

Navegación autónoma para un vehículo sin conductor usando el simulador Webots

Luis David Torres Trejo

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Motivación

- ① Los accidentes viales en su mayoría son provocados por errores humanos.
- ② Los vehículos sin conductor tienen el potencial de disminuir accidentes viales y problemas de tránsito vehicular.
- ③ El desarrollo de vehículos autónomos requiere de instrumentación costosa.
- ④ Los simuladores son una buena alternativa en el desarrollo de sistemas para vehículos sin conductor.

Planteamiento del Problema

Crear un ambiente vial virtual para probar sistemas de:

- Visión artificial
- Control
- Navegación autónoma

Diseñar y desarrollar software para:

- Detección de carriles
- Detección de obstáculos
- Seguimiento de obstáculos
- Control del vehículo

Definir comportamientos para:

- Seguimiento de carriles
- Rebase
- Seguimiento de vehículos

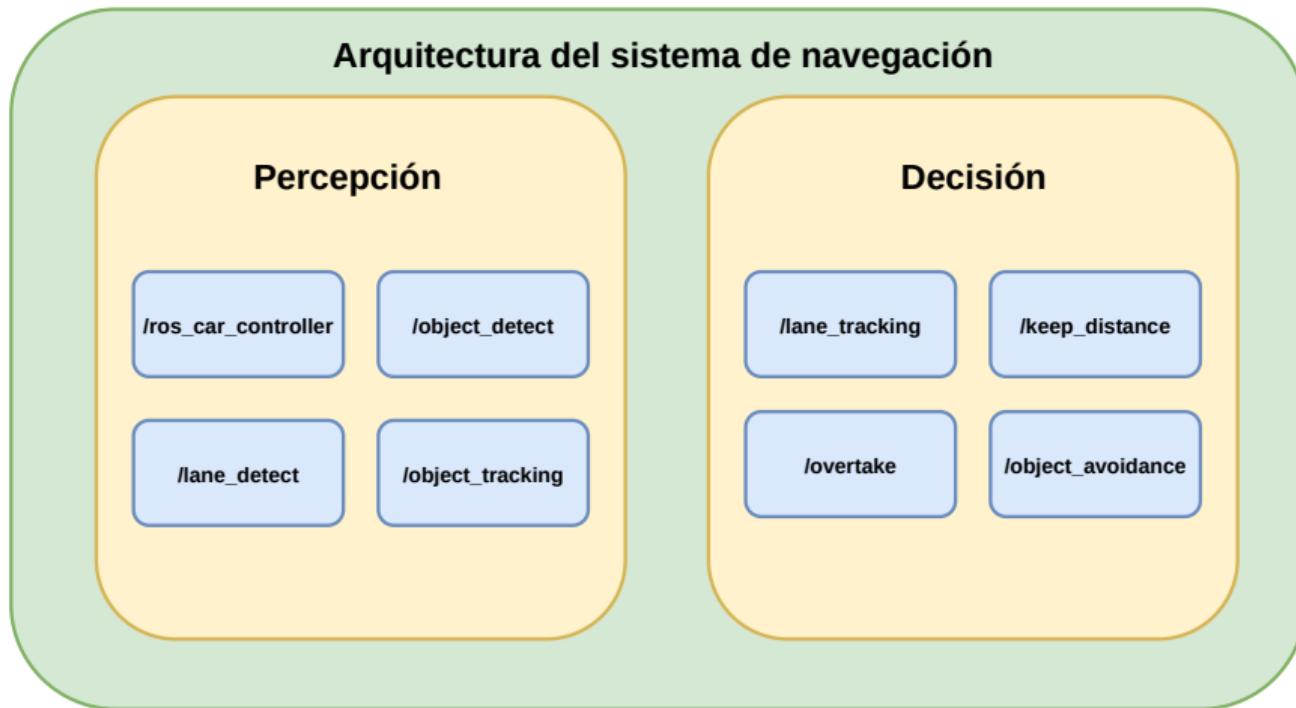
Hipótesis

- Los simuladores son una buena opción en el desarrollo de tecnologías para vehículos autónomos.
- La transformada **Hough** y el detector de bordes de **Canny** son lo suficientemente poderosos para lograr el reconocimiento de carriles en imágenes **RGB**.
- El simulador **Webots** cuenta con las características necesarias para desarrollar sistemas de visión artificial, control y navegación autónoma.
- Las máquinas de estados finitos son apropiadas para la implementación de comportamientos reactivos.
- La navegación autónoma en un ambiente controlado se puede conseguir a través de comportamientos para **seguimiento de carril**, **rebase** y **mantener distancia**.

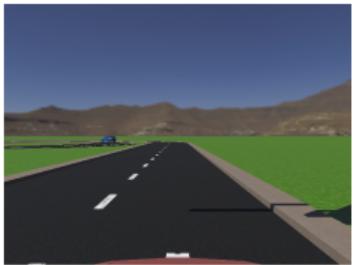
Objetivos

- Diseñar y modelar un ambiente vial urbano para realizar pruebas de navegación autónoma en el simulador **Webots**.
- Utilizar el detector de bordes de **Canny** y transformada **Hough** a fin de desarrollar un algoritmo para detección de carril a partir de imágenes **RGB**.
- Diseñar y desarrollar un sistema de control para el vehículo autónomo.
- Establecer e implementar comportamientos para **seguimiento de carriles, rebase de obstáculos y mantener distancia**.
- Integrar todos los módulos empleando la plataforma **ROS** y el simulador **Webots**.
- Realizar pruebas de **navegación sin obstáculos** y pruebas con **obstáculos** tanto estáticos como en movimiento.

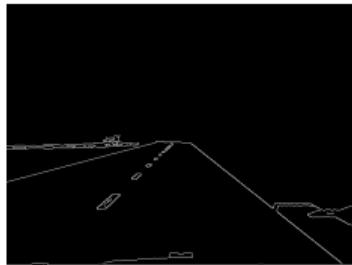
Arquitectura del sistema de navegación del vehículo autónomo



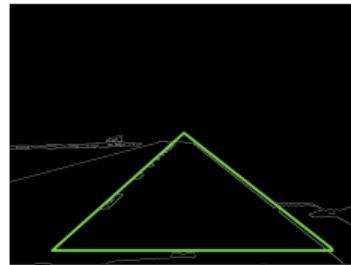
Detección de Carriles



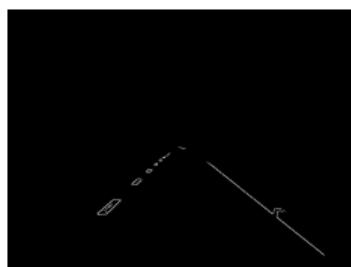
(a) Imagen original
RGB.



(b) Bordes detectados
con Canny.



(c) Geometría de
interés.



(d) Imagen de bordes
delimitada.

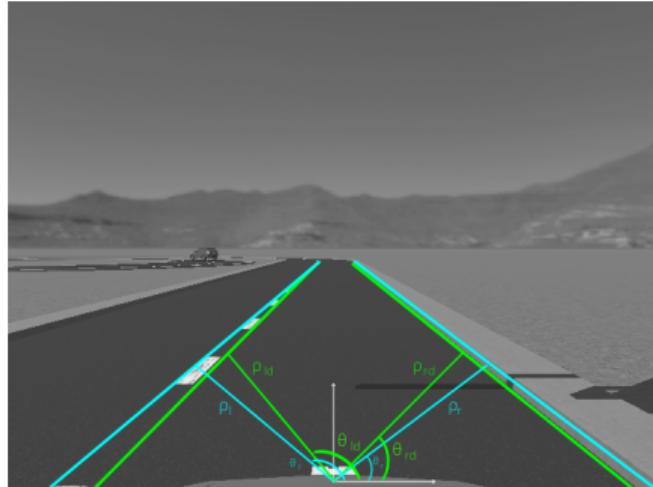


(e) Líneas detectadas
con Hough.



(f) Carriles
detectados.

Movimiento lateral y longitudinal



Leyes de Control.

Movimiento Lateral

$$\delta = K_\rho e_{\rho p} + K_\theta e_{\theta p}$$

Donde:

$$e_{\rho p} = (\rho_{Id} - \rho_I + \rho_{rd} - \rho_r)/2$$

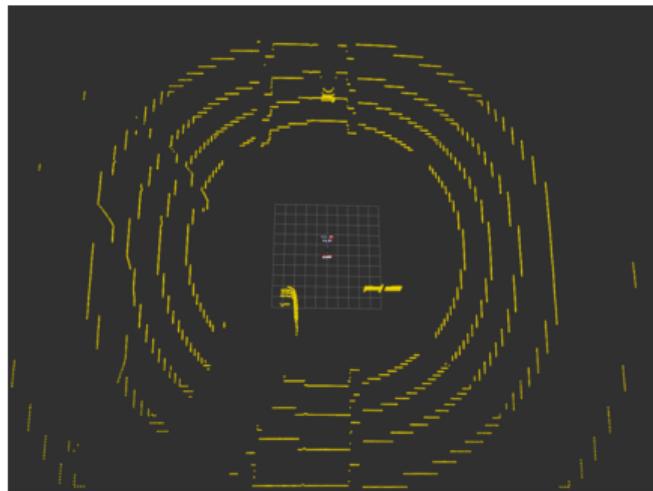
$$e_{\theta p} = (\theta_{Id} - \theta_I + \theta_{rd} - \theta_r)/2$$

Movimiento Longitudinal

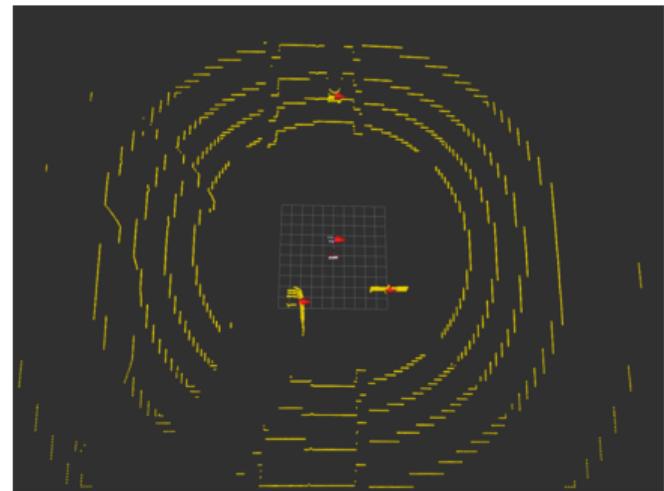
$$v = C$$

Detección de objetos

Utiliza algoritmo *k-means* sobre una nube de puntos.



Nube de puntos del sensor LIDAR.



Nube de puntos agrupada.

Empatado

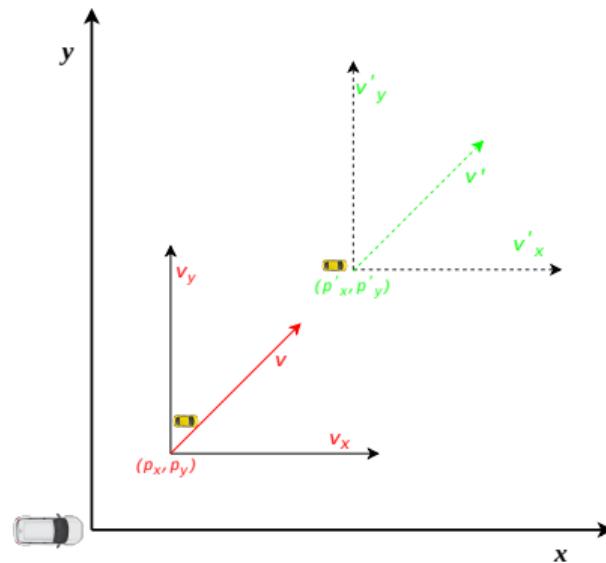
Calcular distancia euclídea.

$$d(c_a, c_p) = \sqrt{(c_{a_x} - c_{p_x})^2 + (c_{a_y} - c_{p_y})^2 + (c_{a_z} - c_{p_z})^2}$$

Criterio de decisión.

$$d(c_a, c_p) = \begin{cases} \text{Son el mismo si,} & d(c_a, c_p) \leq u \\ \text{Son diferentes si,} & d(c_a, c_p) \geq u \end{cases}$$

Estimación de posición y velocidad con el filtro de Kalman extendido



Sistema a estimar

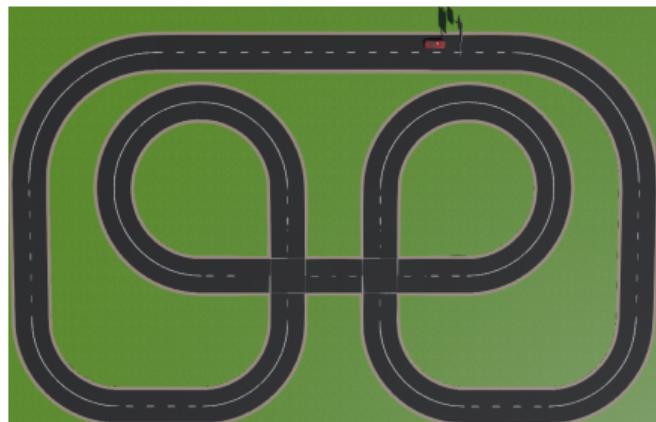
$$\begin{cases} \frac{dp}{dt} = v \\ \frac{dv}{dt} = 0 \end{cases}$$

Caso discreto

$$\begin{cases} p_{k+1} = p_k + \Delta t v_k + w \\ v_{k+1} = v_k + w \end{cases}$$

Crucero

Activo mientras no existan obstáculos en el camino.

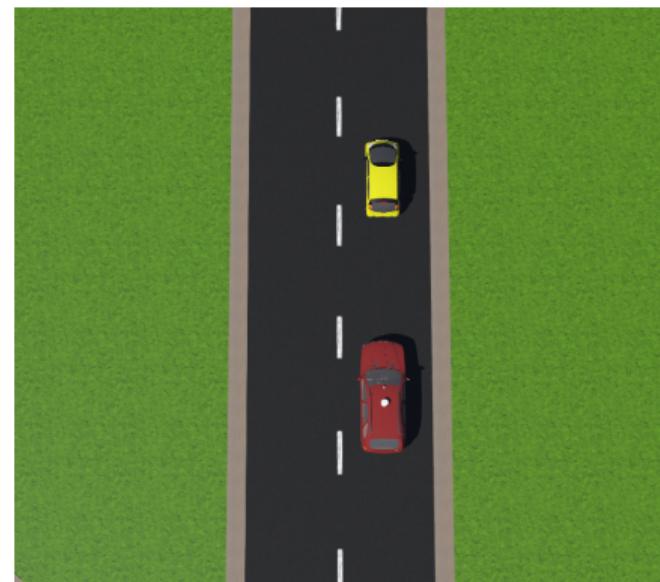


- No existen obstáculos dentro del circuito.
- Velocidad constante en un rango de [30.0, 40.0] [km/h].
- Utiliza leyes de control para movimiento lateral y longitudinal.

Rebasar

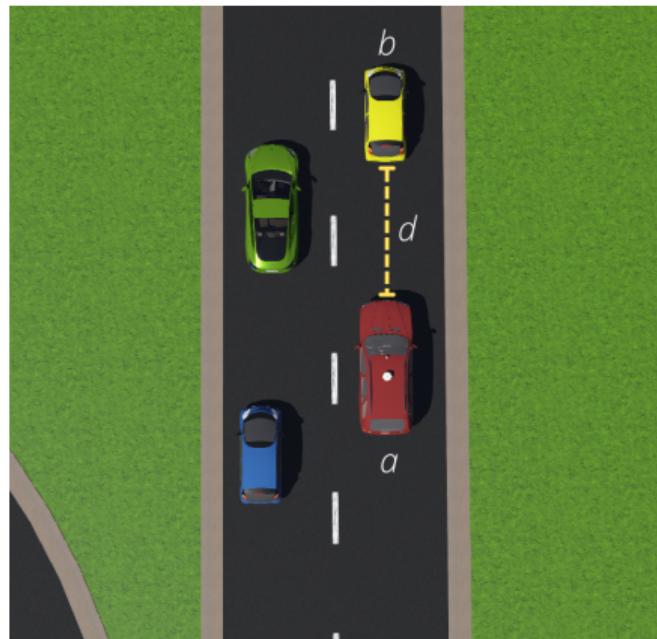
Implementado en lazo abierto y mediante una máquina de estados.

- ① Girar a la izquierda t segundos.
- ② Girar a la derecha $t/2$ segundos para alinear.
- ③ Avanzar en línea recta t segundos.
- ④ Girar a la derecha t segundos.
- ⑤ Girar a la izquierda $t/2$ segundos para alinear.
- ⑥ Avanzar en línea recta t segundos.



Mantener Distancia

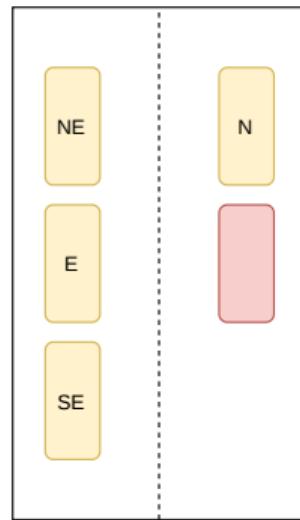
Se utiliza en situaciones de tráfico vial.



- Misma ley de control para movimiento lateral que en Crucero.
- Velocidad variable $v = d * k$.
- Mantiene distancia segura respecto a otros vehículos.

Árbitro

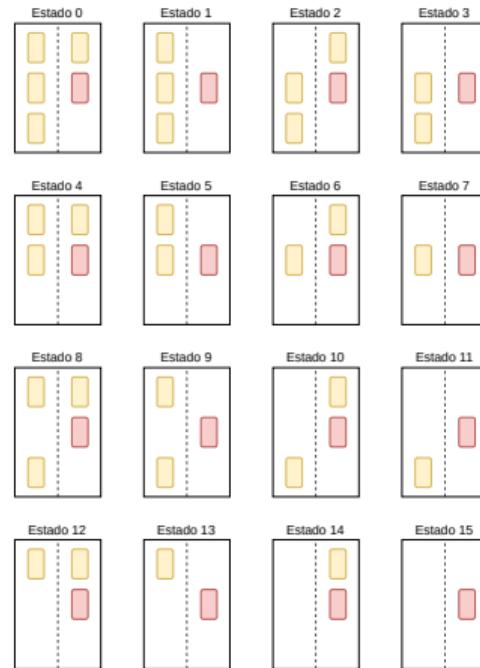
Máquina de estados para toma de decisiones y elección de comportamientos.



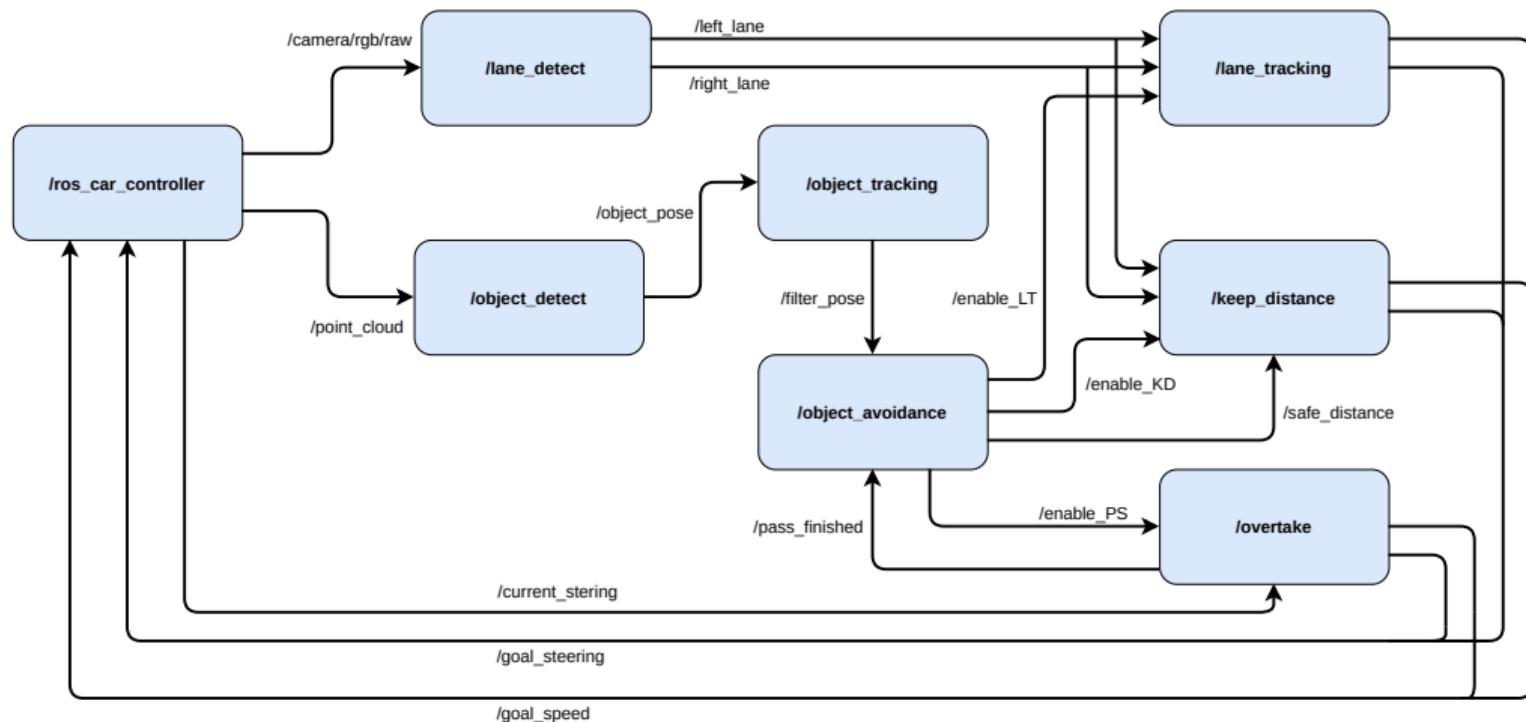
Zonas de obstáculos: Norte (N), Este (E), Noroeste (NE), Sureste (SE).

Estados

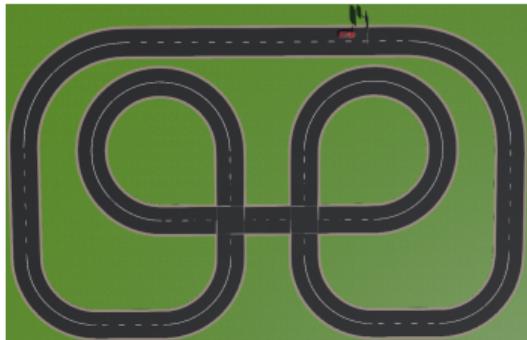
- 8 estados para Crucero.
- 6 estados para Mantener Distancia.
- 2 estados para Rebase.



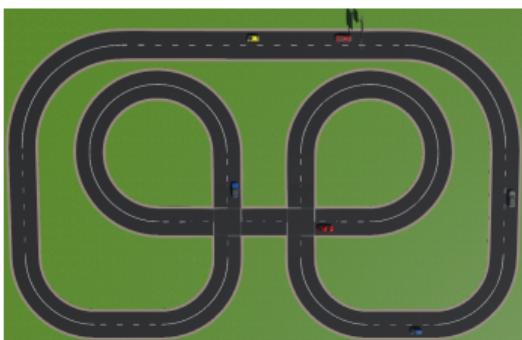
Comunicación entre nodos



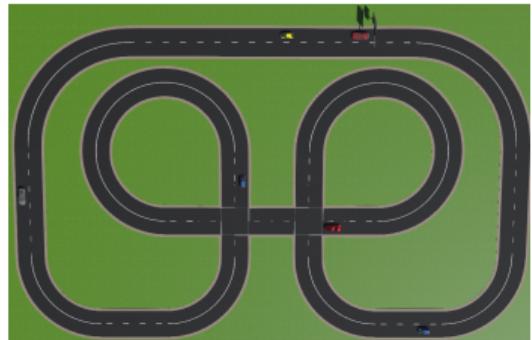
Escenarios de prueba para navegación autónoma



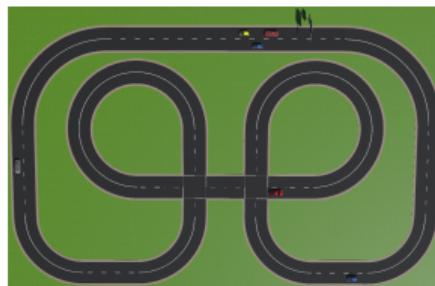
(a) Sin obstáculos.



(b) Con obstáculos estáticos.



(c) Con obstáculos dinámicos.



(d) Mantener Distancia.

Pruebas de navegación autónoma con y sin obstáculos

Navegación	Observaciones
Sin obstáculos (Crucero)	Completa el circuito con velocidad de 30 a 40[km/h].
Con obstáculos estáticos (Rebase)	Completa el circuito en 3 de 10 simulaciones Velocidad de rebase de 20[km/h].
Con obstáculos dinámicos (Rebase)	Completa rebase en 6 de 10 simulaciones Realiza rebases a 20[km/h].
Con obstáculos dinámicos (Mantener Distancia)	Detecta situación de tráfico en 9 de 10 simulaciones Velocidad en función de la distancia.

Conclusiones

- ① El detector de bordes de **Canny** y la transformada **Hough** son efectivas en el desarrollo de sistemas de visión artificial.
- ② Los simuladores son alternativas rentables en el desarrollo de tecnologías relacionadas con conducción autónoma.
- ③ Implementación de comportamientos reactivos para “**Seguir Carril**”, “**Rebase**”, “**Mantener distancia**” y sistema de “**Árbitro**” a través de máquinas de estados finitos.
- ④ Creación de escenarios y ejecución de pruebas para navegación autónoma con y sin obstáculos con el simulador **Webots**.
- ⑤ Integración entre la plataforma **ROS** y el simulador **Webots** de todos los nodos desarrollados.

Contribuciones

TMR 2022

- El ambiente de simulación y vehículo de pruebas instrumentado fueron de ayuda en el desarrollo de la plataforma para las pruebas consideradas en el TMR-AutoModelCar 2022 en su modalidad virtual.
- Participación en la misma categoría del TMR-2022 donde se obtuvo el segundo lugar.

Artículo Iberamia

- El sistema de navegación desarrollado se utilizó para el artículo “Probabilistic logic description of Markov decision processes for modeling driving behaviors in self-driving cars” publicado en Iberamia.