

Vehículo autónomo de configuración Ackermann para seguimiento de carril mediante procesamiento digital de imagen

Julio Alfonso Rodriguez Olea., Ilse Dayana Leon Bon, Ulises Orozco-Rosas*, Kenia Picos

CETYS Universidad, Centro de Innovación y Diseño (CEID), Av. CETYS Universidad No. 4. El Lago, C.P. 22210, Tijuana B.C., México
[julio.rodriguez, ilse.leon]@cetys.edu.mx, [ulises.orozco, kenia.picos]@cetys.mx

Resumen

Este artículo presenta una propuesta de algoritmo para la identificación y seguimiento autónomo de carril de un vehículo de prueba a través del procesamiento de secuencias de imágenes captadas por una videocámara durante su navegación. El robot de configuración Ackerman modifica su ángulo de giro frontal de acuerdo con los datos obtenidos por la cámara, en los cuales se aplican filtros de tono, saturación y matiz (HSV, por sus siglas en inglés) y de contorno para definir los límites de la pista y su centro. Además, se aplica un controlador Proporcional-Derivativo (PD) por medio de código de programación para procurar la incorporación del vehículo al carril en todo punto del camino; utilizando la función de control para calcular la respuesta al error en un instante determinado y modificar de manera correspondiente la orientación actual. Se considera a su vez que el error es el valor de la desviación con respecto al centro del carril. Las pruebas de funcionamiento del prototipo implementado se llevan a cabo en un ambiente parcialmente controlado, donde se ofrece garantía de la ausencia de obstáculos en la pista, pero el sistema es sometido a diferentes condiciones de iluminación, el cual es un reto importante para la detección del camino. Los resultados experimentales muestran el comportamiento del vehículo ante los casos de seguimiento de líneas rectas, curvas y de circuito cerrado.

Palabras clave— Vehículo autónomo, Controlador PD, filtro HSV, filtro de contornos, videocámara.

Abstract

This paper presents an algorithm proposal for the autonomous identification and lane tracking of a test vehicle through the processing of image sequences captured by a video camera during its navigation. The Ackermann configuration robot modifies its front rotation angle according to the data obtained by the camera, in which Hue-Saturation-Value (HSV) and Contours filters are applied to define the borders of the track and its center. Besides, the prototype applies a Proportional-Derivative (PD) controller employing program code to ensure the incorporation of the vehicle to the lane at all points of the path; using the control function to calculate the response to the error at a given time and modify the current orientation correspondingly. At the same time, the error is considered as the value of the deviation concerning the center of the lane. The functionality tests of the implemented prototype are made in a partially controlled environment, where a guarantee of the absence of obstacles on the track is offered but the system is subjected to different lighting conditions, which is an important challenge for lane detection. The experimental results show the behavior of the vehicle in the case of monitoring straight, curved and closed-circuit lines.

Keywords— Autonomous vehicle, PD controller, HSV filter, contours filter, video camera.

1. INTRODUCCIÓN

El área de vehículos autónomos ha crecido significativamente en los últimos años, donde el principal problema a enfrentar en vehículos terrestres de configuración Ackerman es la detección de carril. Esto se debe a que los vehículos convencionales suelen desplazarse en carreteras que contienen marcadores de carril como señalamiento base. Como se menciona en [1] y [2], en este tipo de ambiente las técnicas de detección y seguimiento del indicador de carril constituyen el enfoque central que debe abordarse.

Uno de los propósitos principales de esta área de desarrollo es reducir los accidentes de tránsito y por lo tanto aumentar la seguridad de los usuarios [1]. De hecho, dice [2] que los accidentes en carreteras son causados en un 45% por humanos en cambios de carril imprevistos y colisiones de frente.

Además de la reducción de accidentes, existen otras ventajas de este tipo de sistemas, como un potencial aumento de la capacidad de las carreteras a causa de un aceleramiento más constante de los vehículos y su habilidad de mantener distancias más cortas entre sí sin sacrificar la seguridad.

El seguimiento de un carril por medio de un algoritmo de control se ha implementado ya en el pasado desde diferentes acercamientos. Los pasos que se suelen seguir para ello son:

1. Extraer la información del camino a seguir.
2. Filtrar la información indeseada.
3. Realizar cálculos con la información para posicionar el vehículo en zona segura.

Existen muchas ideas con respecto a la detección de carril usando cámaras rectilíneas, por ejemplo, en [3] se propone utilizar una doble cámara estéreo rectilínea para el seguimiento de trayectoria y la evasión de obstáculos, utilizando transformaciones de la imagen para encontrar las

* Autor para la correspondencia (Corresponding author).

posiciones de los carriles. En [4], se utilizan cámaras rectilíneas para obtener los límites de un carril en una autopista e igualmente se utilizan transformaciones en la imagen para posicionar los bordes de los carriles, señalando que la intensidad de la luz puede provocar alteraciones en los resultados. En [5] se utilizan cámaras rectilíneas y cámaras de tipo vista de ave para detectar los carriles de una autopista. Se utilizan transformadas en la imagen y se calcula una trayectoria ideal para mantenerse en el centro del carril.

En [6] se menciona que a pesar de que un controlador PD tiene una estructura simple, es útil para aproximar labores como lo puede ser el manejo autónomo de vehículos. Al ver que los cálculos de seguimiento del carril han sido cubiertos por otros autores con éxito, se plantea observar los efectos que tendría utilizar un controlador PD junto con cámaras rectilíneas para desarrollar un prototipo que use esta información como guía en su seguimiento de trayectoria. Se siguen los pasos planteados anteriormente como referencia y se propone una combinación de algoritmos y técnicas ya estudiadas para filtrar los carriles.

2. FUNDAMENTOS

En este apartado se definen las características básicas de los filtros digitales que se utilizaron para el procesamiento de las imágenes captadas por la videocámara del vehículo, además de detallar de manera general en qué consiste un controlador Proporcional Derivativo (PD). Estos conceptos son clave para comprender cómo se llevó a cabo la implementación del prototipo y analizar los resultados obtenidos.

2.1 Filtro HSV

La técnica de filtración por HSV (*Hue, Saturation, Value*) permite binarizar una imagen siguiendo criterios de tono, saturación y matiz. De acuerdo con el estudio comparativo de [7], el sistema HSV permite alcanzar una mayor precisión para la identificación de objetos y superficies que otros sistemas. A su vez, [8] indica que el filtrado HSV hace posible obtener información con considerablemente menos ruido que otros filtros como el RGB (*Red, Green, Blue*), que se basa en superposición de color y provoca mayor distorsión. Estos factores inclinaron la balanza a favor del uso de filtrado HSV sobre otros modelos para el procesamiento de imagen del vehículo autónomo. Al establecer rangos en cada parámetro de este sistema es posible crear máscaras que aíslan regiones dentro de una gama de color predeterminada de la imagen. Entonces, la función de este filtro es identificar las zonas que contienen los colores de interés, en este caso, los colores correspondientes a los carriles a detectar.

2.2 Filtro de contornos

El filtro de contornos utilizado es una adaptación del algoritmo de detección de contornos de imágenes binarizadas propuesto por [9], el cual extrae información correspondiente a las relaciones de entorno entre dos tipos de bordes en una imagen binaria:

- Los bordes exteriores, que se definen como el conjunto de puntos externos entre una secuencia de píxeles en estado 1 y una secuencia de píxeles en estado 0 que lo rodea [9].
- Los bordes huecos, que se definen como el conjunto de coordenadas externas que se encuentran entre una serie de píxeles en estado 0 y una secuencia de píxeles en estado 1 que lo rodea [9].

De tal modo, el algoritmo de contornos de imágenes binarizadas aprovecha las propiedades de las imágenes binarias, donde se guarda una relación de uno a uno entre los bordes exteriores y los componentes conectados de píxeles en estado 0 (huecos); o bien, entre los bordes huecos y puntos conectados en estado 1 (no huecos), permitiendo por lo tanto identificar los contornos de la imagen cada vez que ocurre un cambio de valor en tales píxeles.

2.3 Controlador Proporcional Derivativo

El algoritmo de control implementado en el prototipo se conoce como Proporcional Derivativo (PD) y puede representarse idealmente de la siguiente manera [10]:

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} \quad (1)$$

En donde $u(t)$ es la acción de control y $e(t)$ es el error calculado, definiendo este último como la diferencia entre el valor objetivo y el valor actual de la variable controlada. De tal modo, la señal de control se compone de un término proporcional al error y uno proporcional a la derivada del error, con ganancias k_p y k_d respectivamente. Es importante agregar que la parte proporcional al error de la expresión actúa como un indicador de tal error en el instante actual. Asimismo, la parte derivativa puede traducirse como una predicción de los valores de error futuros basado en extrapolación lineal [10].

De manera particular, el dispositivo desarrollado hace uso de este tipo de controlador para calcular ángulos de giro frontal que permitan posicionar el vehículo dentro del carril en un instante de tiempo determinado. La variable de entrada o error es por tanto la diferencia entre el centro del carril y el centro de la cámara, y la variable controlada es el ángulo de giro.

3. SEGUIMIENTO DE CARRIL MEDIANTE PROCESAMIENTO DIGITAL DE IMAGEN

Esta sección del documento describe de manera detallada cómo se llevó a cabo la implementación del vehículo autónomo mediante una descripción de las fases de captura y procesamiento de información del sistema, y la posterior manipulación de los motores de dirección para el alcance del objetivo de seguimiento de carril. La Fig. 1 muestra un diagrama de flujo con el panorama general del algoritmo propuesto para el seguimiento del carril que fue utilizado en el vehículo.

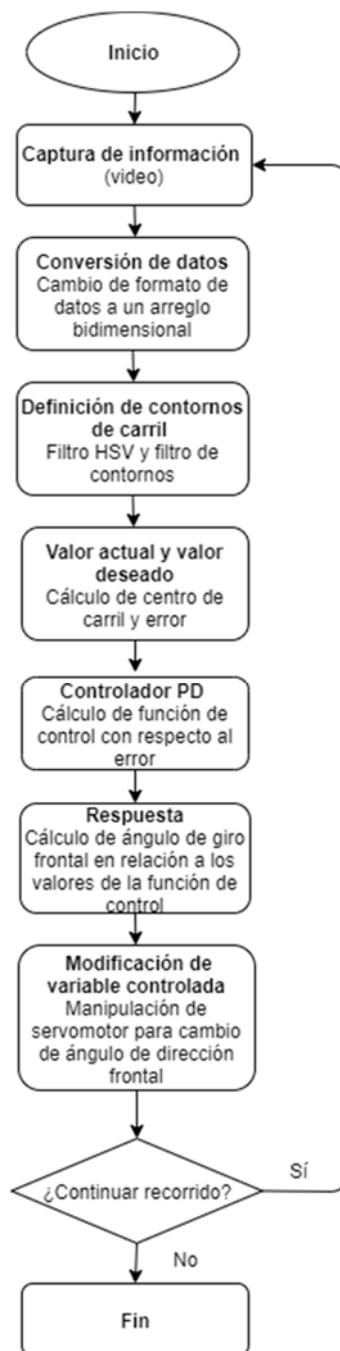


Fig. 1. Algoritmo propuesto para seguimiento de carril.

3.1 Procesamiento de imagen

El procesamiento de imagen se lleva a cabo mayoritariamente mediante el aprovechamiento de la librería de código libre *OpenCV*, la cual, entre muchas otras funciones, contiene métodos para la extracción y formateo de datos capturados por la videocámara, para filtrado digital de imagen y para el despliegue de vídeo en una pantalla de ordenador. Otro factor importante es que esta librería funciona de manera nativa en el sistema operativo *Linux*, lo cual es conveniente porque el prototipo opera con una *Raspberry Pi* que trabaja con el sistema *Raspbian*.

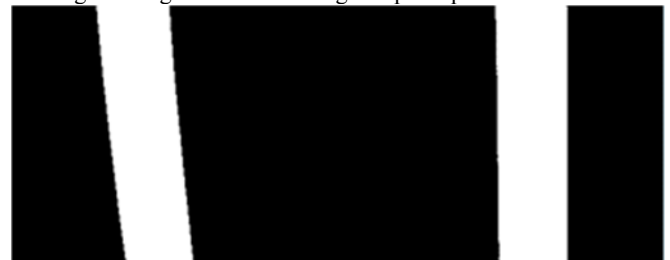
3.1.1 Implementación de filtro HSV

El filtrado tipo HSV es el primero que se implementa en las imágenes captadas; su propósito es aislar las regiones de color rojo, ya que se estableció que los carriles a seguir serán de este color en todos los casos de prueba. En específico, se aplican dos filtros HSV independientes a la imagen para focalizar áreas que tengan diferentes condiciones de iluminación, y por lo tanto diferentes máscaras para los valores de tono, saturación y matiz. Después, las matrices binarias de salida que corresponden a las zonas de color rojo identificadas por cada filtro se suman entre sí mediante una operación lógica tipo OR, abarcando de este modo distintas regiones del espectro y disminuyendo el impacto que pueda ocasionar el ruido inducido por el ambiente. El rango de valores de los parámetros de interés es de 0 a 180 grados para tono (*Hue*), de 0 a 255 para saturación (*Saturation*) y de 0 a 255 para matiz (*Value*). En la Tabla 1 se muestran las máscaras definidas para cada uno de los filtros HSV. Un ejemplo de cómo se ve la imagen después de pasar por este filtro se observa en la Fig. 2.

Tabla 1. Valores del filtro HSV.

Máscaras	<i>Hue</i>	<i>Saturation</i>	<i>Value</i>
Máscara 1	0-16	52-255	60-255
Máscara 2	161-180	23-255	0-255

Fig. 2. Imagen binarizada luego de pasar por el filtro HSV.



3.1.2 Implementación de filtro de contornos

Después de obtener la imagen binarizada de la Fig. 2, es posible aplicar el algoritmo descrito en [9], que como se mencionó anteriormente, utiliza los cambios de valor de la imagen binaria para identificar los contornos en el contenido de una imagen. Por lo tanto, el propósito de esta etapa de filtración es identificar los bordes de los carriles de la pista. En la Fig. 3, se puede apreciar un ejemplo de la salida de este filtro superpuesta con la imagen original.

Fig. 3. Carriles obtenidos de la salida del filtro de contornos, superpuestos sobre la imagen original.



3.1.3 Cálculo del centro

Considerando que el objetivo del vehículo es conservarse dentro de la pista de manera autónoma durante su recorrido, es necesario que se determine periódicamente el centro relativo del carril, de acuerdo con la información de localización de bordes que se obtiene en el filtro anterior. Además, se debe definir la coordenada del centro real de la imagen capturada, la cual puede expresarse mediante (2), donde y_{max} y x_{max} son las dimensiones máximas de la imagen. Por consiguiente, lo que se busca es que el centro relativo del carril se mantenga en la misma coordenada que el centro real en todo momento, siendo este último el valor deseado (*set-point*) del sistema.

$$\text{Centro real } (x, y) = \left(\frac{x_{max}}{2}, \frac{y_{max}}{2} \right) \quad (2)$$

De este modo, el error del sistema se presenta como la diferencia entre el centro real de la cámara y el centro relativo del carril, para el cual se debe calcular un ángulo de giro correspondiente que compense la desviación. El algoritmo para el cálculo del centro relativo consiste en tomar como referencia el centro real definido, y a partir de ese punto realizar un barrido tanto a la derecha como a la izquierda de la imagen, recorriendo todo el eje x . Este barrido se detiene cuando se encuentra un píxel azul, (como los que se muestran superpuestos en la Fig. 3) ya que estos indican el inicio de uno de los contornos. Al encontrar los límites interiores tanto en la izquierda como en la derecha del carril, es posible calcular el centro utilizando (3) donde $X_{derecha}$ y $X_{izquierda}$ representan las distancias obtenidas de los dos barridos con respecto al centro real.

$$\text{Centro relativo} = \frac{(X_{derecha} + X_{izquierda})}{2} \quad (3)$$

Como el error del sistema se calcula mediante una resta entre el centro real de la cámara y el centro relativo del carril, es posible inferir que el signo resultante de la operación determinará la orientación del giro, positivo siendo hacia la derecha y negativo hacia la izquierda. La Fig. 4 muestra la implementación de estos cálculos, donde el círculo verde representa el centro real y el círculo rojo, el centro relativo.

3.2 Controlador

El controlador PD se implementa a través de código con la

lógica planteada en el diagrama de bloques de la Fig. 5 y siguiendo el modelo expuesto en (1) de manera discretizada. Como se mencionó anteriormente, éste abarca los componentes proporcional y derivativo para el cálculo de la función de control, donde el error de entrada $e(t)$ se obtiene del valor deseado o centro real de la cámara $r(t)$ menos el centro del carril calculado con los datos de la imagen captada $h(t)$. La salida es por tanto la función de control correspondiente $c(t)$ que a su vez modifica el ángulo de giro frontal del vehículo $y(t)$.

Fig. 4. Cálculo del centro real (punto verde) y relativo (punto rojo).

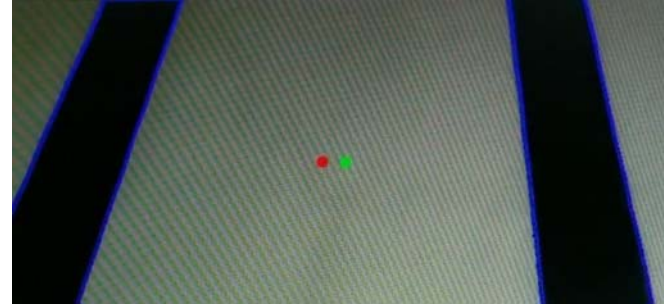
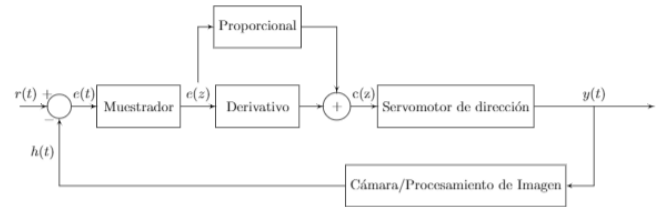


Fig. 5. Diagrama del controlador.



En la Tabla 2 se muestran las ganancias las cuales se obtuvieron a través de un procedimiento de sintonización heurístico, donde los coeficientes proporcional y derivativo se definieron inicialmente como cero y se incrementó el coeficiente proporcional hasta un valor que permita la orientación del dispositivo sin provocar oscilaciones constantes en la salida. Posteriormente, se incrementó el coeficiente derivativo de manera en que las pruebas de desempeño mostraron una mejora en la velocidad de respuesta del sistema sin afectar de forma significativa la precisión en el seguimiento de carril del vehículo.

Tabla 2. Ganancias del controlador PD.

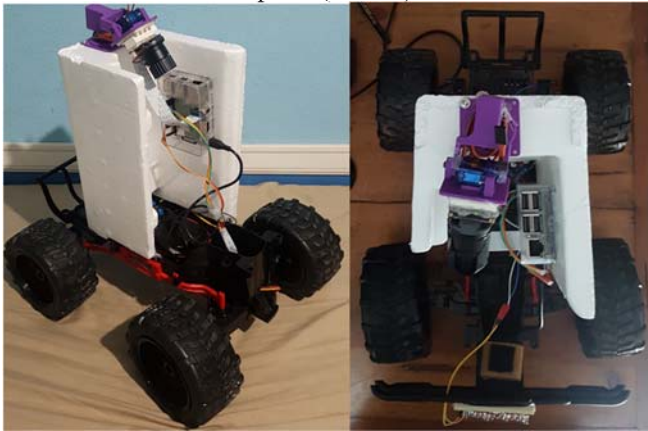
Ganancia Proporcional	Ganancia Derivativa
2	1

4. RESULTADOS

Esta sección del documento se describe de manera detallada los resultados obtenidos de un prototipo que utiliza los

algoritmos y controladores que se describen en el apartado anterior. La Fig. 6 muestra una imagen del prototipo construido para las pruebas, este cuenta con una cámara para procesar la imagen, un vehículo que controla su vuelta por medio de un servomotor, un puente H para controlar la velocidad, una tira de diodos LED para homologar la luz de la cámara y una *Raspberry Pi* que sirve para implementar los algoritmos y enlaces entre los periféricos.

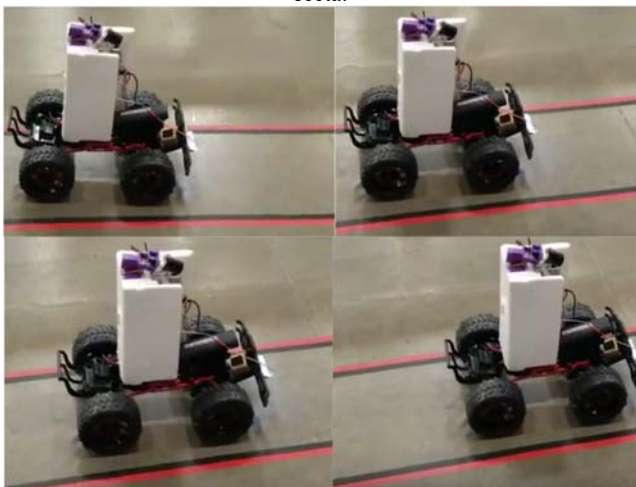
Fig. 6. Prototipo físico utilizado. Vista lateral (izquierda). Vista superior (derecha).



4.1 Seguimiento de carril en línea recta

Las primeras pruebas de seguimiento fueron en carriles rectos de aproximadamente 30 centímetros de ancho. Los resultados observados en estas pruebas permitieron realizar un primer acercamiento para la calibración del controlador, pues en una línea recta es más intuitivo el ajuste. En la Fig. 7, se pueden observar algunas imágenes donde el vehículo sigue la trayectoria recta.

Fig. 7. Resultados obtenidos en términos del desempeño en línea recta.

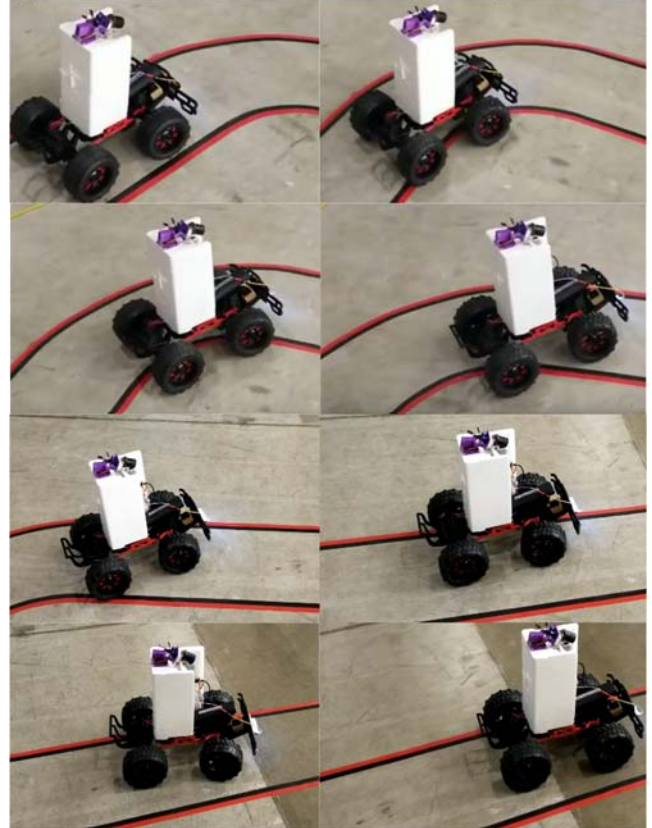


4.2 Seguimiento de carril en curva

Después de que el vehículo es calibrado en el seguimiento de carril con línea recta, se añade un segundo reto que es el

seguimiento de carril en curvas. Una serie de pruebas en curvas verificaron que este objetivo también era completado por el controlador. En la Fig. 8, es posible notar unas imágenes donde el vehículo efectúa la curva de manera correcta.

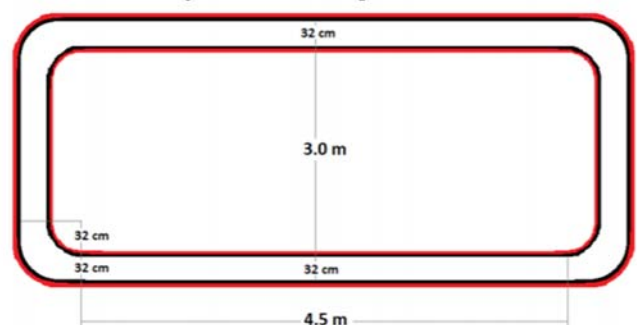
Fig. 8. Resultados obtenidos en términos del desempeño en curvas.



4.3 Seguimiento de carril en pista completa

Como último reto, se implementó una pista como la que se ve en la Fig. 9. Esta pista consta de un recorrido de cuatro curvas y cuatro segmentos rectos por los que el vehículo pasa y se verifica si el modelo está ausente de errores. Se realizaron dos pruebas cronometradas al recorrer una vuelta completa, donde los tiempos obtenidos fueron de 1:43:91 minutos y 1:39:84 minutos.

Fig. 9. Recorrido implementado.



5. CONCLUSIONES

En este trabajo se presentó una solución para el seguimiento de carril en líneas curvas, rectas y circuitos cerrados. Esta propuesta consistió en el uso de un controlador y un algoritmo de detección de carriles para orientar al vehículo a permanecer dentro del carril. A través de los resultados se muestra que los algoritmos planteados son eficaces en ambientes libres de obstáculos e iluminados. El algoritmo de detección predice con efectividad el error de posicionamiento del vehículo mientras exista un poco de iluminación ya sea natural o artificial. El controlador tiene un ajuste favorable para la trayectoria, mientras la detección no presente problemas de iluminación, el vehículo no produce vibraciones dentro del estado estable y se ajusta al carril con cambios pequeños. Después de cumplir con la solución propuesta, un posible trabajo futuro es implementar detección de carril en ambientes con obstáculos y el desarrollo de un control difuso como alternativa al controlador clásico.

6. BIBLIOGRAFÍA

- [1] A. Sotelo, J. Francisco, L. Magdalena, L. Bergasa, L. Boquete, "A Color Vision-Based Lane Tracking System for Autonomous Driving on Unmarked Roads". *Autonomous Robots*, 16(1), pp 95-116, 2004.
- [2] Z. Hongying, K. Onecue, W. Jong-Seob, K. Dong-Joong, "Lane Detection and Tracking based on Annealed Particle Filter". *International Journal of Control, Automation, and Systems*, 12(6), pp 1303-1312, 2014. DOI: 10.1007/s12555-013-0279-2
- [3] M. Bertozzi, A. Broggi, "a parallel real-time stereo vision system for generic obstacle and lane detection". *IEEE Transactions on Image Processing*, 7(1), pp. 62-81, 1998.
- [4] H. Choi, J. Park, W. Choi, S. Oh, "Vision-based fusion of robust lane tracking and forward vehicle detection in a real driving environment". *The Korean Society of Automotive Engineers*, 13(4), pp 653-669, 2012.
- [5] J. Ruyi, K. Reinhard, V. Tobi, W. Shigang, "Lane detection and tracking using a new lane model and distance transform". *Springer-Verlag*, 22(4), pp 721-737, 2011.
- [6] W. Farag, "Complex Trajectory Tracking Using PID Control for Autonomous Driving". *International Journal of Intelligent Transportation Systems Research*, 22(4), pp 1-11, 2019.
- [7] S. Soleimanizadeh, D. Mohamad, T. Saba, A. Rehman., "Recognition of Partially Occluded Objects Based on the Three Different Color Spaces (RGB, YCbCr, HSV)," *3D Research*, 6(22), 2015. DOI: 10.1007/s13319-015-0052-9
- [8] P. Mukund, S. Shankar, "Guided Filter for Grayscale, RGB and HSV color space". *International Journal of Engineering And Computer Science*, 5(12), pp 19381-19384, 2016.
- [9] S. Suzuki, J. K. Abe, "Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing", *Computer Vision, Graphics, and Image Processing*, 30(1), pp. 32-46, 1985. DOI: doi.org/10.1016/0734-189X(85)90016-7

- [10] K. Johan, R. Murray, "Feedback Systems". *Princeton University Press*: United States, 2009.