitrarios. Los resultados de la simulación de este controlador se presentan en la figura 4.9.

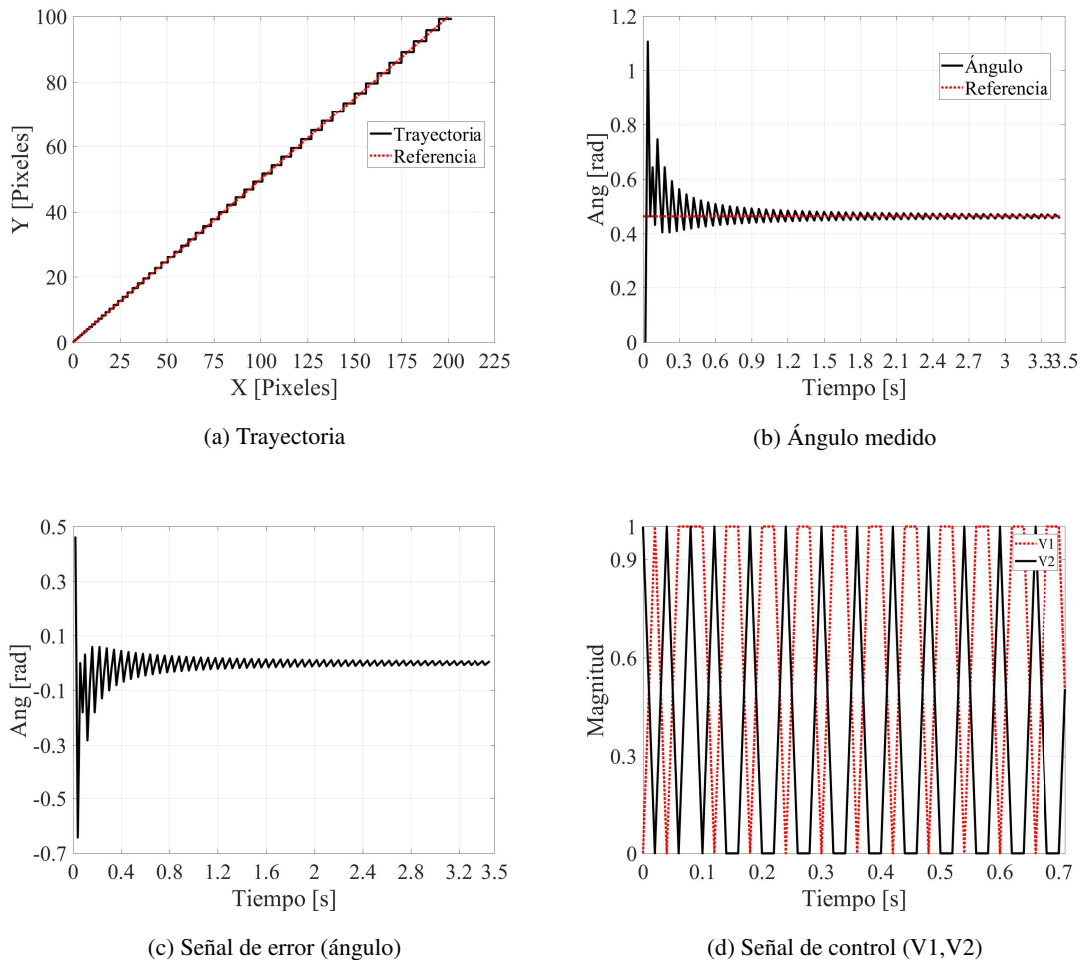


Figura 4.9: Simulación control On/Off

En la figura 4.9a se observa que la trayectoria del vehículo se divide en pequeños tramos en los cuales el valor del error (4.9c) cambia de valores positivos a negativos, el comportamiento de este controlador depende directamente de la frecuencia de muestreo, aumentando su resolución y por lo tanto su precisión, en (4.9d) se observan las velocidades de cada lado del vehículo y su conmutación entre valores de 0 y 1 en magnitud con el fin de simular el encendido y apagado de los motores. Los parámetros de desempeño obtenidos en la simulación de este controlador se presentan en el cuadro 4.1.

Controlador PI					Controlador On/Off				
Ref	Max	PO%	ts (s)	Ess %	Ref	Max	PO%	ts (s)	Ess %
0,464	0,470	1,402	0,360	0,000	0,464	0,745	60,785	0,920	1,553

Cuadro 4.1: Parámetros de desempeño controlador PI y On/Off

A partir de los resultados obtenidos mediante simulación, es posible concluir que el comportamiento del controlador PI presenta mejores condiciones de desempeño que el controlador On/Off, el valor del sobrepico del controlador On/Off es crítico y supera drásticamente el valor esperado para este parámetro, en cuanto a tiempo de establecimiento y error de estado estable el controlador On/Off presenta menor rendimiento respecto al PI, sin embargo los parámetros obtenidos se encuentran dentro de los límites definidos para el proyecto.

Capítulo 5

Implementación

5.1. Implementación con un vehículo

Para la implementación del sistema se construyó un entorno con todos los paraderos propuestos sobre el cual se moverían los vehículos. (5.1) Para el seguimiento de la ruta entregada por el algoritmo se decidió descomponer el problema del movimiento en dos momentos, desplazamiento en línea recta y giro en las intersecciones. Para el desplazamiento en línea recta se implementó el controlador On/Off que realizó el seguimiento de trayectoria con el fin de recorrer los tramos entre paradas.

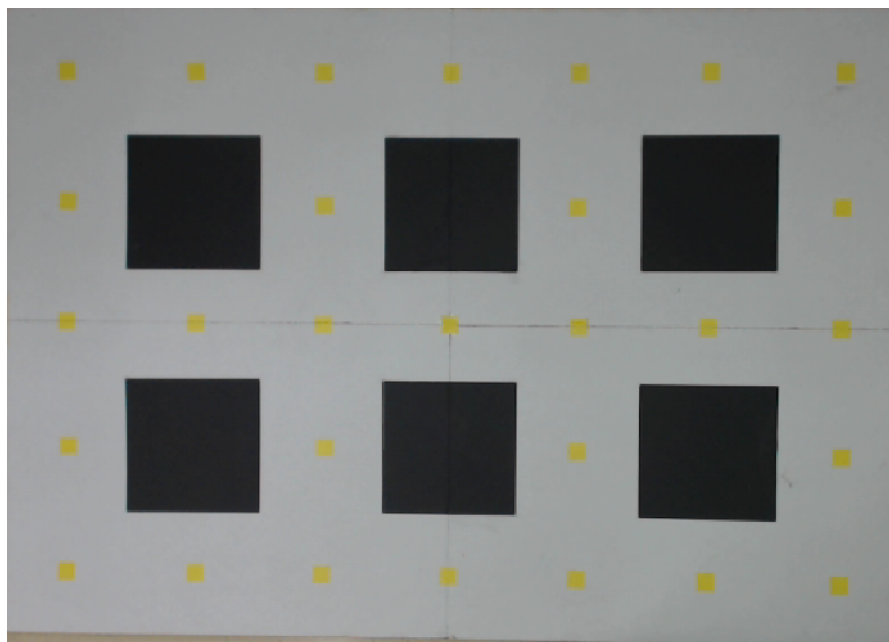


Figura 5.1: Entorno construido

Para el movimiento de giro del vehículo, se diseñó una función que a partir de la detección de los dos puntos del vehículo (frontal y posterior) (figura 5.2) comparara el giro realizado con el giro deseado, en particular, para el cumplimiento de los posibles recorridos se necesitaban tres tipos de giro, a la izquierda, a la derecha y media vuelta en caso de que el vehículo tuviera que devolverse en algún tramo. Con el fin de reducir la cantidad de datos enviados al vehículo y por lo tanto el tiempo de transmisión de las instrucciones, se decidió implementar los tres tipos de giro en una sola dirección es decir, para girar a la derecha debía moverse $\frac{\pi}{2}$ para moverse a la izquierda $\frac{3\pi}{2}$ y para devolverse debía moverse π .

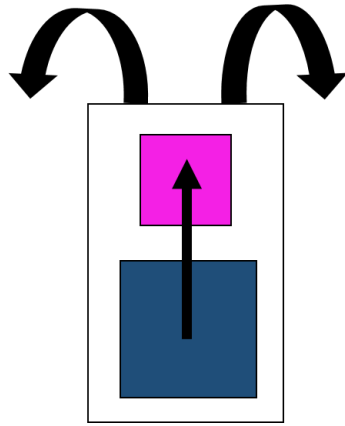


Figura 5.2: Modelo implementado para los vehículos

Por otro lado, la ruta entregada por el algoritmo se encontraba en términos de coordenadas X,Y por lo que fue necesario realizar una transformación de coordenadas de la ruta a pixeles en el entorno para poder identificar cada paradero y enviar los puntos de inicio y fin de cada trayecto a cada uno de los vehículos.

Finalmente, para poder implementar la ruta deseada en cada uno de los vehículos fue necesario descomponer la ruta entregada por el algoritmo en una lista ordenada de instrucciones, esta puede contener cuatro tipos de acción (avanzar, giro a la derecha, giro a la izquierda, media vuelta) las cuales serán ejecutadas por el vehículo, monitoreadas por la cámara y controladas por el computador, adicionalmente la lista contiene las coordenadas (en pixeles) X,Y objetivo de cada tramo.

Para demostrar el funcionamiento del sistema completo a continuación se presenta el desarrollo completo de una ruta y su ejecución.

Inicialmente se define el punto de inicio las paradas de la ruta y el punto de fin, las paradas definidas para la ruta se presentan en el cuadro 5.1.

Filas(Y)	Columnas(X)
5	1
1	1
3	4
2	5
1	6
3	7

Cuadro 5.1: Punto de inicio y paraderos deseados

Estas coordenadas de paradas deseadas entran al bloque algoritmo descrito en el diagrama de bloques general, inicialmente las paradas son ordenadas y a partir de los costos definidos para el grafo del entorno, se ejecuta el algoritmo A* entre cada una de las paradas hasta llegar al punto de fin, finalmente la trayectoria entregada por el algoritmo se presenta en el cuadro 5.2a.

Filas(Y)	Columnas(X)
5	1
4	1
3	1
2	1
1	1
1	2
1	3
2	3
3	3
3	4
3	5
2	5
1	5
1	6
1	7
2	7
3	7

(a) Ruta generada por el algoritmo

Movimiento	X	Y
1	49	298
1	48	222
1	48	148
1	48	69
2	0	0
1	127	70
1	205	70
2	0	0
1	203	152
1	206	224
3	0	0
1	282	224
1	360	226
3	0	0
1	360	152
1	359	70
2	0	0
1	438	70
1	520	70
2	0	0
1	522	153
1	522	228

(b) Lista de instrucciones

Cuadro 5.2: Ruta y lista de instrucciones ruta #1

Con esta trayectoria definida luego se procede a generar la lista de instrucciones para cada movimiento de la ruta (Cuadro 5.2b), esta lista será la base para ejecutar cada movimiento mediante las funciones de línea recta y giro explicadas al inicio de este capítulo.

Finalmente el sistema ejecuta las instrucciones enviándolas al vehículo mediante la comunicación Bluetooth. En la figura 5.3 se presenta la trayectoria deseada como la línea punteada, y la trayectoria ejecutada para cada tramo en línea continua, se observa una adecuada ejecución de la lista de instrucciones, pero con errores apreciables en el seguimiento de cada tramo.

Con el objetivo de cuantificar el error visualizado en la ejecución de la trayectoria, para cada tramo se evaluó el error relativo de distancia entre el punto alcanzado y el punto objetivo, a partir de esto se encontraron los valores de mediana, varianza y desviación estándar, adicionalmente se presenta el error acumulado en términos de la distancia total recorrida por la ruta, los resultados de este análisis estadístico se presentan en el cuadro 5.3.

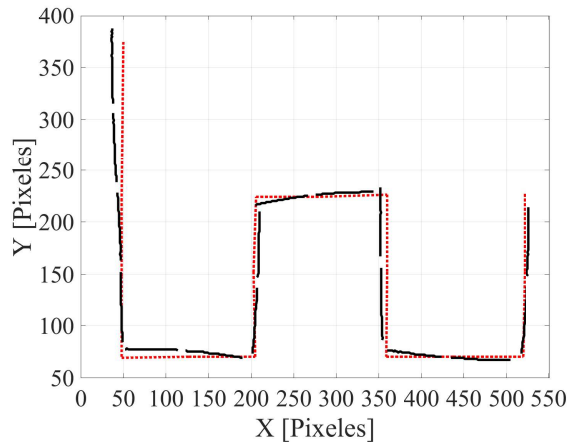


Figura 5.3: Trayectoria obtenida de la ruta # 1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado %
72,028	72,028	11	15	18,601	18,601	25,825
66,008	138,035	9	15	36,094	17,493	26,148
64,281	202,316	3	14	50,412	14,318	24,917
65,490	267,806	7	17	68,797	18,385	25,689
54,452	322,258	13	14	87,902	19,105	27,277
67,067	389,325	15	18	111,332	23,431	28,596
47,043	436,368	0	11	122,332	11,000	28,034
60,133	496,501	3	14	136,650	14,318	27,523
67,476	563,977	13	0	149,650	13,000	26,535
68,029	632,006	15	0	164,650	15,000	26,052
65,276	697,283	3	14	178,968	14,318	25,666
68,264	765,547	6	17	196,996	18,028	25,733
54,037	819,584	13	6	211,313	14,318	25,783
69,260	888,844	16	1	227,345	16,031	25,578
63,071	951,916	1	15	242,378	15,033	25,462
63,388	1015,303	7	15	258,931	16,553	25,503
Mediana					15,532	25,804
Varianza					8,167	1,013
Desviación Estándar					2,858	1,007

Cuadro 5.3: Errores de desplazamiento y mediciones estadísticas

En el cuadro se presenta la distancia del recorrido objetivo para cada tramo (Distancia Obj) y adicionalmente la distancia acumulada al ejecutar cada tramo de la trayectoria (Distancia Acu), seguido de esto a partir de los errores en píxeles de X y Y se presenta el error en distancia para cada tramo (Error Dist) junto con el error de distancia acumulado (Error Dist Acu), finalmente se presenta el error acumulado relativo en porcentaje.

Las medidas estadísticas que se indican en el cuadro evidencian un error acumulado casi constante para cada tramo y en general para toda la ruta, conclusión soportada por los valores de varianza y desviación estándar, adicionalmente se observa que el error de distancia por tramo está directamente relacionado con la distancia objetivo. El error acumulado presentado es generado por los errores de ejecución de las instruc-

ciones de línea recta y giro, en el caso del desplazamiento en línea recta el valor de error de estado estable diferente de cero que presenta el controlador On/Off, para el caso del giro las desviaciones de ángulo final presentadas durante el movimiento. Otro factor importante es la posición inicial del vehículo, ya que no se puede garantizar una ubicación precisa y constante en el momento de iniciar la ruta.

5.2. Implementación con dos vehículos

Para ejecutar el movimiento de dos vehículos de forma simultánea dentro del entorno, la estrategia implementada fue la siguiente: se ingresaron dos grupos de paradas obligatorias y a partir de estas se generaron dos rutas (una para cada vehículo) cada una con una lista de instrucciones asociada, en este caso las listas son recorridas simultáneamente ejecutando una acción por ruta en cada iteración, esto con el objetivo de que ambos vehículos recorran el entorno al mismo tiempo.

Para esta implementación fue necesario considerar la existencia de posibles colisiones dentro del entorno, realizando un análisis de este y los posibles desplazamientos de los vehículos se determinó que las colisiones más probables al momento de ejecutar dos rutas simultáneas eran colisiones en intersección, debido a la descomposición del problema en instrucciones ejecutadas alternadamente se decidió monitorear constantemente la distancia en los ejes X y Y de los puntos frontales de cada vehículo, esto con el objetivo de poder decidir en que momento podía ocurrir una colisión y prevenirla.

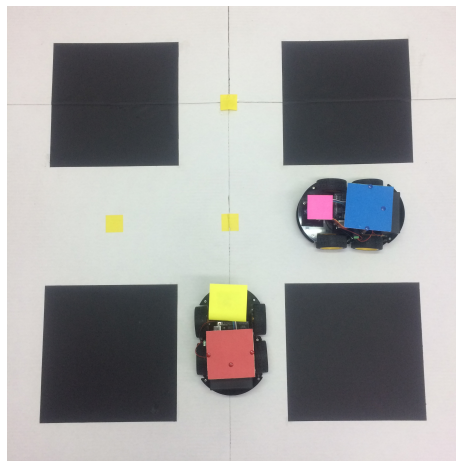


Figura 5.4: Colisión de intersección

La medición de distancia requerida para la implementación de esta estrategia de prevención de colisiones fue posible gracias al procedimiento de calibración de la cámara presentado en la sección 3.4, a partir de los parámetros obtenidos fue posible asociar los puntos X,Y de la imagen a puntos reales en el espacio y calcular su distancia. Se definió entonces como criterio para considerar una posible colisión una distancia menor a 30cm en ambos ejes coordenados, es decir, en el momento de ejecutar alguna instrucción de la lista la función de detección de colisiones evalúa la distancia entre los vehículos, de tal forma que en el momento en que ambos vehículos lleguen simultáneamente a una intersección la distancia entre ellos será menor a la definida en el umbral y se procederá a evitar la colisión.

El método para prevenir la colisión fue entonces, detener la ejecución de la lista de instrucciones durante dos iteraciones para el último vehículo en llegar a la intersección, dando prioridad al otro vehículo, esperando a que este cruce la intersección y así evitar la colisión.

Para demostrar el funcionamiento del sistema, a continuación se presentan los resultados de la ejecución de una ruta con movimiento de los dos vehículos, en esta ruta es posible verificar la estrategia de detección de colisiones ejecutada en el cruce de intersección del nodo $Y = 3$, $X = 5$.

Ruta conjunta 1			
Ruta Azul		Ruta Roja	
Filas(Y)	Columnas(X)	Filas(Y)	Columnas(X)
1	1	5	1
3	1	5	3
3	3	5	5
3	5	3	5
3	7	1	5

Cuadro 5.4: Puntos de inicio y paraderos deseados ruta conjunta 1

Vehículo Azul					Vehículo Rojo				
Ruta	Instrucciones				Ruta	Instrucciones			
1	1	1	49	147	5	1	1	128	374
2	1	1	49	220	5	2	1	206	375
3	1	3	0	0	5	3	1	284	375
3	2	1	128	221	5	4	1	362	377
3	3	1	206	222	5	5	3	0	0
3	4	1	282	222	4	5	1	361	299
3	5	1	362	223	3	5	1	362	223
3	6	1	440	223	2	5	1	361	149
3	7	1	522	223	1	5	1	361	67

Cuadro 5.5: Ruta y lista de instrucciones ruta conjunta #1

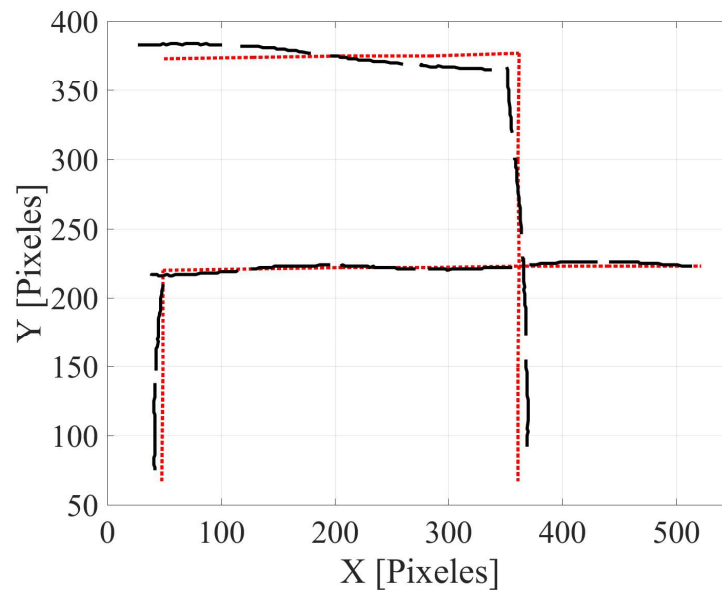


Figura 5.5: Trayectoria obtenida de la ruta conjunta # 1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado %
Ruta Azul						
63,000	63,000	7	9	11,402	11,402	18,098
63,285	126,285	0	10	21,402	10,000	16,947
76,105	202,390	13	1	34,440	13,038	17,017
70,064	272,455	10	2	44,638	10,198	16,384
66,030	338,485	11	1	55,684	11,045	16,451
74,007	412,492	6	1	61,766	6,083	14,974
60,033	472,525	9	3	71,253	9,487	15,079
73,062	545,587	8	0	79,253	8,000	14,526
Mediana					10,099	16,417
Varianza					3,983	1,310
Desviación Estándar					1,996	1,145
Ruta Roja						
74,000	74,000	27	9	28,460	28,460	38,460
62,201	136,201	27	2	55,534	27,074	40,774
64,281	200,482	24	6	80,273	24,739	40,040
63,127	263,609	24	12	107,106	26,833	40,631
48,260	311,868	5	20	127,721	20,616	40,954
55,326	367,195	2	23	150,808	23,087	41,070
57,079	424,274	7	24	175,808	25,000	41,437
63,008	487,282	8	25	202,057	26,249	41,466
Mediana					25,624	40,864
Varianza					5,452	0,840
Desviación Estándar					2,335	0,917

Cuadro 5.6: Errores de desplazamiento y mediciones estadísticas ruta conjunta

En los resultados obtenidos en esta prueba se mantiene la tendencia observada para la ruta individual, un error acumulado similar para cada tramo de la ruta y casi constante a medida que se progresa en el recorrido, nuevamente soportado en las medidas estadísticas de varianza y desviación estándar, se observa un comportamiento con error más alto para el vehículo rojo, este error puede ser causado por una desviación considerable en su punto de inicio.

Capítulo 6

Resultados y protocolo de pruebas

6.1. Línea Recta

Con el objetivo de validar el desempeño del controlador de seguimiento de trayectoria se evaluaron cuatro desplazamientos en diferentes direcciones, para cada uno de estos se obtuvieron las siguientes gráficas: la trayectoria deseada contra la trayectoria ejecutada por el vehículo, las señales de control V1 y V2 equivalentes a la velocidad de las ruedas derecha e izquierda del vehículo, el ángulo medido durante el desplazamiento y la señal de error con el fin de evaluar el seguimiento de la referencia. Al final de esta sección se presenta una compilación de los parámetros de desempeño evaluados para cada una de las pruebas con el fin de realizar una comparación y evaluar el desempeño general del controlador.

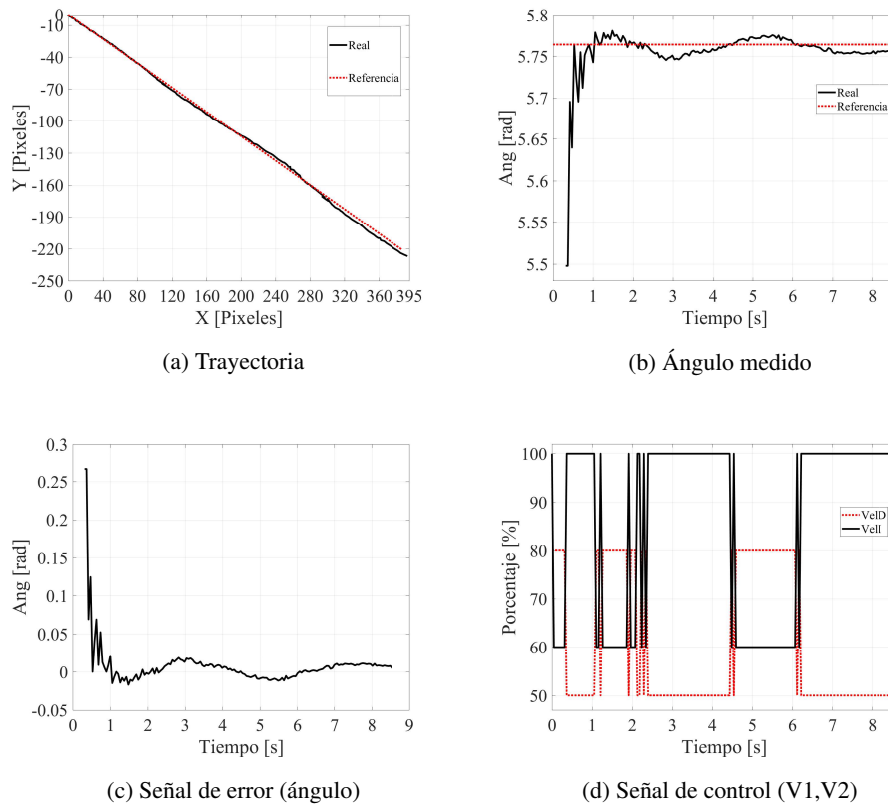
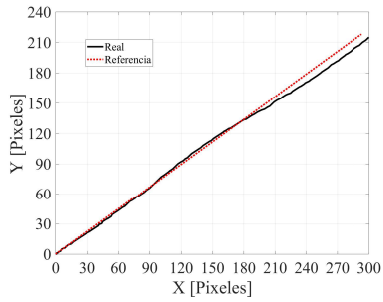
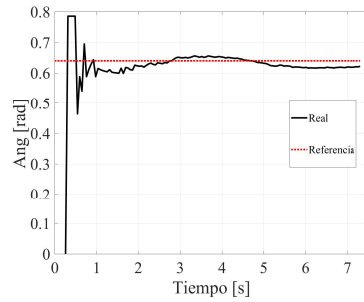


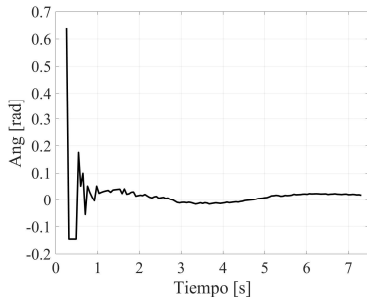
Figura 6.1: Resultados prueba de línea recta 1



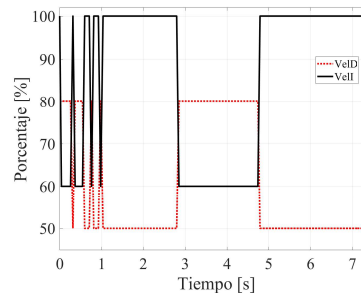
(a) Trayectoria



(b) Ángulo medido

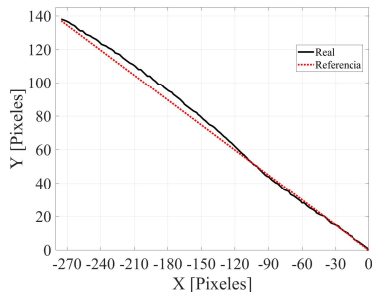


(c) Señal de error (ángulo)

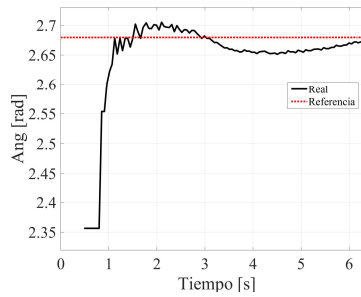


(d) Señal de control (V1,V2)

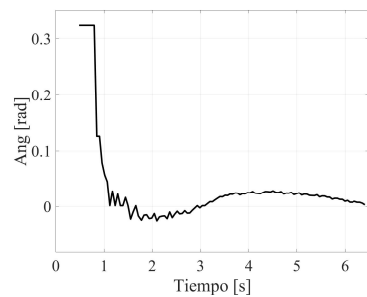
Figura 6.2: Resultados prueba de línea recta 2



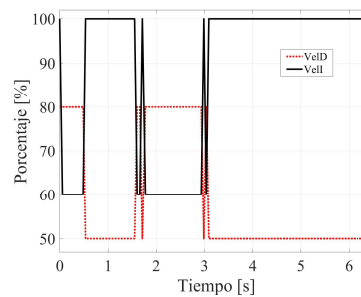
(a) Trayectoria



(b) Ángulo medido



(c) Señal de error (ángulo)



(d) Señal de control (V1,V2)

Figura 6.3: Resultados prueba de línea recta 3

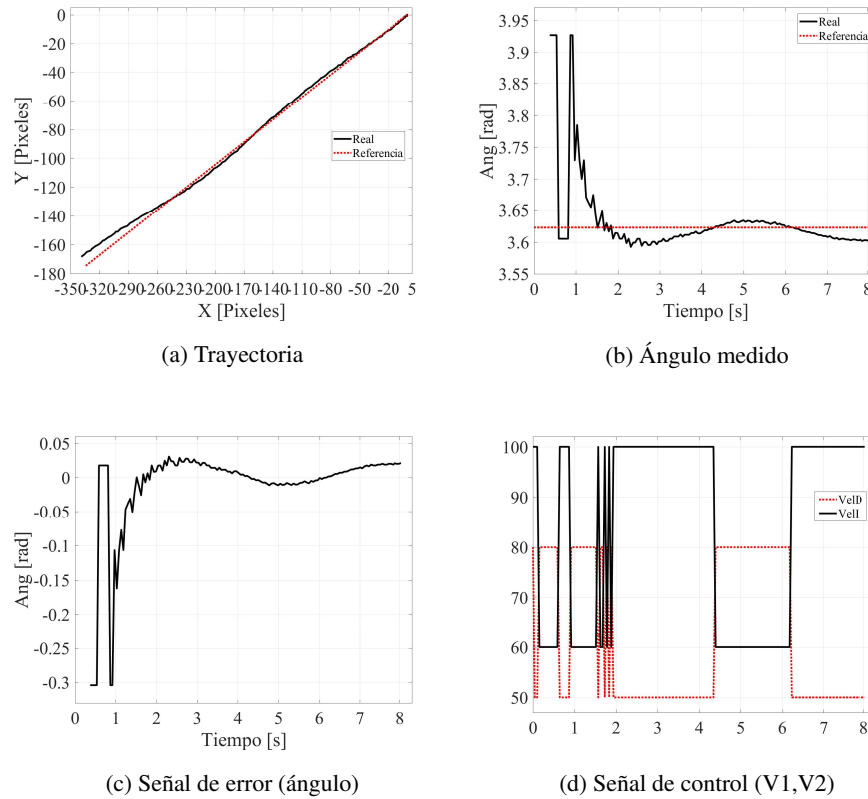


Figura 6.4: Resultados prueba de línea recta 4

En cuanto a la evaluación de las condiciones de desempeño del sistema, en el cuadro 6.1 se presentan los puntos finales alcanzados en cada prueba, los puntos objetivo definidos y el porcentaje de error en cada coordenada. Adicionalmente se presentan las condiciones de desempeño evaluadas, en las cuales se identifican porcentajes de sobrepico entre el 6,4% y 10,1% que cumplen el objetivo planteado para el sistema (<20%), de igual forma los errores de estado estable para cada una de las trayectorias se encuentran por debajo del 10%.

Por otro lado, en la implementación del controlador On/Off se encontró que el tiempo de establecimiento a pesar de no ser menor a un segundo, no afecta el desempeño del sistema, esto debido a la velocidad de ejecución del mismo, por lo tanto no es un parámetro crítico para el sistema.

Línea Recta									
Trayectoria	X Real	Y Real	X Obj	Y Obj	% Error X	% Error Y	% PO	Ts (s)	% Ess
1	393	-227	387	-221	1,550	2,715	6,439	2,900	3,788
2	300	215	293	218	2,389	1,376	8,598	1,440	3,095
3	-275	138	-275	137	0,000	0,730	8,050	4,830	0,929
4	-339	-168	-335	-175	1,194	4,000	10,197	2,473	6,908

Cuadro 6.1: Error y condiciones de desempeño pruebas de línea recta

6.2. Giro

Con el objetivo de evaluar la precisión de la estrategia de giro para el vehículo, se realizaron diferentes pruebas para cada uno de los giros implementados. Para cada giro se realizaron 20 pruebas en las cuales se

midió el ángulo final después de realizar el giro y se comparó el movimiento realizado con el movimiento esperado, los valores de ángulo de giro para cada una de las pruebas se presentan en el cuadro 6.2.

Valores Obtenidos		
Izquierda	Derecha	Media Vuelta
-1,801	1,438	2,912
-1,806	1,346	2,894
-1,718	1,448	2,953
-1,731	1,352	2,948
-1,833	1,403	2,892
-1,900	1,232	2,920
-1,624	1,482	2,965
-2,066	1,368	2,921
-1,809	1,396	2,983
-1,894	1,432	2,843
-1,746	1,407	2,940
-1,866	1,397	2,849
-1,800	1,429	2,879
-1,877	1,461	2,917
-1,654	1,376	2,972
-1,869	1,344	2,830
-1,826	1,414	2,879
-1,869	1,397	2,855
-1,684	1,426	2,902
-1,565	1,414	2,892
Mediana		
-1,808	1,405	2,907
Varianza		
0,012	0,003	0,002
Desviación estándar		
0,111	0,053	0,043

Cuadro 6.2: Valores obtenidos pruebas de giro

A partir de estos valores obtenidos se encontró el error de cada uno de los movimientos, en el caso del giro a la izquierda el ángulo objetivo era $-\frac{\pi}{2}$, para el giro a la derecha $\frac{\pi}{2}$ y para el giro de media vuelta π , se definió el porcentaje de error para cada uno y finalmente se encontró la mediana de los errores porcentuales, los resultados se presentan en el cuadro 6.3.

Se identificó que para el giro a la izquierda se presenta un mayor porcentaje de error en la mediana de las mediciones, esto debido a la estrategia del giro implementada, ya que es necesario un mayor recorrido de giro del vehículo para ejecutar este movimiento.

Giro					
Error Radianes			Error %		
Error Izq	Error Der	Error Media V	Error Izq	Error Der	Error Media V
0,230	0,133	0,229	14,640	8,453	7,298
0,236	0,224	0,247	14,995	14,284	7,874
0,147	0,123	0,188	9,363	7,821	5,991
0,160	0,219	0,193	10,189	13,921	6,159
0,262	0,167	0,250	16,663	10,659	7,958
0,329	0,338	0,222	20,966	21,540	7,060
0,053	0,089	0,177	3,377	5,637	5,629
0,495	0,202	0,221	31,534	12,882	7,035
0,238	0,174	0,158	15,177	11,102	5,037
0,323	0,139	0,299	20,593	8,820	9,516
0,176	0,163	0,201	11,183	10,402	6,408
0,295	0,173	0,293	18,801	11,035	9,322
0,229	0,142	0,262	14,568	9,044	8,344
0,306	0,110	0,225	19,508	6,981	7,160
0,083	0,194	0,169	5,293	12,375	5,383
0,298	0,227	0,312	19,003	14,450	9,934
0,255	0,157	0,263	16,218	9,982	8,368
0,298	0,174	0,287	19,003	11,068	9,126
0,113	0,145	0,240	7,176	9,199	7,642
0,006	0,157	0,250	0,368	10,004	7,946
Mediana					
0,237	0,165	0,235	15,086	10,530	7,470
Varianza					
0,012	0,003	0,002	48,965	11,179	1,892
Desviación estándar					
0,110	0,053	0,043	6,997	3,343	1,375

Cuadro 6.3: Error en giro

En los resultados obtenidos en estas pruebas, se puede identificar uno de los problemas en el momento de realizar la implementación conjunta, los errores de giro del sistema tienen valores de mediana entre el 7,4% y 15% ocasionando giros con errores de hasta 0.23 radianes (13,17°).

Adicionalmente el giro hacia la izquierda presenta valores de varianza y desviación estándar altas respecto a los otros giros, generando una orientación inadecuada del sistema en el momento de ejecutar movimientos de línea recta posteriores a este tipo de giro, el giro que presenta menor error en el momento de su ejecución es el de media vuelta, sin embargo este giro se implementa en reducidas ocasiones dependiendo de la ruta designada por el algoritmo.

6.3. Rutas con un vehículo

Con el fin de evaluar el comportamiento del controlador de seguimiento de trayectoria en diferentes recorridos, se plantearon cuatro rutas distintas y se realizó el recorrido dos veces para cada ruta, las rutas evaluadas se presentan en la figura 6.5. Los cuadros con las paradas obligatorias, la ruta óptima generada por el algoritmo y las listas de instrucciones para cada ruta se presentan en el Anexo 2.

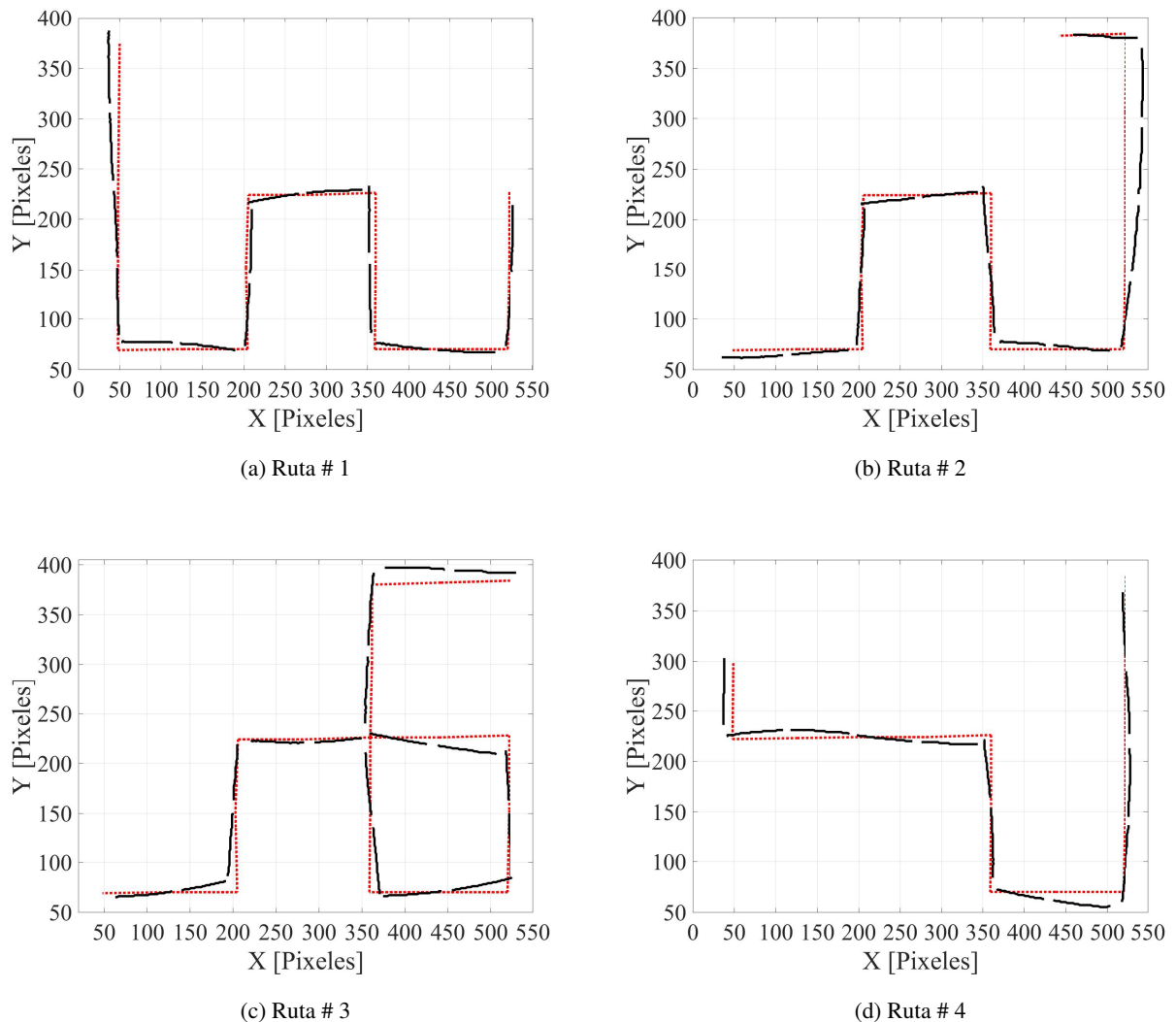


Figura 6.5: Rutas evaluadas con un vehículo

Con los datos tomados en estas ocho pruebas se realizó el mismo análisis presentado en la sección 5.1, a partir de los valores de error finales en cada uno de los tramos se definió un error de distancia con el fin de evaluar el error por tramo y el error acumulado relativo. Los resultados de este análisis de datos se presentan en el cuadro 6.4

En los resultados obtenidos en estas pruebas se evidencia nuevamente un error acumulado alrededor del 25% con varianza y desviación estándar alrededor del 1% para las ocho pruebas. El error de distancia presenta un comportamiento similar pero con valores de varianza y desviación estándar más altos debido a la dependencia directa de este error con la distancia objetivo de cada tramo y cada ruta. Los cuadros completos de error para cada una de las rutas se presentan en el Anexo 3.

	Error Distancia	Error acumulado relativo %	
Mediana	15,532	25,804	Ruta 1.1
Varianza	8,167	1,013	
Desviación Estándar	2,858	1,007	
Mediana	15,083	24,040	Ruta 1.2
Varianza	2,179	1,412	
Desviación Estándar	1,476	1,188	
Mediana	17,263	24,984	Ruta 2.1
Varianza	4,122	0,256	
Desviación Estándar	2,030	0,506	
Mediana	17,029	23,800	Ruta 2.2
Varianza	13,749	1,214	
Desviación Estándar	3,708	1,102	
Mediana	17,491	28,122	Ruta 3.1
Varianza	33,811	1,768	
Desviación Estándar	5,815	1,330	
Mediana	16,217	25,906	Ruta 3.2
Varianza	12,006	1,221	
Desviación Estándar	3,465	1,105	
Mediana	17,117	28,184	Ruta 4.1
Varianza	14,295	1,256	
Desviación Estándar	3,781	1,121	
Mediana	17,263	25,478	Ruta 4.2
Varianza	4,599	0,413	
Desviación Estándar	2,145	0,642	

Cuadro 6.4: Datos estadísticos rutas individuales

6.4. Rutas con dos vehículos

De igual forma con el fin de evaluar el comportamiento del controlador con los dos vehículos simultáneamente, se realizó una ruta adicional en dos ocasiones, la ruta deseada se presenta en el cuadro 6.5 y las trayectorias real y deseada se presentan en la figura 6.6

Ruta conjunta 2			
Ruta Azul		Ruta Roja	
Filas(Y)	Columnas(X)	Filas(Y)	Columnas(X)
3	1	5	1
3	3	5	3
1	3	5	5
1	5	3	5
1	7	4	7
2	7	5	6

Cuadro 6.5: Puntos de inicio y paraderos deseados ruta conjunta 2

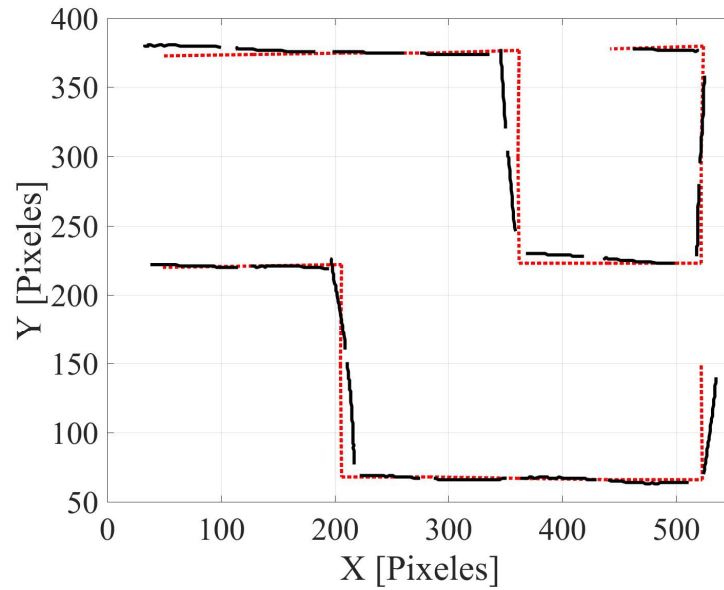


Figura 6.6: Trayectoria obtenida de la ruta conjunta # 2

Con los datos obtenidos en esta prueba se realizó el mismo análisis estadístico presentado en secciones previas con el fin de evaluar el desempeño del controlador, los resultados se presentan en el cuadro 6.6

Medida	Error Distancia	Error acumulado relativo %	Ruta
Mediana	23,000	38,985	Roja 1.2
Varianza	4,732	1,624	
Desviación Estándar	2,175	1,274	
Mediana	11,402	16,910	Azul 1.2
Varianza	14,760	36,246	
Desviación Estándar	3,842	6,020	
Mediana	25,080	39,958	Roja 2.2
Varianza	6,327	0,736	
Desviación Estándar	2,515	0,858	
Mediana	11,443	17,653	Azul 2.2
Varianza	20,134	27,298	
Desviación Estándar	4,487	5,225	

Cuadro 6.6: Datos estadísticos ruta conjunta

A partir de los datos obtenidos en esta prueba se puede concluir que el vehículo que ejecuta la ruta roja mantiene un error acumulado alto para cada ruta recorrida pero con valores constantes, hecho que se ve reflejado en los valores de varianza y desviación estándar, en cambio el vehículo de la ruta azul presenta menor valor de error acumulado en la ejecución de cada ruta pero con una mayor dispersión de los datos.

Capítulo 7

Conclusiones y recomendaciones

Se diseñó y ejecutó completamente el sistema de control de seguimiento de trayectoria para los dos vehículos eléctricos a escala dentro de un entorno controlado determinado. Esto gracias al desarrollo de cada una de las partes definidas para el proyecto y su implementación.

Así mismo, se logró implementar el algoritmo de enrutamiento A* para determinar la ruta optima de un punto a otro pasando por la cantidad de paraderos deseados para cada vehículo, así como la estrategia de segmentación por color de detección de los vehículos para lograr una realimentación efectiva del lazo de control a través de la cámara.

A partir de esto se diseñaron dos tipos de controladores clásicos uno en simulación en el cual se cumplieron los parámetros de desempeño deseados y otro finalmente implementado en el sistema, el cual no logró cumplir con el tiempo de establecimiento propuesto al inicio de este trabajo de grado. Sin embargo dicho parámetro no fue crítico para el sistema.

Por otro lado después de analizar los datos obtenidos de las diferentes pruebas realizadas se pueden concluir dos factores críticos respecto al control de seguimiento de trayectoria, por un lado la técnica de detección implementada presenta variaciones dependiendo del nivel de iluminación del entorno. Por otra parte, la posición inicial de los vehículos en el momento de ejecutar la ruta afecta el desempeño del control ya que no siempre inicia en la misma posición ni alineado con los paraderos de referencia.

Finalmente, este trabajo de grado cuenta con un sistema básico de prevención de colisiones en intersección, sin embargo para trabajos futuros es posible diseñar un sistema de prevención de colisiones que no tenga que detener el sistema general sino que sea capaz de calcular una nueva ruta con ayuda del algoritmo de enrutamiento para alguno de los dos vehículos antes de la colisión. Adicionalmente se propone incluir mediciones de consumo de batería para evaluar la eficiencia energética de las rutas entregadas por el algoritmo, implementar otro tipo de controlador que permita obtener mejores parámetros de desempeño y finalmente, obtener mejores resultados en la implementación con unos vehículos mejor diseñados físicamente que se ajusten a las necesidades del movimiento.

Bibliografía

- [1] (2009, Abril) Estudio prospectivo del sistema integrado de transporte de bogota y la region. Secretaria Distrital de Planeacion.
- [2] (2013) Capitulo para la caracterizacion de la demanda de transporte del sitp con la inclusion de nuevos proyectos de infraestructura. documento tecnico. Alcaldia Mayor de Bogota DC Transmilenio SA. Subgerencia tecnica y de servicios.
- [3] (2013, Agosto) 200 buses hibridos volvo llegaran a la flota de transmilenio. Portafolio. [Online]. Available: <http://www.portafolio.co/negocios/empresas/200-buses-hibridos-volvo-llegaran-flota-transmilenio-83150>
- [4] Z. Dong and M. Li, "A routing method of ad hoc networks based on a-star algorithm," in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, April 2009, pp. 623–626.
- [5] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.
- [6] A. Patel. (2016, Junio) Heuristics. Red Blob Games. [Online]. Available: <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>
- [7] Logitech. (2017) Especificaciones camara web logitech c920. [Online]. Available: <http://www.logitech.com/es-roam/product/hd-pro-webcam-c920>
- [8] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [9] (2017) Smart robot car kit user manual. Elegoo industries. [Online]. Available: <http://www.elegoo.com/download/>
- [10] M. Sauter, *Bluetooth*. John Wiley & Sons, Ltd, 2011, pp. 365–407. [Online]. Available: <http://dx.doi.org/10.1002/9780470978238.ch7>
- [11] (2017) Instrument control toolbox overview. Mathworks. [Online]. Available: <https://www.mathworks.com/products/instrument.html>
- [12] J. R. J. Reyes, "Vehiculos electricos a escala controlados y simulados dentro de un entorno controlado," Master's thesis, Pontificia Universidad Javeriana Bogota, Abril - Noviembre 2013.
- [13] (2017) Pid library documentation. Arduino. [Online]. Available: <https://playground.arduino.cc/Code/PIDLibrary>

Anexo 1

Listados de código y soporte audiovisual

Los códigos implementados en Matlab y en Arduino, junto con los vídeos demostrando el funcionamiento del sistema pueden ser accedidos mediante el siguiente URL

https://www.dropbox.com/sh/w0mt1kw8nox3l0/AADyWWSyH6o39vE2xIUSV_kAa?dl=0

Anexo 2

- Ruta # 1:

Filas(Y)	Columnas(X)
5	1
1	1
3	4
2	5
1	6
3	7

Cuadro 1: Punto de inicio y paraderos deseados

Filas(Y)	Columnas(X)	Movimiento	X	Y
5	1	1	49	298
4	1	1	48	222
3	1	1	48	148
2	1	1	48	69
1	1	2	0	0
1	2	1	127	70
1	3	1	205	70
2	3	2	0	0
3	3	1	203	152
3	4	1	206	224
3	5	3	0	0
2	5	1	282	224
1	5	1	360	226
1	6	3	0	0
1	7	1	360	152
2	7	1	359	70
3	7	2	0	0
		1	438	70
		1	520	70
		2	0	0
		1	522	153
		1	522	228

(a) Ruta generada por el algoritmo

(b) Lista de instrucciones

Cuadro 2: Ruta y lista de instrucciones ruta #1

■ Ruta # 2:

Filas(Y)	Columnas(X)
1	1
1	3
3	3
1	5
1	7
5	6

Cuadro 3: Punto de inicio y paraderos deseados

Filas(Y)	Columnas(X)
1	1
1	2
1	3
2	3
3	3
3	4
3	5
2	5
1	5
1	6
1	7
2	7
3	7
4	7
5	7
5	6

(a) Ruta generada por el algoritmo

Movimiento	X	Y
1	127	70
1	205	70
2	0	0
1	203	152
1	206	224
3	0	0
1	282	224
1	360	226
3	0	0
1	360	152
1	359	70
2	0	0
1	438	70
1	520	70
2	0	0
1	522	153
1	522	228
1	522	306
1	523	384
2	0	0
1	442	382

(b) Lista de instrucciones

Cuadro 4: Ruta y lista de instrucciones ruta #2

■ Ruta # 3:

Filas(Y)	Columnas(X)
5	7
1	7
1	5
4	5
3	3
1	1

Cuadro 5: Punto de inicio y paraderos deseados

Filas(Y)	Columnas(X)
5	7
5	6
5	5
4	5
3	5
3	6
3	7
2	7
1	7
1	6
1	5
2	5
3	5
3	4
3	3
2	3
1	3
1	2
1	1

(a) Ruta generada por el algoritmo

Movimiento	X	Y
1	442	382
1	362	380
2	0	0
1	362	302
1	360	226
2	0	0
1	440	226
1	522	228
3	0	0
1	522	153
1	520	70
3	0	0
1	438	70
1	359	70
3	0	0
1	360	152
1	360	226
2	0	0
1	282	224
1	206	224
2	0	0
1	203	152
1	205	70
3	0	0
1	127	70
1	48	69

(b) Lista de instrucciones

Cuadro 6: Ruta y lista de instrucciones ruta #3

- Ruta # 4:

Filas(Y)	Columnas(X)
4	1
3	2
3	4
1	6
2	7
5	7

Cuadro 7: Punto de inicio y paraderos deseados

Filas(Y)	Columnas(X)	Movimiento	X	Y
4	1	1	48	222
3	1	2	0	0
3	2	1	126	223
3	3	1	206	224
3	4	1	282	224
3	5	1	360	226
2	5	3	0	0
1	5	1	360	152
1	6	1	359	70
1	7	2	0	0
2	7	1	438	70
3	7	1	520	70
4	7	2	0	0
5	7	1	522	153
		1	522	228
		1	522	306
		1	523	384

(a) Ruta generada por el algoritmo

(b) Lista de instrucciones

Cuadro 8: Ruta y lista de instrucciones ruta #4

Anexo 3

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
72,028	72,028	11	15	18,601	18,601	25,825
66,008	138,035	9	15	36,094	17,493	26,148
64,281	202,316	3	14	50,412	14,318	24,917
65,490	267,806	7	17	68,797	18,385	25,689
54,452	322,258	13	14	87,902	19,105	27,277
67,067	389,325	15	18	111,332	23,431	28,596
47,043	436,368	0	11	122,332	11,000	28,034
60,133	496,501	3	14	136,650	14,318	27,523
67,476	563,977	13	0	149,650	13,000	26,535
68,029	632,006	15	0	164,650	15,000	26,052
65,276	697,283	3	14	178,968	14,318	25,666
68,264	765,547	6	17	196,996	18,028	25,733
54,037	819,584	13	6	211,313	14,318	25,783
69,260	888,844	16	1	227,345	16,031	25,578
63,071	951,916	1	15	242,378	15,033	25,462
63,388	1015,303	7	15	258,931	16,553	25,503
Mediana					15,532	25,804
Varianza					8,167	1,013
Desviación Estándar					2,858	1,007

Cuadro 9: Errores de desplazamiento y mediciones estadísticas Ruta 1.1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
72,007	72,007	11	17	20,248	20,248	28,120
68,118	140,124	6	16	37,336	17,088	26,645
65,123	205,247	1	15	52,370	15,033	25,515
68,029	273,277	1	15	67,403	15,033	24,665
62,008	335,285	13	6	81,721	14,318	24,374
67,365	402,650	15	1	96,754	15,033	24,029
64,195	466,845	4	15	112,278	15,524	24,050
63,071	529,916	4	14	126,839	14,560	23,936
62,650	592,566	15	1	141,872	15,033	23,942
68,066	660,632	16	3	158,151	16,279	23,939
66,000	726,632	8	15	175,151	17,000	24,104
70,007	796,639	5	16	191,914	16,763	24,090
64,281	860,920	14	0	205,914	14,000	23,918
70,029	930,948	15	3	221,211	15,297	23,762
65,276	996,225	2	15	236,344	15,133	23,724
64,031	1060,256	5	14	251,210	14,866	23,693
Mediana					15,083	24,040
Varianza					2,179	1,412
Desviación Estándar					1,476	1,188

Cuadro 10: Errores de desplazamiento y mediciones estadísticas Ruta 1.2

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
71,063	71,063	17	3	17,263	17,263	24,292
69,007	140,071	17	2	34,380	17,117	24,545
60,299	200,370	0	15	49,380	15,000	24,644
62,201	262,571	2	15	64,513	15,133	24,570
65,513	328,084	16	4	81,005	16,492	24,690
68,593	396,677	16	12	101,005	20,000	25,463
76,322	472,999	1	18	119,033	18,028	25,166
74,000	546,999	1	16	135,064	16,031	24,692
59,841	606,841	13	3	148,406	13,342	24,455
67,365	674,205	17	11	168,654	20,248	25,015
72,993	747,198	6	17	186,682	18,028	24,984
65,192	812,390	13	15	206,531	19,849	25,423
67,030	879,420	15	14	227,050	20,518	25,818
68,066	947,486	11	14	244,854	17,804	25,842
70,007	1017,494	17	0	261,854	17,000	25,735
Mediana					17,263	24,984
Varianza					4,122	0,256
Desviación Estándar					2,030	0,506

Cuadro 11: Errores de desplazamiento y mediciones estadísticas Ruta 2.1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado %
74,027	74,027	18	6	18,974	18,974	25,631
69,116	143,143	17	1	36,003	17,029	25,152
63,127	206,270	1	15	51,036	15,033	24,743
66,121	272,391	1	13	64,075	13,038	23,523
65,276	337,667	14	3	78,393	14,318	23,216
67,186	404,853	16	1	94,424	16,031	23,323
66,272	471,126	4	15	109,948	15,524	23,337
71,253	542,379	5	16	126,711	16,763	23,362
56,080	598,459	12	4	139,360	12,649	23,286
68,264	666,723	17	2	156,477	17,117	23,470
64,133	730,856	7	16	173,942	17,464	23,800
67,476	798,332	17	14	195,964	22,023	24,547
66,121	864,453	21	15	221,771	25,807	25,655
69,000	933,453	20	14	246,184	24,413	26,374
79,057	1012,510	17	1	263,214	17,029	25,996
Mediana					17,029	23,800
Varianza					13,749	1,214
Desviación Estándar					3,708	1,102

Cuadro 12: Errores de desplazamiento y mediciones estadísticas Ruta 2.2

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado %
72,062	72,062	17	5	17,720	17,720	24,590
70,349	142,412	16	15	39,652	21,932	27,843
72,180	214,592	10	19	61,123	21,471	28,483
67,365	281,957	17	17	85,164	24,042	30,205
76,792	358,749	17	6	103,192	18,028	28,764
68,593	427,342	16	19	128,032	24,839	29,960
49,649	476,990	2	10	138,230	10,198	28,980
68,029	545,020	7	15	154,783	16,553	28,399
72,835	617,855	17	3	172,045	17,263	27,846
68,469	686,324	15	7	188,598	16,553	27,479
71,701	758,025	1	17	205,628	17,029	27,127
66,000	824,025	2	15	220,760	15,133	26,790
58,215	882,240	14	7	236,413	15,652	26,797
61,074	943,314	15	12	255,622	19,209	27,098
65,000	1008,314	14	16	276,882	21,260	27,460
68,659	1076,972	34	16	314,459	37,577	29,198
37,054	1114,026	5	11	326,542	12,083	29,312
58,138	1172,164	14	7	342,195	15,652	29,193
Mediana					17,491	28,122
Varianza					33,811	1,768
Desviación Estándar					5,815	1,330

Cuadro 13: Errores de desplazamiento y mediciones estadísticas Ruta 3.1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
73,027	73,027	16	12	20,000	20,000	27,387
70,029	143,056	15	17	42,672	22,672	29,829
73,246	216,302	4	17	60,136	17,464	27,802
68,066	284,368	6	14	75,367	15,232	26,503
67,897	352,265	14	7	91,020	15,652	25,838
68,469	420,734	15	19	115,227	24,207	27,387
50,090	470,824	1	10	125,277	10,050	26,608
64,031	534,855	4	16	141,770	16,492	26,506
72,180	607,036	17	2	158,887	17,117	26,174
68,184	675,219	16	4	175,379	16,492	25,974
68,469	743,688	3	16	191,658	16,279	25,771
63,388	807,076	6	15	207,814	16,155	25,749
54,231	861,307	13	3	221,155	13,342	25,677
62,032	923,339	15	1	236,189	15,033	25,580
56,143	979,482	2	11	247,369	11,180	25,255
64,195	1043,677	11	16	266,785	19,416	25,562
52,773	1096,450	13	2	279,938	13,153	25,531
65,192	1161,642	15	4	295,462	15,524	25,435
Mediana					16,217	25,906
Varianza					12,006	1,221
Desviación Estándar					3,465	1,105

Cuadro 14: Errores de desplazamiento y mediciones estadísticas Ruta 3.2

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
70,029	70,029	14	16	21,260	21,260	30,359
72,173	142,202	17	1	38,290	17,029	26,926
71,253	213,455	15	9	55,783	17,493	26,133
65,123	278,578	15	15	76,996	21,213	27,639
67,007	345,585	14	19	100,597	23,601	29,109
46,325	391,910	4	12	113,246	12,649	28,896
67,602	459,512	16	16	135,873	22,627	29,569
48,166	507,679	10	2	146,071	10,198	28,772
65,620	573,299	16	8	163,960	17,889	28,599
69,181	642,480	2	17	181,077	17,117	28,184
64,031	706,511	5	15	196,888	15,811	27,868
64,008	770,519	4	14	211,449	14,560	27,442
64,000	834,519	2	15	226,581	15,133	27,151
Mediana					17,117	28,184
Varianza					14,295	1,256
Desviación Estándar					3,781	1,121

Cuadro 15: Errores de desplazamiento y mediciones estadísticas Ruta 4.1

Distancia Obj	Distancia Acu	Error X	Error Y	Error Dist Acu	Error Dist	Error acumulado%
67,007	67,007	11	14	17,804	17,804	26,571
68,264	135,272	17	8	36,593	18,788	27,051
71,113	206,384	17	3	53,855	17,263	26,095
67,268	273,652	17	4	71,320	17,464	26,062
68,029	341,682	16	9	89,677	18,358	26,246
57,559	399,240	1	12	101,719	12,042	25,478
70,029	469,269	3	14	116,037	14,318	24,727
57,871	527,140	14	7	131,689	15,652	24,982
69,260	596,400	16	14	152,949	21,260	25,645
76,531	672,931	5	17	170,670	17,720	25,362
67,007	739,938	6	15	186,825	16,155	25,249
63,127	803,065	2	16	202,950	16,125	25,272
65,123	868,188	4	16	219,442	16,492	25,276
Mediana					17,263	25,478
Varianza					4,599	0,413
Desviación Estándar					2,145	0,642

Cuadro 16: Errores de desplazamiento y mediciones estadísticas Ruta 4.2