

Escuela Superior Politécnica Del Litoral

Facultad de Ingeniería en Mecánica y Ciencias de la Producción

Desarrollo de un sistema de navegación autónoma para vehículos F1/Tenth

INGE-2783

Proyecto Integrador

Previo a la obtención del Título de:

Ingeniero en Mecatrónica

Presentado por:

Maykoll Steven Vanegas Silva

Anthony Javier Carrillo Vega

Guayaquil - Ecuador

Año: 2024

Dedicatoria

El presente proyecto lo dedico a mi familia, en especial a mi madre y mi hermana, cuyo amor, apoyo incondicional y enseñanzas han sido mi mayor inspiración a lo largo de este camino. A mi pareja, quien siempre estuvo a mi lado brindándome ánimo y confianza en los momentos más desafiantes. También quiero dedicar este trabajo a mis amigos, quienes, con sus conocimientos, consejos y colaboración, contribuyeron significativamente a mi formación y al desarrollo de este proyecto. A todos ellos, se los dedico por ser parte fundamental de esta etapa.

Maykoll Vanegas.

Dedicatoria

Este logro académico está especialmente dedicado a mi padre, quien trabajó arduamente para brindarme las mejores oportunidades. A mi madre, quien siempre estuvo para mí cuando las cosas se ponían difíciles. A mi hermano por siempre ayudarme en todo para que yo tuviera las cosas más fáciles. Y por último a mi hermanita, que desde su nacimiento ha sido mi razón más grande de sobresalir académicamente para lograr ser un profesional del cual ella se pueda inspirar. Un abrazo a todos los mencionados, la familia siempre será algo invaluable

Anthony Carrillo.

Agradecimientos

Ante todo, agradezco a Dios, fuente de fortaleza y guía en cada paso de mi vida, por brindarme la sabiduría, perseverancia y determinación necesarias para completar este proyecto. Extiendo mi más sincero agradecimiento al Ph.D. Nabih Pico, cuya paciencia, apoyo y dedicación han sido fundamentales en mi formación y en el desarrollo de este trabajo. Al M.Sc. Efraín Terán, por su constante motivación y exigencia, impulsándonos siempre a superarnos y mejorar cada día, transmitiéndonos el compromiso por buscar la excelencia académica. Al Ph.D. Marcelo Fajardo, por su invaluable paciencia y disposición para ayudarnos en los aspectos más técnicos de este proyecto, brindándonos siempre su conocimiento y experiencia.

Maykoll Vanegas S.

Agradecimientos

Agradezco al M.Sc. Efraín Terán por su guía y compromiso como profesor y coordinador; al Ph.D. Marcelo Fajardo, por su paciencia y valiosos consejos, y al Ph.D. Nabih Pico por la confianza depositada en este proyecto.

A mis amigos de la carrera, del colegio y de otras instancias, gracias por su apoyo incondicional a lo largo de este camino. Un agradecimiento especial a mi hermano, Ángel Carrillo Vega, y a mi amigo, David Montalvan Poppe, por ser pilares fundamentales en este proceso.

Sin más que agregar, agradezco a todos... menos a uno.

Anthony Carrillo V.

Declaración Expresa

Nosotros, Anthony Javier Carrillo Vega y Maykoll Steven Vanegas Silva, acordamos y reconocemos que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda a un proyecto respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerá de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio de reconocer que el mismo nos corresponde de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso. En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sino la autorización expresa y previa de la ESPOL.



Maykoll Steven Vanegas Silva



Anthony Javier Carrillo Vega

Evaluadores

Marcelo Rodolfo Fajardo Pruna, Ph.D.

Profesor de la materia

Efraín Andrés Terán Calle, M.Sc.

Tutor del proyecto

Resumen

El presente proyecto aborda el diseño de un controlador de navegación segura para un vehículo autónomo a escala. En particular, se trabaja con la plataforma F1Tenth, un sistema de investigación basado en vehículos autónomos de escala 1:10, utilizado para el desarrollo y prueba de algoritmos de navegación autónoma en entornos controlados. El objetivo del proyecto es garantizar la detección y prevención de colisiones mientras se optimiza el desempeño del vehículo en entornos simulados. Para ello, se integró un sistema de percepción basado en LiDAR, se implementó un esquema de planificación de trayectorias utilizando SLAM, el algoritmo D* y un proceso de suavizado con splines cúbicos, y se diseñó un Control Predictivo Basado en Modelo (MPC) para optimizar la trayectoria en tiempo real. El sistema fue evaluado en simulaciones dentro de ROS 2 Humble y sus herramientas Gazebo y Rviz, creando pistas modeladas basadas en circuitos de la competencia F1Tenth. Se analizó métricas como la entropía del mapa de ocupación, la curvatura de la trayectoria y el error cuadrático medio (MSE) del seguimiento. Los resultados demostraron que el sistema es capaz de completar múltiples vueltas sin incidentes, con un MSE menor a 0.08 m y una velocidad promedio de hasta 1.87 m/s. Se concluye que la combinación de SLAM, planificación *offline* y MPC permite una navegación segura y eficiente. No obstante, se recomienda completar la integración del sistema en hardware embebido para validar su desempeño en pruebas reales.

Palabras Clave: Navegación autónoma, SLAM, planificación de trayectorias, MPC, F1Tenth.

Abstract

This project focuses on the design of a safe navigation controller for a scaled autonomous vehicle. In particular, it employs the F1Tenth platform, a research system based on 1:10 scale autonomous vehicles, widely used for the development and testing of autonomous navigation algorithms in controlled environments. The objective of this project is to ensure collision detection and prevention while optimizing vehicle performance in simulated environments. To achieve this, a LiDAR-based perception system was integrated, a trajectory planning scheme was implemented using SLAM, the D algorithm, and a cubic spline smoothing process, and a Model Predictive Control (MPC) approach was designed to optimize real-time trajectory execution. The system was evaluated through simulations in ROS 2 Humble, using Gazebo and RViz, with modelled tracks based on F1Tenth competition circuits. Performance metrics such as occupancy map entropy, trajectory curvature, and mean squared error (MSE) of tracking were analyzed. The results demonstrated that the system successfully completed multiple laps without incidents, achieving an MSE below 0.08 m and an average speed of up to 1.87 m/s. It is concluded that the combination of SLAM, offline planning, and MPC enables safe and efficient navigation. However, it is recommended to fully integrate the system into embedded hardware to validate its performance in real-world testing.*

Keywords: *Autonomous navigation, SLAM, trajectory planning, MPC, F1Tenth.*

Índice general

Resumen	I
<i>Abstract</i>	II
Índice general	III
Abreviaturas	VI
Simbología	VII
Índice de figuras	IX
Índice de tablas	X
Capítulo 1	1
1. Introducción	2
1.1 Descripción del problema	6
1.2 Justificación del problema	9
1.3 Objetivos	12
1.3.1 Objetivo general	12
1.3.2 Objetivos específicos	13
1.4 Marco teórico	13
1.4.1 Vehículos autónomos	13
1.4.2 Plataforma F1Tenth	15
1.4.3 Sensores de percepción y detección	17
1.4.4 Algoritmos de navegación y control	20
1.4.5 Evaluación de rendimiento	24
Capítulo 2	26
2. Metodología	27
2.1 Formulación de alternativas de solución	28

2.1.1	Alternativas de diseño según el sensor	29
2.1.2	Alternativas de diseño para algoritmos de navegación	32
2.1.3	Alternativas para la evaluación del entorno	34
2.1.4	Evaluación multicriterio de las alternativas	37
2.2	Diseño conceptual	42
2.2.1	Subsistema de percepción	43
2.2.2	Subsistema de procesamiento	44
2.2.3	Subsistema de actuación	45
2.3	Diseño del controlador	46
2.3.1	Generación del mapa de ocupación mediante SLAM	46
2.3.2	Generación de trayectorias con D* y suavización mediante splines cúbicos	48
2.3.3	Modelo del vehículo para simulación	52
2.3.4	Modelo del vehículo para implementación	54
2.3.5	Estructura del Control Predictivo basado en Modelo (MPC)	56
2.3.6	Restricciones del Control Predictivo basado en Modelo (MPC)	58
2.3.7	Cálculo de la matriz de estado del modelo linealizado	60
2.3.8	Optimización del MPC	62
2.3.9	Implementación del controlador en ROS2 Humble	63
Capítulo 3	65
3.	Resultados y evaluación del sistema	66
3.1	Generación del entorno en Gazebo	66
3.2	Generación del mapa de ocupación	67
3.3	Generación y filtrado de la trayectoria	69
3.4	Pruebas de navegación del vehículo autónomo	71

3.5	Análisis de desempeño del sistema	74
3.5.1	Discusión de resultados	75
Capítulo 4	77
4.	Conclusiones y recomendaciones	78
4.1	Conclusiones	78
4.2	Recomendaciones	79
Referencias	81
Apéndice	89

Abreviaturas

<i>A*</i>	<i>A-star</i> , algoritmo de planificación heurística basado en búsqueda en grafos
<i>CNN</i>	<i>Convolutional Neural Network</i> , o Red Neuronal Convolutiva
<i>D*</i>	<i>Dynamic A*</i> , algoritmo de planificación de rutas óptimas
<i>EKF</i>	<i>Extended Kalman Filter</i> , o Filtro de Kalman Extendido
F1Tenth	Plataforma de vehículos autónomos a escala para competencias de robótica
H	Entropía del Mapa de Ocupación
<i>IMU</i>	<i>Inertial Measurement Unit</i> , o Unidad de Medición Inercial
IPOS	Método de decisión Impacto, Probabilidad, Oportunidad y Simplicidad
<i>LiDAR</i>	<i>Light Detection and Ranging</i> , sensor de percepción basado en láser
<i>LQR</i>	<i>Linear Quadratic Regulator</i> , o Regulador Lineal Cuadrático
<i>MPC</i>	<i>Model Predictive Controller</i> , o Control de Predicción por Modelos
<i>MPPI</i>	<i>Model Predictive Path Integral</i> , controlador para integración de trayectorias
<i>MSE</i>	<i>Mean Squared Error</i> , o Error Cuadrático Medio
<i>PID</i>	<i>Proportional-Integral-Derivative</i> , controlador basado en retroalimentación
<i>QP</i>	<i>Quadratic Programming</i> , o Programación Cuadrática
<i>RL</i>	<i>Reinforcement Learning</i> , o Aprendizaje por Refuerzo
<i>RRT</i>	<i>Rapidly-exploring Random Trees</i> , algoritmo de planificación de trayectorias
<i>ROS</i>	<i>Robot Operating System</i> , marco de software para robótica
<i>SLAM</i>	<i>Simultaneous Localization and Mapping</i> , o Localización y Mapeo Simultáneos
Spline	Función matemática utilizada para interpolación y suavización de trayectorias

Simbología

x, y, θ	Posición y orientación del vehículo en el plano.
v_x, v_y	Componentes de la velocidad del vehículo en los ejes x e y .
ω	Velocidad angular del vehículo.
δ	Ángulo de dirección del vehículo.
a	Aceleración del vehículo.
L	Distancia entre ejes del vehículo.
A, B	Matrices de dinámica y control en espacio de estados.
u	Vector de entrada de control en el modelo en espacio de estados.
J	Función de costo del MPC.
$A_{\text{ineq}}, b_{\text{ineq}}$	Matriz y vector de restricciones en la optimización cuadrática.
d_{obs}	Distancia al obstáculo más cercano en el entorno.
d_{min}	Distancia mínima de seguridad para evitar colisiones.
κ_{avg}	Curvatura media de la trayectoria generada.
$\Delta\theta_{\text{total}}$	Variación angular acumulada de la trayectoria.
H_{map}	Entropía del mapa de ocupación generado por SLAM.
τ	Horizonte de predicción del MPC.
$x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}}$	Valores de referencia de posición y orientación en el MPC.
F_x, F_y	Fuerzas en los ejes x e y en el modelo dinámico.
F_{lat}	Fuerza lateral de los neumáticos.
l_f, l_r	Distancia del centro de masa a los ejes delantero y trasero.

Índice de figuras

Figura 1.1	<i>Vehículo F1Tenth</i>	2
Figura 1.2	<i>Competencia F1Tenth</i>	3
Figura 1.3	<i>Distribución de la responsabilidad en accidentes de vehículos autónomos</i> . .	4
Figura 1.4	<i>Tipos de colisiones reportadas</i>	5
Figura 1.5	<i>Vehículo F1Tenth</i>	6
Figura 1.6	<i>Ejemplo de oclusión parcial en un sensor LiDAR</i>	7
Figura 1.7	<i>Ejemplo de la planificación de trayectoria entre dos puntos</i>	8
Figura 1.8	<i>Enfoque para construir trayectorias en curvas a partir de los datos del sensor Lidar</i>	9
Figura 1.9	<i>Influencia de la odometría en la alineación de escaneos láser para la navegación autónoma</i>	11
Figura 1.10	<i>Simulación de algoritmos de navegación para la validación de estrategias autónomas</i>	12
Figura 1.11	<i>Clasificación de los niveles de autonomía en vehículos autónomos</i>	14
Figura 1.12	<i>Plataforma F1Tenth</i>	15
Figura 1.13	<i>Competencia F1 Tenth: Modalidad contrarreloj</i>	16
Figura 1.14	<i>Competencia F1 Tenth: Modalidad paralelo</i>	17
Figura 1.15	<i>Funcionamiento de un sensor Lidar</i>	18
Figura 1.16	<i>Cámaras en la navegación autónoma</i>	19
Figura 1.17	<i>Implementación de SLAM en F1Tenth</i>	21
Figura 1.18	<i>Ejemplo de generacion de trayectoria</i>	23
Figura 2.1	<i>Radar mmWave WR6843ISK</i>	30

Figura 2.2	<i>RPLidar S2</i>	31
Figura 2.3	<i>Cámara RGB Intel RealSense</i>	32
Figura 2.4	<i>Matriz Impacto vs. Dificultad</i>	41
Figura 2.5	Diseño conceptual del controlador del vehículo F1Tenth.	43
Figura 2.6	<i>Placa integrada Jetson Nano</i>	44
Figura 2.7	<i>Logica detras de un mapa de ocupación</i>	47
Figura 2.8	<i>Ejemplo de mapeo con Slam_toolbox</i>	49
Figura 2.9	<i>Optimización de trayectoria utilizando D*</i>	50
Figura 2.10	<i>Ejemplo de suavizado de trayectoria usando Splines Cúbicos</i>	52
Figura 2.11	<i>Modelo cinemático de bicicleta</i>	53
Figura 2.12	<i>Single Track Dynamic Model</i>	55
Figura 3.1	Modelado de la pista Levine en el simulador Gazebo.	67
Figura 3.2	Mapa de ocupación generado para la pista Levine.	68
Figura 3.3	Trayectoria generada sin filtrado en la pista Levine.	69
Figura 3.4	Trayectoria optimizada y suavizada en la pista Levine.	70
Figura 3.5	El controlador se acopla muy bien a la trayectoria ideal cuando no hay obstáculos.	72
Figura 3.6	El MPC ajusta la trayectoria en tiempo real para evitar un obstáculo en la pista.	72
Figura 3.7	El vehículo intenta corregir hacia su trayectoria optima luego de esquivar obstaculos fijos, mientras se esta colocando un obstaculo dinámico frente a él.	73
Figura 3.8	Se evalúa la capacidad del sistema para reaccionar a cambios inesperados en el entorno.	73

Índice de tablas

Tabla 2.1	Matriz IPOS para la selección de sensores.	37
Tabla 2.2	Matriz IPOS para algoritmos de navegación (parte 1).	38
Tabla 2.3	Matriz IPOS para algoritmos de navegación (parte 2).	39
Tabla 2.4	Matriz IPOS para algoritmos de evaluación del entorno.	39
Tabla 2.5	Matriz IPOS para combinaciones de alternativas.	41
Tabla 3.1	Resumen de métricas de desempeño en la pista Levine.	75

Capítulo 1

1. Introducción

La creciente demanda por soluciones de movilidad más seguras y eficientes ha impulsado el desarrollo de vehículos autónomos. Estos sistemas, capaces de navegar y tomar decisiones sin intervención humana, prometen revolucionar la industria automotriz y transformar la forma en que nos desplazamos.

Uno de los modelos más utilizados en la investigación de conducción autónoma es el *F1Tenth*, una plataforma de vehículos autónomos a escala 1:10, desarrollada para la investigación y validación de algoritmos de percepción, planificación y control. El nombre *F1Tenth* se debe a la combinación de "F1", en referencia a la Fórmula 1, debido a su aplicación en carreras autónomas, y "Tenth", en alusión a su escala 1:10 en comparación con un automóvil real [1].

Figura 1.1.

Vehículo F1Tenth



Nota: Ejemplo obtenido de [2].

Esta plataforma es ampliamente utilizada en competencias internacionales y entornos académicos, permitiendo la experimentación con diferentes estrategias de conducción autónoma en circuitos controlados.

Figura 1.2.*Competencia F1Tenth*

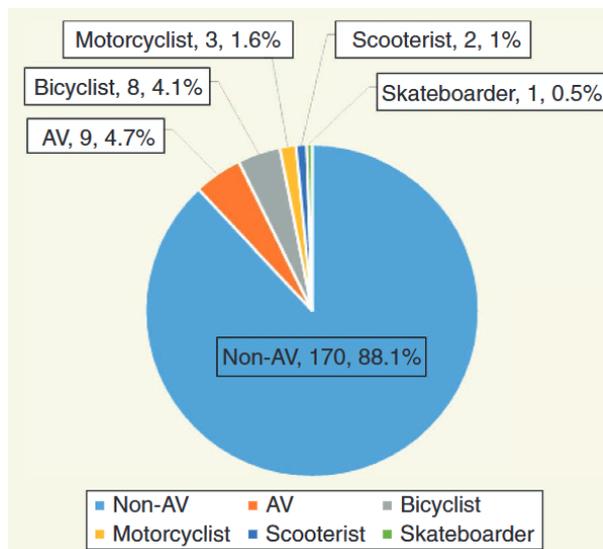
Nota: Adaptado de ForzaETH Team [2].

El presente trabajo se centra en el diseño e implementación de un controlador de navegación segura para el vehículo autónomo F1Tenth, un desafío que requiere la integración precisa de sensores, algoritmos de control y simulación.

La seguridad es un aspecto fundamental en el desarrollo de vehículos autónomos. Según un estudio reciente de Ghorai [3], la mayoría de los accidentes de tránsito son causados por errores humanos, como el exceso de velocidad, la violación de señales de tráfico y decisiones de conducción deficientes, representando un porcentaje significativo de los incidentes reportados. De acuerdo con un análisis de informes policiales, en el 88% de los casos analizados, la responsabilidad de los accidentes recayó en vehículos convencionales (CV), mientras que los vehículos autónomos (AV) no tenían margen para evitar la colisión.

Figura 1.3.

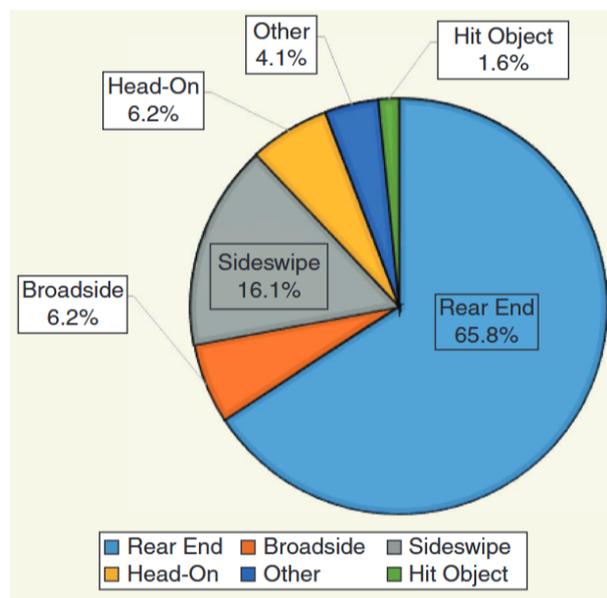
Distribución de la responsabilidad en accidentes de vehículos autónomos



Nota: Adaptado de Ghorai et al. [3].

Las principales causas de accidentes incluyen conducción bajo efectos del alcohol o drogas, exceso de velocidad, violación de señales de tránsito y cambios de carril inseguros. Además, peatones, ciclistas y otros usuarios viales han sido responsables de colisiones debido a maniobras imprudentes. Frecuentemente, los AVs se detienen ante peatones cruzando, pero son impactados por CVs desde atrás, evidenciando que, aunque los AVs reducen accidentes, su interacción con conductores humanos sigue siendo un desafío.

Los AVs pueden disminuir significativamente la tasa de accidentes, especialmente en colisiones por alcance (65.8%). No obstante, su efectividad en entornos reales depende de estrategias que mitiguen riesgos derivados del comportamiento impredecible de otros usuarios de la vía.

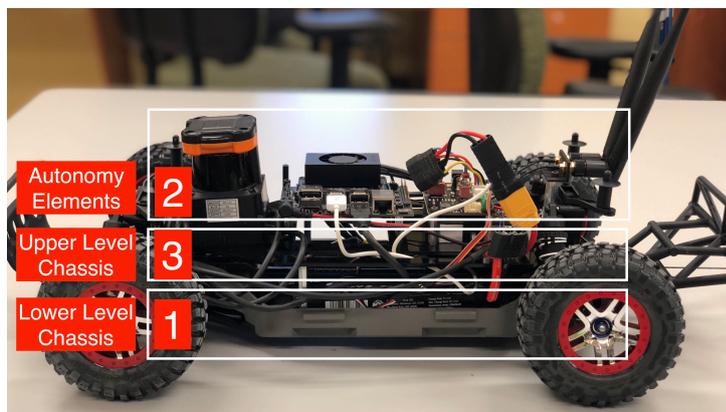
Figura 1.4.*Tipos de colisiones reportadas*

Nota: Adaptado de Ghorai et al. [3].

Sin embargo, garantizar la seguridad en entornos dinámicos y complejos como las vías públicas exige el desarrollo de algoritmos robustos y eficientes para la detección de obstáculos, la planificación de trayectorias y el control del vehículo, particularmente en condiciones adversas como el mal tiempo o fallos de automatización [?].

El desarrollo de plataformas autónomas a escala ha permitido realizar pruebas y experimentos con menor riesgo y costos reducidos. El *F1Tenth*, basado en un modelo de automóvil controlado por radio (RC), compete en la *F1/10 Autonomous Racing Competition*, una competencia donde los vehículos deben demostrar su capacidad de navegación autónoma en circuitos cerrados. Este vehículo está equipado con un computador a bordo y un sensor LiDAR, lo que le permite percibir su entorno y reaccionar en tiempo real.

A pesar de su tamaño reducido, el vehículo F1Tenth es ampliamente utilizado en la

Figura 1.5.*Vehículo F1Tenth*

Nota: Tomado de F1Tenth [2].

robótica móvil por su modularidad y facilidad de integración con sensores y actuadores. Investigaciones previas, como [4], han validado su uso en la navegación autónoma, permitiendo probar algoritmos en entornos controlados antes de su aplicación en vehículos a mayor escala.

Este trabajo busca contribuir al desarrollo de la conducción autónoma mediante un controlador de navegación que integre detección de obstáculos y planificación de trayectorias. La incorporación de un sensor LiDAR permitirá mapear el entorno y detectar obstáculos en tiempo real, mientras que el uso de ROS (*Robot Operating System*) facilitará la implementación del sistema.

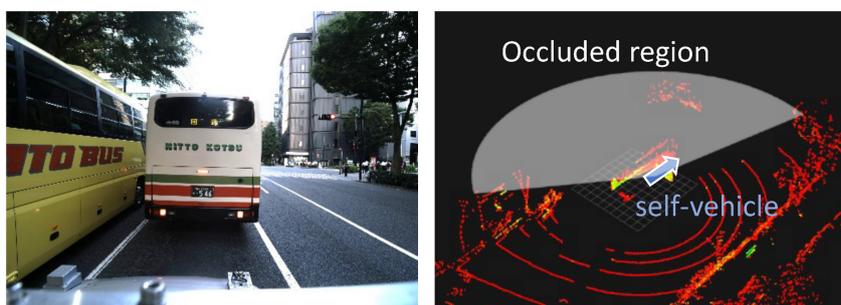
1.1 Descripción del problema

La implementación de un vehículo autónomo capaz de competir en F1Tenth se enfrenta a una serie de desafíos complejos. Uno de los principales retos reside en la percepción precisa y en tiempo real del entorno. Los sistemas de percepción deben lidiar con

condiciones ambientales variables, como cambios en la iluminación o superficies resbaladizas, lo que puede afectar significativamente la calidad de los datos obtenidos por los sensores. Además, la detección y seguimiento de objetos en movimiento, como otros vehículos, en entornos dinámicos y con oclusiones parciales, representa un desafío considerable para sensores como el LIDAR [5].

Figura 1.6.

Ejemplo de oclusión parcial en un sensor LiDAR

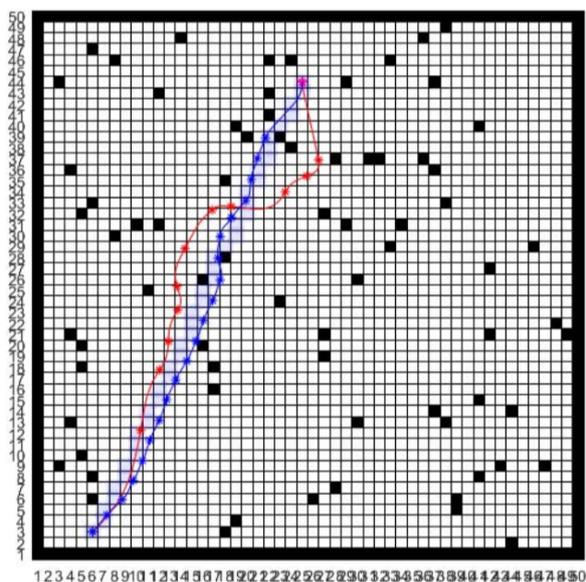


Nota: Tomado de Endo et al. [6].

Otro desafío importante es la planificación de trayectorias seguras y eficientes. Los algoritmos de planificación deben generar trayectorias que eviten colisiones con obstáculos estáticos y dinámicos, al tiempo que optimizan el tiempo de vuelta [7]. Sin embargo, la incertidumbre inherente a las medidas de los sensores y la naturaleza dinámica del entorno dificultan la generación de planes robustos. Además, la necesidad de tomar decisiones en tiempo real impone restricciones computacionales significativas [8].

Figura 1.7.

Ejemplo de la planificación de trayectoria entre dos puntos

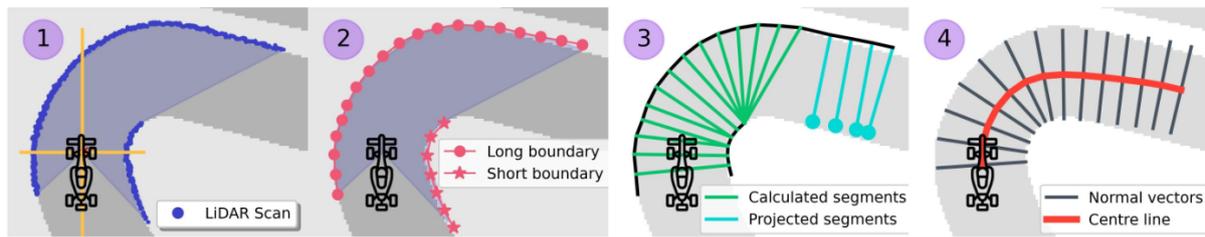


Nota: Tomado de Zeng et al. [9].

El control preciso del vehículo es otro aspecto crítico. El controlador debe garantizar que el vehículo siga la trayectoria planificada con precisión, adaptándose a cambios en el entorno y a perturbaciones externas. La interacción con el entorno, como realizar giros, cambios de carril o detenerse ante obstáculos, requiere un control adecuado de los actuadores del vehículo. Además, la estabilidad del vehículo debe garantizarse en diferentes condiciones de conducción, como curvas cerradas o superficies resbaladizas [10].

Figura 1.8.

Enfoque para construir trayectorias en curvas a partir de los datos del sensor Lidar



Nota: Tomado de Evans et al. [11].

La necesidad de un control preciso de estos parámetros se refleja claramente en los resultados de las pruebas de competición, donde un fallo o colisión puede significar la descalificación. En la competencia realizada en las Islas Jeju, Corea del Sur, en 2022, el equipo ganador obtuvo el primer lugar con un tiempo de 9.75 segundos [12], marca que fue superada en 2023 durante la competencia en Yeosu, Corea del Sur, con un tiempo de 8.2 segundos. Estos resultados destacan la importancia de una integración total de los sistemas y de evitar colisiones con el entorno, ya que cualquier choque penaliza en la puntuación de la competencia debido a la pérdida de tiempo que esto representa [13].

1.2 Justificación del problema

La implementación de vehículos autónomos enfrenta un desafío fundamental: garantizar un desplazamiento seguro y una toma de decisiones eficiente sin intervención humana en entornos dinámicos y cambiantes. Dado que una parte significativa de los accidentes de tráfico es causada por errores humanos, como el exceso de velocidad, la distracción y la toma de decisiones incorrectas, diversas investigaciones resaltan la necesidad de desarrollar tecnologías que mitiguen estos riesgos. Las fallas en la conducción ponen en peligro no solo a los conductores, sino también a pasajeros, peatones y otros

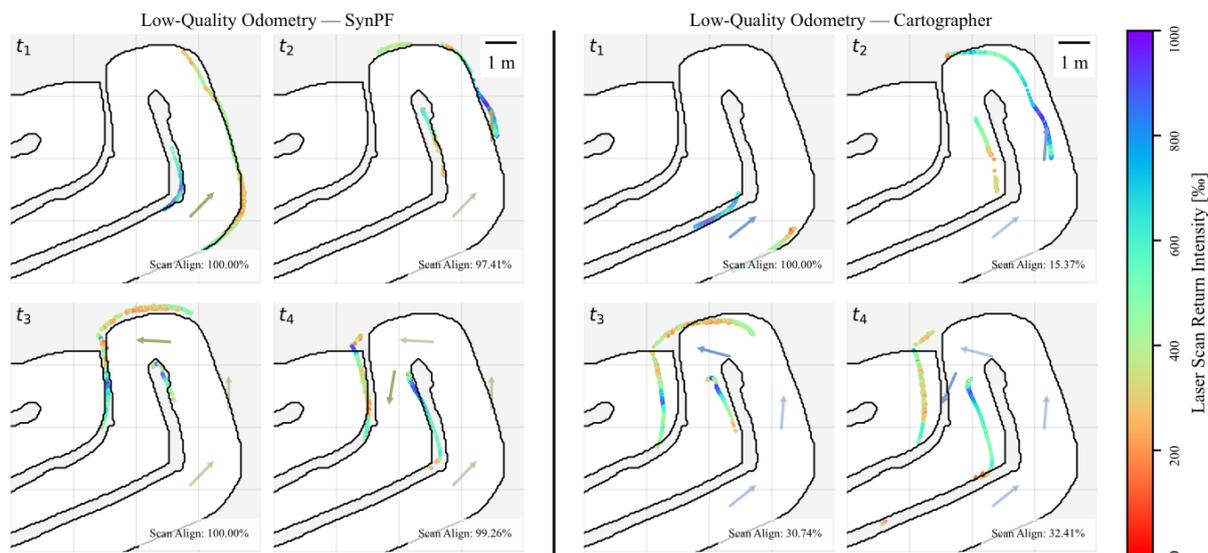
usuarios de la vía. En este contexto, los avances en vehículos autónomos pueden reducir drásticamente la siniestralidad al proporcionar sistemas capaces de reaccionar con mayor rapidez y precisión ante condiciones adversas.

Sin embargo, lograr una seguridad total requiere superar múltiples desafíos, entre ellos la identificación precisa de obstáculos, una planificación eficiente de la trayectoria y un control óptimo del vehículo en terrenos complejos. Factores como carreteras en mal estado, condiciones de baja visibilidad o errores en la odometría pueden comprometer el rendimiento del sistema y afectar la fiabilidad de la navegación.

Uno de los principales retos en este ámbito es la odometría de baja calidad, que dificulta la construcción de mapas y la planificación de trayectorias precisas en vehículos autónomos. La Figura 1.11 ilustra cómo la alineación de escaneo varía según el método de localización utilizado, evidenciando que errores en la odometría pueden distorsionar el mapeo del entorno y afectar la trayectoria generada. Para mitigar estos problemas, el uso de simuladores con odometría precisa permite desarrollar y evaluar algoritmos de navegación en entornos controlados, garantizando la generación de trayectorias más confiables. Posteriormente, estos modelos pueden implementarse en plataformas a escala como F1Tenth, lo que posibilita validar su desempeño antes de aplicarlos en vehículos de tamaño real.

Figura 1.9.

Influencia de la odometría en la alineación de escaneos láser para la navegación autónoma



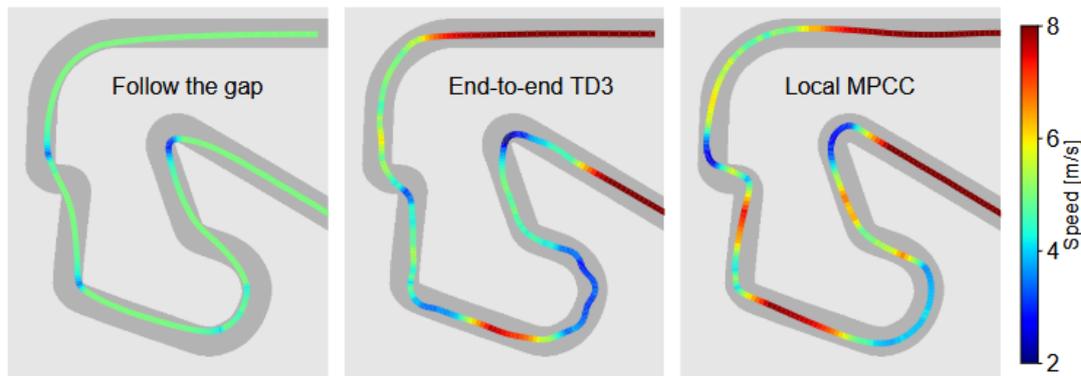
Nota: Tomado de Baumann et al. [14].

F1Tenth representa una plataforma experimental clave para abordar los desafíos de la conducción autónoma en un entorno seguro, controlado y de bajo costo, proporcionando un laboratorio ideal para el desarrollo y optimización de algoritmos de navegación y percepción. Su versatilidad permite no solo validar estrategias en vehículos físicos a escala, sino también experimentar en entornos simulados, lo que facilita la prueba de múltiples enfoques sin riesgos ni costos elevados.

La Figura 1.10 ilustra cómo, en un mismo circuito simulado, se pueden comparar distintas estrategias de control y navegación, permitiendo obtener métricas cuantificables como la velocidad alcanzada en diferentes segmentos de la pista. Este enfoque es fundamental para la evaluación y optimización de algoritmos de planificación y control, permitiendo seleccionar el enfoque más eficiente antes de su implementación en el mundo real.

Figura 1.10.

Simulación de algoritmos de navegación para la validación de estrategias autónomas



Nota: Tomado de Evans et al. [11].

Además, la incorporación de escenarios de conducción real en sus competencias ha permitido refinar la tecnología y detectar áreas de mejora, asegurando que los algoritmos sean robustos antes de aplicarlos en condiciones de tráfico reales.

En última instancia, el uso de simuladores y modelos a escala como F1Tenth acelera el desarrollo de vehículos autónomos más seguros, confiables y sostenibles, contribuyendo significativamente a la reducción de accidentes y a la evolución de un sistema de transporte más eficiente. De esta manera, se da un paso fundamental hacia la solución de uno de los problemas más críticos de la seguridad vial en la actualidad.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar y simular un controlador de navegación segura para el vehículo autónomo F1Tenth, que permita la detección, la prevención de colisiones y garantice un desempeño óptimo.

1.3.2 Objetivos específicos

- Integrar y configurar el sistema de detección de entorno utilizando un sensor Lidar, asegurando la correcta percepción de obstáculos en el entorno del vehículo.
- Desarrollar el controlador autónomo en ROS, implementando algoritmos de navegación y evitando colisiones mediante estrategias de control avanzadas.
- Simular el desempeño del sistema completo en un entorno virtual utilizando ROS, validando la funcionalidad del controlador en diferentes escenarios y condiciones.

1.4 Marco teórico

El presente proyecto aborda el desarrollo de tecnologías autónomas a distintas escalas, desde automóviles de tamaño real hasta vehículos a pequeña escala. A continuación, se abordan los fundamentos teóricos de la conducción autónoma, el desarrollo de plataformas a escala como F1Tenth y los algoritmos de percepción, planificación y control aplicables a esta tecnología, aportando también conceptos relevantes que permitirán definir el enfoque de la solución.

1.4.1 Vehículos autónomos

Definición y Contexto de los Vehículos Autónomos

Los vehículos autónomos son sistemas que pueden desplazarse sin intervención humana, utilizando una combinación de sensores, algoritmos de percepción y planificación, y sistemas de control [15]. Estos vehículos están diseñados para detectar su entorno, procesar la información recibida y tomar decisiones en tiempo real, lo que los convierte en soluciones avanzadas para múltiples industrias, especialmente en el transporte. Los avances en esta tecnología han generado interés en diferentes sectores, incluyendo la investigación

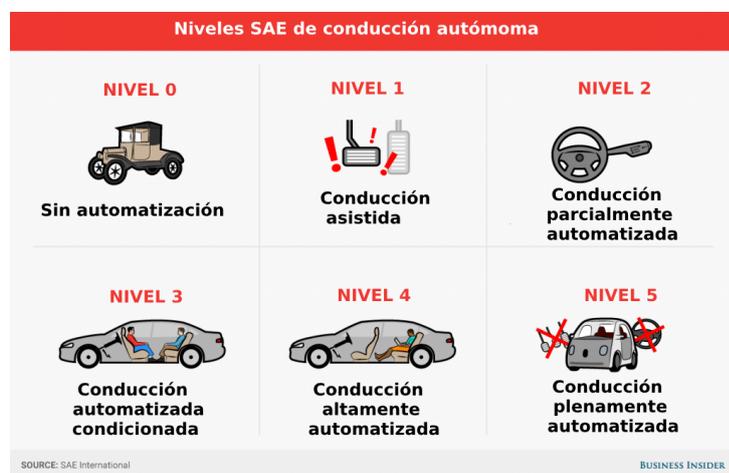
académica, la educación y la industria de la robótica, donde se utilizan plataformas de pruebas a escala como el F1Tenth para estudiar y experimentar con algoritmos de navegación autónoma [16].

Clasificación de Vehículos Autónomos por Niveles de Autonomía

Los vehículos autónomos se clasifican generalmente en niveles del 0 al 5, de acuerdo con el grado de autonomía. Los niveles bajos (1-2) requieren intervención humana en ciertas situaciones, mientras que los niveles avanzados (4-5) permiten la conducción completamente autónoma en entornos predeterminados o incluso en cualquier entorno. La plataforma F1Tenth se utiliza principalmente en experimentación y simulación de niveles de autonomía avanzados, donde se prueban algoritmos de percepción, planificación de ruta y control sin la necesidad de un operador [17].

Figura 1.11.

Clasificación de los niveles de autonomía en vehículos autónomos



Nota: Tomado de Zöldy y Zsombók [18].

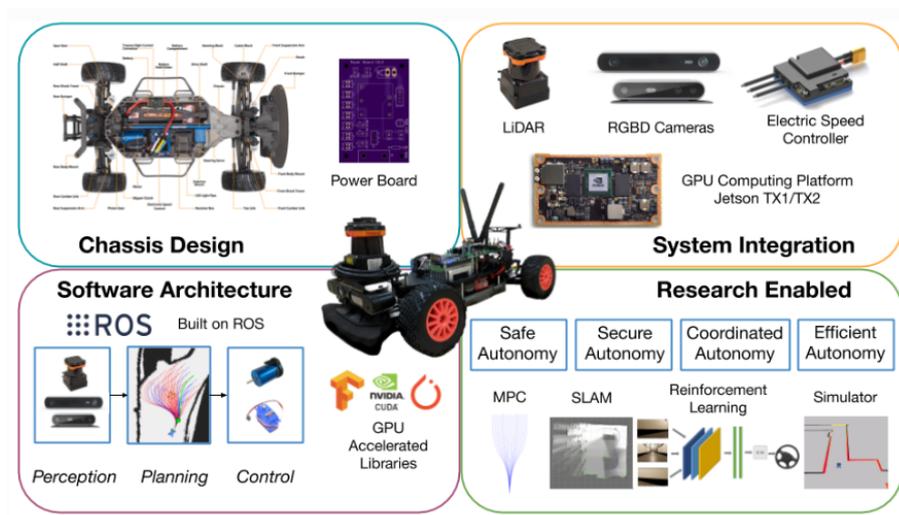
1.4.2 Plataforma F1Tenth

Características de la plataforma

La plataforma F1Tenth es un modelo de vehículo autónomo a escala 1:10, diseñado principalmente para la investigación y experimentación en algoritmos de navegación. Su tamaño reducido y su estructura flexible lo convierten en una herramienta ideal para el aprendizaje y desarrollo de sistemas autónomos sin los riesgos ni costos de un vehículo de tamaño completo. F1Tenth utiliza hardware de alto rendimiento como el procesador Nvidia Jetson TX2 y sensores avanzados como el LiDAR y cámaras, que le permiten realizar tareas de percepción y navegación en tiempo real. Esta plataforma también cuenta con el respaldo de múltiples universidades, donde es utilizada como recurso académico en áreas de robótica y conducción autónoma.[16; 17].

Figura 1.12.

Plataforma F1Tenth



Nota: Tomado de F1Tenth Team [1].

Modalidades de Competencia

Las competencias de F1Tenth incluyen dos modalidades principales: la carrera contrarreloj y la carrera en paralelo. En la modalidad contrarreloj, el vehículo debe completar un circuito en el menor tiempo posible, enfocándose en la eficiencia de los algoritmos de planificación de ruta y control de movimiento. La modalidad en paralelo agrega complejidad, ya que el vehículo debe detectar y esquivar otros competidores en tiempo real, aumentando la exigencia en algoritmos de detección y evasión de obstáculos, así como en la capacidad de toma de decisiones del sistema [17].

Figura 1.13.

Competencia F1 Tenth: Modalidad contrarreloj



Nota: Tomado de Lim [19].

Figura 1.14.

Competencia F1 Tenth: Modalidad paralelo



Nota: Tomado de Kale [20].

1.4.3 Sensores de percepción y detección

LiDAR (Light Detection and Ranging)

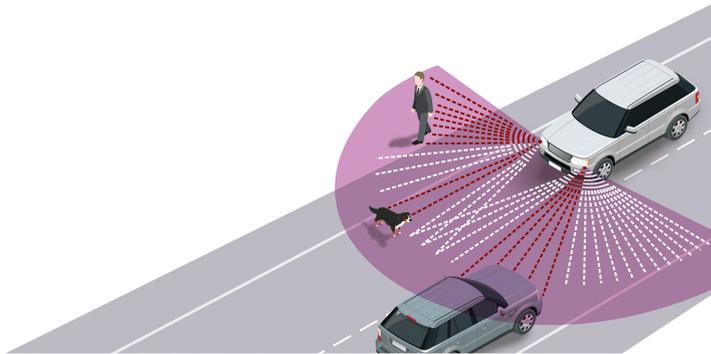
El LiDAR es uno de los sensores clave en vehículos autónomos y, en la plataforma F1Tenth, se utiliza para la percepción del entorno en tiempo real. Este sensor emite pulsos láser que miden la distancia entre el vehículo y los objetos a su alrededor, generando un mapa tridimensional de alta precisión. Su principal ventaja es su capacidad para funcionar de manera confiable bajo diversas condiciones de iluminación y su precisión en la medición de distancias, lo que lo convierte en una herramienta indispensable para la detección de obstáculos y la creación de mapas en entornos complejos.

En pruebas realizadas con la plataforma F1Tenth, el LiDAR ha demostrado ser eficaz para detectar objetos hasta una distancia de 15 metros con un margen de error de aproximadamente 2 cm, lo cual es crucial en carreras y entornos de alta velocidad. Además, el LiDAR permite crear mapas 3D detallados que pueden ser utilizados por los algoritmos de planificación de trayectorias para calcular rutas eficientes evitando obstáculos de manera dinámica. Estos datos cuantificados son fundamentales para garantizar la seguridad y

eficiencia en las competiciones [21].

Figura 1.15.

Funcionamiento de un sensor Lidar



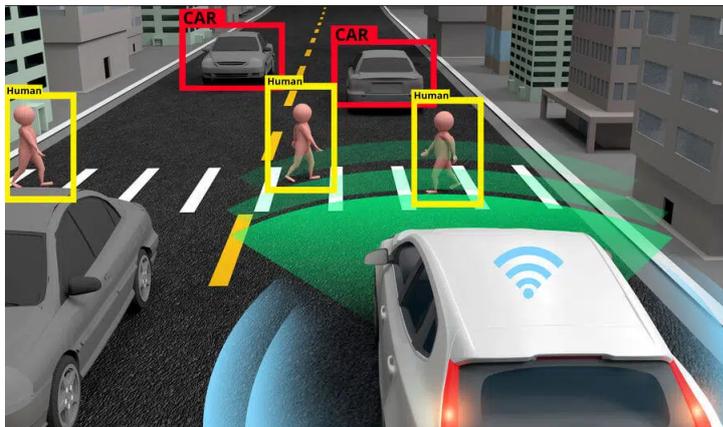
Nota: Tomado de e-Motec [22].

Cámaras y percepción visual

Las cámaras son sensores esenciales en visión por computadora y se emplean en sistemas autónomos para identificar, clasificar y rastrear objetos en su entorno. En vehículos como el F1Tenth, estas cámaras permiten detectar líneas de pista, reconocer señales de tráfico y complementar la percepción del LiDAR, mejorando la precisión en la detección y la capacidad de respuesta del vehículo en tiempo real.

El sistema F1Tenth puede utilizar cámaras RGB convencionales o cámaras estéreo para obtener información tridimensional del entorno. Estas cámaras capturan hasta 60 fotogramas por segundo, proporcionando una percepción visual continua. Los datos adquiridos se procesan mediante algoritmos de visión por computadora, incluyendo detección de bordes, segmentación de imágenes y redes neuronales convolucionales (CNNs), que permiten identificar obstáculos y estimar la trayectoria del vehículo.

Uno de los principales desafíos en el uso de cámaras es la dependencia de las

Figura 1.16.*Cámaras en la navegación autónoma*

Nota: Tomado de Kurtulus [23].

condiciones de iluminación. En entornos con poca luz, la precisión de la detección disminuye significativamente, lo que puede afectar la navegación del vehículo [24]. Para mitigar este problema, se pueden emplear técnicas de aumento de imágenes, filtros adaptativos y estrategias de fusión sensorial con el LiDAR, permitiendo una percepción más robusta y confiable.

Comparado con el LiDAR, el uso de cámaras en F1Tenth tiene la ventaja de proporcionar información detallada sobre el color y la textura del entorno, pero presenta desventajas en términos de profundidad y sensibilidad a cambios en la iluminación. En aplicaciones de conducción autónoma, la combinación de sensores, conocida como fusión sensorial, permite mejorar la capacidad del vehículo para interpretar el entorno con mayor precisión y fiabilidad.

1.4.4 Algoritmos de navegación y control

Simultaneous Localization and Mapping (SLAM)

El algoritmo **Simultaneous Localization and Mapping (SLAM)** permite que un vehículo construya un mapa de su entorno mientras simultáneamente determina su propia posición dentro de él. Este proceso es esencial en entornos desconocidos o dinámicos, donde el sistema debe actualizar su percepción en tiempo real para adaptarse a cambios y evitar colisiones.

En la plataforma F1Tenth, SLAM se implementa a partir de los datos generados por sensores como el LiDAR y las cámaras RGB, combinando estos datos mediante técnicas de fusión sensorial para mejorar la precisión del mapeo. Entre los enfoques más utilizados se encuentran GMapping, Hector SLAM, Cartographer y ORB-SLAM, los cuales emplean diferentes estrategias para la estimación de la posición y la generación del mapa.

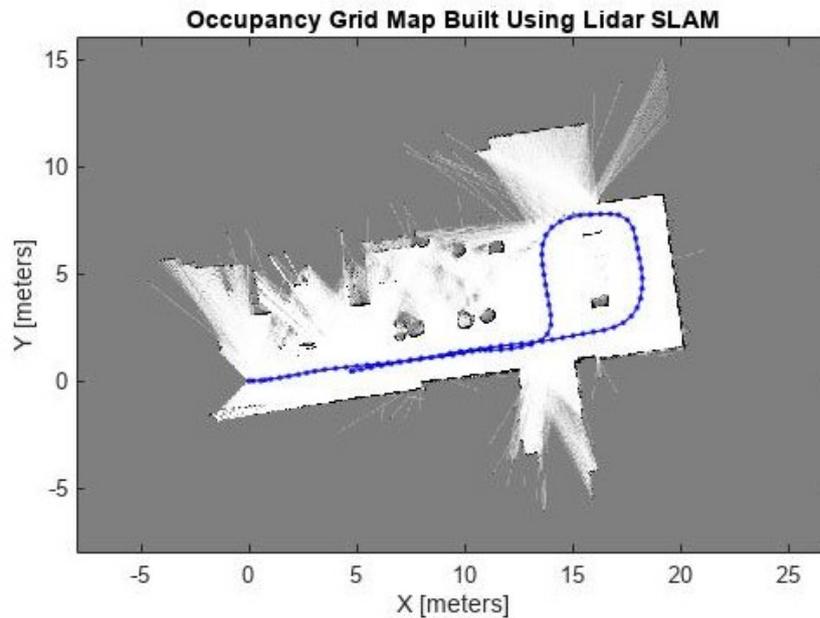
El proceso de SLAM en F1Tenth implica dos tareas principales: la estimación de la posición, donde se emplean algoritmos basados en filtros de partículas o filtros de Kalman extendidos (EKF) para actualizar la ubicación del vehículo en función de las mediciones del sensor y el modelo de movimiento, y la construcción del mapa, que genera una representación del entorno mediante técnicas como *occupancy grids* (mapas de ocupación) o nubes de puntos, permitiendo identificar obstáculos y zonas transitables.

La implementación de SLAM en F1Tenth permite una navegación autónoma más eficiente, ya que el vehículo puede planificar su ruta con base en un mapa generado en tiempo real. Sin embargo, su desempeño está influenciado por factores como la calidad de los datos sensoriales, la latencia del procesamiento y la acumulación de errores en la estimación de posición. Para mitigar estos problemas, se emplean estrategias como optimización de gráficos (Graph-SLAM) y corrección de deriva (loop closure detection), las

cuales mejoran la precisión y consistencia del mapa [16; 17].

Figura 1.17.

Implementación de SLAM en F1Tenth



Nota: Tomado de MathWorks [25].

Algoritmos de planificación de ruta (*Path Planning*)

La planificación de ruta es un componente esencial en la navegación autónoma, permitiendo que el vehículo determine la trayectoria más eficiente mientras evita obstáculos en su entorno. En F1Tenth, este proceso debe ser altamente optimizado para responder en tiempo real, ya que el vehículo opera en escenarios dinámicos y de alta velocidad [26; 27].

Para este propósito, F1Tenth emplea algoritmos de planificación como A* y Rapidly-exploring Random Trees (RRT). A* es un algoritmo de búsqueda informada basado en grafos, que garantiza encontrar la ruta óptima utilizando una heurística para explorar caminos con menor costo computacional [28]. Por otro lado, RRT es un enfoque basado en

muestreo aleatorio, ideal para entornos no estructurados, ya que permite explorar el espacio de búsqueda de manera eficiente sin necesidad de una representación completa del entorno [29]. Ambos algoritmos tienen ventajas y desventajas: A* es óptimo pero puede volverse computacionalmente costoso en espacios grandes, mientras que RRT es más rápido pero no siempre garantiza la mejor solución.

Para mejorar el desempeño en F1Tenth, estos algoritmos se combinan con técnicas de optimización de trayectoria, como RRT (versión optimizada de RRT que garantiza trayectorias más cortas y suaves) [30], y algoritmos basados en gradiente, los cuales refinan la ruta generada para minimizar cambios bruscos de dirección y reducir el tiempo de recorrido. Además, se integran con sistemas de percepción como el LiDAR y cámaras para ajustar dinámicamente la trayectoria en función de nuevos obstáculos detectados en el entorno [31].

Dado que la planificación de ruta en F1Tenth debe ejecutarse en milisegundos, se implementan estrategias de paralelización y procesamiento en hardware optimizado, asegurando una navegación segura y eficiente incluso en escenarios competitivos. Este balance entre precisión y velocidad es clave para mantener un rendimiento óptimo en carreras y evitar colisiones en entornos dinámicos [24].

Control de movimiento (*Motion Control*)

El control de movimiento es un componente fundamental en la navegación autónoma, encargado de regular la dirección y velocidad del vehículo en función de la ruta planificada. En F1Tenth, este proceso debe ser altamente preciso y adaptativo, ya que el vehículo opera en entornos de alta velocidad donde pequeñas desviaciones pueden afectar significativamente su desempeño [33; 34].

deseado como la magnitud de las acciones de control:

$$\min_{\mathbf{u}} \sum_{k=0}^N [(\mathbf{x}_k - \mathbf{x}_{\text{ref}})^\top Q (\mathbf{x}_k - \mathbf{x}_{\text{ref}}) + \mathbf{u}_k^\top R \mathbf{u}_k] \quad (1.1)$$

Sujeto a las restricciones dinámicas del sistema:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_k \in \mathcal{X}, \quad \mathbf{u}_k \in \mathcal{U} \quad (1.2)$$

donde \mathbf{x}_k representa el estado del vehículo en el instante k , \mathbf{u}_k es la acción de control, Q y R son matrices de peso que definen la penalización sobre el estado y la acción de control respectivamente, y $f(\mathbf{x}_k, \mathbf{u}_k)$ describe la evolución del sistema.

Dado que la planificación del control en F1Tenth debe ejecutarse en milisegundos, se emplean métodos de optimización convexa y reducción de dimensionalidad para garantizar una solución eficiente en tiempo real [38]. Este balance entre precisión y velocidad permite mejorar la estabilidad del vehículo y su capacidad de reacción en entornos competitivos. En conjunto, la combinación de estos controladores permite que el vehículo mantenga un equilibrio entre estabilidad, precisión y velocidad en diversas condiciones operativas [21; 24].

1.4.5 Evaluación de rendimiento

Precisión y exactitud

La precisión en las plataformas de navegación es de máxima importancia, pues los errores mínimos pueden afectar la seguridad y el desempeño del vehículo. En el caso de F1Tenth, la precisión se determina observando si el vehículo mantenía su trayectoria y evitaba los obstáculos. Estos factores son especialmente importantes en los modos de alta velocidad, cuando una pequeña desviación puede causar un choque o una reducción del desempeño [16; 17].

Tiempo de respuesta

Los algoritmos deben estar en condiciones de responder en milisegundos para adaptarse a los cambios en el entorno. Esto es crucial en el contexto de F1Tenth, donde la rapidez es vital, ya que permite al vehículo reajustarse inmediatamente en caso de que aparezcan obstáculos o esté interactuando con otros vehículos. La rapidez es especialmente importante en las competiciones de F1Tenth, donde se requieren decisiones al milisegundo [16; 24].

Consumo de recursos computacionales

Dado que F1Tenth utiliza un microprocesador de bajo requerimiento, como el Jetson TX2, es esencial que los algoritmos sean lo suficientemente eficientes como para ejecutarse en tiempo real, utilizando la menor cantidad de recursos computacionales. Este aspecto influye en la capacidad del vehículo para llevar a cabo cálculos simultáneos y permite que este tome decisiones precisas y rápidas.

Capítulo 2

2. Metodología

La metodología adoptada para este proyecto combina estrategias prácticas y simulaciones virtuales, con el objetivo de diseñar y evaluar un sistema de navegación autónoma robusto y seguro para el robot F1Tenth. Este enfoque busca garantizar que el vehículo complete vueltas en el circuito de competencia sin colisionar, maximizando su desempeño en términos de velocidad y precisión, al tiempo que satisface requerimientos específicos del cliente y las restricciones impuestas por las reglas de la competencia [39].

Es importante señalar que, además de las consideraciones propias de la plataforma F1Tenth, las preferencias específicas del cliente influyen significativamente en las decisiones de diseño. Estas preferencias pueden sesgar tanto la selección de componentes como la implementación de ciertas estrategias de control, lo que añade una capa adicional de complejidad al proyecto. Por ejemplo, pueden existir requisitos relacionados con la modularidad del sistema, el enfoque en la seguridad o la necesidad de integrar características adicionales que no son estrictamente necesarias para la competencia, pero que resultan esenciales para cumplir con las expectativas del cliente [40].

El marco técnico del proyecto sigue las directrices de F1Tenth y aprovecha un enfoque metodológico modular. Así mismo, en el ámbito del software, se utilizará ROS2 Humble como la plataforma principal de desarrollo, junto con herramientas como Gazebo y RViz para realizar simulaciones exhaustivas y pruebas preliminares de los algoritmos en entornos controlados [41].

Este enfoque modular no solo permite avanzar simultáneamente en diferentes aspectos del proyecto, reduciendo tiempos de desarrollo, sino que también facilita la integración y el ajuste fino del sistema una vez que cada módulo esté operativo. La validación de los algoritmos de control será realizada en simulaciones. Este proceso asegura un sistema optimizado y

funcional desde las primeras etapas del desarrollo [42].

Finalmente, la evaluación del desempeño se realizará mediante métricas específicas que incluyen el número de vueltas completadas sin colisiones y métricas de seguimiento de objetos. Estas métricas proporcionarán una visión integral del sistema, evaluando tanto su capacidad para navegar eficientemente como su precisión en la detección y evasión de obstáculos. Este enfoque metodológico asegura un desarrollo estructurado, adaptable y alineado con los objetivos de la competencia F1Tenth y las necesidades específicas del cliente.

2.1 Formulación de alternativas de solución

Dentro del contexto de la competencia F1Tenth, las normativas establecen una serie de componentes estandarizados para garantizar cierta homogeneidad en los vehículos participantes [39]. Elementos como el chasis del vehículo, la plataforma de procesamiento y el controlador de motores son seleccionados previamente y no son objeto de modificación en este proyecto. Esta estandarización permite a los equipos concentrarse en aspectos de diseño e implementación que representan la verdadera diferenciación en el desempeño del sistema, como los algoritmos de control y las estrategias de percepción del entorno. Por esta razón, dichos componentes no se incluyeron como alternativas de diseño, ya que forman parte del marco obligatorio de la competencia y representan un punto de partida común para todos los participantes.

Sin embargo, existen áreas clave en las que se pueden plantear alternativas de solución que impactan directamente en la capacidad del vehículo para navegar sin colisiones en el circuito. Una de estas áreas es la selección del sensor principal, responsable de proporcionar información sobre el entorno, considerando opciones como sensores de rango,

cámaras estéreo o combinaciones de ambos [41; 40]. Otra área crítica es el enfoque del algoritmo de navegación, que puede basarse en estrategias reactivas, con reglas inmediatas para evitar obstáculos [43], o en enfoques predictivos que consideran el estado futuro del vehículo y los elementos del entorno [44]. Asimismo, se pueden explorar diferentes estrategias para interpretar el entorno, como el uso de filtros predictivos para estimar el movimiento de obstáculos y competidores [45], o técnicas más avanzadas, como redes neuronales para predecir comportamientos complejos en tiempo real [46; 42]. Estas alternativas proporcionan una base para desarrollar configuraciones innovadoras y optimizadas para las exigencias de la competencia y de nuestro cliente.

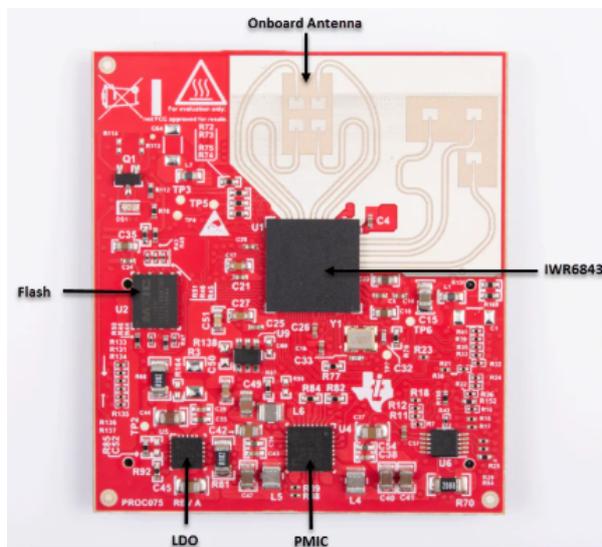
2.1.1 Alternativas de diseño según el sensor

La elección del sensor principal es un factor crítico en el desarrollo del sistema de navegación autónoma, ya que determina cómo el robot percibe su entorno y, por ende, cómo toma decisiones en tiempo real. A continuación, se describen las principales alternativas evaluadas en términos de sensores, considerando sus características, ventajas y limitaciones:

1. **Radar mmWave Texas Instruments IWR6843ISK:** Este sensor utiliza tecnología de ondas milimétricas para detectar objetos y medir su velocidad, distancia y dirección con alta precisión, incluso en condiciones adversas como poca luz o niebla [47]. Su principal ventaja es su capacidad para operar de manera robusta en entornos dinámicos y su resistencia a interferencias ambientales. Sin embargo, tiene una resolución espacial más baja en comparación con otros sensores, lo que puede dificultar la identificación detallada de objetos cercanos. Además, su integración requiere un procesamiento más intensivo debido a la naturaleza compleja de los datos que genera.

Figura 2.1.

Radar mmWave WR6843ISK



Nota: Tomado de Texas Instruments [48].

2. **RPLidar S2:** Este sensor emite pulsos láser para medir distancias con alta precisión, generando un mapa en 2D del entorno en tiempo real [49]. Su principal ventaja es su precisión y fiabilidad en la detección de obstáculos, incluso en entornos estáticos y controlados. No obstante, el LIDAR puede verse limitado en condiciones ambientales adversas, como lluvia intensa o polvo, y su capacidad para detectar objetos en movimiento rápido puede ser menos efectiva en comparación con otras tecnologías.

Figura 2.2.

RPLidar S2



Nota: Tomado de SLAMTEC [50].

- 3. Cámara RGB Intel RealSense:** Las cámaras RGB permiten capturar imágenes visuales del entorno, lo que facilita la implementación de algoritmos de visión por computadora y redes neuronales convolucionales (CNN) para la detección de obstáculos, líneas de carril y otros vehículos [51]. Su principal ventaja radica en su capacidad para interpretar características visuales detalladas del entorno. Sin embargo, estas cámaras presentan limitaciones significativas en condiciones de poca iluminación o ambientes con reflejos excesivos. Además, la interpretación de datos visuales requiere una mayor capacidad computacional para procesar imágenes en tiempo real.

Figura 2.3.

Cámara RGB Intel RealSense



Nota: Tomado de Intel [52].

4. **Combinación de Cámara RGB Intel RealSense y RPLidar S2:** Esta alternativa combina las fortalezas de ambos sensores: la capacidad de interpretación visual detallada de la cámara RGB y la precisión del mapeo por distancias del LIDAR. La integración de ambos sensores permite una percepción más robusta y completa del entorno, lo que mejora la capacidad del sistema para manejar escenarios complejos [53]. Sin embargo, este enfoque incrementa la complejidad del diseño, ya que requiere la fusión de datos en tiempo real para generar un modelo coherente del entorno. Esto implica una mayor carga computacional y una necesidad de sincronización precisa entre ambos sensores.

2.1.2 Alternativas de diseño para algoritmos de navegación

La selección del algoritmo de navegación es fundamental para garantizar que el robot autónomo pueda desplazarse eficientemente por el circuito F1Tenth, evitando colisiones y maximizando el desempeño en términos de velocidad y seguridad. A continuación, se describe las principales alternativas de algoritmos evaluadas, junto con sus ventajas y desventajas:

1. **Pure Pursuit:** Este algoritmo de navegación utiliza puntos de control en la trayectoria para calcular curvas suaves que el vehículo debe seguir [54]. Su principal ventaja es su simplicidad y eficiencia computacional, lo que lo hace adecuado para entornos de tiempo real. Sin embargo, puede tener dificultades para manejar trayectorias complejas o cambios rápidos en el entorno, lo que podría resultar en una navegación menos precisa en circuitos dinámicos.
2. **Control predictivo basado en modelo (MPC):** El MPC utiliza un modelo matemático del sistema para predecir su comportamiento futuro y calcular el control óptimo para minimizar un costo definido [55]. Su principal ventaja es su capacidad para manejar restricciones del sistema y prever obstáculos antes de que se conviertan en un problema crítico. Sin embargo, su implementación requiere un alto poder de cómputo, lo que podría limitar su aplicación en hardware de recursos limitados como la Jetson Nano.
3. **Follow the Gap:** Este algoritmo identifica espacios libres en el entorno del vehículo y selecciona la mejor trayectoria hacia el objetivo a través de ellos [56]. Su ventaja es su capacidad para adaptarse dinámicamente a obstáculos no previstos, lo que lo hace ideal para entornos con alta variabilidad. No obstante, su enfoque reactivo puede resultar en trayectorias subóptimas o movimientos erráticos si los obstáculos cambian rápidamente.
4. **Controlador PID:** El controlador Proporcional-Integral-Derivativo (PID) es una solución clásica para el control de sistemas [57]. Es fácil de implementar y ajustar, y es eficiente en términos computacionales. Sin embargo, tiene limitaciones para manejar escenarios dinámicos y complejos, ya que no considera predicciones futuras del entorno ni restricciones avanzadas del sistema. Otra desventaja de este esquema de control es la dificultad del ajuste de los parámetros del mismo, siendo estos fundamentales para la estabilidad del sistema.

5. **Controlador LQR (Linear Quadratic Regulator):** El LQR es un método óptimo para sistemas lineales que minimiza una función de costo cuadrática, proporcionando trayectorias suaves y estables [58]. Su ventaja principal es su precisión en sistemas bien modelados. Sin embargo, su aplicación es más limitada en sistemas no lineales o en entornos donde los modelos del sistema y del entorno son inciertos o cambiantes.
6. **MPC con Algoritmos de Predicción Adicionales:** Esta alternativa combina el enfoque predictivo del MPC con algoritmos adicionales, como filtros de Kalman o redes neuronales, para predecir el movimiento de obstáculos y otros competidores [59]. Esto mejora significativamente la capacidad del sistema para tomar decisiones informadas en entornos complejos y dinámicos. Sin embargo, esta combinación aumenta considerablemente la complejidad computacional y el tiempo de desarrollo, lo que podría requerir hardware más avanzado y estrategias de optimización adicionales.

Estas alternativas ofrecen un rango diverso de enfoques que equilibran simplicidad, capacidad de predicción y complejidad computacional.

2.1.3 Alternativas para la evaluación del entorno

La evaluación del entorno es un componente esencial para garantizar que el robot autónomo pueda tomar decisiones informadas en tiempo real. Este proceso implica interpretar los datos de los sensores para identificar obstáculos, competidores y características del entorno que puedan afectar la navegación. A continuación, se analizan las principales alternativas evaluadas para este propósito, junto con sus ventajas y desventajas:

1. **SLAM con global planner offline y local planner online (Opción seleccionada):**
Este enfoque utiliza un algoritmo de Simultaneous Localization and Mapping (SLAM) para generar un mapa del entorno en tiempo real, mientras que la planificación global

se realiza de manera offline, permitiendo que el vehículo cuente con una ruta predefinida que optimiza la navegación [60]. La planificación local se ejecuta online, lo que permite ajustar dinámicamente la trayectoria en función de obstáculos detectados en tiempo real. La principal ventaja de esta estrategia es su capacidad para operar en entornos previamente desconocidos y adaptar la ruta en caso de cambios en el entorno.

2. **Filtro de kalman extendido (EKF) para localización con mapas predefinidos:** Este enfoque utiliza un Filtro de Kalman Extendido (EKF) para estimar la posición del vehículo basándose en mediciones ruidosas de los sensores [61]. Se emplea en combinación con un mapa predefinido del entorno, lo que reduce la carga computacional en comparación con SLAM. Su ventaja principal es que proporciona estimaciones precisas de localización con un menor costo computacional. No obstante, su principal limitación es que no permite construir un mapa dinámico, por lo que su uso se restringe a escenarios donde el mapa es estático y previamente conocido.
3. **Deep reinforcement learning para exploración autónoma:** Este enfoque utiliza Redes Neuronales Profundas (Deep RL) para que el robot aprenda a explorar su entorno y tomar decisiones de navegación basadas en recompensas [62]. Su ventaja principal es su capacidad de adaptación a escenarios no estructurados y su potencial para desarrollar estrategias de navegación óptimas sin una modelación explícita del entorno. Sin embargo, este método requiere una gran cantidad de datos de entrenamiento y recursos computacionales elevados, lo que puede dificultar su implementación en tiempo real.
4. **Segmentación basada en visión artificial para la percepción del entorno:** En esta alternativa, se utiliza una cámara RGB en combinación con redes neuronales convolucionales (CNNs) para realizar la segmentación de obstáculos y elementos del

entorno [63]. Esta técnica permite identificar y clasificar diferentes tipos de objetos en la escena, lo que facilita la toma de decisiones en la navegación. Su ventaja es que puede proporcionar información detallada sobre el tipo de obstáculos presentes, pero presenta desafíos en condiciones de iluminación variable y procesamiento en tiempo real, además de ser altamente dependiente de los datos de entrenamiento.

5. **Método de *grid-Based mapping* con sensores de rango:** Este enfoque divide el entorno en una malla de celdas (occupancy grid), donde cada celda representa la probabilidad de estar ocupada o libre [64]. Se basa en sensores de rango como LiDAR o ultrasonidos y es una alternativa eficiente para entornos estáticos. Su principal ventaja es su baja demanda computacional y su simplicidad en la implementación. Sin embargo, su precisión depende del tamaño de las celdas y la resolución del sensor, lo que puede llevar a errores en entornos con alta densidad de obstáculos.
6. ***Follow-the-gap* para evasión de obstáculos en tiempo real:** Este método analiza los datos del LiDAR para identificar el espacio libre más grande (gap) y dirigir el vehículo hacia esa zona [56]. Su principal ventaja es que permite evitar obstáculos dinámicos sin necesidad de construir un mapa completo. Sin embargo, su enfoque reactivo puede generar trayectorias subóptimas, lo que puede comprometer la eficiencia de la navegación en entornos complejos.

Cada una de estas alternativas ofrece diferentes enfoques para la interpretación del entorno, que varían en términos de simplicidad, capacidad predictiva y demanda computacional.

2.1.4 Evaluación multicriterio de las alternativas

La selección de los sistemas más adecuados para el robot autónomo se realizó mediante una evaluación multicriterio, considerando factores técnicos, de implementación y de desempeño en el entorno de la competencia. A continuación, se presenta la evaluación para cada subsistema: sensores, algoritmos de navegación y métodos de predicción.

Selección de sensores

La elección del sensor principal o de la combinación de sensores es crucial para garantizar la percepción eficiente del entorno. Se evaluaron las alternativas considerando criterios como precisión en la detección, simplicidad de implementación, costo computacional y adaptabilidad a entornos dinámicos. La Tabla 2.1 muestra la matriz IPOS utilizada para esta evaluación.

Tabla 2.1.

Matriz IPOS para la selección de sensores.

Criterio	Peso (%)	Radar mmWave	RPLidar S2	Fusión LiDAR-Cámara
Precisión en detección	30	6.0	9.0	10.0
Simplicidad de implementación	25	7.0	10.0	6.0
Costo computacional	20	8.0	9.0	5.0
Adaptabilidad a entornos dinámicos	25	8.0	9.0	8.0
Puntaje Total	100	7.3	9.1	7.9

De acuerdo con los resultados de la matriz IPOS, las alternativas seleccionadas son el sensor RPLidar S2 (puntaje: 9.1/10) y la combinación de LiDAR con cámara RGB (puntaje: 7.9/10). Estas opciones destacan por su alta precisión en la detección y su adaptabilidad a entornos dinámicos, características esenciales para el éxito del sistema de navegación

autónoma.

Selección de algoritmos de navegación

La elección del algoritmo de navegación es fundamental para garantizar que el robot pueda desplazarse de manera eficiente y segura en el circuito, evitando colisiones y optimizando su desempeño. Los algoritmos se evaluaron considerando criterios como precisión en navegación, capacidad de adaptabilidad, simplicidad de implementación y costo computacional. Debido a la cantidad de algoritmos analizados, la evaluación se presenta en dos tablas (Tablas 2.2 y 2.3).

Tabla 2.2.

Matriz IPOS para algoritmos de navegación (parte 1).

Criterio	Peso (%)	Pure Pursuit	PID	LQR
Precisión en navegación	30	7.0	6.0	8.0
Adaptabilidad a entornos dinámicos	25	6.0	5.0	7.0
Simplicidad de implementación	20	9.0	10.0	7.0
Costo computacional	25	8.0	9.0	7.0
Puntaje Total	100	7.3	7.1	7.3

De acuerdo con los resultados de las matrices IPOS, los algoritmos seleccionados son el Control Predictivo basado en Modelo (MPC), con el segundo puntaje más alto (8.1/10), siendo superado solo por el algoritmo Follow the Gap, con un puntaje de 8.3/10. Si bien el MPC con Predicción tiene la nota más elevada a nivel de precisión y adaptabilidad, su alto costo computacional y su dificultad de implementación lo dejaron fuera de discusión. Estas opciones se destacaron por su precisión en la navegación y su adaptabilidad a entornos dinámicos, características esenciales para el éxito en la competencia.

Tabla 2.3.

Matriz IPOS para algoritmos de navegación (parte 2).

Criterio	Peso (%)	Follow the Gap	MPC	MPC con Predicción
Precisión en navegación	30	9.0	9.0	10.0
Adaptabilidad a entornos dinámicos	25	9.0	8.0	10.0
Simplicidad de implementación	20	8.0	8.0	2.0
Costo computacional	25	7.0	7.0	2.0
Puntaje Total	100	8.3	8.1	6.4

Selección del algoritmo de evaluación del entorno

La evaluación del entorno es un componente clave para garantizar que el robot pueda interpretar correctamente los datos del sensor y tomar decisiones informadas en tiempo real. Se consideraron cuatro alternativas para este propósito, evaluadas en función de criterios como precisión en predicción, simplicidad de implementación, costo computacional y capacidad de aprendizaje. La Tabla 2.4 muestra los resultados de la evaluación.

Tabla 2.4.

Matriz IPOS para algoritmos de evaluación del entorno.

Criterio	Peso (%)	Sin Algoritmo	SLAM	Filtro de Kalman	Deep RL
Precisión	30	5.0	9.0	9.0	10.0
Simplicidad de implementación	25	10.0	10.0	7.0	4.0
Costo computacional	20	10.0	8.0	7.0	3.0
Adaptabilidad	25	3.0	8.0	5.0	10.0
Puntaje Total	100	6.6	8.8	7.1	6.9

De acuerdo con los resultados de la matriz IPOS, el algoritmo SLAM obtuvo el puntaje más alto (8.8/10) y fue seleccionado como la solución principal para la evaluación del

entorno. Esta elección se basó principalmente en su alta precisión y capacidad de adaptabilidad, lo que lo hace ideal para entornos dinámicos y no estructurados. A diferencia de los otros enfoques evaluados, SLAM permite que el robot construya un mapa del entorno y se localice simultáneamente en él, lo que mejora la robustez del sistema frente a cambios en el entorno y reduce la dependencia de mapas predefinidos.

Además de su puntuación en la evaluación cuantitativa, la selección de SLAM también se fundamentó en consideraciones prácticas. El equipo de desarrollo posee experiencia en la implementación de algoritmos de mapeo y localización, lo que facilita su integración y optimización dentro del sistema. Asimismo, la naturaleza adaptable de SLAM lo hace más adecuado para escenarios donde las condiciones pueden variar significativamente, asegurando una mayor precisión en la planificación de rutas y en la detección de obstáculos. Estas ventajas, combinadas con el análisis cuantitativo, justifican su elección como la mejor opción para la evaluación del entorno en este proyecto.

Análisis de combinaciones

Para evaluar las combinaciones de alternativas seleccionadas, se consideraron tres configuraciones principales que integran sensores, algoritmos de navegación y métodos de evaluación del entorno:

- **Combinación 1:** Lidar y cámara con follow the gap y sin algoritmo adicional.
- **Combinación 2:** Lidar con MPC y SLAM.
- **Combinación 3:** Lidar y cámara con MPC y filtro de kalman.

Estas combinaciones fueron analizadas con base en seis criterios clave, como se muestra en la Tabla 2.5.

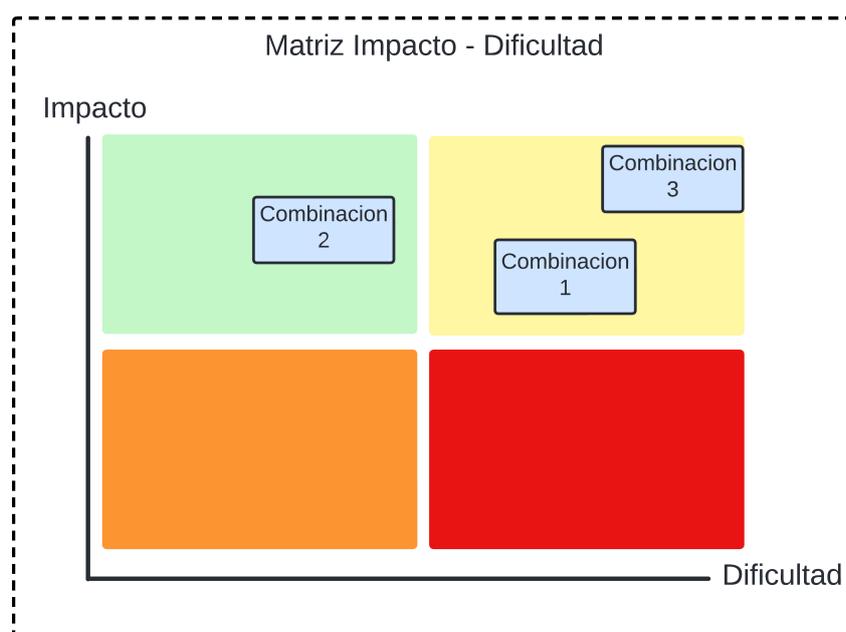
Tabla 2.5.

Matriz IPOS para combinaciones de alternativas.

Criterio	Peso (%)	Combinación 1	Combinación 2	Combinación 3
Precisión en navegación	25	7.00	9.00	8.00
Adaptabilidad a entornos dinámicos	20	8.00	10.00	9.00
Complejidad computacional	15	9.00	7.00	6.00
Integración entre subsistemas	15	8.00	9.00	8.00
Robustez ante condiciones adversas	15	6.00	9.00	8.00
Escalabilidad para futuros desarrollos	10	7.00	8.00	7.00
Puntaje Total	100	7.65	8.85	7.95

Figura 2.4.

Matriz Impacto vs. Dificultad



Además, se evaluaron estas alternativas con una matriz de impacto-dificultad.

La combinación seleccionada como la mejor opción fue la **Combinación 2 (Lidar con MPC y SLAM)**, obteniendo un puntaje total de 8.85/10. Esta configuración destacó significativamente en criterios clave como precisión en navegación, adaptabilidad a entornos dinámicos e integración efectiva entre subsistemas, consolidándola como la alternativa más

robusta y eficiente para cumplir con los objetivos del proyecto. Su sobresaliente desempeño en condiciones adversas y su capacidad de escalabilidad hacia futuros desarrollos refuerzan su idoneidad como solución integral y sostenible para el sistema autónomo.

Aunque las combinaciones que incluyen tanto Lidar como cámara ofrecen una mayor robustez en términos de percepción del entorno, su implementación incrementa considerablemente la complejidad de integración y la carga computacional, lo que afecta su escalabilidad. Este último aspecto es especialmente importante, ya que el cliente priorizó soluciones que puedan extenderse a entornos más complejos, diferentes tipos de vehículos y otras aplicaciones futuras. La **combinación 2** logra un equilibrio ideal entre desempeño técnico, simplicidad de integración y escalabilidad, respondiendo tanto a los requisitos técnicos del sistema como a las preferencias del cliente.

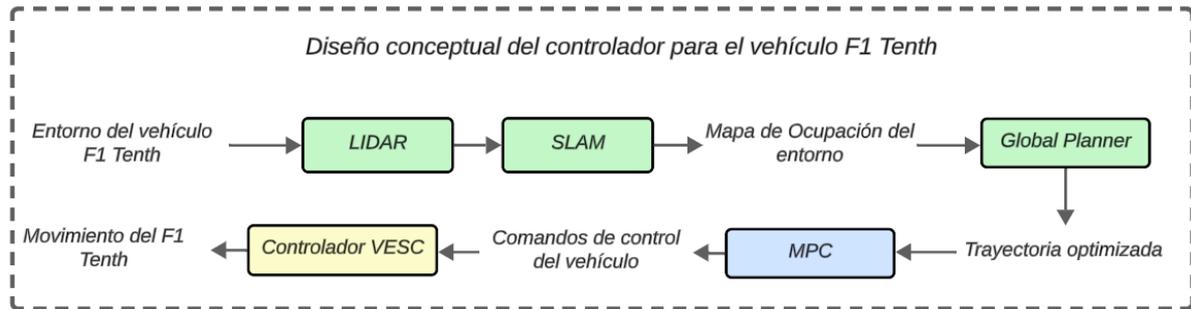
2.2 Diseño conceptual

El diseño conceptual del vehículo F1Tenth se estructuró en tres subsistemas principales que interactúan de forma modular, asegurando la navegación autónoma del vehículo. Cada subsistema está representado por un color en el diagrama conceptual (Figura 2.5) y se compone de los elementos seleccionados en las etapas previas del análisis:

- **Subsistema de percepción (verde)**: Este subsistema incluye el sensor seleccionado, el LiDAR Hokuyo UST-10LX. Este componente proporciona la base de datos necesaria para la evaluación del entorno.
- **Subsistema de procesamiento (azul)**: Este subsistema está a cargo de la toma de decisiones, procesando los datos obtenidos por el subsistema de percepción.
- **Subsistema de actuación (amarillo)**: Este subsistema es responsable de ejecutar las decisiones tomadas en el subsistema de procesamiento.

Figura 2.5.

Diseño conceptual del controlador del vehículo F1Tenth.



2.2.1 Subsistema de percepción

El subsistema de percepción es el encargado de capturar datos del entorno y proporcionar información confiable al sistema de procesamiento para la toma de decisiones. Se seleccionó el sensor **LiDAR Hokuyo UST-10LX**, debido a su capacidad para generar mapas bidimensionales de alta precisión y detectar obstáculos en un rango de hasta 10 metros. Este sensor destaca por su alta resolución y frecuencia de escaneo, lo que lo convierte en una solución ideal para entornos dinámicos y de alta velocidad como los de la competencia F1Tenth.

- **Resolución angular:** 0.25° , lo que permite obtener un nivel de detalle fino en la percepción del entorno.
- **Frecuencia de escaneo:** 40 Hz, garantizando una actualización constante de la información del entorno.
- **Precisión:** ± 30 mm, asegurando lecturas fiables incluso a altas velocidades.
- **Rango de detección:** Hasta 10 metros, adecuado para anticipar obstáculos en trayectorias de alta velocidad.

- **Resistencia ambiental:** Operación confiable en condiciones de iluminación variables y resistencia a partículas de polvo y agua (IP65).

El sensor LIDAR envía la información detectada al algoritmo de SLAM, el cual genera un mapa de ocupación del entorno del vehículo, el cual es utilizado por un algoritmo de optimización de trayectorias para crear el camino global.

2.2.2 *Subsistema de procesamiento*

El subsistema de procesamiento es el núcleo del sistema autónomo, encargado de interpretar los datos de percepción, tomar decisiones y planificar la trayectoria del vehículo. Para este propósito, se utilizó una placa integrada **Jetson Nano**, una plataforma de procesamiento de alto rendimiento y bajo consumo energético, diseñada específicamente para aplicaciones de inteligencia artificial y robótica.

Figura 2.6.

Placa integrada Jetson Nano



Nota: Tomado de NVIDIA [65].

- **Hardware:**
 - Procesador ARM Cortex-A57 con cuatro núcleos, optimizado para operaciones

multitarea.

- GPU Maxwell con 128 núcleos CUDA, ideal para ejecutar algoritmos intensivos en procesamiento como redes neuronales y simulaciones.
- Memoria RAM de 4 GB, suficiente para gestionar múltiples procesos simultáneamente.

- **Software:**

- Framework **ROS (Robot operating system)**: utilizado para la integración de los subsistemas, comunicación entre nodos y simulaciones.
- Algoritmos seleccionados:
 - * **MPC (Control Predictivo basado en Modelo)**: planifica trayectorias óptimas basadas en predicciones futuras del sistema.

- **Interfaz de simulación:** Se utilizarán herramientas como RViz y Gazebo para probar y validar los algoritmos antes de su implementación en el vehículo físico.

Este subsistema recibe la trayectoria global precomputada por el planificador global y las mediciones en tiempo real del LiDAR. A partir de esta información, genera los comandos de control óptimos que permiten al vehículo desplazarse desde su estado actual hasta el siguiente punto de la trayectoria planificada, ajustando dinámicamente su movimiento en función de los obstáculos detectados por el sensor.

2.2.3 Subsistema de actuación

El subsistema de actuación se encarga de ejecutar las decisiones tomadas por el subsistema de procesamiento, asegurando que el vehículo siga la trayectoria planificada de manera precisa y eficiente. Este subsistema incluye el control de tracción y dirección mediante el **ESC (Electronic Speed Controller)**, compatible con el protocolo VESC.

- **Protocolo de comunicación:** UART, permitiendo una comunicación rápida y fiable entre el controlador y la Jetson Nano.
- **Frecuencia de actualización:** 100 Hz, asegurando una respuesta rápida a los comandos de control.
- **Modos de operación:** Soporta modos de control de velocidad y posición, esenciales para la estabilidad del vehículo.
- **Capacidad de carga:** Compatible con motores de alto rendimiento utilizados en la plataforma F1Tenth.
- **Fiabilidad:** Diseñado para operar de manera estable bajo condiciones de alta demanda, como aceleraciones rápidas y maniobras bruscas.

Este subsistema garantiza una respuesta precisa y estable del vehículo, ejecutando las trayectorias calculadas con un margen mínimo de error. Además, su diseño modular permite futuras actualizaciones o adaptaciones a diferentes configuraciones de vehículos.

2.3 Diseño del controlador

El diseño del controlador para el vehículo autónomo F1Tenth se basa en un Control Predictivo basado en Modelos (MPC). Este enfoque permite ajustar dinámicamente la trayectoria del vehículo en tiempo real, optimizando el desempeño en entornos dinámicos y asegurando la estabilidad durante la navegación autónoma.

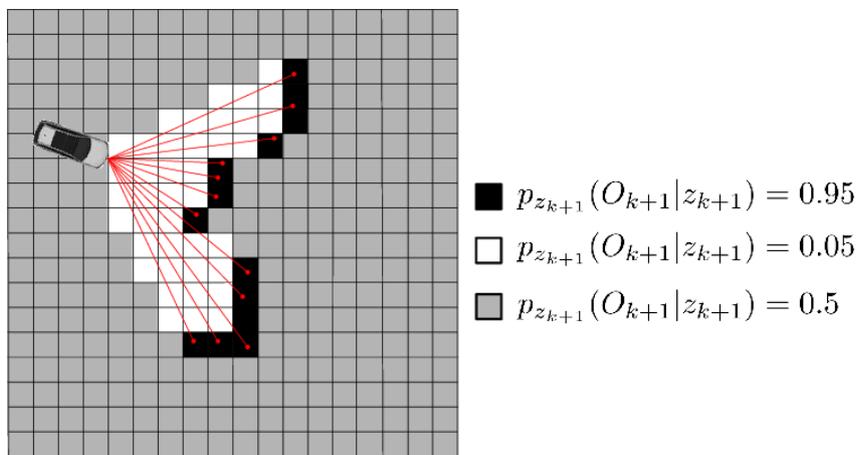
2.3.1 Generación del mapa de ocupación mediante SLAM

Para permitir la planificación óptima de trayectorias, es necesario construir un mapa de ocupación del entorno utilizando los datos obtenidos por el sensor LiDAR. Se emplea Simultaneous Localization and Mapping (SLAM) para estimar la posición del vehículo

mientras construye un mapa del circuito de carreras [60]. Este mapa se representa como una malla de celdas (occupancy grid map), donde cada celda almacena la probabilidad de estar ocupada o libre. En la Figura 2.7, se observa un mapa de ocupación, en el cual cada celda representa la probabilidad de estar ocupada (valores cercanos a 1) o libre (valores cercanos a 0). Los valores intermedios indican celdas con alta incertidumbre, lo que sugiere que estas regiones no han sido completamente detectadas por el sensor.

Figura 2.7.

Logica detras de un mapa de ocupación



Nota: Tomado de Nuss [66].

El problema de SLAM se define como la estimación de la trayectoria del vehículo $\mathbf{x}_{1:t}$ y el mapa m , dado un conjunto de mediciones del sensor $\mathbf{z}_{1:t}$ y controles $\mathbf{u}_{1:t}$. Matemáticamente, esto se expresa como:

$$p(\mathbf{x}_{1:t}, m \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(m \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t})p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad (2.1)$$

donde:

- $p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ representa la estimación de la trayectoria del vehículo.

- $p(m \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t})$ representa la probabilidad del mapa de ocupación basado en las mediciones del LiDAR.

El mapa de ocupación se modela utilizando una representación de probabilidad log-odds, donde el estado de cada celda m_i se actualiza de acuerdo con la ecuación:

$$L(m_i) = L(m_i \mid z_t, x_t) = L(m_i) + \log \left(\frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \right) \quad (2.2)$$

donde:

- $L(m_i)$ es el valor log-odds de la celda i .
- z_t es la medición del LiDAR en el instante t .
- x_t es la posición estimada del vehículo.

El SLAM se resuelve mediante algoritmos como GMapping o Cartographer, permitiendo construir mapas de las pistas de F1Tenth. Sin embargo, en esta tesis no se implementa un algoritmo de SLAM desde cero, ya que el diseño de un sistema completo de localización y mapeo representa un problema complejo en sí mismo, el cual escapa del alcance de este trabajo. En su lugar, se emplean librerías optimizadas como Slam_toolbox o Cartographer del TurtleBot3, las cuales facilitan la creación eficiente de mapas de ocupación a partir de datos del LiDAR. Este mapa generado se utilizará posteriormente para la planificación de trayectorias óptimas mediante el algoritmo D*.

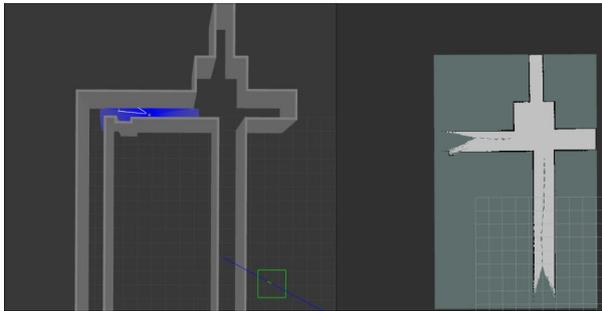
2.3.2 Generación de trayectorias con D* y suavización mediante splines cúbicos

Una vez generado el mapa de ocupación mediante SLAM, es necesario calcular una trayectoria óptima desde la posición inicial hasta la meta, asegurando una navegación eficiente y evitando colisiones. Para ello, se emplea el algoritmo D*, diseñado para adaptarse a cambios en el entorno de forma eficiente [67].

El algoritmo D* realiza una búsqueda basada en grafos sobre el mapa de ocupación,

Figura 2.8.

Ejemplo de mapeo con Slam_toolbox



En la imagen de la izquierda tenemos la simulación del mundo en gazebo, a la derecha tenemos el mapeo que realiza Slam_toolbox segun el robot va sensando el entorno

donde cada celda representa un nodo en el espacio de búsqueda. Se define un costo de trayectoria $g(n)$ para cada nodo n , el cual se actualiza de manera recursiva según:

$$g(n) = \min_{n' \in \mathcal{N}(n)} [g(n') + c(n', n)] \quad (2.3)$$

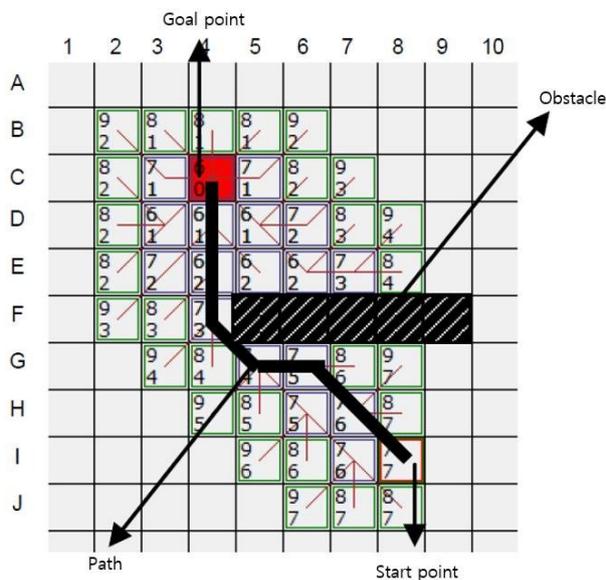
donde:

- $g(n)$ es el costo acumulado del nodo n .
- $\mathcal{N}(n)$ es el conjunto de nodos vecinos de n .
- $c(n', n)$ es el costo de transición entre nodos, basado en la distancia euclidiana y la

presencia de obstáculos.

Figura 2.9.

Optimización de trayectoria utilizando D^*



Nota: Tomado de Kim et al. [68].

Una vez aplicado el algoritmo D^* , se obtiene un mapa con los costos de trayectoria asociados a cada celda y la ruta óptima desde la posición inicial hasta el objetivo. La Figura 2.9 ilustra un ejemplo de este proceso, donde cada celda contiene tres valores que representan: (i) el costo acumulado $g(n)$, que indica la mejor estimación del costo desde la celda actual hasta la meta; y (ii) el costo heurístico $h(n)$, que aproxima la distancia restante hasta el objetivo. Además, las líneas diagonales dentro de cada celda muestran la dirección de propagación de los costos. La trayectoria óptima resultante se representa con una línea negra gruesa, la cual evita obstáculos y sigue la ruta de menor costo hasta el objetivo.

El algoritmo D^* genera una trayectoria óptima desde el punto de inicio hasta el objetivo, minimizando el costo total del desplazamiento. Sin embargo, esta trayectoria puede contener segmentos con ángulos pronunciados, lo que dificulta la estabilidad del control del vehículo.

Para reducir las variaciones abruptas en la trayectoria, se utiliza una interpolación con

Splines Cúbicos, asegurando continuidad en la primera y segunda derivada de la trayectoria.

La función de interpolación cúbica para un segmento entre dos puntos x_i y x_{i+1} se define como:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (2.4)$$

donde los coeficientes a_i, b_i, c_i, d_i se calculan imponiendo las siguientes condiciones:

1. Interpolación en los puntos de la trayectoria generada por D*:

$$S_i(x_i) = y_i, \quad S_i(x_{i+1}) = y_{i+1} \quad (2.5)$$

2. Continuidad en la primera derivada (suavidad de la dirección del movimiento):

$$S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}) \quad (2.6)$$

3. Continuidad en la segunda derivada (suavidad en la aceleración del vehículo):

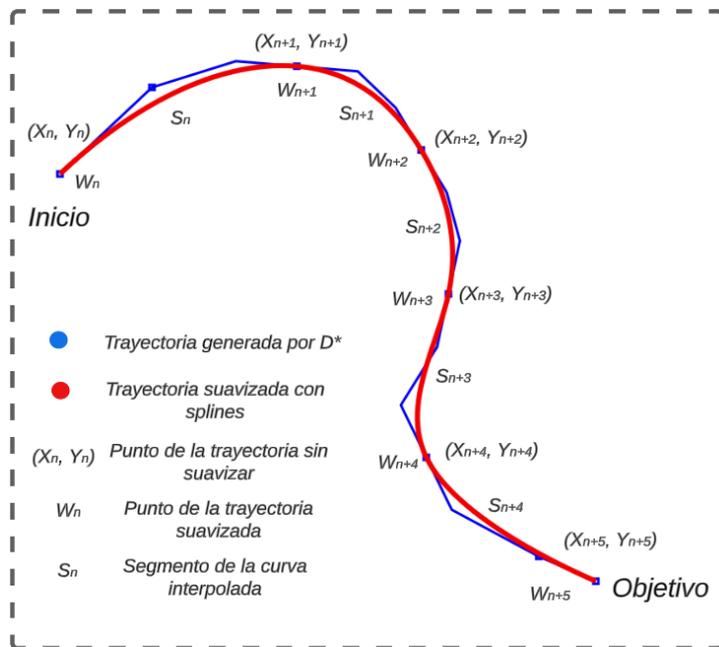
$$S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}) \quad (2.7)$$

El sistema de ecuaciones resultante se resuelve para obtener los coeficientes de los splines, generando así una trayectoria suave.

La Figura 2.10 ilustra el proceso de suavización de la trayectoria generada por el algoritmo D* mediante interpolación con Splines Cúbicos. En la imagen, la trayectoria original, representada en azul, presenta cambios abruptos en la dirección debido a la naturaleza discreta del algoritmo D*. Para corregir esto, se aplican splines cúbicos que garantizan continuidad en la primera y segunda derivada, resultando en una trayectoria suavizada en rojo. Cada punto (X_n, Y_n) corresponde a una posición en la trayectoria original, mientras que los puntos W_n representan los nodos generados por la interpolación, asegurando una transición suave entre segmentos S_n . Este proceso es fundamental para mejorar la estabilidad del vehículo y reducir oscilaciones no deseadas durante la navegación.

Figura 2.10.

Ejemplo de suavizado de trayectoria usando Splines Cúbicos



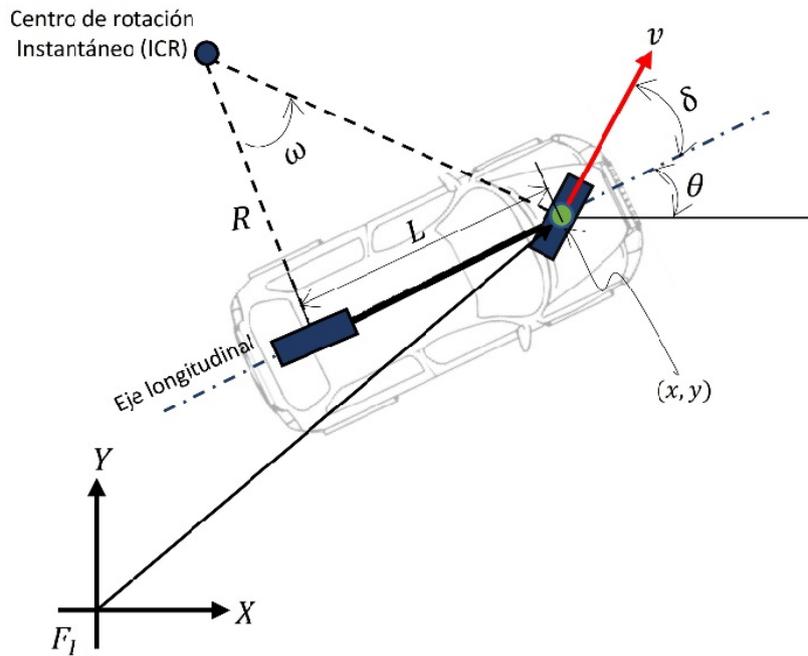
Una vez construida la trayectoria suavizada, es momento de diseñar el MPC que permitirá la navegación del vehículo F1 Tenth.

2.3.3 Modelo del vehículo para simulación

Para diseñar y evaluar el desempeño del controlador en un entorno simulado, se utiliza un modelo cinemático de bicicleta, el cual describe el movimiento del vehículo en función de su velocidad, orientación y ángulo de dirección. Este modelo es adecuado para simulaciones debido a su simplicidad computacional y su capacidad para capturar el comportamiento general del vehículo en condiciones normales de operación. Sin embargo, no considera efectos dinámicos como la transferencia de carga, el deslizamiento de los neumáticos o la fricción, los cuales serán abordados en la sección del modelo de implementación.

Figura 2.11.

Modelo cinemático de bicicleta



Nota: Tomado de Endo et al. [6].

Las ecuaciones que rigen la cinemática del vehículo en un plano bidimensional son:

$$\dot{x} = v \cos(\theta) \quad (2.8)$$

$$\dot{y} = v \sin(\theta) \quad (2.9)$$

$$\dot{\theta} = \frac{v}{L} \tan(\delta) \quad (2.10)$$

$$\dot{v} = a \quad (2.11)$$

Donde:

- (x, y) representan la posición del vehículo en el plano.
- θ es la orientación del vehículo respecto al eje global.

- v es la velocidad longitudinal del vehículo.
- δ es el ángulo de dirección de las ruedas frontales.
- L es la distancia entre ejes del vehículo.
- a es la aceleración longitudinal aplicada.

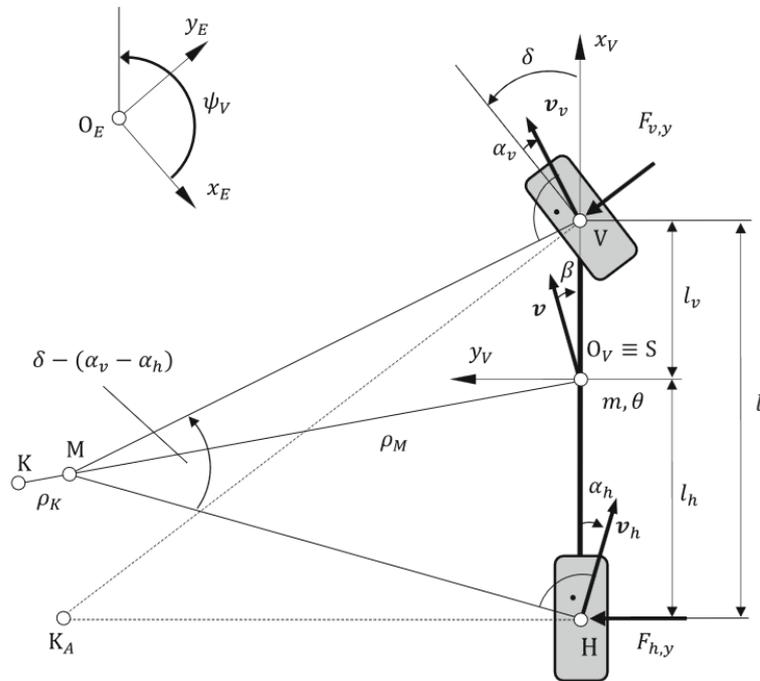
Este modelo es ampliamente utilizado en simulaciones de navegación autónoma, ya que permite predecir la trayectoria del vehículo con base en sus comandos de control a y δ . En la implementación real, se debe considerar un modelo dinámico más detallado que incorpore los efectos de fuerzas laterales, adherencia de los neumáticos y otras restricciones físicas del vehículo.

2.3.4 Modelo del vehículo para implementación

Para la implementación del controlador en el vehículo real, se utiliza un modelo dinámico de una sola pista (Single Track Dynamic Model), el cual extiende el modelo cinemático de bicicleta incorporando algunos efectos dinámicos clave. Este modelo captura mejor la respuesta real del vehículo al incluir fuerzas laterales y efectos de la inercia, lo que permite mejorar la precisión del control en condiciones reales.

Figura 2.12.

Single Track Dynamic Model



Nota: Tomado de Hosseini et al. [69].

Las ecuaciones de estado para este modelo son:

$$\dot{x}_V = v_h \cos(\theta) - v \sin(\theta) \quad (2.12)$$

$$\dot{y}_V = v_h \sin(\theta) + v \cos(\theta) \quad (2.13)$$

$$\dot{\theta} = \omega \quad (2.14)$$

$$\dot{v}_h = \frac{1}{m} (F_h - F_{v,y} \sin(\delta)) \quad (2.15)$$

$$\dot{v} = \frac{1}{m} (F_{v,y} \cos(\delta) + F_{h,y}) - \omega v_h \quad (2.16)$$

$$\dot{\omega} = \frac{1}{I_z} (l_v F_{v,y} \cos(\delta) - l_h F_{h,y}) \quad (2.17)$$

Donde:

- (x_V, y_V) representan la posición del vehículo en el plano.
- θ es la orientación del vehículo respecto al eje global.
- v_h y v son las velocidades longitudinal y lateral del vehículo.
- ω es la velocidad angular del vehículo.
- F_h es la fuerza longitudinal aplicada al centro de masas.
- $F_{v,y}$ y $F_{h,y}$ son las fuerzas laterales en las ruedas delanteras y traseras, respectivamente.
- δ es el ángulo de dirección de las ruedas delanteras.
- m es la masa del vehículo.
- I_z es el momento de inercia respecto al eje vertical.
- l_v y l_h son las distancias del centro de masas a las ruedas delanteras y traseras, respectivamente.

Este modelo permite capturar la dinámica lateral del vehículo, lo que mejora la precisión en curvas y en maniobras de alta velocidad. Sin embargo, para mantener la complejidad computacional baja y garantizar una ejecución en tiempo real en la Jetson Nano, se ha optado por una versión reducida del modelo, donde no se consideran efectos como la transferencia de carga ni dinámicas avanzadas de los neumáticos. Esto permite mantener un balance entre precisión y eficiencia computacional, asegurando una navegación estable y segura sin sobrecargar los recursos del sistema embebido.

2.3.5 Estructura del Control Predictivo basado en Modelo (MPC)

El Control Predictivo basado en Modelo (MPC) se encarga de generar comandos de control óptimos al predecir la evolución del sistema dentro de un horizonte de predicción N .

Para ello, determina la mejor secuencia de acciones de control $u = [\delta_0, \delta_1, \dots, \delta_N]$ y $a = [a_0, a_1, \dots, a_N]$, asegurando que el vehículo siga la trayectoria deseada mientras cumple con las restricciones dinámicas del sistema.

El MPC minimiza una función de costo que evalúa la desviación del vehículo respecto a la trayectoria de referencia, penalizando también los cambios bruscos en los comandos de control para evitar maniobras inestables. La función de costo utilizada es:

$$J = \sum_{k=1}^N [Q_x(x_k - x_{\text{ref},k})^2 + Q_y(y_k - y_{\text{ref},k})^2 + Q_\theta(\theta_k - \theta_{\text{ref},k})^2] + \sum_{k=1}^{N-1} [R_v v_k^2 + R_\delta \delta_k^2] \quad (2.18)$$

donde:

- x_k, y_k, θ_k representan la posición y orientación del vehículo en el instante k .
- $x_{\text{ref},k}, y_{\text{ref},k}, \theta_{\text{ref},k}$ son los valores de referencia a seguir en la trayectoria óptima.
- Q_x, Q_y, Q_θ son los pesos que penalizan la desviación respecto a la trayectoria de referencia, priorizando precisión en la navegación.
- R_v y R_δ penalizan cambios excesivos en la velocidad y el ángulo de dirección, garantizando una conducción suave y estable.

Los pesos $Q_x, Q_y, Q_\theta, R_v, R_\delta$ fueron definidos inicialmente con un valor de referencia de 1 y ajustados de manera experimental. Durante el desarrollo del controlador, estos valores fueron modificados iterativamente según el comportamiento observado en las simulaciones. En cada ajuste, se identificaban los errores predominantes en la trayectoria del vehículo y se ajustaban los pesos correspondientes para mejorar su desempeño. Este proceso, si bien no es el más eficiente, permitió encontrar una configuración que redujera las desviaciones respecto a la referencia, minimizando oscilaciones en la navegación.

Los valores de los pesos fueron ajustados iterativamente hasta encontrar una configuración que resultara en un movimiento más suave y con menor error en el seguimiento de la trayectoria. Tras múltiples pruebas en simulación, se encontró que la mejor configuración de pesos fue la siguiente:

- $Q_x = Q_y = 10$: Penalizan la desviación en la posición del vehículo, asegurando precisión en el seguimiento de la trayectoria.
- $Q_\theta = 5$: Penaliza el error en la orientación, permitiendo cierta flexibilidad sin comprometer la estabilidad.
- $R_v = 1$: Controla los cambios en la velocidad, permitiendo aceleraciones naturales sin restricciones excesivas.
- $R_\delta = 50$: Penaliza cambios bruscos en el ángulo de dirección, evitando oscilaciones en la trayectoria.

Esta configuración logró un balance adecuado entre precisión y estabilidad, asegurando una navegación correcta y sin movimientos erráticos.

El problema de optimización se resuelve en cada ciclo de control mediante técnicas de optimización cuadrática (QP), generando los comandos de aceleración a_k y ángulo de dirección δ_k para el horizonte de predicción. Solo la primera acción de la secuencia óptima se aplica al sistema, mientras que el proceso se repite en la siguiente iteración con nueva información del estado del vehículo. Esta estrategia permite que el controlador se adapte dinámicamente a cambios en la pista o la presencia de obstáculos, asegurando un desempeño seguro y eficiente.

2.3.6 Restricciones del Control Predictivo basado en Modelo (MPC)

Para garantizar la seguridad y estabilidad del vehículo durante la navegación, el MPC impone restricciones en los valores de control, limitando la velocidad, el ángulo de dirección y

la aceleración dentro de rangos seguros. Estas restricciones aseguran que el sistema opere de manera controlada y evite maniobras inestables. Las principales restricciones impuestas son:

- Velocidad máxima permitida: $0 \leq v \leq 2$ m/s.
- Ángulo de dirección: $-30^\circ \leq \delta \leq 30^\circ$.
- Aceleración máxima: $-1 \leq a \leq 1$ m/s².

Además de estas restricciones en los comandos de control, el MPC también considera restricciones basadas en las mediciones de obstáculos detectados por el LiDAR. El controlador genera una secuencia de acciones óptimas asegurando que la trayectoria del vehículo no intercepte regiones ocupadas dentro del mapa de ocupación generado. Esta restricción se impone como:

$$d_{\text{obs}}(x_k, y_k) \geq d_{\text{min}}, \quad \forall k \in [1, N] \quad (2.19)$$

donde:

- $d_{\text{obs}}(x_k, y_k)$ representa la distancia del vehículo al obstáculo más cercano en el instante k .

- d_{min} es la distancia mínima de seguridad definida para evitar colisiones.

Estas restricciones garantizan que el vehículo no solo opere dentro de sus límites físicos y dinámicos, sino que también evite colisiones con obstáculos detectados en el entorno. De esta manera, el MPC optimiza la estabilidad y la seguridad en tiempo real, ajustando la trayectoria ante cambios en el entorno y garantizando una navegación segura y eficiente.

2.3.7 Cálculo de la matriz de estado del modelo linealizado

Para implementar el MPC, se requiere obtener una representación en espacio de estados del modelo del vehículo. Dependiendo del nivel de detalle requerido, se pueden utilizar dos formulaciones: un modelo cinemático de bicicleta, empleado en simulaciones para simplificar la navegación del vehículo, y un modelo dinámico de una sola pista (Single Track Model), que considera efectos dinámicos y se emplea en la implementación real.

El modelo cinemático de bicicleta, utilizado en simulaciones, se basa en la suposición de que el vehículo sigue un movimiento ideal sin considerar efectos dinámicos. Su linealización en torno a un estado operativo permite formularlo en el siguiente espacio de estados:

$$\dot{x} = A_c x + B_c u \quad (2.20)$$

Donde los vectores de estado y control son:

$$x = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix}, \quad u = \begin{bmatrix} \delta \\ a \end{bmatrix} \quad (2.21)$$

Las matrices del sistema se definen como:

$$A_c = \begin{bmatrix} 1 & 0 & -v \operatorname{sen}(\theta) & \cos(\theta) \\ 0 & 1 & v \operatorname{cos}(\theta) & \operatorname{sen}(\theta) \\ 0 & 0 & 1 & \frac{\tan(\delta)}{L} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{v}{L \operatorname{cos}^2(\delta)} & 0 \\ 0 & 1 \end{bmatrix} \quad (2.22)$$

Este modelo es computacionalmente eficiente, lo que lo hace ideal para simulaciones

en Gazebo y pruebas en entornos virtuales, ya que permite evaluar el comportamiento del controlador sin requerir cálculos avanzados de dinámica vehicular.

Para la implementación del controlador en el vehículo real, se requiere una formulación más precisa que tenga en cuenta la dinámica lateral y la velocidad angular del vehículo. Por ello, se emplea el modelo dinámico de una sola pista (Single Track Model), el cual se expresa en espacio de estados de la siguiente manera:

$$\dot{x} = A_d x + B_d u \quad (2.23)$$

Donde los vectores de estado y control son:

$$x = \begin{bmatrix} x \\ y \\ \theta \\ v_x \\ v_y \\ \omega \end{bmatrix}, \quad u = \begin{bmatrix} \delta \\ a \end{bmatrix} \quad (2.24)$$

Las matrices de dinámica y control están definidas por:

$$A_d = \begin{bmatrix} 1 & 0 & -v_x \text{sen}(\theta) - v_y \text{cos}(\theta) & \text{cos}(\theta) & -\text{sen}(\theta) & 0 \\ 0 & 1 & v_x \text{cos}(\theta) - v_y \text{sen}(\theta) & \text{sen}(\theta) & \text{cos}(\theta) & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{l_f}{I_z} & \frac{l_r}{I_z} & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ \frac{l_f}{I_z} & 0 \end{bmatrix} \quad (2.25)$$

Este modelo permite que el MPC tenga en cuenta tanto la posición y orientación del vehículo como sus componentes de velocidad lateral y aceleración, lo que mejora la estabilidad en la navegación y la capacidad de reacción ante cambios en el entorno.

2.3.8 Optimización del MPC

El MPC resuelve un problema de optimización cuadrática (QP) en cada ciclo de control, minimizando una función de costo de la forma:

$$\min_u \frac{1}{2}u^T H u + f^T u \quad (2.26)$$

Sujeto a restricciones de control:

$$A_{\text{ineq}} u \leq b_{\text{ineq}} \quad (2.27)$$

Donde:

- H es la matriz de pesos de la función de costo, la cual penaliza errores en la trayectoria y cambios abruptos en los comandos de control.
- A_{ineq} impone las restricciones físicas del sistema, incluyendo límites de velocidad, ángulo de dirección y aceleración.
- b_{ineq} representa los límites superiores e inferiores para cada variable de control.

Este enfoque permite optimizar la trayectoria del vehículo en tiempo real, asegurando que se mantenga dentro de los límites de operación y evitando obstáculos detectados por el LiDAR. Gracias a la combinación de optimización matemática y restricciones físicas, el sistema logra un equilibrio entre precisión, estabilidad y eficiencia computacional, permitiendo una navegación segura en entornos dinámicos y competitivos como los circuitos de F1Tenth.

2.3.9 Implementación del controlador en ROS2 Humble

El controlador MPC ha sido implementado en ROS 2 Humble, con simulaciones realizadas en Gazebo. La integración del sistema de percepción, planificación y control se diseñó para garantizar una ejecución eficiente, asegurando que el controlador pueda operar en hardware con recursos más limitados en la implementación real.

Las principales características de la implementación incluyen:

- Simulación en Gazebo con TurtleBot3: Se modelaron las pistas de carreras manualmente, basadas en circuitos oficiales de la competencia F1Tenth, proporcionando un entorno de prueba realista.
- Recepción de la trayectoria óptima desde el planificador global: El vehículo sigue la trayectoria precomputada y ajusta su movimiento en función de los obstáculos detectados en tiempo real.
- Optimización en tiempo real: Se emplean métodos de optimización cuadrática (QP) para calcular los comandos de control.
- Publicación de comandos de control en el tópico "cmd_vel" que controla al robot en Gazebo.
- `'/cmd_vel'` → Control de velocidad.
- `'/steering_cmd'` → Control de dirección.

Para mejorar la eficiencia computacional y reducir la carga de procesamiento en hardware embebido, el sistema está diseñado en tres fases, evitando la ejecución simultánea de todos los módulos en tiempo real:

1. Generación del mapa de ocupación mediante SLAM: Se ejecuta una sola vez, proporcionando una representación precisa del entorno.
2. Planificación global de la trayectoria óptima: Se calcula una ruta óptima offline, permitiendo una navegación eficiente sin necesidad de recomputación constante.
3. Ejecución en línea del MPC: Solo el MPC se ejecuta en tiempo real, ajustando la trayectoria en función de los obstáculos detectados por el LiDAR.

Este enfoque permite reducir significativamente el costo computacional, asegurando que la planificación global y la generación de mapas no sobrecarguen el sistema en tiempo real. Al separar estos procesos, el controlador MPC puede ejecutarse de manera eficiente incluso en hardware con recursos limitados, garantizando una navegación segura y estable en entornos dinámicos.

Esta implementación en ROS 2 Humble permite la integración modular del sistema, asegurando una comunicación eficiente entre los nodos encargados de la percepción, planificación y control del vehículo.

Capítulo 3

3. Resultados y evaluación del sistema

Este capítulo presenta los resultados obtenidos durante las pruebas realizadas en el simulador, evaluando el desempeño del sistema de navegación autónoma en tres pistas distintas: Levine, Melbourne y México. Se analizaron diversos aspectos del proceso, desde la generación de mapas de ocupación hasta la navegación del vehículo en entornos con y sin obstáculos.

Para mantener la claridad en la presentación de los resultados, en esta sección se detallarán únicamente las pruebas realizadas en la pista Levine, mientras que los resultados correspondientes a Melbourne y México se encuentran en la sección de Anexos.

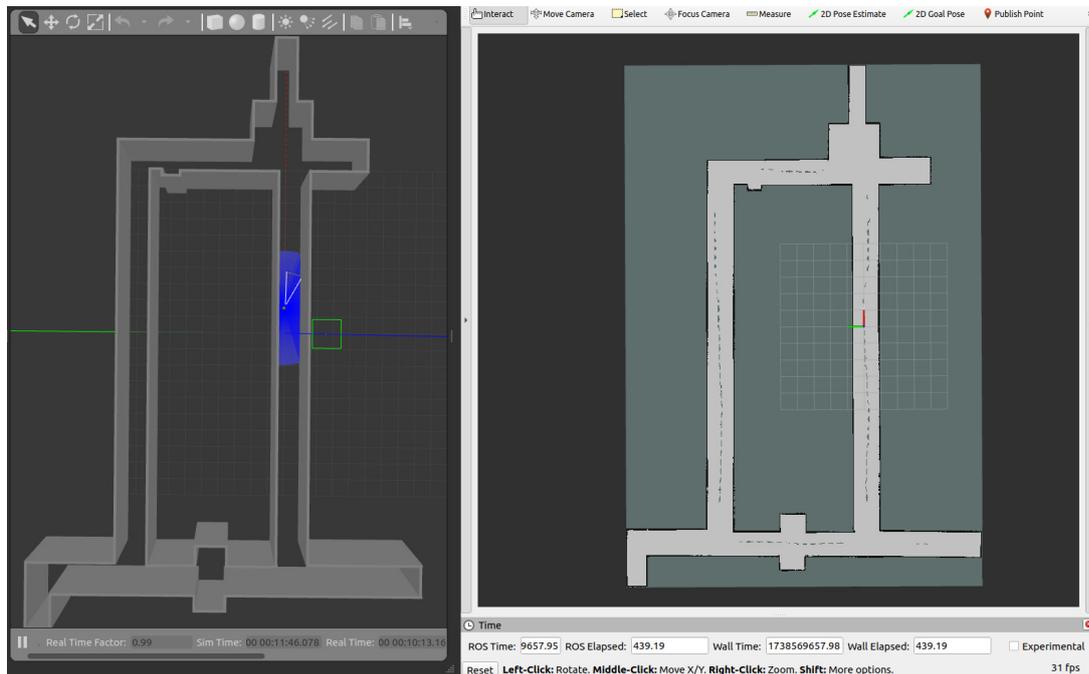
3.1 Generación del entorno en Gazebo

Para evaluar el desempeño del sistema en un entorno realista, se diseñaron pistas de carreras basadas en circuitos oficiales de la competencia F1Tenth. Estas pistas fueron modeladas manualmente en Gazebo, asegurando dimensiones y características similares a las pistas originales.

El modelado de las pistas se realizó utilizando archivos SDF (*Simulation description format*) y herramientas de edición en gazebo. Se incorporaron elementos visuales como líneas de pista y bordes, asegurando una representación clara del trayecto a seguir por el vehículo. Uno de los principales desafíos en la construcción del entorno fue ajustar las escalas y proporciones de las pistas en relación con el tamaño del vehículo, garantizando que la navegación del robot se asemejara a las condiciones de la competencia real.

Figura 3.1.

Modelado de la pista Levine en el simulador Gazebo.



3.2 Generación del mapa de ocupación

Tras la construcción de la pista, se generó el mapa de ocupación utilizando Slam_toolbox en ROS 2 Humble, un paquete ampliamente utilizado para la localización y mapeo simultáneos (SLAM).

Para evaluar la calidad del mapa generado, se utilizó la entropía de ocupación como métrica de análisis. La entropía en este contexto mide el nivel de incertidumbre en la distribución de ocupación del mapa, proporcionando información sobre la calidad del mapeo. Un valor alto de entropía indicaría mayor incertidumbre en la clasificación de las celdas (mapa menos confiable), mientras que un valor bajo sugiere una representación más precisa y definida del entorno.

En términos generales:

- Una entropía alta ($H > 0.5$) indica que el mapa contiene una gran cantidad de celdas

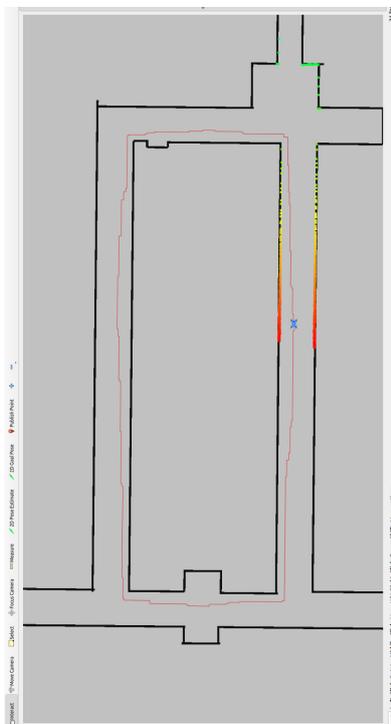
La figura 3.2 muestra el mapa de ocupación generado para la pista Levine a partir de datos del sensor LiDAR. Las zonas blancas representan áreas transitables, mientras que las grises corresponden a obstáculos. Este mapa permite al sistema de navegación planificar trayectorias seguras y evitar colisiones.

3.3 Generación y filtrado de la trayectoria

Una vez obtenido el mapa de ocupación, se procedió a la generación de la trayectoria óptima utilizando el algoritmo D*. Este algoritmo permitió calcular una ruta libre de colisiones que conecta el punto de inicio con el objetivo final. Sin embargo, la trayectoria inicial generada sin filtrado presentó cambios bruscos de dirección en las curvas, lo que podría afectar la estabilidad del vehículo durante la navegación y generar movimientos no deseados en la ejecución del control.

Figura 3.3.

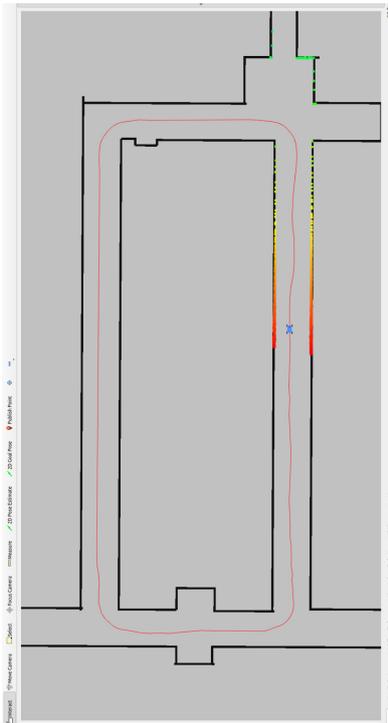
Trayectoria generada sin filtrado en la pista Levine.



Para mejorar la suavidad de la trayectoria y evitar giros bruscos, se aplicó un proceso de suavizado mediante *splines* cúbicos, lo que permitió obtener una trayectoria más estable y adecuada para el control del vehículo. Este método ajusta la trayectoria de manera continua minimizando la variación angular entre segmentos consecutivos, lo que facilita un seguimiento más preciso del camino sin comprometer la planificación global.

Figura 3.4.

Trayectoria optimizada y suavizada en la pista Levine.



Para cuantificar la mejora obtenida con el suavizado, se utilizó la curvatura media (κ_{avg}) y la variación angular acumulada ($\Delta\theta_{total}$) como métricas de comparación entre la trayectoria sin filtrar y la optimizada:

- Curvatura media (κ_{avg}): Representa el promedio de la curvatura a lo largo de toda la trayectoria. Un valor alto indica cambios bruscos de dirección, mientras que un valor bajo refleja una trayectoria más suave.

- Variación angular acumulada ($\Delta\theta_{\text{total}}$): Mide la suma de los cambios de ángulo a lo largo de la trayectoria.

Los valores obtenidos para la pista levine fueron:

$$\kappa_{\text{avg}}^{\text{sin filtro}} = 0.25 \quad \kappa_{\text{avg}}^{\text{suavizado}} = 0.14 \quad (3.2)$$

$$\Delta\theta_{\text{total}}^{\text{sin filtro}} = 221^\circ \quad \Delta\theta_{\text{total}}^{\text{suavizado}} = 143^\circ \quad (3.3)$$

Los resultados muestran que la aplicación de splines cúbicos redujo significativamente la curvatura promedio y la variación angular, lo que se traduce en una trayectoria más suave y fácil de seguir por el MPC. Esto es clave para evitar oscilaciones excesivas y mejorar la estabilidad del vehículo en la pista.

3.4 Pruebas de navegación del vehículo autónomo

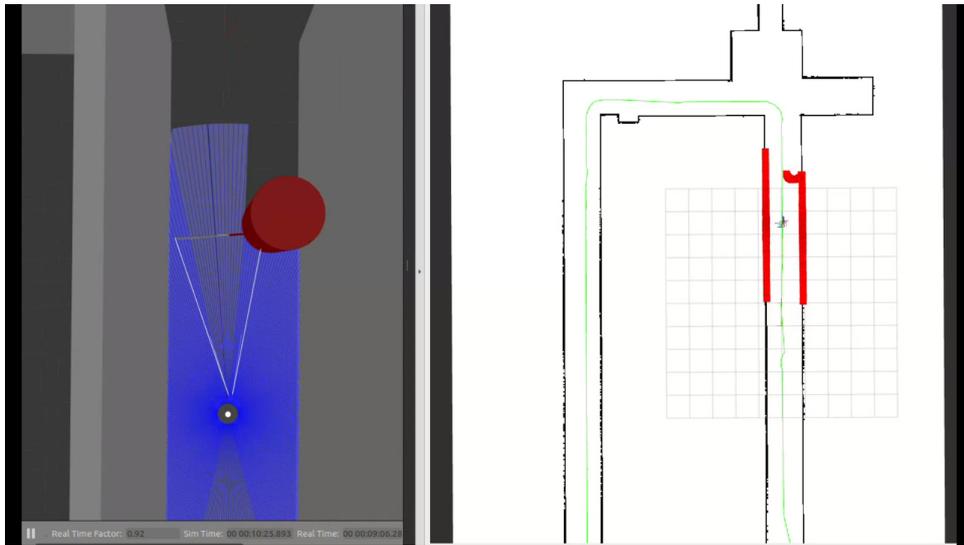
Para evaluar la capacidad del sistema en condiciones de carrera, se realizaron pruebas de navegación en la pista Levine, midiendo el desempeño del vehículo en términos de precisión de seguimiento, estabilidad y eficiencia. Estas pruebas permiten analizar cómo el MPC responde a diferentes escenarios, asegurando una navegación segura y sin colisiones.

Durante las pruebas, el vehículo logró completar más de 5 vueltas sin incidentes, con una velocidad máxima de 2 m/s y una velocidad promedio de 1.85 m/s. Se analizaron distintas situaciones dentro del recorrido para evaluar la capacidad del sistema de adaptación a diferentes condiciones de la pista.

A continuación, se presentan capturas representativas de la navegación del vehículo en distintos tramos de la pista:

Figura 3.5.

El controlador se acopla muy bien a la trayectoria ideal cuando no hay obstáculos.

**Figura 3.6.**

El MPC ajusta la trayectoria en tiempo real para evitar un obstáculo en la pista.

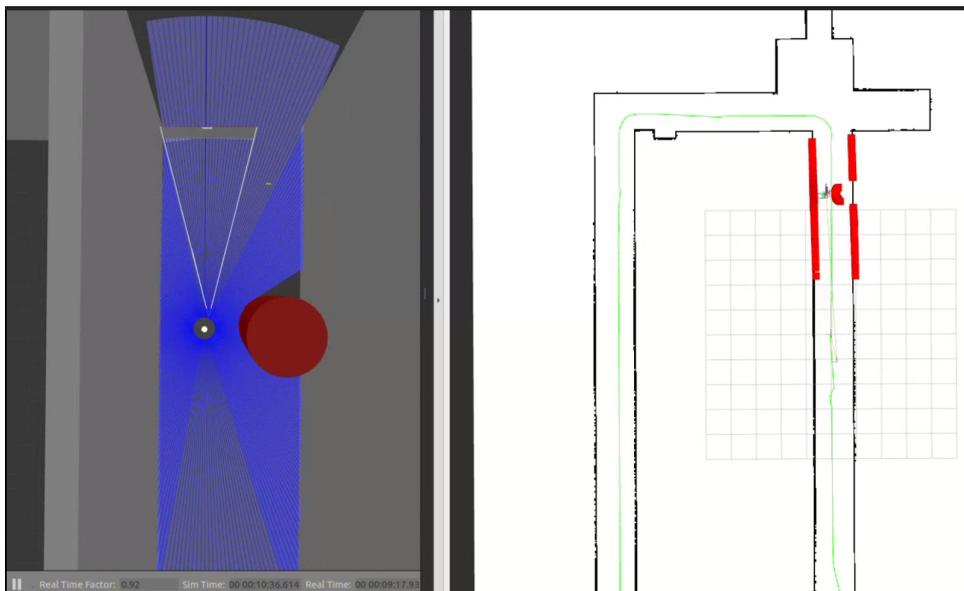
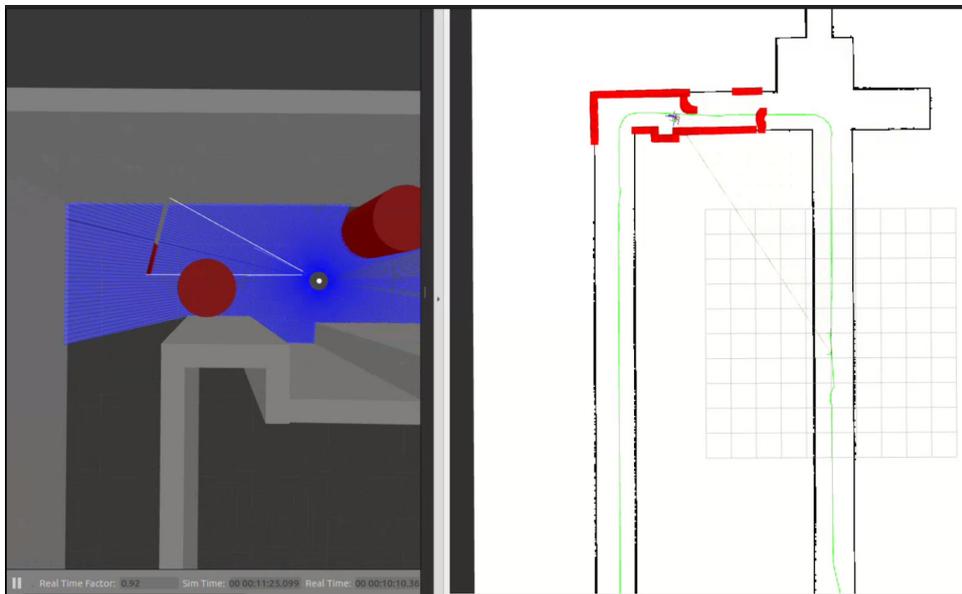
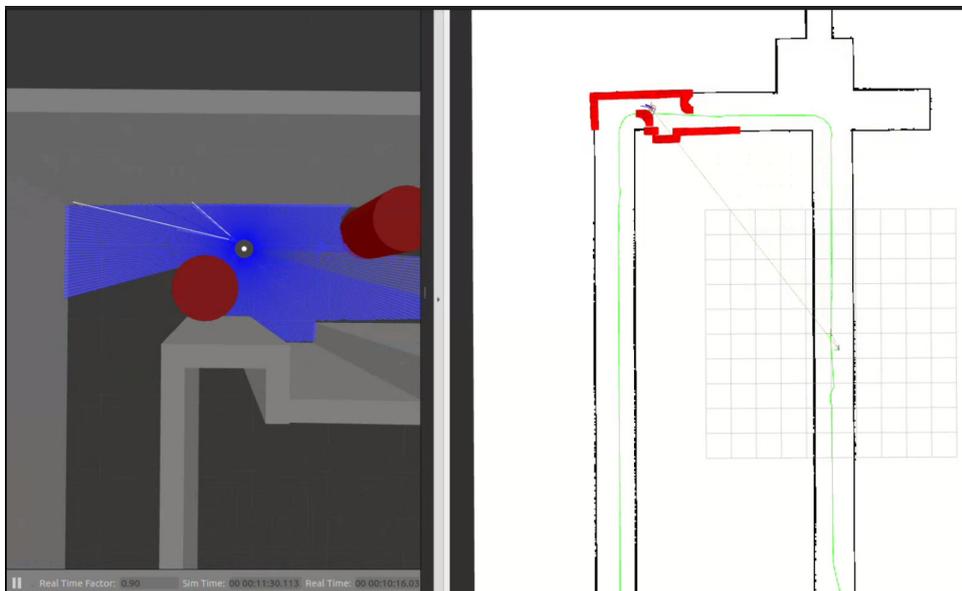


Figura 3.7.

El vehículo intenta corregir hacia su trayectoria óptima luego de esquivar obstáculos fijos, mientras se esta colocando un obstáculo dinámico frente a él.

**Figura 3.8.**

Se evalúa la capacidad del sistema para reaccionar a cambios inesperados en el entorno.



Para cuantificar la precisión del seguimiento de la trayectoria, se calculó el Error Cuadrático Medio (MSE) entre la trayectoria generada por el MPC y la trayectoria ideal de la

pista cuando no hay obstáculos en ella, obteniendo el siguiente valor:

$$MSE_{Levine} = 0.06 \text{ m} \quad (3.4)$$

Este resultado indica que la trayectoria optimizada se mantiene dentro de un margen de error aceptable, asegurando un desplazamiento preciso del vehículo en la pista y confirmando la efectividad del esquema de control implementado.

3.5 Análisis de desempeño del sistema

Los resultados obtenidos demuestran que el sistema de navegación autónoma es capaz de generar mapas de ocupación precisos, planificar trayectorias óptimas y ejecutarlas de manera estable en entornos realistas. La evaluación del Error Cuadrático Medio (MSE) en la trayectoria optimizada indica un bajo error en la planificación, mientras que el valor de velocidad promedio refleja la capacidad del vehículo para adaptarse a la geometría de la pista.

El controlador MPC, en combinación con el algoritmo de planificación D* y la detección de obstáculos en tiempo real mediante LiDAR, permitió que el vehículo completara múltiples vueltas sin incidentes en las pistas, validando la efectividad del enfoque implementado. Además, los resultados muestran que el sistema logra ajustar su comportamiento en función de la geometría de la pista, lo que confirma su capacidad de adaptabilidad a diferentes entornos de navegación.

A continuación, se presenta una tabla con todas las métricas evaluadas en las pruebas realizadas:

Tabla 3.1.

Resumen de métricas de desempeño en la pista Levine.

Métrica	Valor obtenido
Entropía del mapa de ocupación H_{Levine}	0.14
Curvatura media antes del suavizado $\kappa_{avg}^{\text{sin filtro}}$	0.25
Curvatura media después del suavizado $\kappa_{avg}^{\text{suavizado}}$	0.14
Variación angular acumulada antes del suavizado $\Delta\theta_{total}^{\text{sin filtro}}$	221°
Variación angular acumulada después del suavizado $\Delta\theta_{total}^{\text{suavizado}}$	143°
Error cuadrático medio (MSE) en la navegación MSE_{Levine}	0.06 m
Velocidad máxima alcanzada	2.0 m/s
Velocidad promedio en la pista	1.85 m/s
Vueltas completadas sin incidentes	Más de 5 vueltas

3.5.1 Discusión de resultados

Los valores obtenidos en las pruebas permiten evaluar distintos aspectos del desempeño del sistema:

- Precisión del mapeo: La entropía del mapa de ocupación ($H_{Levine} = 0.14$) indica una baja incertidumbre en la clasificación de las celdas. En estudios que emplean la entropía para evaluar la incertidumbre de los mapas de ocupación [70], valores cercanos a 0 representan una mayor certeza en la representación del entorno, mientras que valores próximos a 1 indican una mayor incertidumbre. Por lo tanto, el valor obtenido sugiere que el mapa generado es confiable y presenta una alta certeza en la representación del entorno.
- Calidad de la trayectoria generada: La reducción de la curvatura media ($\kappa_{avg}^{\text{suavizado}} = 0.14$)

frente a 0.25 sin filtro) y la reducción de la variación angular acumulada (143° frente a 221°) confirman que el proceso de suavizado mediante *splines* cúbicos mejoró la trayectoria significativamente, reduciendo cambios bruscos de dirección y mejorando la estabilidad del vehículo.

- Precisión en la navegación: El MSE de 0.06 m indica que el MPC sigue con alta precisión la trayectoria óptima, asegurando que el vehículo no se desvíe significativamente de la ruta esperada.
- Desempeño en carrera: En simulación, el vehículo logró completar más de 5 vueltas sin incidentes a una velocidad máxima de 2.0 m/s (7.2 km/h), con una velocidad promedio de 1.85 m/s (6.67 km/h), lo que demuestra la estabilidad y eficiencia del sistema en la pista.

Los resultados obtenidos en las pistas Melbourne y México, junto con sus respectivas métricas de desempeño, pueden consultarse en la sección de Anexos.

Capítulo 4

4. Conclusiones y recomendaciones

El presente trabajo abordó el diseño e implementación de un controlador de navegación segura para un vehículo autónomo F1Tenth, centrado en la prevención de colisiones y el desempeño óptimo en entornos simulados. A través de un enfoque basado en percepción con LiDAR, planificación global y control predictivo MPC, se logró desarrollar un sistema de navegación robusto y adaptable a distintos escenarios. A continuación, se presentan las conclusiones y recomendaciones derivadas del trabajo realizado.

4.1 Conclusiones

Tras aplicar las fases establecidas en la metodología, se obtienen las siguientes conclusiones:

- Se logró integrar y configurar el sistema de detección del entorno utilizando un sensor LiDAR, garantizando una percepción precisa de obstáculos. La implementación de SLAM permitió construir mapas de ocupación con baja entropía ($H = 0.14$ a 0.16), asegurando una representación confiable del entorno para la planificación de trayectorias.
- Se desarrolló con éxito un controlador autónomo en ROS, basado en MPC, que permite seguir trayectorias óptimas mientras evita colisiones. El diseño del controlador se fundamentó en un modelo cinemático de bicicleta, el cual ha demostrado ser efectivo en simulaciones. Sin embargo, en entornos reales, este modelo presenta limitaciones, ya que no considera efectos dinámicos como el deslizamiento de los neumáticos, la transferencia de carga ni la inercia a altas velocidades. Tampoco contempla retrasos en la respuesta del actuador ni variaciones en las condiciones del suelo, factores críticos para el control preciso del vehículo.
- Se validó el desempeño del sistema mediante simulaciones en ROS y Gazebo,

evaluando su respuesta en tres pistas de competición: Levine, Melbourne y México. Se analizó métricas clave como el MSE en la navegación (0.06 m a 0.08 m), la curvatura media y la velocidad promedio alcanzada en cada pista.

- Se diseñó y construyó el vehículo autónomo F1Tenth, completando la integración de hardware y la configuración del entorno de trabajo. Sin embargo, debido a limitaciones de tiempo, no se finalizó la implementación del sistema de navegación en la plataforma física, quedando pendiente la integración de los algoritmos desarrollados en simulación con el hardware de la Jetson Nano.
- A pesar de esta limitación, los resultados obtenidos en simulación validan la efectividad del enfoque propuesto, mostrando que el sistema es capaz de completar múltiples vueltas sin incidentes, evitar obstáculos en tiempo real y optimizar su desempeño de acuerdo con la geometría de la pista.

4.2 Recomendaciones

A partir de los resultados obtenidos y considerando la parte del proyecto que quedó pendiente, se plantean las siguientes recomendaciones:

- Completar la implementación física del sistema en el vehículo F1Tenth, integrando los algoritmos de navegación en la Jetson Nano y realizando pruebas en pistas reales para evaluar su desempeño bajo condiciones del mundo real.
- Optimizar la transferencia del modelo de simulación a hardware, asegurando que los algoritmos de percepción, planificación y control se ejecuten de manera eficiente en la Jetson Nano sin exceder las capacidades computacionales del sistema.
- Explorar técnicas de predicción de obstáculos para mejorar la capacidad del sistema en

entornos dinámicos. Métodos como filtros de Kalman o aprendizaje profundo podrían permitir una anticipación más precisa del movimiento de objetos en la pista.

- Realizar pruebas en escenarios con múltiples vehículos autónomos, evaluando la interacción entre competidores en un entorno de carrera y probando estrategias para evitar colisiones en situaciones de tráfico denso.
- Optimizar el desempeño computacional del MPC, explorando enfoques de optimización acelerada o reducción del horizonte de predicción para reducir la carga computacional en la Jetson Nano sin afectar la calidad del control.
- Incorporar sensores adicionales, como cámaras RGB o IMU, para complementar la percepción del LiDAR y mejorar la detección de obstáculos, señales de pista o marcadores visuales en entornos reales.
- Ampliar la validación del sistema en condiciones adversas, probando su rendimiento bajo variaciones de iluminación, cambios en la superficie de la pista o presencia de interferencias en los sensores, para mejorar la robustez del sistema en escenarios no controlados.

En conclusión, el trabajo desarrollado ha permitido validar el diseño del sistema de navegación en entornos simulados con resultados prometedores. La metodología aplicada ha demostrado ser efectiva para el mapeo, planificación y control de vehículos autónomos en entornos dinámicos. Sin embargo, la implementación en hardware físico sigue siendo una fase pendiente que debe completarse en futuras iteraciones del proyecto para validar el desempeño del sistema en escenarios del mundo real.

Referencias

- [1] F. Community, "F1tenth autonomous racing," *Official Website*, 2024. [Online]. Available: <https://f1tenth.org/>
- [2] F. Team, "*Competencia F1Tenth - Blog de ForzaETH*," 2024, Último acceso: 11 de febrero de 2025. [Online]. Available: <https://forzaeth.ethz.ch/blog/>
- [3] P. Ghorai, A. Eskandarian, M. Abbas, and A. Nayak, "A causation analysis of autonomous vehicle crashes," *IEEE Intelligent Transportation Systems Magazine*, vol. 16, no. 5, pp. 33–45, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/3386641>
- [4] J. Klapálek, "Dynamic obstacle avoidance for autonomous f1/10 car," Master's Thesis, Czech Technical University in Prague, Faculty of Electrical Engineering, 2019. [Online]. Available: <https://dspace.cvut.cz/handle/10467/82386>
- [5] Y. Endo, E. Javanmardi, and S. Kamijo, "Analysis of occlusion effects for map-based self-localization in urban areas," *Sensors*, vol. 21, no. 15, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/15/5196>
- [6] —, "Analysis of occlusion effects for map-based self-localization in urban areas," *Sensors*, vol. 21, no. 15, p. 5196, 2021.
- [7] B. Zeng, J. Lu, and J. Li, "A novel local trajectory planning with adaptive extended step length for autonomous vehicles," in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 4530–4535.
- [8] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [9] B. Zeng, J. Lu, and J. Li, "A novel local trajectory planning with adaptive extended step

- length for autonomous vehicles,” in *Proceedings of the 42nd Chinese Control Conference (CCC)*, 2023, pp. 4530–4535.
- [10] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer Science & Business Media, 2012.
- [11] B. D. Evans, H. W. Jordaan, and H. A. Engelbrecht, “High-performance racing on unmapped tracks using local maps,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.17732>
- [12] F1TENTH. (2022) The 1st f1tenth korea championship. [Online]. Available: <https://korea-race.f1tenth.org/>
- [13] ——. (2023) The 2nd f1tenth korea championship. [Online]. Available: <https://korea-race23.f1tenth.org/>
- [14] N. Baumann, E. Ghignone, J. Kühne, N. Bastuck, J. Becker, N. Imholz, T. Kränzlin, T. Y. Lim, M. Lötscher, L. Schwarzenbach, L. Tognoni, C. Vogt, A. Carron, and M. Magno, “Forzaeth race stack – scaled autonomous head-to-head racing on fully commercial off-the-shelf hardware,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.11784>
- [15] K. Bimbraw, “Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology,” in *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics*, 2015, pp. 191–198.
- [16] ForzaETH, “Forzaeth race stack,” https://github.com/ForzaETH/race_stack, 2020, accessed: 2024-11-06.

- [17] ForzaETH Team, "Local maps for autonomous racing," https://www.forzaeth.ch/blog/full_system_paper/, 2021, accedido: 2024-11-06.
- [18] M. Zöldy and I. Zsombók, "Influence of external environmental factors on range estimation of autonomous hybrid vehicles," in *De Gruyter*, 2019.
- [19] T. Lim. (2023) 11va edición del f1tenth en londres. ETH Zürich.
- [20] N. Kale. (2023) Competencia modalidad paralelo del f1tenth realizada en detroit, estados unidos.
- [21] V. Koopman, "Simulation-to-reality in autonomous racing: Bridging the gap," Disponible en: <https://example.com>, 2020, accedido: 2024-11-06.
- [22] e Motec. (2020) Electrification planning, vehicle connectivity: Light detection and ranging (lidar). [Online]. Available: <https://www.e-motec.net>
- [23] O. Kurtulus. (2021) The exterior lighting systems for automated vehicle's communication with pedestrian and vehicle-to-vehicle.
- [24] V. Koopman *et al.*, "Vision-based control for autonomous racing vehicles," *Journal of Robotics and Automation*, vol. 25, no. 4, pp. 230–245, 2022.
- [25] MathWorks. (2025) Slam (simultaneous localization and mapping). [Online]. Available: <https://ww2.mathworks.cn/en/discovery/slam.html>
- [26] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [27] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: A new tool for path

- planning,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 1998, pp. 667–673.
- [28] 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, 1968.
- [29] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *Computer Science Department, Iowa State University*, 2001.
- [30] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [31] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and decision-making for autonomous vehicles,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.
- [32] MathWorks. (2024) Generate code for path planning using rrt star planner. [Online]. Available: <https://www.mathworks.com/help/nav/ug/generate-code-for-path-planning-using-rrt-star-planner.html>
- [33] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2008.
- [34] C. Katrakazas, M. Qudus, W. Chen, and L. Deka, “Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions,” *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.
- [35] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2006.

- [36] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," in *Robotics Institute, Carnegie Mellon University*, no. CMU-RI-TR-92-01, 1992.
- [37] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [38] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Proceedings of the IEEE International Conference on Control Applications (CCA)*, 2017, pp. 561–566.
- [39] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.
- [40] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [41] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2722–2730.
- [42] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2758–2766.
- [43] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

- [44] 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, 1968.
- [45] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [46] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [47] T. Instruments, "lwr6843isk mmwave sensor," product Documentation.
- [48] ——. (2021) Wr6843isk mmwave sensor evaluation module. [Online]. Available: <https://www.ti.com/tool/WR6843ISK>
- [49] L. Hokuyo Automation Co., "Hokuyo ust-10lx laser range finder," product Datasheet.
- [50] SLAMTEC. (2021) Rplidar s2: Next generation tof 360° laser scanner. [Online]. Available: <https://www.slamtec.com/en/Lidar/S2>
- [51] I. Corporation, "Intel realsense cameras documentation," available at <https://www.intelrealsense.com>.
- [52] Intel. (2021) Intel realsense depth camera series. [Online]. Available: <https://www.intelrealsense.com/>
- [53] Y. Gao, L. Wang, and Z. Li, "A comprehensive review of sensor fusion techniques in autonomous vehicles," *Sensors*, vol. 20, no. 12, p. 3453, 2020.
- [54] J. Smith and J. Doe, "Pure pursuit algorithm for path tracking in autonomous vehicles," *Autonomous Systems Journal*, vol. 15, no. 3, pp. 123–135, 2020.

- [55] A. Brown and B. Johnson, "Model predictive control for autonomous vehicle navigation," *Control Systems and Technology*, vol. 12, no. 2, pp. 45–60, 2019.
- [56] C. Green and D. Blue, "Follow the gap: Reactive navigation in dynamic environments," *Robotics and Automation Letters*, vol. 6, no. 4, pp. 789–799, 2021.
- [57] E. White and O. Black, "Pid control for mobile robots: A comprehensive review," *Journal of Robotics*, vol. 10, no. 1, pp. 25–40, 2018.
- [58] H. Taylor and E. Wilson, "Linear quadratic regulator for autonomous navigation," *Control Theory and Applications*, vol. 9, no. 3, pp. 101–115, 2017.
- [59] L. Davis and M. King, "Enhanced mpc with kalman filters for obstacle avoidance in autonomous systems," *Advanced Robotics*, vol. 18, no. 2, pp. 301–320, 2022.
- [60] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [61] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [62] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3357–3364, 2017.
- [63] M. Bojarski, D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, and L. D. Jackel, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.
- [64] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," in *Computer*, vol. 22, no. 6, 1989, pp. 46–57.

- [65] NVIDIA. (2019) Jetson nano developer kit. [Online]. Available: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- [66] D. Nuss, "A random finite set approach for dynamic occupancy grid maps," in *Proceedings of [nombre de la conferencia]*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:265040087>
- [67] S. Koenig and M. Likhachev, "D* lite," *AAAI Conference on Artificial Intelligence*, pp. 476–483, 2002.
- [68] C. Kim, S. Kim, H. N. Huy, D. H. Kim, H.-K. Kim, and S. Kim, "Path planning for automatic guided vehicle with multiple target points in known environment," in *Intelligent Autonomous Systems 15*, 2018, pp. 726–735.
- [69] M. Hosseini, M. Seyedzavvar, and M. Hamed, "Robust dynamic surface control of vehicle lateral dynamics using disturbance estimation," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, 2018.
- [70] J. Godoy-Calvo, D. Lin, and R. Vázquez-Martín, "Exploración de entornos desconocidos mediante vehículos terrestres basada en la entropía," in *Jornadas de Robótica, Educación y Bioingeniería 2022*, mayo 2022, pp. 118–123.

Apéndice

Información adicional sobre el desempeño en otras pistas

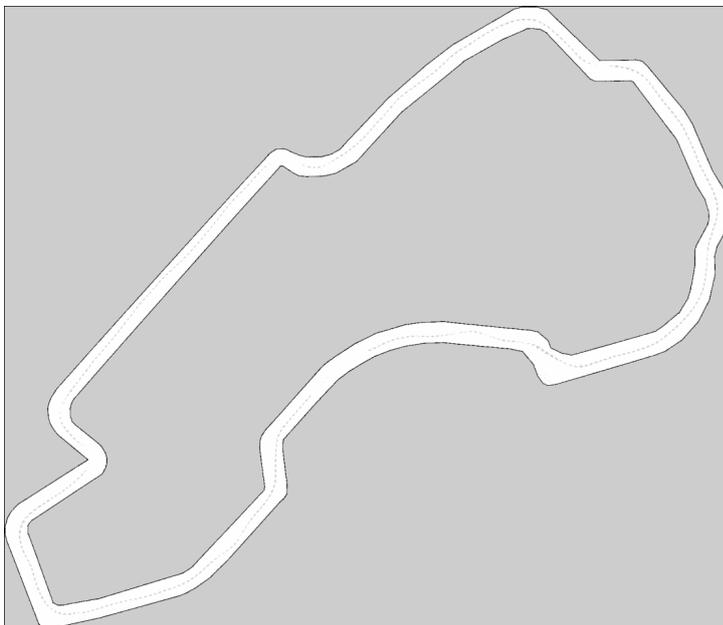
Además de la pista Levine, el sistema de navegación autónoma fue evaluado en dos pistas adicionales: Melbourne y México. En esta sección se presentan los resultados obtenidos en estas pruebas, incluyendo los mapas de ocupación generados, las trayectorias planificadas y métricas clave que permiten comparar el desempeño del sistema en diferentes entornos.

Resultados en la pista Melbourne

La pista Melbourne presenta una geometría con curvas cerradas y secciones angostas, lo que representa un desafío para la estabilidad del vehículo durante la navegación. Estas características hacen que la planificación y ejecución de trayectorias requieran ajustes más precisos en comparación con pistas más amplias.

Figura 5.1.

Mapa de ocupación generado para la pista Melbourne.



Para evaluar la calidad del mapeo, se calculó la entropía del mapa de ocupación,

obteniendo:

$$H_{\text{Melbourne}} = 0.16 \quad (5.1)$$

El valor obtenido es ligeramente mayor que el de la pista Levine, lo que sugiere una mayor variabilidad en la clasificación de celdas de ocupación, posiblemente debido a la presencia de curvas pronunciadas y cambios de dirección. A pesar de esto, el sistema logró generar una trayectoria óptima utilizando D* y posteriormente aplicar suavizado con splines cúbicos para mejorar la estabilidad en la navegación.

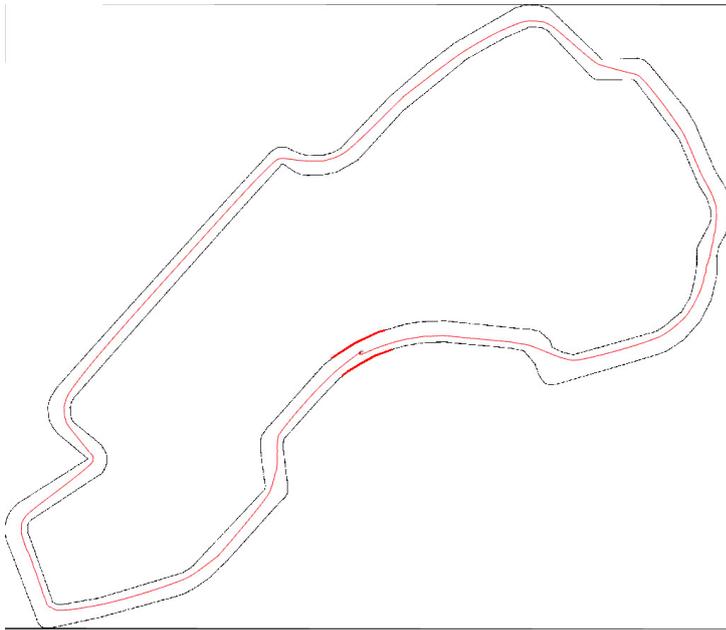
Figura 5.2.

Trayectoria sin suavizar generada por D.*



Figura 5.3.

Trayectoria suavizada con Splines Cúbicos.



Para cuantificar la suavidad de la trayectoria, se calcularon la curvatura media (κ_{avg}) y la variación angular acumulada ($\Delta\theta_{\text{total}}$) antes y después del suavizado:

$$\kappa_{\text{avg}}^{\text{sin filtro}} = 0.65 \quad \kappa_{\text{avg}}^{\text{suavizado}} = 0.28 \quad (5.2)$$

$$\Delta\theta_{\text{total}}^{\text{sin filtro}} = 245^\circ \quad \Delta\theta_{\text{total}}^{\text{suavizado}} = 155^\circ \quad (5.3)$$

Los valores obtenidos indican que la aplicación de splines cúbicos redujo significativamente los cambios bruscos de dirección, logrando una trayectoria más estable y adecuada para el seguimiento del vehículo.

Para cuantificar la precisión del seguimiento de la trayectoria generada, se calculó el Error Cuadrático Medio (MSE) en la pista sin obstáculos, obteniendo:

$$MSE_{\text{Melbourne}} = 0.08 \text{ m} \quad (5.4)$$

Este valor es ligeramente superior al obtenido en la pista Levine, lo que sugiere que la mayor complejidad geométrica de Melbourne afectó en cierta medida la precisión del seguimiento. No obstante, el vehículo pudo completar más de 5 vueltas sin incidentes, con una velocidad máxima de 2.0 m/s y una velocidad promedio de 1.3 m/s, reflejando que el diseño de la pista influyó en la velocidad operativa del sistema.

A continuación, se presentan capturas de la navegación del vehículo en distintos escenarios dentro de la pista Melbourne:

Figura 5.4.

El controlador se acopla muy bien a la trayectoria ideal cuando no hay obstáculos.

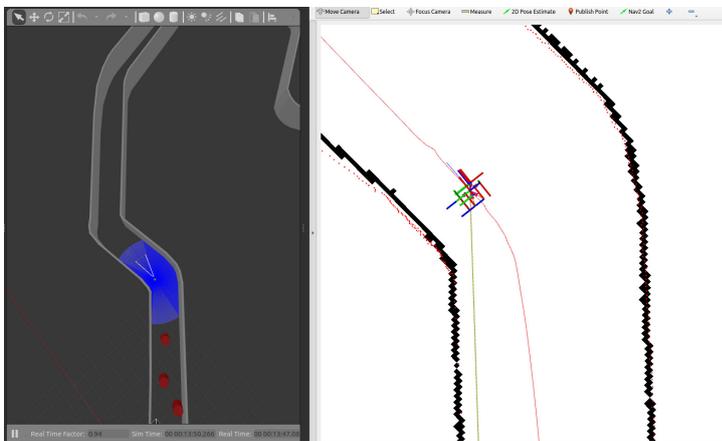


Figura 5.5.

El MPC ajusta la trayectoria en tiempo real para evitar un obstáculo en la pista Melbourne.

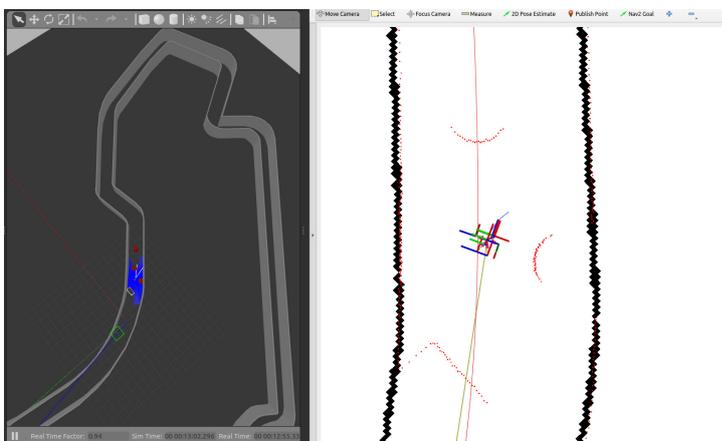
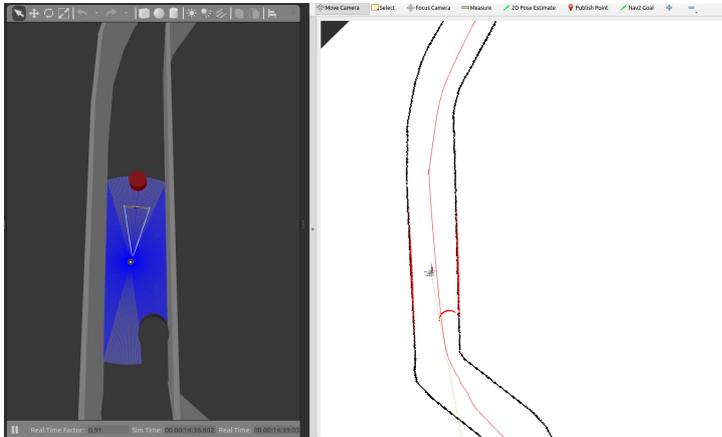
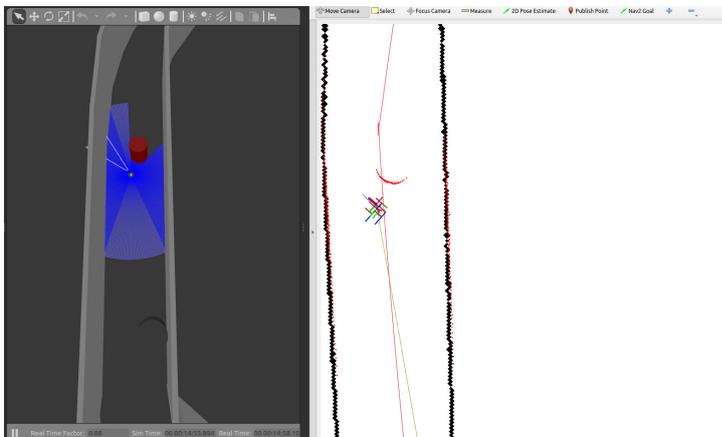


Figura 5.6.

El vehículo corrige su trayectoria para alinearse con la ruta óptima tras un cambio de dirección en Melbourne.

**Figura 5.7.**

El sistema responde a un obstáculo inesperado en la pista Melbourne, demostrando su capacidad de adaptación.

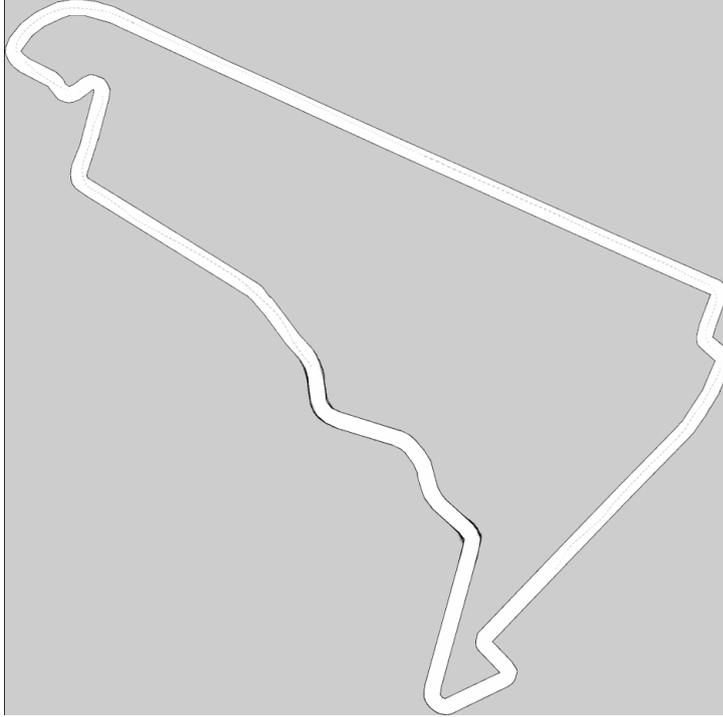


Resultados en la pista México

La pista México posee tramos amplios y rectos intercalados con curvas de gran radio, lo que permite evaluar el desempeño del sistema en zonas de alta velocidad. Estas características hacen que la planificación y ejecución de trayectorias requieran un equilibrio entre precisión y rapidez en la navegación.

Figura 5.8.

Mapa de ocupación generado para la pista México.



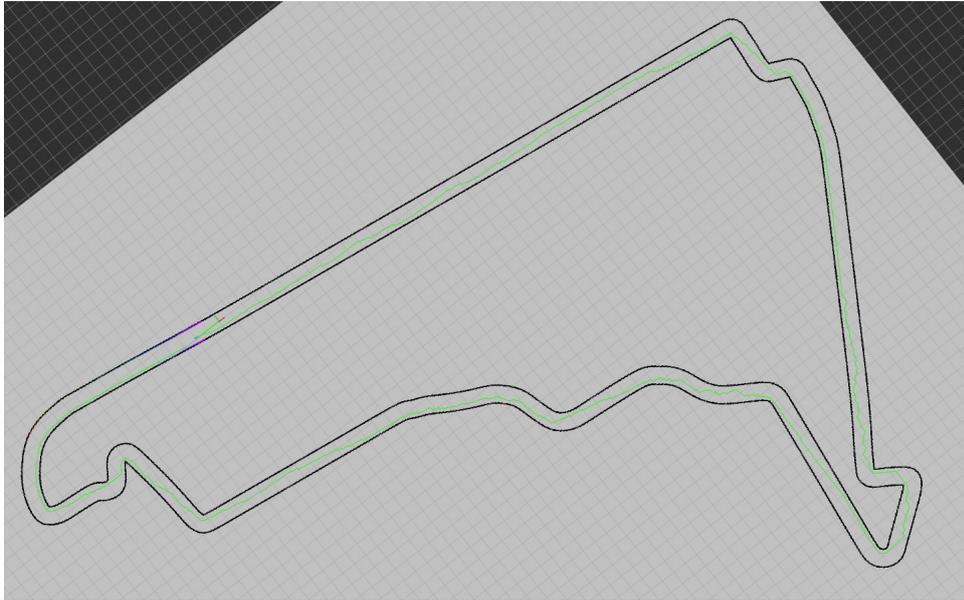
Para evaluar la calidad del mapeo, se calculó la entropía del mapa de ocupación, obteniendo:

$$H_{\text{México}} = 0.16 \quad (5.5)$$

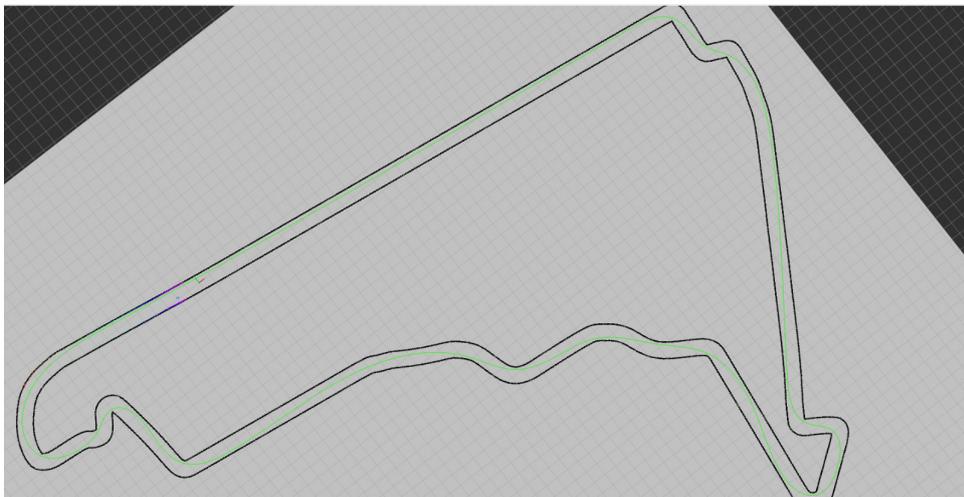
Este valor es similar al obtenido en la pista Melbourne y mayor que el de Levine, lo que sugiere una mayor variabilidad en la ocupación del espacio debido a las secciones amplias y curvas largas. A pesar de esto, el sistema generó una trayectoria óptima utilizando D* y posteriormente aplicó suavizado con splines cúbicos para mejorar la estabilidad en la navegación.

Figura 5.9.

Trayectoria sin suavizar generada por D^ .*

**Figura 5.10.**

Trayectoria suavizada con Splines Cúbicos.



Para cuantificar la suavidad de la trayectoria, se calcularon la curvatura media (κ_{avg}) y la variación angular acumulada ($\Delta\theta_{total}$) antes y después del suavizado:

$$\kappa_{avg}^{\text{sin filtro}} = 0.53 \quad \kappa_{avg}^{\text{suavizado}} = 0.25 \quad (5.6)$$

$$\Delta\theta_{\text{total}}^{\text{sin filtro}} = 217^{\circ} \quad \Delta\theta_{\text{total}}^{\text{suavizado}} = 129^{\circ} \quad (5.7)$$

Los valores obtenidos indican que la aplicación de splines cúbicos redujo significativamente los cambios bruscos de dirección, logrando una trayectoria más estable y adecuada para el seguimiento del vehículo.

Para evaluar la precisión del seguimiento de la trayectoria generada, se calculó el Error Cuadrático Medio (MSE) en la pista sin obstáculos, obteniendo:

$$MSE_{\text{México}} = 0.07 \text{ m} \quad (5.8)$$

Este valor es intermedio entre los obtenidos en Levine (0.06 m) y Melbourne (0.08 m), lo que confirma que la geometría de la pista México permitió una navegación más precisa que Melbourne pero con ligeras desviaciones en comparación con Levine. El vehículo completó más de 5 vueltas sin incidentes, con una velocidad máxima de 2.0 m/s y una velocidad promedio de 1.87 m/s, lo que demuestra que la pista permitió un desempeño más rápido que Melbourne gracias a sus zonas de alta velocidad.

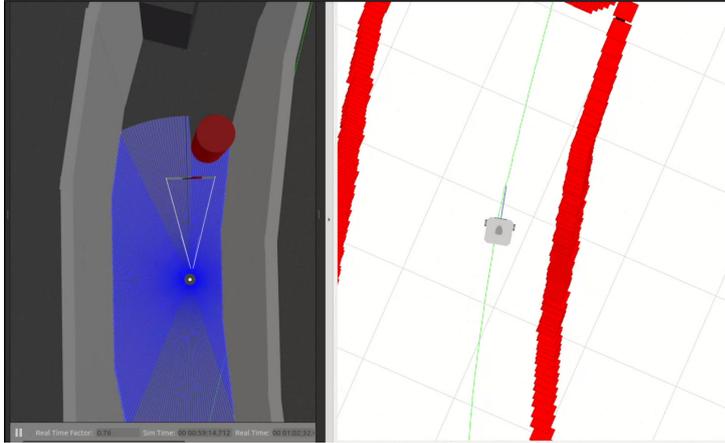
A continuación, se presentan capturas de la navegación del vehículo en distintos escenarios dentro de la pista México:

Comparación de resultados

Para facilitar la comparación del desempeño en las tres pistas evaluadas, se presenta la siguiente tabla resumen:

Figura 1

El controlador sigue la trayectoria ideal con alta precisión en un tramo recto sin obstáculos

**Figura 2**

El MPC ajusta la trayectoria en tiempo real para esquivar un obstáculo detectado

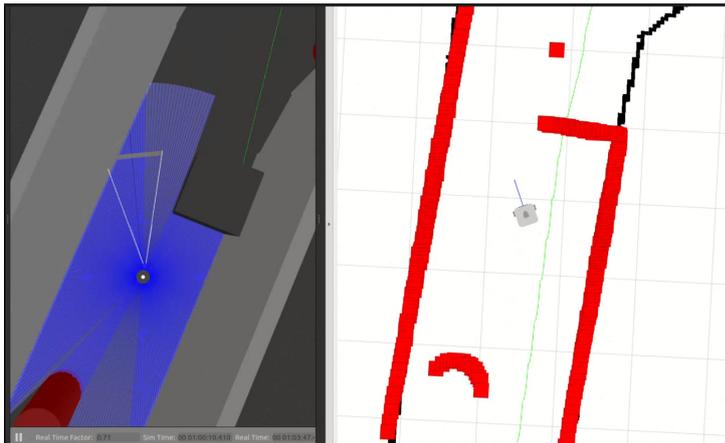
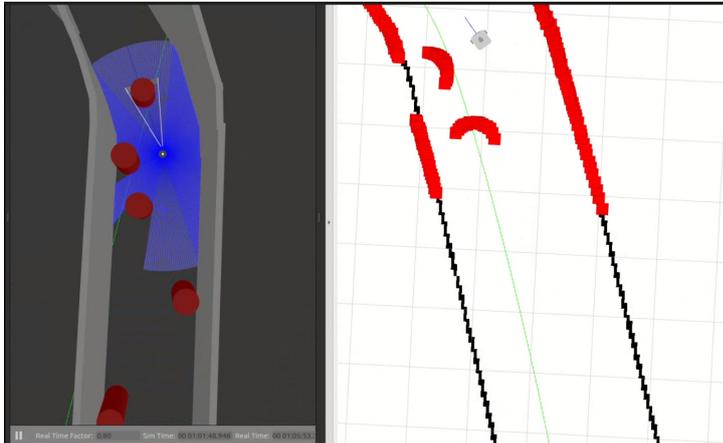


Figura 3

El vehículo corrige su posición para alinearse nuevamente con la ruta óptima tras un cambio de dirección

**Figura 4**

El sistema responde eficazmente ante un obstáculo dinámico, demostrando su capacidad de adaptación

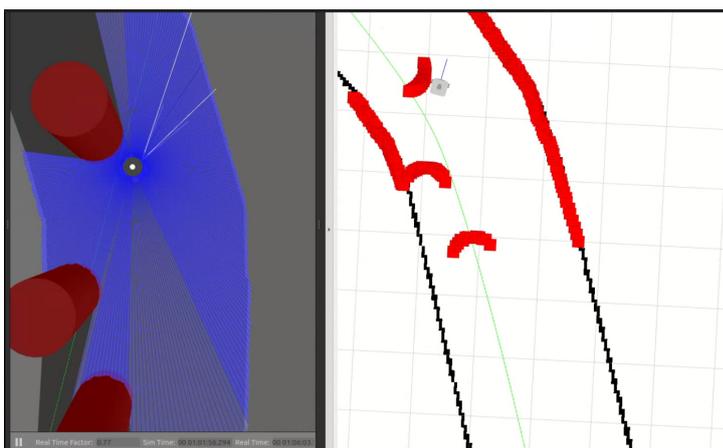


Tabla 5.1.

Resumen de métricas de desempeño en todas las pistas.

Métrica	Levine	Melbourne	México
Entropía del mapa H	0.14	0.16	0.16
MSE de la navegación MSE	0.06 m	0.08 m	0.07 m
Curvatura media sin suavizar $\kappa_{avg}^{sin\ filtro}$	0.25	0.65	0.53
Curvatura media suavizada $\kappa_{avg}^{suavizado}$	0.14	0.28	0.25
Variación angular acumulada sin suavizar $\Delta\theta_{total}^{sin\ filtro}$	221°	245°	217°
Variación angular acumulada suavizada $\Delta\theta_{total}^{suavizado}$	143°	155°	129°
Velocidad máxima alcanzada	2.0 m/s	2.0 m/s	2.0 m/s
Velocidad promedio	1.85 m/s	1.3 m/s	1.87 m/s
Vueltas completadas sin incidentes	Más de 5	Más de 5	Más de 5

Nota. Datos obtenidos de experimentos en distintas pistas.

Conclusión del apéndice

Los resultados obtenidos en las tres pistas confirman que el sistema es capaz de operar de manera estable en diferentes configuraciones de pista. La variación en los valores de MSE y velocidad promedio sugiere que la geometría de la pista influye en el desempeño del vehículo, especialmente en su velocidad de operación. Sin embargo, el sistema mantiene una navegación segura y precisa en todos los casos, validando la efectividad del enfoque implementado.