industriales
etsii

Escuela Técnica Superior de Ingeniería Industrial

Universidad Politécnica de Cartagena

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Fusión sensorial Cámara y LiDAR para el seguimiento de múltiples objetivos. Aplicación a vehículos de conducción autónoma.

Camera & LiDAR sensory fusion for multi-target tracking. Application to autonomous driving vehicles.

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS INDUSTRIALES.

Autor: Francisco Miranda Alcaraz

Director: Pedro J. Navarro Lorente

Cartagena, 22 de abril de 2021

Universidad Politécnica de Cartagena

industriales etsii UPCT

Fusión sensorial Cámara y LiDAR para el seguimiento de múltiples objetivos. Aplicación a vehículos de conducción autónoma.

Camera & LiDAR sensory fusion for multi-target tracking. Application to autonomous driving vehicles.

Autor: Francisco Miranda Alcaraz

Director: Pedro J. Navarro Lorente

Cartagena, 22 de abril de 2021

Resumen

Con este Trabajo de Fin de Estudios se pretende realizar un estudio global de corte científico sobre las técnicas de percepción, detección y localización, utilizadas en la industria de los vehículos autónomos. Se incidirá en mayor medida sobre aquellos sistemas que, combinados, permitan una sencilla y correcta captación del entorno del automóvil, como los sensores de imagen (comúnmente conocidos como cámaras) y los sensores de detección por láser pulsado (LiDAR). Además, se aplicarán estos conocimientos técnicos para llevar a cabo simulaciones de los sensores del prototipo "Cloud Incubator Car", desarrollado por la Universidad Politécnica de Cartagena.

Este trabajo surge debido al creciente interés, tanto propio como de la industria, por la implementación de los vehículos autónomos en nuestras calles. Este hecho podría suponer no solo un aumento significativo de la seguridad vial (eliminando de raíz el factor accidental humano) sino también una reducción de emisiones contaminantes y tiempos de transporte.

Actualmente, el número de empresas que apuestan por este tipo de tecnología va en aumento. Algunas, como Tesla, propias del sector de la automoción, ya diseñan y manufacturan sus propios vehículos autónomos. Otras, como Apple, Intel o AMD, propias del sector de la informática, la electrónica y los dispositivos móviles, han dado el salto recientemente a la creación de algoritmos y microprocesadores destinados de forma exclusiva a la conducción autónoma.

Teniendo estos datos en mente, se pretende estudiar las diferentes técnicas de fusión de datos y hacer uso de la herramienta "Automated Driving Toolbox™", incluida en el entorno de programación de MATLAB®, para simular distintas situaciones de conducción que puedan surgir durante las pruebas en campo de un vehículo autónomo real. Asimismo, se modelará la fusión de sensores correspondiente al prototipo de la UPCT como una combinación de distintos tipos de cámaras y sensores LiDAR.

La intención de este documento es, en primer lugar, ser una fuente de información básica y actualizada sobre el estado del arte en materia de conducción autónoma; en segundo lugar, actuar como base bibliográfica para la simulación y modelado de fusión sensorial aplicada a los vehículos autónomos mediante MATLAB®; y, por último, servir de potencial punto de partida para proyectos futuros.

Abstract

The aim of this Final Degree Project is to carry out a scientific global study on perception, detection and location techniques used in the autonomous vehicle industry. Furthermore, this project will go more deeply in those systems which, combined, allow for a simple capture of the car environment, such as image sensors (cameras) or pulsed laser detection sensors (LiDAR). In addition, this technical knowledge will be applied to make different road simulations based on the "Cloud Incubator Car", an autonomous prototype developed by the Polytechnic University of Cartagena.

This work arises due to the growing interest, both mine and from the industry, for the implementation of autonomous vehicles in our streets. Not only could this mean a significant increase in road safety (rooting out the accidental human factor) but also a reduction of polluting emissions and transport times.

Nowadays, the number of companies betting on this type of technology is increasing. Some, such as Tesla, have already manufactured their own autonomous vehicles. Others, like Apple, Intel, or AMD, that come from the computer, electronics and mobile devices sector, have recently made the leap to creating algorithms and microprocessors exclusively for autonomous driving.

Bearing this data in mind, it is intended to study different data fusion techniques and make use of the 'Automated Driving Toolbox™', as part of the MATLAB® programming environment, to simulate different driving situations that may happen during actual driving field tests. The sensory fusion corresponding to the prototype vehicle will be modelled as a combination of different types of cameras and one LiDAR sensor.

The intention of this document is, firstly, to be a source of basic and updated information on the state of art in autonomous driving; secondly, to act as a bibliographic basis for the simulation of sensory fusion applied to driving using MATLAB®; and eventually, serve as a potential starting point for future projects.

Agradecimientos

A mi familia y a mi novia, por brindarme infinito apoyo incondicional.

A mis amigos, por ayudar a levantarme en los malos momentos.

A mi tutor Pedro J. Navarro, por guiarme con paciencia y sabiduría en este proyecto.

A ti, lector, por dar una oportunidad a la tecnología, la ingeniería y la ciencia.

Índice general

Índice de figuras

3.5.	Cámara infrarroja LWIR.	*https_//www.expo21xx.com/sensor/16834_st2_infrared_cameras_sensors/default.htm*
3.6.	Cámara de tiempo de vuelo (ToF).	*https://es.wikipedia.org/wiki/Cámara_de_tiempo_de_vuelo#/media/Archivo:PMD_CamBoard.png*
3.7.	Cámara térmica industrial.	*https://www.directindustry.es/prod/workswell-sro/product-113875-1736727.html*
3.8.	Cámara profesional de visión 360º.	*https://filmora.wondershare.com/virtual-reality/top-10-professional-360-degree-cameras.html*
3.9.	Sensores ultrasónicos de automóvil desarrollados por Bosch.	*https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/construction-zone-assist/ultrasonic-sensor/*
3.10.	Sensor LiDAR Velodyne HDL64E y nube de puntos generada mediante dicho sensor.	*https://velodynelidar.com/products/hdl-64e/* *https://proyectoidis.org/laser-lidar/*
3.11.	Comparación de las características de algunos sensores exteroceptivos (I).	*https://www.mdpi.com/1424-8220/19/3/648/htm*
3.12.	Comparación de las características de algunos sensores exteroceptivos (II).	*https://www.mdpi.com/1424-8220/19/3/648/htm*
3.13.	Fusión de datos. Esquema y terminología.	*esi.uclm.es/www/dvallejo/TFE/TFM_David_Frutos.pdf*
3.14.	Fusión de sensores en un vehículo utilitario autónomo.	*https://medium.com/lansaar/what-is-sensor-fusion-cb596f7cf446*
3.15.	Proceso de fusión de datos según la Junta de Directores de Laboratorios.	*esi.uclm.es/www/dvallejo/TFE/TFM_David_Frutos.pdf*
3.16.	Esquema de sistema de control difuso (recursivo).	*upcommons.upc.edu/bitstream/handle/2099.1/4376/03.pdf?sequence=28&isAllowed=y*
3.17.	Esquema del algoritmo bayesiano de fusión de datos.	*upcommons.upc.edu/bitstream/handle/2099.1/4376/03.pdf?sequence=28&isAllowed=y*
3.18.	Esquema del algoritmo de fusión de datos de Dempster-Shafer.	*upcommons.upc.edu/bitstream/handle/2099.1/4376/03.pdf?sequence=28&isAllowed=y*
3.19.	Algoritmo recursivo del filtro de Kalman.	*https://es.wikipedia.org/wiki/Filtro_de_Kalman*
3.20.	Nube de puntos generada a partir de un sensor LiDAR 3D de alta definición.	*https://www.cursosteledeteccion.com/donde-descargar-datos-lidar/*

5.1.	Izquierda: Cloud Incubator Car. Derecha: Renault Twizy.	*F.P.* *https://blog.japemasa.com/renault-twizy-urbano-pequeno-electrico/*
5.2.	Modificaciones realizadas sobre el sistema de dirección y frenado.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.3.	Arquitectura de comunicaciones del CIC.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.4.	Arquitectura del CIC a alto nivel.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.5.	Sistemas de percepción del CIC.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.6.	Definición de la dirección inicial de búsqueda en el algoritmo SCP.	*F.P.*
5.7.	Cálculo de trayectorias mediante el algoritmo SCP (I).	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.8.	Cálculo de trayectorias mediante el algoritmo SCP (II).	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.9.	Algoritmo SCP aplicado a un mapa real (I).	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.10.	Algoritmo SCP aplicado a un mapa real (II).	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.11.	Búsqueda de puntos clave sobre los obstáculos del entorno.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.12.	Algoritmo de Harris para la búsqueda de puntos clave en la ROI del vehículo.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.13.	Trayectorias finales generadas sobre el entorno del vehículo.	*https://repositorio.upct.es/bitstream/handle/10317/7755/2018-11.pdf*
5.14.	Magnitudes físicas por defecto del vehículo definido en DSD.	*F.P.*
5.15.	Medidas del Renault Twizy.	*https://www.renault.es/electricos/twizy/datos-tecnicos.html*
5.16.	Medidas del sensor LiDAR Velodyne HDL64E.	*https://www.goetting-agv.com/dateien/downloads/63-9194_Rev-G_HDL-64E_S3_Spec%20Sheet_Web.pdf*
5.17.	Definición física del CIC mediante DSD.	*F.P.*
5.18.	Parámetros del Sensor LiDAR en DSD.	*F.P.*

F.P: Fuente propia.

Listado de acrónimos

Acrónimo	Significado
ABS	Sistema Antibloqueo de Ruedas (*Antiblockiersystem*)
ADAS	Sistemas Avanzados de Asistencia al Conductor (*Advanced Driving Assistance Systems*)
ADT	Automated Driving Toolbox™
API	Interfaz de Programación de Aplicaciones (*Application Programming Interface*)
CA	Conducción Autónoma
CAN	Red de Control de Área (*Controller Area Network*)
CAV	Vehículos de Conducción Autónoma
CCD	Dispositivo de Carga Acoplada (*Charged-Coupled Device*)
CEA	Comisariado Europeo del Automóvil
CIC	Cloud Incubator Car
DARPA	Agencia de Proyectos de Investigación Avanzados de la Defensa (*Defense Advanced Research Projects Agency*)
DC	Operador Dependiente del Contexto
DFIG	Grupo Informativo de Fusión de Datos (*Data Fusion Information Group*)
DGT	Dirección General de Tráfico
DIW	Instituto Alemán para la Investigación Económica (*Deutsche Institut für Wirtschaftsforschung*)
DSD	*Driving Scenario Designer*
ECU	Unidad de Control de Motor (*Engine Control Unit*)
EE.UU.	Estados Unidos de América
EKF	Filtro de Kalman Extendido (*Extended Kalman Filter*)
EMS	Visión Sacádica Multifocal basada en Expectativas (*Expectation-Based Multifocal Saccadic Vision*)
GLONASS	Sistema de Navegación Global por Satélite
GPL	Licencia Pública Global (*Global Public License*)
GPS	Sistema de Posicionamiento Global (*Global Positioning System*)

GTL	*Ground Truth Labeler*
HIL	*Hardware-in-the-loop*
IA	Inteligencia Artificial
ICCC	Operador Independiente del Contexto con Comportamiento Constante
ICCV	Operador Independiente del Contexto con Comportamiento Variable
IMU	Unidad de Medición Inercial (*Inertial Measurement Unit*)
JDL	Junta de Directores de Laboratorios (*Joint Directors of Laboratories*)
KF	Filtro de Kalman
LiDAR	*Light Detection and Ranging*
LKAS	Sistema de Seguimiento de Líneas Viales (*Lane Keeping Assist System*)
LRS	Sistema de percepción de largo alcance (*Long Range System*)
LWIR	Infrarrojo de Onda Larga (*Long-Wave Infrarred*)
MIL	*Model-in-the-loop*
MIT	Instituto Tecnológico de Massachusetts
OGM	Mapeado mediante celdas de ocupación (*Occupancy Grid Mapping*)
OICA	Organización Internacional de Constructores de Automóviles
PCAP	*Packet Capture*
PDNR	*Park, Driver, Neutral, Reverse*
PRM	Mapa de Carreteras generado por Probabilidad (*Probabilistic Roadmap*)
Radar	*Radio Detection and Ranging*
RAE	Real Academia Española de la Lengua
RANSAC	Algoritmo de Consenso de Muestra Aleatoria (*Random Sample Consensus Algorithm*)
RCS	Sección Transversal de Radar (*Radar Cross-Section*)
ROI	Región de Interés (*Region of Interest*)
ROS	*Robot Operating System*
RRT	Árboles de Exploración Rápida Aleatoria (*Rapidly Exploring Random Trees*)
RTK	Navegación Cinética Satelital en Tiempo Real (*Real Time Kinematic*)

SAE	Sociedad de Ingenieros de la Automoción
SCP	Búsqueda de Puntos Cruzados (*Search for Cross Points*)
SIL	*Software-in-the-loop*
SLAM	Localización y Mapeado Simultáneos (*Simultaneous Location and Mapping*)
SRS	Sistema de Percepción de Corto Alcance (*Short Range System*)
SWIR	Infrarrojo de Onda Corta (*Short-Wave Infrarred*)
TCP	Protocolo de Control de Transmisión (*Transmission Control Protocol*)
ToF	Tiempo de Vuelo (*Time of Flight*)
UDP	Protocolo de Datagramas de Usuario (*User Datagram Protocol*)
UE	Unión Europea

Capítulo 1.

Introducción y objetivos

En este capítulo, se pretende dar al lector una visión global del vehículo autónomo: presentar sus fundamentos, las ventajas e inconvenientes que supone su implantación en el mundo y establecer de forma concisa el rumbo hacia donde se dirige la industria respecto al desarrollo de este tipo de automóviles. Además, se establecerán los objetivos del trabajo, tanto de forma general como específica, y la estructura del presente documento.

1.1. Introducción

1.1.1. El vehículo autónomo

Un vehículo autónomo, definido de forma informal, es cualquier sistema de transporte que permite simular las capacidades humanas durante la conducción. Estos vehículos se basan principalmente en la utilización de técnicas de detección en tiempo real para recolectar datos que puedan ser procesados por un sistema informático. Tras la ejecución de los algoritmos correspondientes, el vehículo autónomo en cuestión debe ser capaz de circular en consecuencia de forma precisa, evitando obstáculos, detectando e interpretando las diferentes señales viales y simulando las decisiones humanas.

El desarrollo de este tipo de vehículos no ha sido exclusivo de la era de la información, en la que nos encontramos actualmente. El ser humano siempre ha deseado innovar, y desde la creación del automóvil a finales del siglo XIX, se ha estudiado la posibilidad de introducir funcionalidades adicionales que mejoren la experiencia del usuario, la comodidad, o en este caso, la seguridad.

En los últimos años, el mundo ha presenciado grandes avances en materia de conducción autónoma. En el año 2015, eran ya más de 100 las grandes empresas de todo el mundo que apostaban por el desarrollo, de forma directa o indirecta, de coches autónomos y su implementación en la industria. Según datos de Intel y Strategy Analytics, los vehículos autónomos generarán aproximadamente 7 billones de dólares hasta el año 2050, posicionándose como una de las principales actividades industriales del mundo moderno [1].

Actualmente, la conducción autónoma se apoya en tres grandes disciplinas científicas que se interconectan entre sí:

- La **visión por computador**, encargada de convertir las imágenes del mundo real en datos numéricos interpretables por ordenadores convencionales.
- La **fusión de datos** (más concretamente, la **fusión de sensores**), mediante la cual se consigue información mucho más fiable, precisa y coherente, que será el punto de partida para los algoritmos implementados en los vehículos autónomos.

- La **inteligencia artificial** y las **redes neuronales**, siendo estas las herramientas más modernas en materia de conducción autónoma, en continuo desarrollo.

Tras un estudio del Instituto Alemán para la Investigación Económica (DIW), basado en el número de patentes registradas hasta el año 2017 relacionadas con la conducción autónoma, se llegó a la conclusión de que las marcas líderes en el desarrollo de este ámbito son las representadas en el siguiente gráfico (*ver Figura 1.2*) [2].

Cabe destacar que, aunque existen empresas como Tesla dedicadas al desarrollo y manufactura de forma íntegra de vehículos eléctricos con algún tipo de autonomía, existen otras muchas únicamente centradas en proveer a los gigantes de la automoción (como Daimler, BMW o Ford) de sistemas y tecnologías que puedan ser usados en sus prototipos, como ADAS, dispositivos de detección del entorno, o simplemente algoritmos de toma de decisiones. Con esto, se expande mucho más el mundo de la conducción autónoma, haciendo posible que empresas inicialmente no destinadas a la automoción puedan involucrarse en estos proyectos y apostar por el avance de la tecnología.

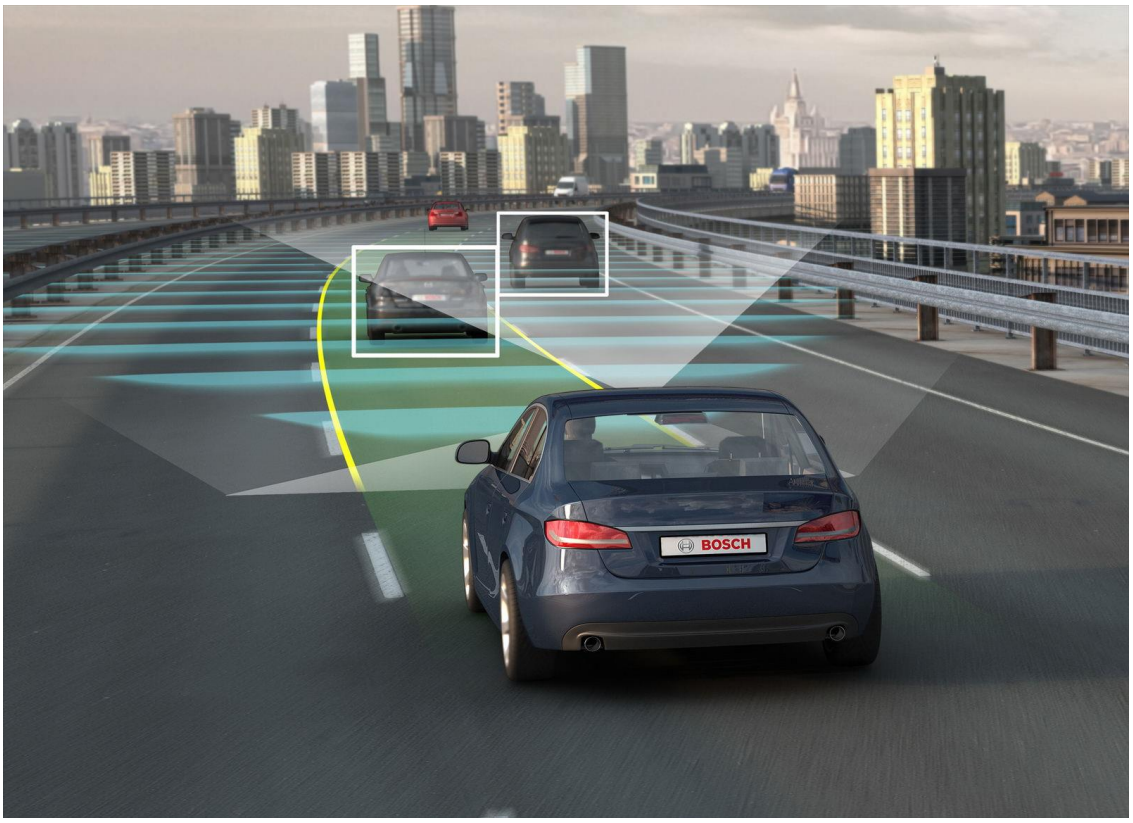


Figura 1.1. *La visión por computador, la fusión de datos y la inteligencia artificial son los principales pilares de los vehículos autónomos de la actualidad.*

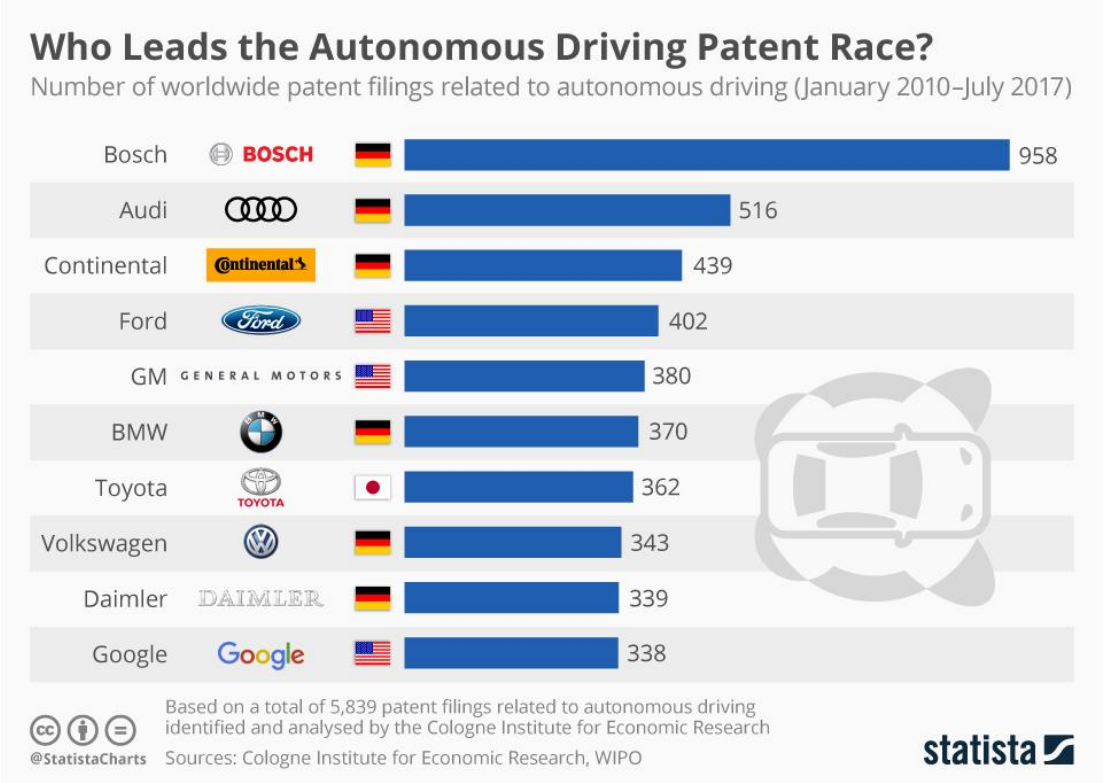


Figura 1.2. *Marcas líderes en el sector de la conducción autónoma, clasificadas por número de patentes desarrolladas, según el Instituto Alemán de Investigación Económica.*

1.1.2. Contexto socioeconómico y problemática actual

Tras la invención del automóvil, gran parte del esfuerzo y el trabajo de las empresas del mundo de la automoción ha estado destinado a la mejora de la seguridad durante la acción de transporte. Desde la creación del habitáculo indeformable, hasta la implantación relativamente reciente en todo vehículo de sistemas como el ABS, o el "Airbag", pasando por el desarrollo, simple pero efectivo, del cinturón de seguridad; cada una de estas mejoras incrementaba de forma progresiva la seguridad al volante.

Desgraciadamente, el número de automóviles matriculados asciende cada año y, con él, la cantidad de accidentes de tráfico. La causa de estos es, en la mayoría de los casos, un sistema para el que no existe mejora posible: el ser humano.

Según los datos expuestos en la jornada "*El factor humano en la causalidad de los accidentes de tráfico*" celebrada en 2017 en el Congreso de los Diputados, y organizada por la Comisión de Seguridad Vial de la DGT, entre un 71 y un 91% de los accidentes se deben a errores humanos que podrían haber sido evitados durante la conducción [3].

De esta forma, la implementación del coche autónomo pretende mejorar la seguridad en cada desplazamiento, eliminando el error humano, utilizando sistemas robustos que actúen de forma rápida y precisa ante cualquier imprevisto. Sin embargo, la transición tecnológica hacia la conducción autónoma conlleva tanto ventajas como diversos inconvenientes, que deben ser analizados para obtener una visión global de la situación.

La Organización Internacional de Constructores de Automóviles (OICA) diferencia las ventajas de la utilización de vehículos autónomos según el ámbito en el que presentan mejoras sustanciales frente a la conducción tradicional [4].

- **En el ámbito de seguridad vial**, se evitarían en su totalidad los errores humanos.
- **En el ámbito de organización del tráfico**, se optimizaría el flujo de vehículos y se ahorraría tiempo en los desplazamientos, aumentando la productividad y los tiempos de ocio.
- **En el ámbito de medioambiente**, la implantación de la conducción autónoma reduciría las emisiones contaminantes.
- **En el ámbito social**, la conducción autónoma ofrecería apoyo a conductores inseguros y, además, supondría un aumento de la movilidad de las personas de edad avanzada.
- **En el ámbito industrial**, la conducción autónoma promete innovación constante y el desarrollo de un nuevo sector.

A simple vista, puede parecer que la implementación de vehículos autónomos solo puede ser beneficiosa, aunque la realidad es que también presenta diversas desventajas difíciles de resolver a corto plazo.

- Tras un accidente, pueden surgir problemas relacionados con la aplicación de seguros.
- El precio de los coches autónomos, al menos en su etapa inicial de distribución al público, es muy elevado debido a la complejidad de los sistemas utilizados.
- Existirá una transición de profesionales desde el sector del automóvil tradicional al sector del vehículo autónomo, mediante la cual se perderán puestos de trabajo.
- Aparece un nuevo marco legal complejo sobre los posibles accidentes provocados por un vehículo autónomo. El Instituto Tecnológico de Massachussets (MIT), en colaboración con MediaLab, pone a disposición pública distintas pruebas, con el objetivo de valorar las decisiones que debería tomar un vehículo autónomo en caso de accidente inevitable [5].

Teniendo en mente estos inconvenientes, los expertos aseguran que la mejor forma de garantizar una transición tecnológica segura es introducir el vehículo autónomo de forma progresiva.

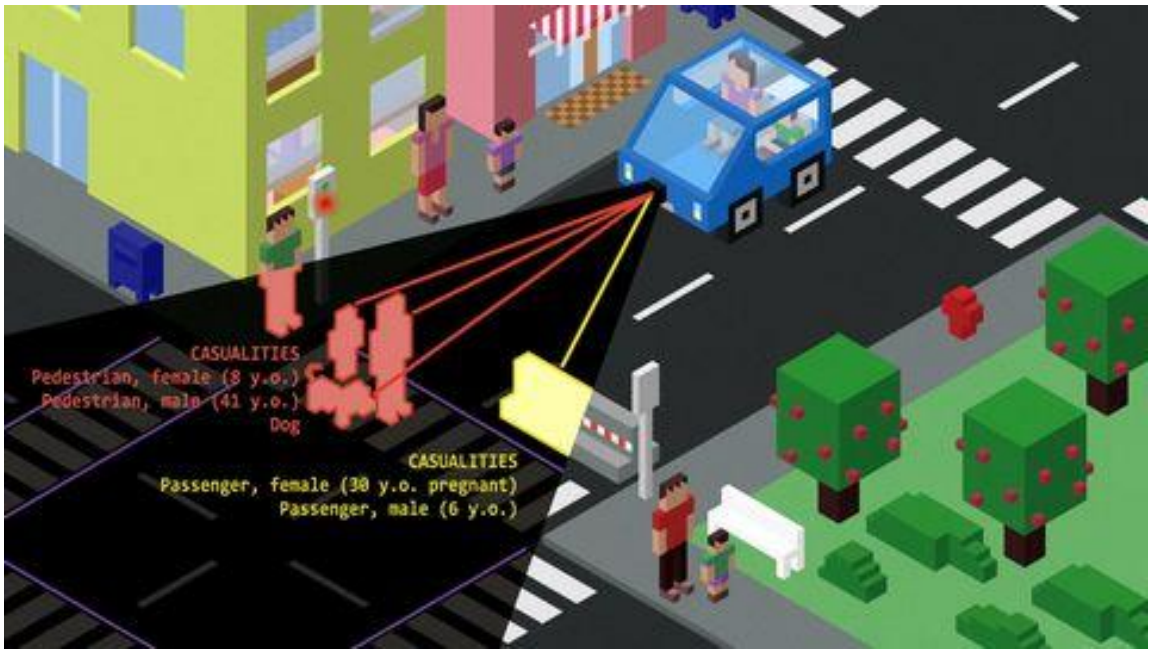


Figura 1.3. *Representación del dilema de la doble moral en entornos de conducción autónoma.*

1.1.3. Niveles de automatización en la conducción

El Comisariado Europeo del Automóvil (CEA) define los niveles de la conducción autónoma como una manera de cuantificar el grado de automatización implantado en vehículos autónomos de cualquier índole. Existen seis niveles de conducción autónoma, que dependen de tres agentes primarios fundamentales: el conductor como factor humano, el sistema de conducción autónoma y los sistemas convencionales del vehículo (entendiendo estos últimos como cada uno de los elementos propios de un vehículo tradicional).

Los niveles de automatización en la conducción fueron postulados por primera vez por la Sociedad de Ingenieros de la Automoción (SAE), y están regidos por el documento J3016, siendo este la referencia de la industria en materia de conducción autónoma [6].

Por lo tanto, se pueden definir estos seis niveles de la forma que sigue:

- **Nivel SAE 0**. Vehículos sin automatización en la conducción o vehículos "convencionales": en este tipo de automóviles, el conductor realiza todas las tareas que engloba la conducción, como acelerar, frenar, cambiar de marcha o controlar la dirección. También están incluidos en este nivel aquellos vehículos que hagan uso de asistentes a la conducción cuyas funciones no impliquen el control parcial del automóvil (por ejemplo, vehículos con sistemas de detección de distancias y objetos).
- **Nivel SAE 1**. Vehículos con automatización parcial: los vehículos incluidos en este nivel presentan asistentes a la conducción que pueden hacerse cargo del control longitudinal o lateral del vehículo, aunque nunca de ambos al mismo tiempo. Un vehículo con controlador activo de velocidad, o un sistema LKAS,

estaría incluido en este nivel. Actualmente, la mayoría de los vehículos manufacturados en la industria presentan este grado de automatización.

- **Nivel SAE 2**. Vehículos con automatización parcial: en este nivel se incluyen aquellos automóviles que hacen uso de sistemas de control avanzados para actuar sobre los movimientos longitudinal y lateral, aunque el conductor sigue siendo el único responsable de la tarea de conducción. Los vehículos de este nivel no están programados para actuar frente a obstáculos imprevistos, sino que presentan automatismos para tomar el control en situaciones recurrentes, como el aparcamiento o el control del vehículo en autovía. El primer coche con nivel 2 de automatización fue el Mercedes-Benz Clase S 2013 (*ver Figura 1.4*), pionero en la implementación de un asistente de conducción en atascos de tráfico. Hoy en día, son muchas las empresas que ofrecen este tipo de vehículos [4].
- **Nivel SAE 3**. Vehículos con automatización condicional: los automóviles pertenecientes a este nivel tienen la capacidad de moverse de forma autónoma, aunque el conductor debe actuar si existe un obstáculo o una situación imprevista.
- **Nivel SAE 4**. Vehículos con alta autonomía: en este nivel, el sistema de conducción autónoma permite guiar el vehículo de forma ininterrumpida, sin la necesidad de que el conductor tome el mando ante obstáculos o elementos viales imprevistos. Estos sistemas están programados para actuar en situaciones de peligro, analizando la información del entorno y respondiendo para minimizar los daños en caso de accidente. Sin embargo, los vehículos de nivel 4 poseen limitaciones geográficas, es decir, están programados para actuar en un entorno conocido. Cuando se sobrepasan dichos límites geográficos, es el conductor el que debe tomar el mando. Actualmente, existen ciertos prototipos de la mano de empresas como Waymo o Tesla que pertenecen a este nivel de automatización.
- **Nivel SAE 5**. Vehículos con autonomía y automatización total: en este último nivel, el sistema de conducción autónoma tiene un ámbito de funcionamiento global. Los vehículos pueden responder ante todo tipo de situaciones, superar cualquier obstáculo o actuar en consecuencia para llegar a la situación de mínimo riesgo, al igual que un conductor humano. Estos vehículos pueden prescindir de conductor en todo momento, y de elementos muy arraigados en la cultura del automóvil, como volante o pedales.

A primera vista, la aparición de los vehículos autónomos de nivel 5 en el mercado es todavía impensable para los próximos cinco años. Sin embargo, todas las señales apuntan a que no es algo del todo imposible. En un futuro, ya sea a medio o a largo plazo, los vehículos completamente autónomos se abrirán paso para ofrecer nuevas facilidades, comodidad, pero, sobre todo, seguridad.

Figura 1.4. *Mercedes Benz Clase S 2013. Primer vehículo de nivel SAE 2 comercializado*

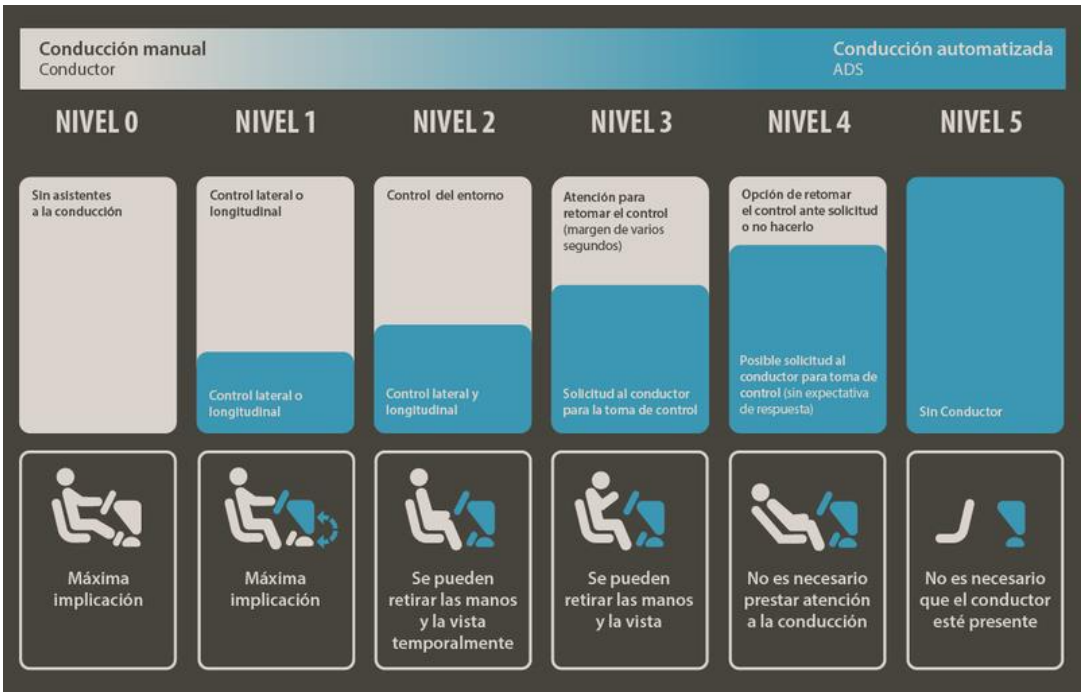


Figura 1.5. *Niveles de conducción autónoma.*

1.2. Objetivos

1.2.1. Objetivo general

El objetivo general de este trabajo es el estudio de los principales algoritmos y sistemas de percepción del entorno utilizados en los automóviles autónomos de la actualidad. Los esfuerzos se centrarán, sobre todo, en presentar la fusión de sensores "Cámara + LiDAR" como una herramienta sencilla y suficiente para la obtención de información del entorno de un prototipo de vehículo autónomo. Para ello, se hará uso de "Automated Driving Toolbox®", un paquete de herramientas dedicado exclusivamente al desarrollo de sistemas para la conducción autónoma, perteneciente al entorno de programación de MATLAB®, para realizar diversas simulaciones del "Cloud Incubator Car" [7], un prototipo diseñado por la Universidad Politécnica de Cartagena.

1.2.2. Requerimientos y estructura del documento

Una vez planteado el objetivo principal del Trabajo Fin de Grado, se realizará un recorrido por las diferentes tareas que cumplimentarán el proyecto:

- En primer lugar, se llevará a cabo un seguimiento de la historia del vehículo autónomo, así como un estudio del estado del arte sobre los sistemas actuales de percepción, localización y detección de la industria del automóvil.
- En segundo lugar, se realizará una aproximación a la fusión de datos aplicada a la conducción autónoma, traducida como fusión de sensores, y se explicarán los algoritmos principales usados para llevar a cabo el análisis de la información recibida.
- En tercer lugar, se plantearán las ventajas y fundamentos de programación de "Automated Driving Toolbox®" como herramienta de simulación vial para vehículos autónomos.
- En cuarto lugar, se hará un estudio básico del diseño del vehículo de la Universidad, el "Cloud Incubator Car", y se realizará un recorrido por sus principales herramientas de percepción del entorno. Asimismo, se llevará a cabo el modelado de este mediante MATLAB®.
- En quinto lugar, se realizarán simulaciones de la fusión de sensores del vehículo, planteando un entorno vial lo más aproximado a la realidad posible, y desarrollando un algoritmo que permita el seguimiento de objetivos durante la conducción.
- Por último, se presentarán las conclusiones del proyecto.

Capítulo 2.

Historia de la conducción autónoma

En este capítulo, se realizará un repaso de los más grandes acontecimientos históricos y avances relacionados con la conducción autónoma. Asimismo, se definirán conceptos que serán de gran importancia para el entendimiento de los siguientes capítulos.

2.1. El nacimiento de la conducción autónoma

En el año 1925, un ingeniero eléctrico de la armada de Estados Unidos llamado Francis Houdina, desarrolló un primer vehículo al que comúnmente se le considera el "precursor de los automóviles autónomos". El vehículo en cuestión, manufacturado por la empresa Houdina Radio Control, pudo maniobrar por la isla de Manhattan durante únicamente 19 kilómetros, hasta que colisionó con la parte frontal de otro vehículo. Aunque no se le puede considerar un vehículo autónomo propiamente dicho (ya que estaba controlado por su creador mediante un sistema de radio), fue calificado como el primer automóvil que podía realizar prácticamente todas las acciones necesarias para maniobrar por una ciudad sin un piloto a bordo [8].

Durante la década de los años 20 se dieron varios experimentos similares, todos con resultados cuestionables, y no fue hasta el año 1939 cuando Norman Bel Geddes, el diseñador industrial y teatral nacido en Michigan y fiel representante del futurismo, presentó en la exposición universal de Nueva York su primer proyecto de coche eléctrico conducido de forma autónoma mediante raíles. Este fue el principal detonante para que los gigantes de la automoción como General Motors se interesaran por estos avances, y, por consiguiente, por el nuevo futuro de la tecnología [9].

En 1977, el laboratorio japonés Tsukuba de ingeniería mecánica construyó el primer vehículo semiautomático, capaz de captar elementos de la conducción mediante dos cámaras y una computadora. Este vehículo pudo circular a aproximadamente 30 km/h con la ayuda de raíles, como predijo el futurista Bel Geddes.

Sin embargo, la mayoría de los avances en vehículos autónomos se registraron durante el último tramo del siglo XX, y se deben a la figura de Ernst D. Dickmanns y a su equipo. Dickmanns fue un ingeniero aeronáutico alemán, profesor en la Bundeswehr Universität München y experto tanto en inteligencia artificial como en visión por computador dinámica, que lideró el proyecto y manufactura del primer vehículo autónomo de la era moderna.

A principios de la década de los 80, su equipo comenzó a trabajar con una furgoneta Mercedes-Benz a la que equiparon con cámaras y otros sensores, y modificaron para que tanto la dirección, como los pedales, frenos y demás sistemas del automóvil pudieran

trabajar conectados a un controlador, cargado previamente con una aplicación software y algoritmos de conducción autónoma.

Este proyecto se alargó hasta el año 1987, cuando la furgoneta en cuestión, conocida como "VaMoRs", fue capaz de maniobrar por una autovía sin tráfico de Bavaria a una velocidad aproximada de 100 km/h de forma completamente autónoma, aunque siempre con supervisión [10].

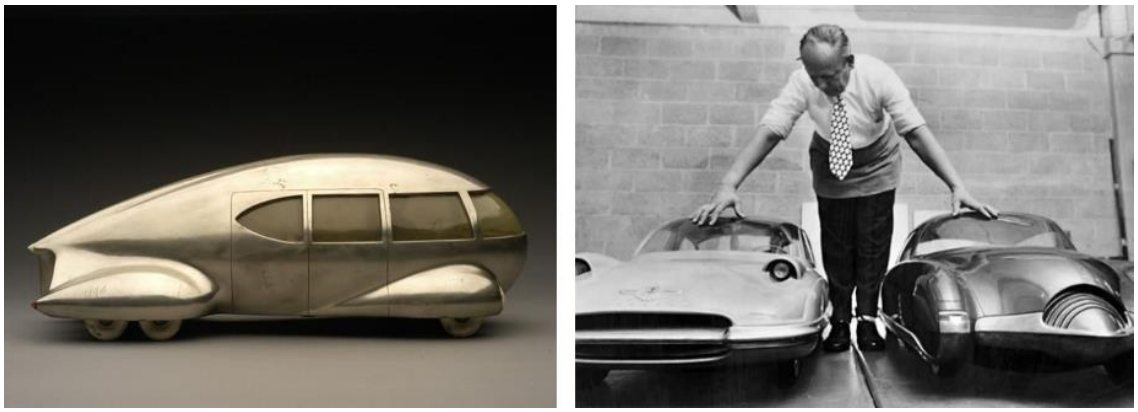


Figura 2.1. *Izquierda, vehículo autónomo de Norman Bel Geddes. Derecha, Ernst Dickmanns con dos prototipos (1979).*

Figura 2.2. *Izquierda, furgoneta autónoma Mercedes-Benz. Derecha, interior modificado de la furgoneta "VaMoRs".*

El desafío fundamental que presentaba la construcción y utilización de este tipo de vehículos en la década de los 80 era la alta velocidad a la que las cámaras y sensores debían detectar los elementos viales, comparada con las bajas velocidades de procesamiento de las computadoras de la época. Por lo tanto, Dickmanns tuvo que ingeniar distintas estrategias para poder utilizar dichos sensores y reaccionar a tiempo real. Se utilizaron algoritmos creados por el propio Dickmanns que simulaban los movimientos sacádicos del ojo, obteniendo información únicamente de zonas específicas de las imágenes en movimiento, como de las líneas viales, descartando otro tipo de datos. Además, mediante sistemas estadísticos, se pudo modelar el comportamiento del vehículo para eliminar ruido y prevenir comportamientos erróneos.

Este hito fue posible gracias a las empresas automovilísticas de toda Europa, lideradas por Daimler-Benz AG, que consiguieron poner en marcha el programa "*EUREKA*

Prometheus" (aprobado por la Comisión Europea) mediante el cual se destinaban 749 millones de dólares a las investigaciones en automoción [11].

A partir de entonces, los vehículos de conducción autónoma comenzaron a cambiar a pasos agigantados. Los sistemas utilizados para captar las imágenes comenzaron a modernizarse, dando lugar a muchas de las técnicas de captación del entorno que conocemos hoy en día. El equipo de Dickmanns continuó cooperando con Daimler-Benz, y aproximadamente 5 años más tarde, en 1992, se iniciaron las pruebas en vías con tráfico. Esto era esencial para poder depurar los algoritmos de conducción desarrollados, y para conseguir introducir los vehículos autónomos en la industria.

En 1995, el Mercedes-Benz Clase S W140 rediseñado por Ernst Dickmanns, se presentó como culmen del programa Prometheus. Este vehículo, junto con otros dos desarrollados por Daimler, recorrieron más de 1000 kilómetros de forma autónoma, tanto en las autopistas parisinas de tres carriles, como en las *Autobahn* alemanas, en las que no existe límite de velocidad para turismos.

Este vehículo no solo demostró que podía circular de forma pasiva, sino que también podía realizar adelantamientos, cambios de carril, elección de salidas, e incluso detectar la velocidad a la que debía circular en cada momento. Para conseguir esto, se utilizaron sistemas de cámaras profesionales tanto en la parte posterior como anterior del coche [9].

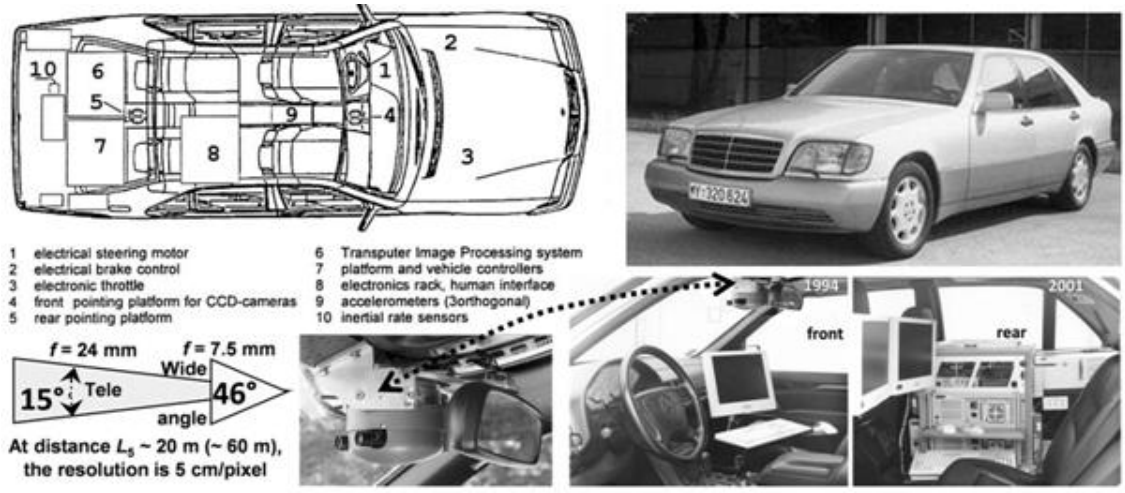


Figura 2.3. *Mercedes-Benz Clase S W140 modificado por Ernst Dickmanns y su equipo.*

A partir de 1995, los cambios en la industria del automóvil autónomo fueron constantes. Dickmanns siguió trabajando para Daimler-Benz, obteniendo cada vez más mejoras en sus vehículos. Por aquel entonces, un 95% de la conducción podía realizarse de forma autónoma. Hasta el año 2004, las investigaciones se centraron en perfeccionar la conducción por carreteras secundarias, asfalto inadecuado, campo a través, e incluso para evitar obstáculos de pequeño tamaño. Fue entonces cuando apareció la Visión EMS (*Expectation-based Multifocal Saccadic Vision*), desarrollada por la Bundeswehr Universität München. Estos nuevos sistemas se presentaron como la tercera generación de sistemas de visión artificial utilizados para la automoción, y seguían utilizando el

"Enfoque 4D" (*4D Approach*) que planteaba Dickmanns, basado en la estimación y modelado de la curvatura de la carretera [12].

Cabe destacar que, a la par de las investigaciones de Dickmanns y su equipo, existió otra corriente de desarrollo de vehículos autónomos impulsada tanto por la Carnegie Mellon University como por la Agencia de Investigación de Prototipos Avanzados de la Defensa, perteneciente al ejército de EE.UU. (DARPA). En 1997, este proyecto culminó con la demostración de su vehículo, que maniobró durante más de 4501 km del país, estableciendo un nuevo récord, con un 98% de la conducción realizado de forma completamente autónoma [13].

2.2. El vehículo autónomo en la actualidad

En la era actual de la información y las nuevas tecnologías, el desarrollo de vehículos autónomos se ha llevado al siguiente nivel. En el año 2014, un automóvil de competición Audi RS7 fue modificado para maniobrar de forma autónoma por el circuito de carreras de Hockenheim, al sur de Frankfurt. El coche en cuestión utilizó tanto cámaras como sistemas GPS, sensores láser, radiotransmisores y radares, que consiguieron que el vehículo circulara a una velocidad de 240 km/h, completando el circuito cinco segundos antes que un piloto humano [14]. Un año después, la compañía californiana Google dio el salto, y realizó pruebas con una flota de 25 vehículos autónomos diseñados en colaboración con Lexus, que circularon durante el día por la ciudad de Mountain View sin superar los 40 km/h. Estos recorrieron más de un millón de kilómetros de forma conjunta en aproximadamente nueve días [15].

Figura 2.4. *Prototipo de vehículo autónomo SAE 4 desarrollado por Waymo.*

Hoy en día, son muchas las marcas y empresas que apuestan por este tipo de tecnología e invierten grandes cantidades de tiempo y dinero para conseguir acercar el futuro a la humanidad. Algunas, como la anteriormente mencionada Google, Apple o AMD, ni siquiera son empresas destinadas a la automoción. Sin embargo, a pesar de todos estos esfuerzos, aún existe cierta incertidumbre sobre la implantación de este tipo de transporte en el mercado moderno.

Tanto la empresa Waymo (filial de Google) como Uber centran sus esfuerzos en desarrollar vehículos autónomos SAE 4 (nivel 4 de la conducción autónoma) basados en la tecnología de vallado virtual o *geofencing* que puedan actuar como transporte público autónomo en zonas urbanas. A finales de 2018, la propia Waymo comenzó un proyecto piloto para ofrecer este servicio en zonas de la ciudad de Phoenix (Arizona, EE.UU) [16].

"Algunas compañías están centrando sus esfuerzos en el desarrollo de otros ámbitos de funcionamiento en modo autónomo, tanto en coches como en camiones; por ejemplo, determinadas rutas por autopista. Otras, como por ejemplo Tesla o Audi, también trabajan para llevar las soluciones tecnológicas de la conducción autónoma al vehículo particular en un sentido más general, con el objetivo de evolucionar hacia el nivel 5, aunque no se espera que este sea una realidad a corto o medio plazo", comenta Juan Carlos López, redactor de la revista de tecnología *Xataka*, tras el Salón del Automóvil de Ginebra [17].

En la actualidad, no existen vehículos comercializados que presenten un grado de autonomía SAE 5, sin embargo, son una prioridad para cientos de empresas de la automoción. La mayoría de los pronósticos (*ver Tabla 1.1*) coinciden en que llevará cierto tiempo hacer que estos vehículos sean vistos como algo habitual en nuestras calles, no por el hecho de poseer tecnologías muy avanzadas (es bien sabido que esta se desarrolla de manera casi exponencial) sino por la ruptura de la convencionalidad en la cultura del automóvil [18].

Tabla 2.1. Previsiones sobre la implementación de los vehículos autónomos.

Fuente	Nivel SAE 5	Entorno CAVs
Underwood, 2014	2025-2035	2040-2060
Godsmark, 2015	2020-2025	2020-2030
Shladover, 2016	2075	¿?
Zmud, 2017	2025-2030	¿?
Bloomberg, 2017	2028-2030	2040-2060
Litman, 2018	2020-2040	2060-2080
Kuhnert, 2018	2025-2030	¿?
Gehrke, 2018	2018-2021	2040-2050
SSCTCC, 2018	2040-2050	2040-2060
Shaheen, 2018	2023-2040	2045-2070

Fuente: *Martínez-Díaz, M; Soriguera, F. (2018). "Autonomous vehicles: theoretical and practical challenges".* ***Publicado por****: Elsevier & ScienceDirect.*

Capítulo 3.

Revisión científica del estado del arte

El objetivo de este capítulo es proporcionar al lector conocimientos actualizados sobre las tecnologías más utilizadas en vehículos autónomos. Se hará una revisión de los métodos y técnicas de percepción, localización, construcción de mapas y planificación de trayectorias, cuyo uso aplicado a la conducción es ciertamente común. Asimismo, se realizará una primera aproximación a la fusión de datos y fusión de sensores, definiendo sus fundamentos y los algoritmos más comunes.

3.1. Sistemas físicos (I): dispositivos de percepción del entorno

Un vehículo autónomo tiene como objetivo, de forma básica y en primera instancia, obtener una información de su entorno. En segundo plano, esta información es procesada y, mediante los algoritmos y la programación correspondiente, el vehículo en cuestión puede actuar en consecuencia.

Por lo tanto, podemos decir que la percepción, definida por la Real Academia Española (RAE), como "*acción y efecto de percibir*" o "*sensación interior que resulta de una impresión material hecha en nuestros sentidos*", es la tarea primaria de un vehículo autónomo, además de transportar personas o carga.

Los sensores son, en definitiva, los órganos de los sentidos del propio vehículo autónomo, y se encargan de captar la información de este para que pueda ser procesada posteriormente.

En el ámbito científico, los sistemas de percepción se clasifican según el estímulo que se desea percibir. De esta forma, podemos encontrar:

- **Sistemas de percepción visual**: cuyo principal exponente son las cámaras alojadas en los vehículos.
- **Sistemas de percepción espacial**: como el LiDAR 3D, o sistemas de ultrasonidos.
- **Sistemas de percepción auditiva**: como los micrófonos o los captadores piezoeléctricos.
- **Sistemas de percepción térmica**: como las cámaras térmicas, termopares o termistores.
- **Sistemas de percepción de posición absoluta**: como los giroscopios.
- **Sistemas de percepción de posición relativa**: como los acelerómetros.
- **Sistemas de percepción del campo magnético**: como las brújulas digitales.

Los sensores pueden ser clasificados según su tipo de funcionamiento en digitales y analógicos. Los sensores digitales transforman la información captada en impulsos eléctricos con valor discreto, por ejemplo, en una señal de 0 o 5 V (encendido o apagado). Sin embargo, los sensores analógicos son capaces de medir magnitudes intermedias, es decir, un rango continuo de valores.

Para decidir si un sensor es adecuado para nuestro sistema, es conveniente conocer ciertas características de estos, como la fiabilidad (capacidad del sensor de obtener datos estables, y semejantes a los de la realidad), la precisión (capacidad del sensor para medir valores muy próximos entre sí), el rango de valores (definición de los valores máximos y mínimos que el sensor es capaz de medir), la velocidad de muestreo (tiempo entre mediciones llevadas a cabo por el sensor) o el coste.

Algunos factores, como la fiabilidad, pueden ser mejorados mediante técnicas como la redundancia de sensores; otros, como la velocidad de muestreo o el rango de medidas, deben ser cuidadosamente analizados para que la elección de los sensores sea adecuada.

Los sensores pueden ser clasificados, a su vez, según las magnitudes que son capaces de medir (*ver Tabla 3.1*).

Tabla 3.1. Clasificación de sensores según su magnitud de medida.

Tipo de sensor	Ejemplos
Sensores lumínicos	Fotorresistores, celdas fotovoltaicas, diodos fotorreceptores, fototransistores, dispositivos de carga acoplada, cámaras o sensores de imagen.
Sensores de presión o fuerza	Microinterruptores, galgas extensiométricas, sensores de fuerza piezoeléctricos, sensores de contacto.
Sensores de sonido	Micrófonos, captadores piezoeléctricos, rangers ultrasónicos.
Sensores de posición absoluta, relativa, distancia y proximidad	Sensores ultrasónicos, sensores de distancia por haz infrarrojo, sensores de proximidad capacitivos e inductivos, radares, LiDAR, tacómetros, encoders, giroscopios, acelerómetros, sensores pendulares, contactos de mercurio.
Sensores de temperatura	Termistores, termopares, termorresistencias, diodos térmicos, pirosensores.
Sensores de humedad	Sensores capacitivos, resistivos, módulos integrados.
Sensores de magnetismo	Sensores de Efecto Hall, brújulas digitales y electrónicas, interruptores magnéticos.
Sensores de ubicación	GPS, GLONASS, Galileo, radiobalizas.

Fuente: *Pallás Areny, Ramón. (2003). "Sensores y acondicionadores de señal".* ***Publicado por****: Universidad Politécnica de Cataluña.*

Los vehículos autónomos de la actualidad basan su capacidad sensorial en unos pocos dispositivos, suficientes para captar con cierta exactitud el entorno que los rodea. Los sensores deben poder producir una vista tanto perceptiva como de ubicación del entorno para que el vehículo sea capaz de tomar decisiones en tiempo real. Estos sensores se clasifican en dos tipos principales: los sensores exteroceptivos, que se utilizan para percibir el entorno y para calcular la distancia entre determinados objetos, y los sensores propioceptivos, utilizados para medir valores del propio sistema de transporte y movimiento, por ejemplo, la velocidad, posición de las ruedas, ángulos de articulación, etc. [19].

En la siguiente sección, analizaremos los sensores más utilizados en vehículos autónomos:

3.1.1. Sensores exteroceptivos

Se denomina sistema exteroceptivo a los receptores o sensores que permiten captar estímulos de origen exterior. Al igual que la piel, los órganos oculares o auditivos, este tipo de sensores permite percibir información del entorno. De esta forma, podemos encontrar:

- **Cámaras o sensores de imagen:**

Los sensores de imagen, conocidos de forma habitual como cámaras, se basan en el principio de funcionamiento del ojo humano para obtener información del entorno. La conducción autónoma actual depende del desarrollo y utilización de este tipo de sensores, siendo sus principales herramientas de percepción. La ventaja fundamental de estos sistemas es su capacidad de detectar colores y texturas. Además, el uso de un conjunto de cámaras permite determinar la distancia que existe entre objetos. Asimismo, son sistemas relativamente baratos, comparados con otros sensores usados en automoción. Sin embargo, la capacidad de procesamiento de datos requerida para su aplicación debe ser elevada (se utilizan junto a complejos algoritmos de visión artificial).

Las cámaras usadas en vehículos autónomos generan un conjunto de imágenes superpuestas del entorno visible del automóvil que permiten captar información procedente de obstáculos, líneas de carretera, señales de tráfico, peatones u otros vehículos en circulación [20]. Sin embargo, son sistemas sensibles a la intensidad lumínica, por lo que los fabricantes coinciden en que su uso debe estar acompañado de otro tipo de sensores mediante técnicas de fusión de datos. Los principales tipos de cámaras se recogen en la siguiente tabla.

Tabla 3.2. Tipos de cámaras.

Cámara	Descripción
Matriciales o de área (3.1)	Están formadas por una matriz de diodos CCD que convierte la energía lumínica en energía eléctrica. Han sido utilizadas históricamente en cine y televisión, y actualmente son el principal elemento de visión artificial en la industria.
Lineales (3.2)	Construyen la imagen línea a línea a través de un sensor que se desplaza respecto a la imagen u objeto a capturar. Este tipo de cámaras fueron específicamente desarrolladas para la inspección de materiales fabricados en continuo, como en la industria textil.
De alta velocidad (3.3)	En determinadas aplicaciones, como en el caso de la conducción autónoma, es necesaria la utilización de sensores que sean capaces de detectar un mayor número de fotogramas para un mejor análisis de datos. Existen cámaras capaces de capturar hasta 1 millón de fotogramas por segundo.
3D (3.4)	Este tipo de cámaras permite realizar diferentes perfiles de imagen de un objeto, para, tras el procesamiento adecuado, recrear la imagen binocular en tres dimensiones, como si del ojo humano se tratara. Este procedimiento puede ser útil en la detección de medidas de profundidad del entorno.
Infrarrojas (3.5)	Estas cámaras son capaces de detectar longitudes de onda más allá del espectro visible, entre los 750 y 14000 nanómetros. Su uso está ampliamente extendido en la industria, aunque se trata de sistemas con un elevado coste.
Térmicas (3.6)	Las cámaras térmicas pertenecen a la familia de las cámaras infrarrojas, con la diferencia de que estas detectan una longitud de onda de entre 8000 y 14000 nanómetros (LWIR), correspondiente al calor corporal. Son usadas en gran medida para detectar la presencia de formas de vida animal.
Panorámicas o de visión 360º (3.7)	Este tipo de cámaras, que trabajan usualmente en el espectro visible, se utilizan para obtener fotografías superpuestas, formando una imagen global del entorno. Su uso está ampliamente extendido dentro de los sistemas de mapeado virtual, y creación de escenarios 3D.
De Tiempo de Vuelo (3.8)	Este tipo de cámaras se basa en la tecnología TOF, que permite calcular el tiempo transcurrido entre la emisión y la recepción de un haz de luz infrarroja, para conseguir imágenes en tres dimensiones. Se trata de sistemas sencillos y fáciles de aplicar, siendo uno de los principales métodos de detección en vehículos autónomos.

Fuente: *EDS Robotics. (2020). "Cámaras industriales, tipos y funcionamiento".* ***Obtenido de****: https://www.edsrobotics.com/blog/camaras-industriales/*

Tabla 3.2. Tipos de cámaras.

Figura 3.1. ***Cámara de sensor matricial***

Figura 3.2. ***Cámara de sensor lineal***

Figura 3.3. ***Cámara de alta velocidad***

Figura 3.4. ***Cámara de visión 3D***

Figura 3.5. ***Cámara infrarroja LWIR***

Figura 3.6. ***Cámara de Tiempo de Vuelo***

Figura 3.7. ***Cámara térmica industrial***

Figura 3.8. *Cámara industrial 360º*

- **Sensores ultrasónicos:**

Los sensores de ultrasonidos emiten ondas sonoras fuera del espectro auditivo, que impactan con los objetos del entorno, y son recogidas por un receptor. Mediante la diferencia de tiempos entre la onda emitida y la recibida, se pueden estimar de forma precisa las distancias a las que se encuentran los elementos del entorno. La eficacia de estos sensores alcanza su máximo cuando el vehículo autónomo circula a bajas velocidades. Además, funcionan especialmente bien con obstáculos cercanos.

Los sensores de ultrasonidos son relativamente baratos, comparados con los expuestos en este apartado. Se trata de sistemas robustos y seguros, con años de desarrollo a sus espaldas. Actualmente, son usados en sistemas de aparcamiento automático y detección de obstáculos en ruta. Su principal desventaja reside en que las ondas de sonido necesitan un medio para propagarse, por lo que cambios en la temperatura o la humedad del aire pueden afectar al diagnóstico de medición, aunque existen algoritmos desarrollados para lidiar con este tipo de problemas.

Figura 3.9. *Sensores ultrasónicos de automóvil desarrollados por Bosch.*

- **Radar:**

El radar es un tipo de sensor que utiliza ondas de radio para medir la distancia, el ángulo y la velocidad de determinados objetos, basándose en los principios de la radiación electromagnética. La palabra radar es un término derivado del acrónimo *Radio Detection and Ranging* (Detección y medición por ondas de radio).

Este tipo de sensor ha sido utilizado históricamente en gran parte de las aeronaves y embarcaciones modernas. Su principio de funcionamiento reside en la emisión de un impulso de radio, que se refleja en el entorno, y es recibido de nuevo por el emisor. Gracias a este eco electromagnético, se puede obtener gran cantidad de información. El uso de las ondas de radio permite a los radares detectar objetos fuera del espectro visible.

Cuanta mayor sea la frecuencia del radar, mayor será la resolución de este, ya que permitirá al sensor captar más número de pulsos en el mismo tiempo. Usualmente, los radares son clasificados como sensores de medio alcance (entre 50 y 100 m). Su uso ofrece muchas ventajas en el ámbito de la conducción autónoma: al ser sensores muy

desarrollados, son sistemas relativamente robustos, con un costo mucho menor que el de los sensores LiDAR.

- **LiDAR:**

La palabra LiDAR es el acrónimo de *Light Detection and Ranging* (Detección y Medición Lumínica). Estos dispositivos, relativamente modernos, se basan en el mismo principio que las cámaras de tiempo de vuelo. Utilizan una fuente de iluminación propia, usualmente un láser de alta frecuencia, que se refleja en el entorno y es captado de nuevo por el sistema, midiendo el tiempo que tarda el haz lumínico en realizar el recorrido. Estas mediciones se pueden utilizar para generar entornos 3D precisos, ya que los sensores LiDAR de la actualidad son capaces de generar más de 150000 pulsos por segundo. Además, son sensores de largo alcance, con un rango de medición de más de 250 m.

Los sensores LiDAR ofrecen muchos beneficios para los vehículos autónomos, debido a su precisión en la detección. Usualmente son utilizados para generar nubes de puntos de alta resolución, obteniendo así una imagen real y precisa del entorno del automóvil. Asimismo, los datos generados por esta nube de puntos pueden ser procesados por un sistema informático. Es común encontrar este tipo de sistemas en vehículos que realizan las tareas de percepción y mapeo del entorno de forma simultánea (SLAM, *Simultaneous Location and Mapping*).

Al ser una tecnología recientemente implementada en vehículos autónomos, aún no se han explorado completamente sus posibilidades, en parte debido a su alto coste y baja disponibilidad. Los sensores LiDAR son complejos sistemas de espejos mecánicos y fuentes lumínicas, que ofrecen una visión de 360º, pero pueden llegar a costar decenas de miles de euros [21].

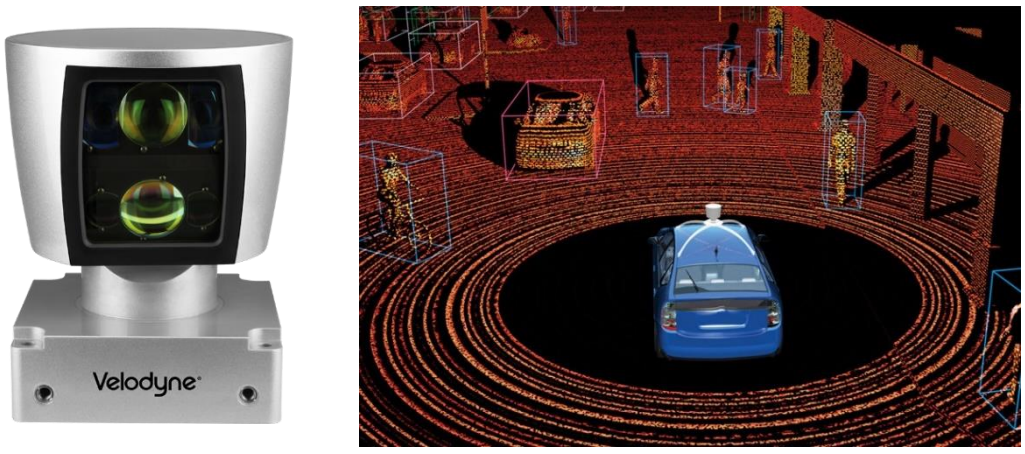


Figura 3.10. *Sensor LiDAR Velodyne HDL-64E y nube de puntos generada con dicho sensor.*

3.1.2. Sensores propioceptivos

Se denominan sensores propioceptivos a aquellos usados para captar estímulos del interior del vehículo autónomo, o ligados a su posición intrínseca en el espacio. Si realizamos una analogía, los sensores propioceptivos serían aquellos capaces de captar el

dolor en el cuerpo humano (indicando que existe algún fallo en nuestro organismo) o aquellos capaces de reconocer la posición de cada parte de nuestro cuerpo en todo momento, como el sistema vestibular. De esta forma, en materia de conducción autónoma podemos encontrar:

- **GPS (Sistema de posicionamiento global):**

El GPS y sus derivados, como los sistemas GLONASS, Galileo o Beidou, son sistemas de navegación por satélite que proveen de geolocalización tanto a dispositivos móviles como a vehículos autónomos, con un error de medida de hasta unos metros (dependiendo de la intensidad de la señal y la posición de los satélites). Fue desarrollado por el Departamento de Defensa de Estados Unidos, y actualmente es el sistema de posicionamiento global más extendido en el mundo, estando presente en casi todos los dispositivos de uso común, como los teléfonos móviles, las tabletas o las computadoras.

Para la detección de la posición del dispositivo, este recibe información de al menos cuatro satélites localizados en la órbita terrestre, con la hora de salida de cada señal. Después, calcula la diferencia de tiempos entre las señales y el dispositivo receptor, obteniendo las distancias a cada uno de los satélites. Finalmente, se utiliza el método de trilateración inversa, que hace uso de geometría triangular y circular, para calcular la posición relativa de cada satélite respecto al dispositivo receptor.

La principal desventaja del uso de GPS en sistemas de conducción autónoma es que la señal de los satélites puede verse afectada por diversos factores, como edificios altos en los alrededores del vehículo, túneles, árboles, etc. [22].

Tabla 3.3. Comparación de sistemas de posicionamiento global.

	GPS	GLONASS	Galileo	Beidou
Satélites	24	24	30	35
Precisión	7,8 m (civil) 5,9 m (militar)	7,4 m (civil) 4,5 m (militar)	1,0 m (civil) 0,01 m (militar)	10 m (civil) 0,1 (militar)
Cobertura	Global	Global	Global	República Popular de China
Período de órbita	11 h 58 min	11 h 15 min	14 h	12 h 53 min
Altura	26650 km	19100 km	23222 km	21150 km
Propietario	EE.UU.	Rusia	Unión Europea	China

Fuentes: *Eisfeller, B; Ameres, G; Kropp, V; Sanroma, D. (2007) "Performance of GPS, GLONASS, and Galileo".* ***Publicado por****: In Photogrammetric Week '07, Germany*

Chen H; Huang, Y; Chiang, K; Yang, M. (2009) "The performance comparison between GPS and Beidou/Compass: A perspective from Asia". ***Publicado por****: Instituto de Ingeniería de China.*

- **IMU (Unidad de medición inercial):**

Las unidades de medición inercial, conocidas por sus siglas en inglés IMU, son dispositivos electrónicos capaces de medir velocidad, orientación y fuerzas gravitacionales del sistema en el que se encuentra instalado. La IMU es utilizada en sistemas de navegación de aviones, barcos, vehículos no tripulados, y coches autónomos.

Estos sistemas consisten en tres acelerómetros, giróscopos y magnetómetros, uno por cada eje espacial ortogonal, que permiten al vehículo calcular su velocidad lineal, aceleración y ángulo de inclinación frente a su posición estándar de ruta. La principal desventaja de estos sistemas es el requerimiento de GPS para corregir determinados errores que afectan a sus mediciones. Mientras que las Unidades de Medición Inercial proporcionan información de la posición relativa del vehículo y sus derivados, como la velocidad, no son capaces de informar sobre la posición global, de la que se encargan los sistemas de GPS. Además, estos sistemas realizan mediciones basadas en la posición anteriormente calculada, por lo que puede existir un error acumulativo conocido como "error de deriva", que debe ser continuamente corregido mediante los sistemas de posicionamiento global.

- **Encoders**:

Los encoders son dispositivos electromecánicos que convierten la posición lineal o angular de un eje en una señal, ya sea analógica o digital, mediante un transductor. Proporcionan información sobre la posición, la dirección y la velocidad del sistema. Existen dos tipos principales de encoders: los incrementales, que generan un número determinado de pulsos por revolución, y los absolutos, que generan un flujo de bits correspondiente a una posición discreta.

Estos sensores suelen utilizarse para obtener datos de velocidad, e incluso la distancia recorrida desde un punto de partida conocido. Son sensores baratos y fáciles de implementar, sin embargo, pueden producir errores, debido a que estos se encuentran ligados al giro de las ruedas.

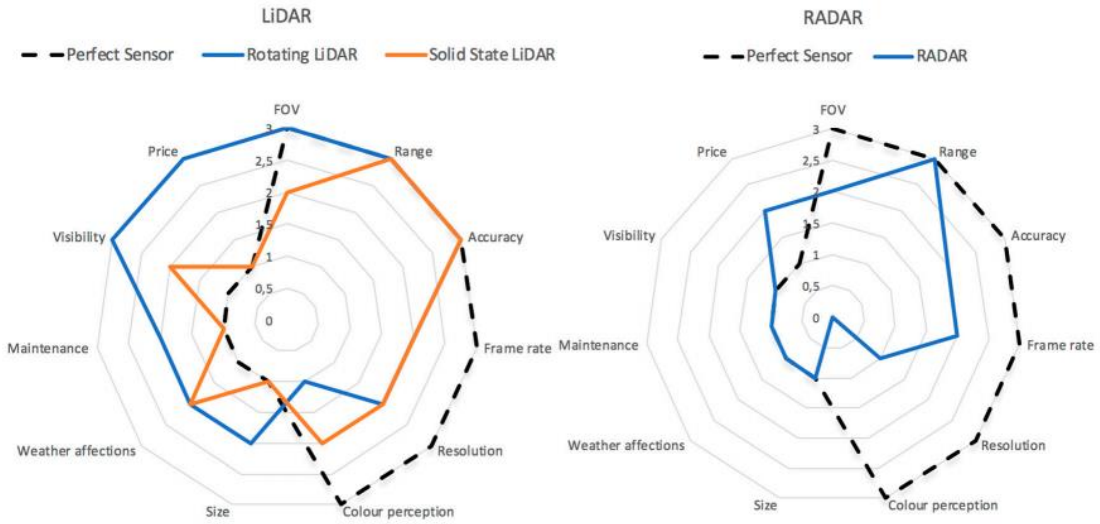


Figura 3.11. *Comparación de las características de algunos sensores exteroceptivos (1).*

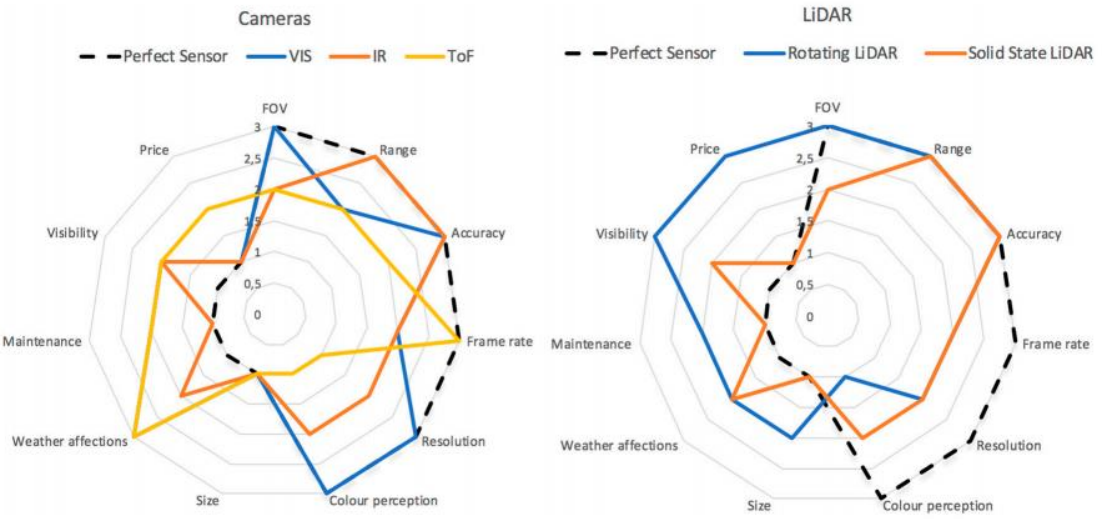


Figura 3.12. *Comparación de las características de algunos sensores exteroceptivos (2).*

3.2. Sistemas físicos (II): introducción a la fusión sensorial

3.2.1. ¿Qué es la fusión de datos?

La fusión de datos o fusión de información es un conjunto de técnicas en diferentes ámbitos de aplicación, que se encargan de gestionar la información recibida de diversas fuentes o dispositivos, con el fin de obtener unos resultados fiables basados en el mundo exterior. Su principal objetivo es conseguir resultados prácticos de mejor calidad que los obtenidos mediante una sola fuente.

El número de sistemas actuales en los que se aplica la fusión de datos es elevado debido a su carácter multidisciplinar; desde el ámbito militar, para el cual fueron originalmente desarrolladas este tipo de técnicas, hasta la diagnosis médica, la robótica y el control de procesos. Cabe destacar que, desde principios de los años 2000, esta técnica ha sido uno de los principales pilares de trabajo de las empresas destinadas al mundo de la conducción autónoma [23].

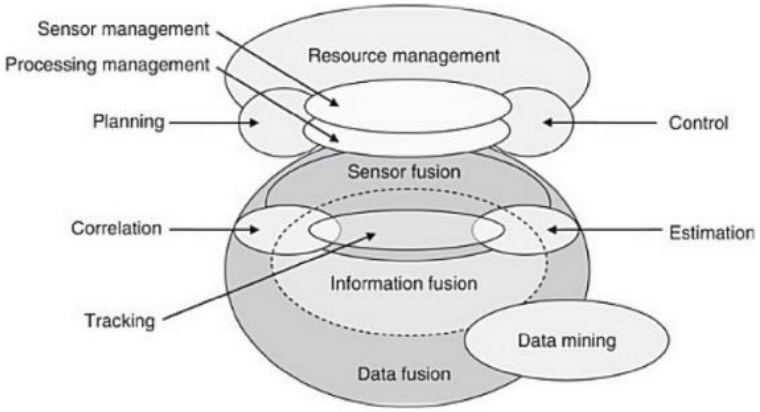


Figura 3.13. *Fusión de datos. Esquema y terminología.*

3.2.2. Fusión de sensores. Ventajas sobre los sistemas de percepción mono-sensor

Los vehículos autónomos, tal y como se ha presentado en el *Apartado 1.1*, poseen una gran cantidad de sensores que proporcionan información en bruto, es decir, sin ningún tipo de tratamiento aplicado. Las técnicas de fusión de sensores se encargan, mayormente, de la gestión y clasificación de esta información [24], y comprenden un subconjunto dentro de la fusión de datos.

Aunque puede parecer que el costo computacional de este tipo de técnicas es muy elevado frente a los sistemas mono-sensor, estos últimos presentan diversas desventajas, que aquí se exponen:

- El nivel de incertidumbre es invariable, es decir, las mediciones realizadas por los sensores poseen un nivel de fiabilidad que no puede ser modificado. Si un sistema mono-sensor genera un error, este será imposible de eliminar mediante técnicas tradicionales.
- La mayoría de los sensores y transductores miden una única magnitud física. Consecuentemente, se pierde información del entorno que podría ser percibida.
- Sin sistemas redundantes, el fallo del sensor implica el fallo total del sistema.

Por lo tanto, es necesario el uso de sistemas de fusión de sensores para obtener una visión completa del entorno de los vehículos autónomos, además de la aplicación de redundancia de dispositivos (que reduce la incertidumbre del sistema y lo protege de fallos de funcionamiento).

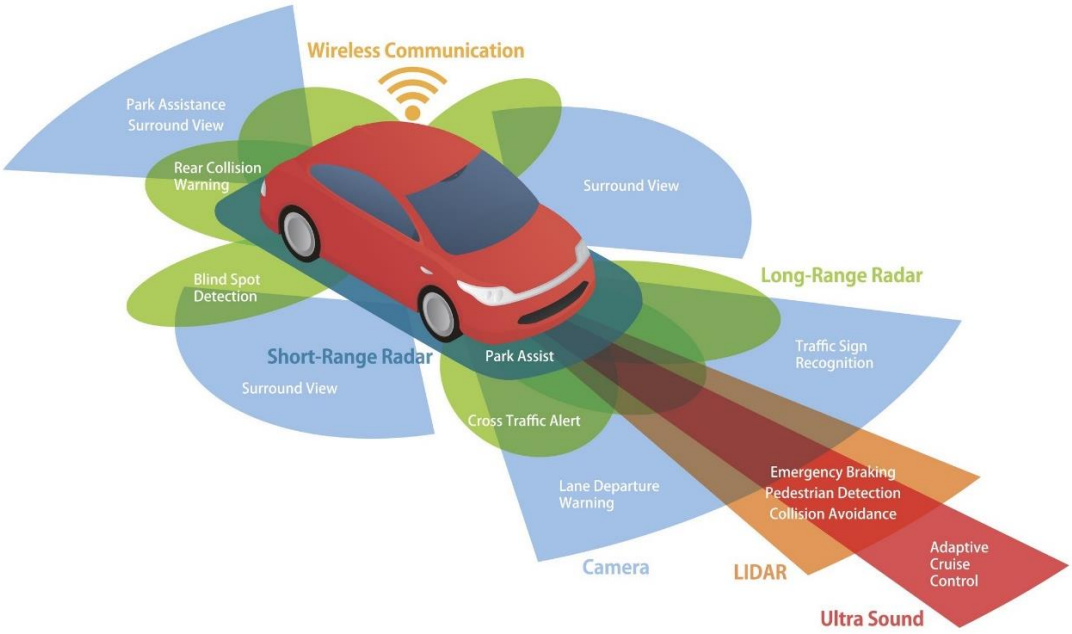


Figura 3.14. *Fusión de sensores en un vehículo utilitario autónomo.*

3.2.3. Niveles de fusión

La Junta de Directores de Laboratorios (JDL) creó, a mediados de la década de los 80, el *Data Fusion Information Group* (DFIG), que definió la fusión de datos como *"un proceso integral de tratamiento de datos, que se divide en niveles, desde la captura o percepción de los datos del entorno, hasta el resultado final"*. De esta forma podemos encontrar seis niveles diferentes [23] (*ver Figura 3.15*):

- **Nivel 0**: alineación de datos: este nivel comprende el pre-procesado de datos, métodos como el filtrado, la corrección, o la clasificación temporal de los mismos. Es por esto por lo que se conoce como "alineación" de datos por parte de las fuentes [23].
- **Nivel 1**: evaluación de objetos: en este nivel se lleva a cabo una combinación de los datos de los sensores para obtener características y propiedades básicas del objeto evaluado, como la posición, la velocidad, etc. Los algoritmos usados en este nivel van desde la combinación de modelos físicos y estadísticos para obtener el estado del objeto, hasta métodos paramétricos (como los modelos de Bayes o Dempster-Shafer) y no paramétricos o de lógica difusa (redes neuronales y Machine Learning) [23].
- **Nivel 2**: evaluación de la situación: en este nivel, se lleva a cabo la interpretación de los resultados del nivel anterior. Los algoritmos más comunes se basan en la inteligencia artificial y en el razonamiento automático. De esta forma, se lleva a cabo una agregación de datos, interpretación contextual, y finalmente, una evaluación global multi-perspectiva. En el ámbito de la conducción autónoma, las tareas realizadas en este nivel son, por ejemplo, el razonamiento por parte del sistema de la situación de conducción en la que se encuentra el vehículo (adelantamiento, atasco, aparcamiento, etc.) y su entorno (condiciones climatológicas, obstáculos, etc.) [23].
- **Nivel 3**: evaluación del futuro: en este nivel, se realizan proyecciones de futuro del objeto analizado, a partir de la situación actual. Para ello, suelen usarse modelos predictivos, inteligencia artificial y estimación estadística [23].
- **Nivel 4**: proceso de refinamiento: en este nivel, se monitoriza todo el proceso de fusión de datos para mejorar el rendimiento de este a tiempo real, mediante la optimización, la modelización sensorial y la inteligencia artificial. Es una etapa necesaria en el proceso de fusión, que utiliza algoritmos basados en los sistemas de control avanzados [23].
- **Nivel 5**: refinamiento cognitivo: en este nivel se lleva a cabo la mejora del sistema de fusión mediante la proyección de modelos en los que la representación de los resultados permite al operador identificar errores [23].

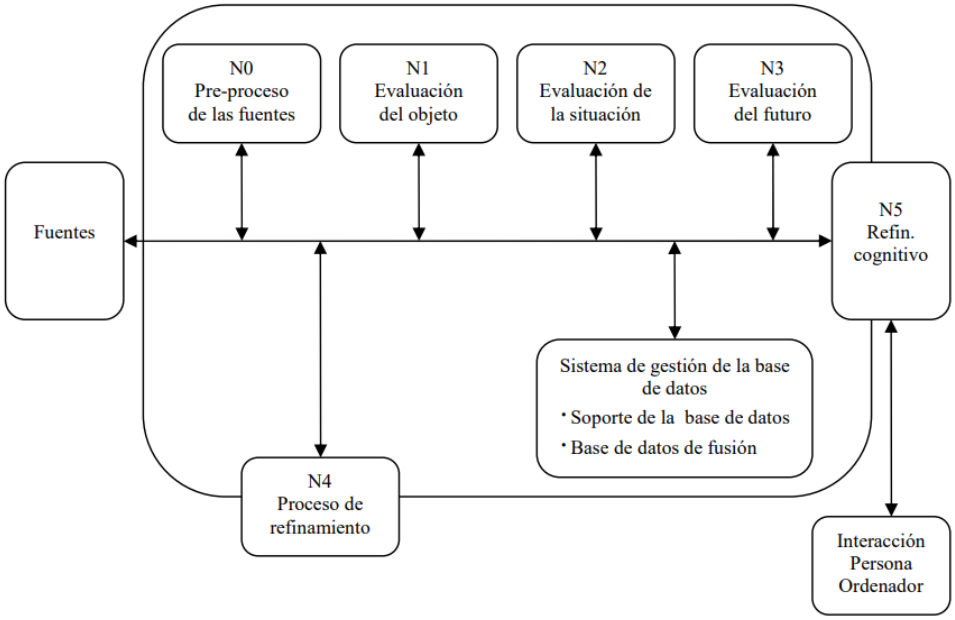


Figura 3.15. *Proceso de fusión de datos según la JDL.*

3.2.4. Introducción a las técnicas de fusión de datos

En este apartado, se clasificarán de forma general las diferentes técnicas de fusión de la actualidad, para posteriormente describir con mayor precisión las más utilizadas, tal y como plantea D. Abeijón en su tesis sobre fusión de datos publicada por la Universidad Politécnica de Cataluña [25] Para esto, deben definirse en primer lugar los siguientes conceptos:

- x_i representa el grado de credibilidad/probabilidad de la información generada por la fuente o sensor i.
- I representa el intervalo de los valores de x_i.
- $F(x_i, \dots, x_n)$ representa el operador de combinación de la información.

Existen dos clasificaciones principales de los algoritmos de fusión de datos, que se exponen a continuación:

Clasificación según el comportamiento del algoritmo

En primer lugar, se supondrá la fusión de dos tipos de información. De esta forma, $x_i = (x, y)$, y, de forma general, $I = [0,1]$. La aplicación de estos métodos para la fusión de más de dos tipos de información se realiza de manera análoga.

Según lo planteado por I. Bloch y H. Maitre en su informe *"Data Fusion in 2D and 3D Image Processing: An Overview*" (1996) [26], se realizará una primera clasificación según el comportamiento de los operadores que son aplicados en los algoritmos de fusión de datos:

- Un operador de fusión de datos se considera **severo**, cuando se comporta de forma conjuntiva, es decir, reduciendo la información y dando mayor fiabilidad a los sensores más restrictivos.

$$F(x, y) \leq \min(x, y)$$

- Un operador de fusión de datos se considera **prudente** si se comporta de forma comprometida, es decir, proporcionan una información intermedia entre los diferentes valores obtenidos por los sensores.

$$\min(x, y) \leq F(x, y) \leq \max(x, y)$$

- Un operador se considera **indulgente**, si el operador se comporta de forma disjunta, es decir, dando mayor credibilidad a la información menos incierta.

$$F(x, y) \geq \max(x, y)$$

Realizada esta clasificación, podemos distinguir tres tipos de operadores, incluidos en los algoritmos de fusión de datos de la actualidad:

- **Operador Independiente del Contexto con Comportamiento Constante (ICCC):**

Esta primera clase está compuesta por aquellos algoritmos que cumplen solo una de las tres condiciones anteriores, sin tener en cuenta información adicional procedente del exterior o de la situación actual del vehículo.

$$\forall (x, y) \in I^2;\ F(x, y) \leq \min(x, y)$$

$$\forall (x, y) \in I^2;\ \min(x, y) \leq F(x, y) \leq \max(x, y)$$

$$\forall (x, y) \in I^2;\ F(x, y) \geq \max(x, y)$$

Las técnicas más utilizadas con un comportamiento independiente del contexto son el teorema de Bayes, el método de Dempster-Shafer y algunos algoritmos de lógica difusa.

- **Operador Independiente del Contexto con Comportamiento Variable (ICCV):**

Esta segunda clase, incluye aquellos algoritmos que son independientes del contexto, pero poseen un comportamiento variable, en función de los valores de información recibidos (x, y). Se pueden comportar de manera severa, prudente o indulgente, dependiendo del tipo de información generada por los sensores.

Algunas de las técnicas más utilizadas con comportamiento variable son algunos algoritmos de lógica difusa y de inteligencia artificial.

- **Operador Dependiente del Contexto (DC):**

Esta tercera clase, incluye aquellos operadores que no solo dependen de los valores de información recibida, sino también del comportamiento global del sistema, de información adicional procedente del exterior o de las características de las fuentes de

datos. Poseen mayor complejidad en su aplicación, y requieren, en gran medida, del uso de inteligencia artificial.

Clasificación según la lógica matemática

Según D. Hall y S. McMullen (2005) [23], las técnicas de fusión de datos se pueden clasificar según la lógica matemática que utilizan. Es aquí donde se encuentran las principales diferencias entre los diferentes algoritmos utilizados en materia de conducción autónoma. Esta lógica matemática representa la forma de atribuir el nivel de incertidumbre a los resultados.

Por lo tanto, podemos encontrar:

- **Lógica probabilística**:

Estos modelos matemáticos son los más difundidos. Cuentan con una gran base teórica, basada en la teoría de la probabilidad. Requieren poca capacidad computacional. Dentro de estos métodos se encuentran los algoritmos basados en el teorema de Bayes y las leyes clásicas de la probabilidad.

- **Lógica epistémica**:

Los modelos basados en evidencias se basan en el uso de niveles de creencia o confianza sobre las funciones de probabilidad, que representan el conocimiento que se tiene del entorno del sistema. Estos métodos pueden dar lugar a respuestas que contradicen a los modelos probabilísticos. Entre estos modelos se encuentra el de Dempster-Shafer.

- **Lógica difusa**:

Estos modelos matemáticos comparan dos valores aleatorios, que se encuentran relacionados entre sí y contextualizados. Este tipo de lógica fue formulada por primera vez por el matemático e ingeniero Lotfi A. Zadeh [27], y se adaptan de una forma más fiel al mundo real mediante la utilización de cuantificadores, además de los operadores lógicos probabilísticos clásicos. Los algoritmos de inteligencia artificial y conjuntos difusos utilizan operadores con este tipo de lógica.

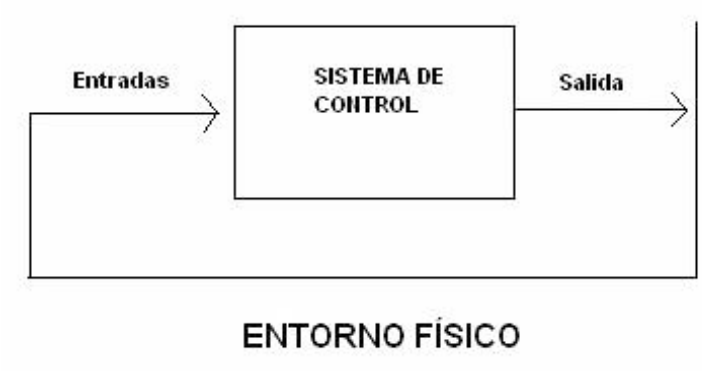


Figura 3.16. *Esquema de sistema de control difuso (recursivo).*

Otros tipos de clasificaciones

Existen otros tipos de clasificaciones de las técnicas de fusión de datos que no se explicarán en profundidad:

- Según la relación de las fuentes de los datos de entrada, los métodos de fusión de datos se clasifican en complementarios, redundantes o cooperativos. Esta clasificación fue propuesta por el ingeniero y académico británico H.F. Durrant-Whyte [28].
- Según la naturaleza de los tipos de datos de entrada y salida, propuesta por Dasarathy [29].
- Según la abstracción de los tipos de datos: nivel de señales, nivel de pixel, características y símbolos.
- Según el tipo de arquitectura del sistema: centralizada, descentralizada, distribuida o jerárquica.

3.2.5. Algoritmos más utilizados

En este apartado, se describirán las técnicas de fusión de datos más utilizadas. Se realizará únicamente una introducción a los conjuntos difusos y la inteligencia artificial, siendo estos temas de mayor complejidad que escapan de la temática de este trabajo.

Métodos bayesianos

Los métodos bayesianos de fusión de datos se basan en la lógica probabilística y utilizan operadores ICCC, para realizar un razonamiento basado en el conocimiento a priori (causa-efecto). Esta teoría fue postulada en 1763 tras la muerte del matemático inglés Thomas Bayes. La inferencia bayesiana permite, además, calcular la veracidad de una hipótesis a posteriori (efecto-causa) [25].

Si definimos E como el evento a evaluar, y x_1 y x_2 como los elementos de información obtenidos por los sensores, la expresión general probabilística del método de Bayes es la siguiente:

$$p(E|x_1,\dots,x_n) = p^2(E)\frac{\prod_{i=1}^{n} p(x_i|E)}{\prod_{i=1}^{n} p(x_i)} \qquad (3.1)$$

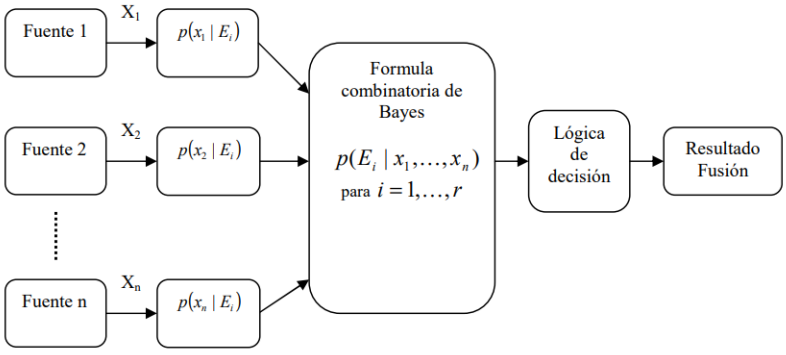


Figura 3.17. *Esquema de la fusión de datos mediante métodos bayesianos.*

Algoritmo de Dempster-Shafer

La inferencia de Dempster-Shafer es un método de fusión de datos basado en la lógica epistémica o de la evidencia. Es utilizado, principalmente, cuando existe cierta incertidumbre en la suma de información obtenida mediante métodos probabilísticos. De esta forma, el algoritmo de Dempster encuentra la intersección de los eventos identificados por los sensores o fuentes de información [30].

El algoritmo general de este método puede expresarse de la siguiente forma:

$$k = \sum m_1(B_1)m_2(B_2) \dots m_n(B_n) \qquad (3.2)$$

Siendo m_i las funciones de "masa" o credibilidad para cada sensor, y B_i las medidas de conflicto entre cada una de las fuentes. Altos valores de k se traducen en un grado de conflicto alto entre la información obtenida por los sensores, siendo un factor de la calidad de la combinación o fusión de datos. Además, en la práctica se definen factores de credibilidad, plausibilidad y comunalidad, que modifican el resultado final de la suma de información [31].

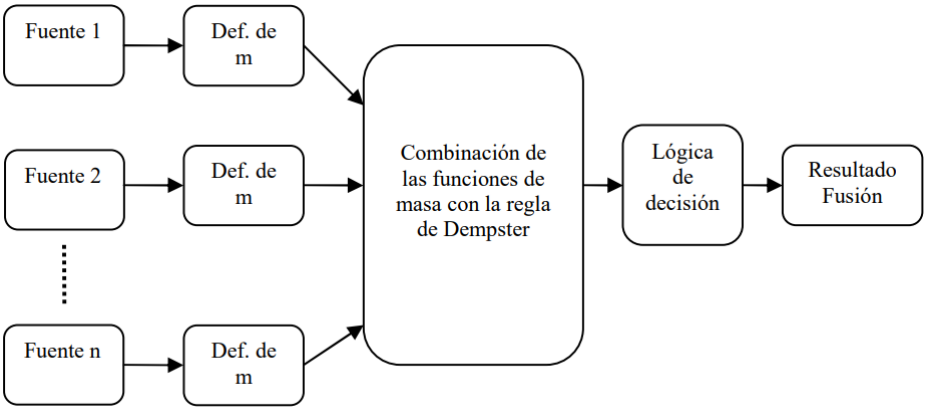


Figura 3.18. *Esquema de la fusión de datos mediante el método de Dempster-Shafer.*

Filtros de Kalman

El filtro de Kalman, y el filtro de Kalman extendido son técnicas de estimación creadas por el ingeniero y matemático Rudolf E. Kalman. Son usadas principalmente en la fusión de datos de bajo nivel, pero poseen numerosas aplicaciones, desde la guía y el control de vehículos hasta el procesamiento de señales con ruido blanco.

Estas técnicas permiten la estimación de estados de un proceso, a partir de la ecuación siguiente, siendo z una función del momento k, de un estado del sistema x.

$$x(k+1) = \Phi(k)x(k) + G(k)u(k) + \omega(k) \qquad (3.3)$$

$$z(k) = H(k)x(k) + v(k) \qquad (3.4)$$

Se define $\Phi(k)$ como la matriz de transición de estados, $G(k)$ como la matriz de transición de entrada, $u(k)$ como el vector de datos de entrada, $H(k)$ como la matriz de

muestreo y los vectores $\omega(k)$ y $v(k)$ como las variables Gaussianas de ruido blanco con valores promedio nulos [32].

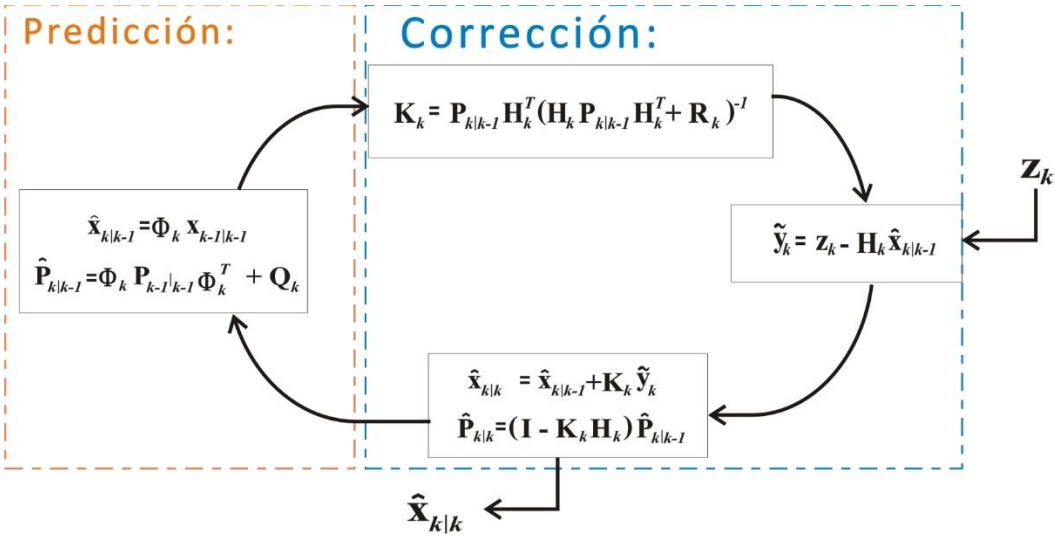


Figura 3.19. *Algoritmo recursivo del filtro de Kalman.*

Conjuntos difusos

Este tipo de técnicas, basadas en la lógica difusa (*Fuzzy Logic*), son efectivas cuando los procesos físicos sobre los cuales queremos obtener los datos a fusionar son difíciles de modelar mediante ecuaciones y paradigmas matemáticos, permitiendo incorporar un nivel de incertidumbre o "vaguedad" en el proceso de decisión final. Estableciendo determinadas normas, los conjuntos difusos permiten construir un sistema eficaz de fusión de datos incluso cuando no se ha entendido el funcionamiento matemático del proceso.

Estas técnicas resultan muy útiles bajo las siguientes condiciones:

- Si existen problemas de definición en el modelo matemático que impiden el análisis correcto de todos los datos a fusionar.
- Si el sistema está compuesto de sensores con gran nivel de incertidumbre, de poca calidad o que transmiten información poco representativa.

Inteligencia artificial

La inteligencia artificial, conocida también por sus siglas IA, es un conjunto de tecnologías y técnicas que permiten el desarrollo o simulación de técnicas de razonamiento, comprensión y aprendizaje por parte de las máquinas. De manera coloquial, se puede decir que los sistemas de inteligencia artificial no solo intentan imitar las capacidades humanas, sino utilizarlas para llevar a cabo tareas de computación avanzadas.

El desarrollo de la IA presenta grandes ventajas en materia de fusión de datos y conducción autónoma, no solo en capacidad y velocidad de computación, sino también en la resolución de problemas mediante la inferencia estadística. Las técnicas de IA utilizadas para la fusión de datos son:

- **Sistemas expertos**: simulan las capacidades de un operario experto en la materia, resolviendo problemas de ejecución de los algoritmos y supervisando la información obtenida por parte de los sensores.
- **Redes bayesianas**: se basan en la inferencia estadística para llevar a cabo la fusión de datos de manera más rápida y eficaz.
- **Redes neuronales**: presentan un nuevo paradigma de aprendizaje y procesamiento automático, inspirado en el sistema nervioso biológico. Las redes neuronales son capaces de aprender de experiencias previas para resolver problemas.

Aunque actualmente los sistemas de IA mejoran con creces la velocidad y capacidad de procesamiento, merece la pena destacar que la inteligencia artificial es una técnica moderna, que está lejos de estar completamente desarrollada [33].

3.3. Sistemas físicos (III): modelado dinámico del entorno

Una vez procesada la información procedente de las técnicas de fusión de datos, se realiza un modelado dinámico del entorno del vehículo autónomo, que permite definirlo de forma matemática.

Actualmente, las técnicas de modelado dependen casi exclusivamente de los sensores LiDAR. El uso de cámaras para el desarrollo de esta tarea, aun siendo posible, añade complejidad debido a que su correcto funcionamiento está determinado por las condiciones lumínicas. Existen multitud de sistemas de modelado del entorno que utilizan sensores LiDAR como herramienta principal. Sin embargo, nos centraremos únicamente en los sistemas de modelado terrestres de tipo móvil, usualmente montados en vehículos autónomos y otros medios de transporte, como trenes de alta velocidad o grandes embarcaciones.

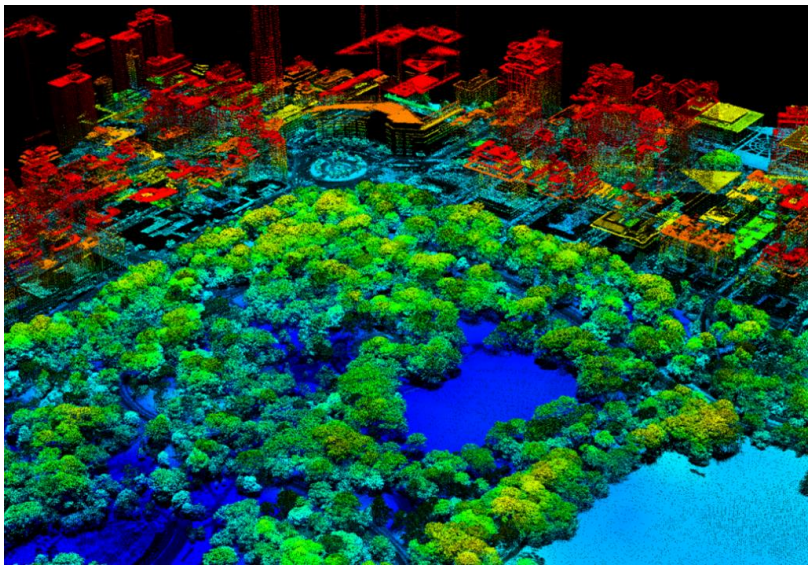


Figura 3.20. *Nube de puntos generada a partir de un LiDAR 3D.*

Como se ha mencionado anteriormente, los sistemas LiDAR crean una nube de puntos que permite posicionar cada objeto del entorno en un entorno virtual de tres dimensiones [34].

Estos puntos proporcionan más información aparte de la posición de una entidad en el espacio:

- Cada punto tiene asociada una intensidad de pulso recibido. De esta forma se puede conocer la capacidad de reflexión y absorción del objeto en cuestión, y generar una imagen en escala de grises del entorno.
- Según el número de recepciones del punto en cuestión y su ángulo respecto al emisor, se puede determinar en qué posición se encuentra el objeto sobre el que rebota el haz de pulsos.
- Los puntos pueden ser clasificados para determinar qué tipo de objeto se está detectando: suelo, obstáculos, edificios, etc.
- Cada punto posee información sobre la fecha, la hora y la posición del sistema de posicionamiento global al realizar la medición.

Figura 3.21. *Imágenes generadas a partir de la intensidad de un pulso láser.*

Gracias a estas características, los sistemas autónomos son capaces de obtener información del entorno y crear un modelo tridimensional del mismo que permita fijar puntos de referencia útiles para su navegación [34].

Otros dispositivos utilizados para el modelado dinámico del entorno son las cámaras, tal y como se ha presentado anteriormente. Sin embargo, precisan de mayor capacidad de computación para llevar a cabo esta tarea, a pesar de ser sensores relativamente más baratos que los LiDAR. Usualmente se requiere de cámaras estereoscópicas 3D a color

capaces de capturar diversas imágenes simultáneamente desde distintos ángulos, y así obtener una imagen tridimensional fiable del entorno. Estas cámaras son normalmente montadas en el frontal y lateral del vehículo. En caso de no disponer de un sensor LiDAR, se requiere además de cámaras traseras para obtener una visión global de la situación.

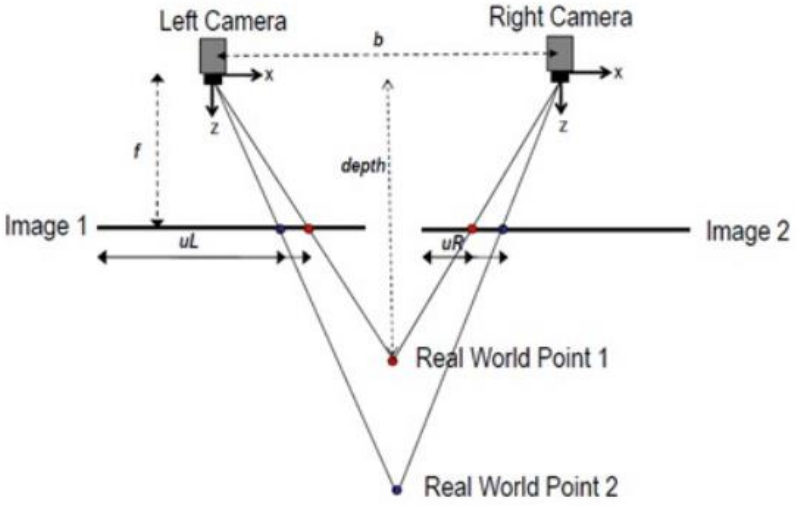


Figura 3.22. *Esquema de un sistema de cámara 3D para el modelado del entorno de un vehículo autónomo.*

No todos los fabricantes de vehículos autónomos prefieren utilizar sensores LiDAR para la tarea de modelado. Empresas como Mercedes-Benz y Volvo, propias de la industria de la automoción, defienden el uso de cámaras frente a este tipo de sensores [35].

3.4. Sistemas lógicos: localización, construcción de mapas y planificación de trayectorias

En este apartado, se definirán aquellas tareas realizadas por el vehículo autónomo que no requieran necesariamente de un sistema físico de percepción para llevarse a cabo. Estas tareas, como la construcción de mapas de circulación, la planificación de las trayectorias, o la toma de decisiones durante situaciones de conducción, representan la parte lógica del vehículo. Mientras que la percepción y el modelado se realizan mediante los "órganos de los sentidos" del vehículo, estas tareas tienen lugar en el "cerebro" del mismo.

3.4.1. Localización y construcción de mapas

Durante la conducción autónoma, es preciso que el vehículo conozca en todo momento su posición global, determinada mediante los sistemas de posicionamiento expuestos anteriormente; y su posición local, es decir, su situación respecto a la carretera o vía por la que deben circular, y respecto a la de su objetivo final.

Actualmente, el GPS es el sistema más extendido para llevar a cabo la tarea de posicionamiento global, debido a su facilidad de uso y la utilización abierta de sus

satélites. Sin embargo, no es el más preciso, y conlleva un error que en algunos casos puede aumentar hasta los 100 m [36].

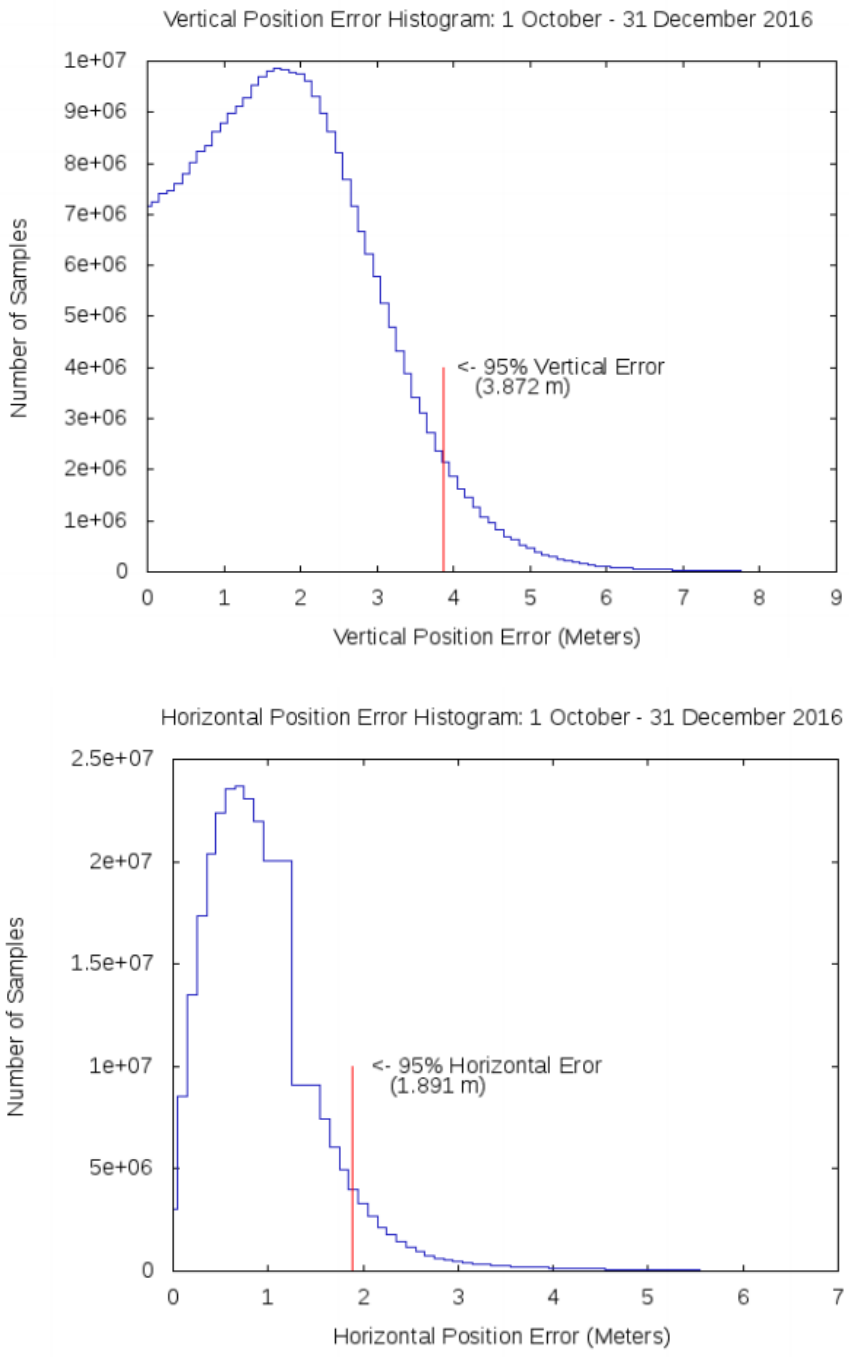


Figura 3.23. *Histogramas de frecuencia de error vertical y horizontal en la utilización de GPS.*

En base a los datos obtenidos de los sistemas de posicionamiento global, es necesario que los vehículos autónomos lleven a cabo la tarea de posicionamiento local, es decir, la construcción en tiempo real de mapas que permitan fijar trayectorias de recorrido, teniendo en cuenta el entorno modelado mediante los sistemas de percepción.

Este proceso no es sencillo, y requiere usualmente estrategias avanzadas de localización y mapeado simultáneos (SLAM, *Simultaneous Location and Mapping*), basadas en la inferencia estadística y en el teorema de Bayes [37].

Filtro de Kalman Extendido

El algoritmo más utilizado en los procesos de SLAM es el filtro de Kalman extendido. Es una generalización del filtro de Kalman presentado en el *Apartado 3.2.5*, y fue introducido por los investigadores matemáticos Randall Smith, Matthew Self y Peter Cheeseman en la década de los 80.

El uso de este algoritmo, conocido usualmente como SLAM-EKF, requiere un mapa en el que las entidades que lo compongan puedan ser definidas mediante los parámetros del sistema. Al ser uno de los algoritmos más extendidos, ha sido mejorado durante años en sistemas de conducción autónoma, por lo que actualmente existen multitud de soluciones para paliar sus puntos débiles [38].

Algoritmos de mapeado mediante cuadrículas

La familia de algoritmos basados en OGM (*Occupancy Grid Mapping*), fueron introducidos por Hans Moravec y Albert Elfes en la década de los 80 [39]. Se basan en la discretización del espacio conocido mediante unidades básicas denominadas “celdas”. Una vez dividido el entorno visible, se clasifican estas celdas y se les asigna un valor binario (vacías u ocupadas), a través de métodos probabilísticos.

Este tipo de técnicas parten de la suposición de conocimiento de la posición del vehículo, por lo que es necesaria la aplicación de métodos de posicionamiento con antelación. Además, su eficacia depende casi exclusivamente del tamaño de definición de las celdas (cuanto más pequeñas se definan, mayor precisión se conseguirá en la construcción del mapa virtual, pero el coste computacional aumentará notablemente).

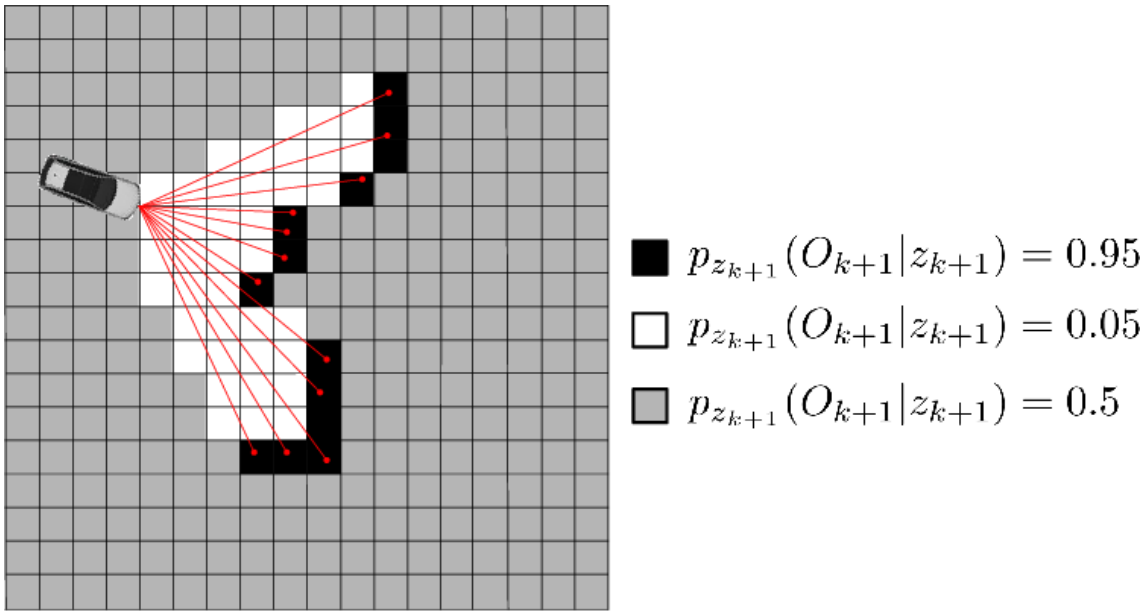


Figura 3.24. *Generación de mapas topológicos mediante el método de OGM.*

Otros algoritmos de construcción de mapas

Existen otros algoritmos de construcción de mapas de los que no se tratará en este trabajo, debido tanto a su complejidad como a su menor relevancia respecto a los definidos anteriormente. Entre estos se encuentran:

- ***Coverage Maps***: son métodos basados en celdas de ocupación. Su fundamento es el mismo que el de los métodos de OGM, sin embargo, al definir probabilidades de ocupación de una celda, los mapas topológicos construidos se ajustan más a la realidad, debido a que resuelven el problema que presentaban celdas parcialmente ocupadas.
- **DP-SLAM**: este tipo de métodos utilizan filtros de partículas para obtener una estimación más realista de los mapas construidos.
- ***Visual* SLAM**: es un método de construcción de mapas relativamente reciente (apareció durante el año 2005). Se centra en el uso de sensores visuales para llevar a cabo un mapeado más preciso.
- ***Active* SLAM**: utiliza la exploración activa, es decir, la construcción de mapas del entorno al mismo tiempo que se ejecutan algoritmos de planificación de trayectorias, con el objetivo de minimizar el cómputo creando un escenario de la forma más eficiente posible.

3.4.2. Planificación de trayectorias

El tercer y último paso, una vez obtenidos tanto el modelo matemático como el mapa del entorno, es proveer al vehículo autónomo de una planificación óptima de la trayectoria de conducción, además de la capacidad de actuar tal y como lo haría un humano ante los problemas que puedan surgir durante la misma, siendo esta última tarea la más complicada.

No solo se requiere un cálculo de la trayectoria que el vehículo debe tomar, sino un estudio cinemático y dinámico preciso del conjunto vehículo-entorno, para asegurar que dicha trayectoria pueda ser físicamente aplicable (aunque obtengamos una trayectoria teóricamente posible en una curva cerrada, esta no podrá ser realizada si nuestro vehículo vuelca debido a las condiciones de velocidad y adherencia con la carretera).

Un vehículo autónomo puede ser descrito, de forma simplificada, como un sólido rígido que se mueve en un plano. Las variables que lo definen son:

- Separación entre los ejes de dirección o longitud efectiva, L.
- Velocidad lineal, v.
- Ángulo de giro, ϕ.

Tal y como se define en la *Figura 3.25*.

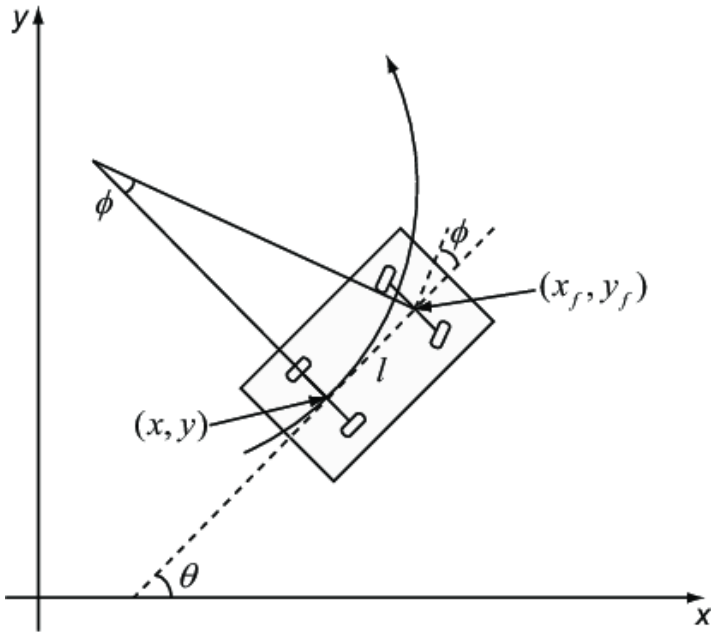


Figura 3.25. *Modelo cinemático simplificado de un vehículo.*

De esta manera, se pueden definir las siguientes ecuaciones de la variación de la posición del centro de masas del vehículo:

$$\dot{x} = f(x, y, \theta, L, v, \phi) \qquad (3.5)$$

$$\dot{y} = f(x, y, \theta, L, v, \phi) \qquad (3.6)$$

$$\dot{\theta} = f(x, y, \theta, L, v, \phi) \qquad (3.7)$$

Una vez definido el modelo cinemático del vehículo en cuestión, se utilizan algoritmos para realizar el recorrido por el espacio 2D o 3D de forma segura y óptima. Para esto, se divide el problema en dos partes fundamentales: la planificación de trayectorias global y la planificación de trayectorias local, estando la primera centrada en estimar la mejor ruta posible para que el vehículo llegue a su destino, y la segunda, en modificar dicha ruta para evitar posibles obstáculos imprevistos [40].

Existe un gran número de algoritmos usados para la planificación de trayectorias de conducción. Además, no existe ninguno que predomine sobre otro, debido a que cada tipo de vehículo autónomo requiere la resolución de multitud de problemas concretos de naturaleza variable. Usualmente, este tipo de algoritmos comienza con la simplificación del modelo del vehículo hasta una configuración básica en el espacio euclídeo.

Debido a la naturaleza básica de este documento, no se definirán de forma exhaustiva cada uno de los algoritmos existentes, sino que se hará una breve introducción a sus características principales. Según J.C. Latombe [41] los algoritmos de planificación de trayectorias se pueden clasificar en:

- ***Roadmaps***: este tipo de algoritmos de planificación de trayectorias describen el espacio de configuraciones del vehículo a partir de una red de curvas unidimensionales, que componen el espacio de soluciones buscadas. Los

principales exponentes de este conjunto de algoritmos son los grafos de visibilidad y los diagramas de Voronoi.

- **Descomposición en celdas**: este tipo de métodos dividen el espacio de configuraciones del vehículo en celdas, similares a las usadas en los métodos de construcción de mapas. Cuando dos celdas comparten una arista o límite común, se crea un segmento de la trayectoria final del vehículo, que queda definida como un arco. Algunos algoritmos de este tipo son la descomposición vertical y las técnicas quadtree (o de árbol de nodos).
- **Campos de potencial**: estos métodos se basan en la creación de funciones escalares sobre el espacio de configuraciones del vehículo, utilizando el gradiente del campo para crear una trayectoria admisible.
- **Probabilísticos**: los métodos de este tipo utilizan configuraciones aleatorias para identificar el entorno, y posteriormente recurrir a la inferencia estadística sobre las pruebas realizadas para generar trayectorias óptimas. Algunos de estos métodos son los árboles de exploración aleatorios (*Rapidly Exploring Random Trees*, RRT) y los mapas de trayectorias probabilísticos (*Probabilistic Roadmaps*, PRM).
- **Mediante el uso de curvas *Splines***: estos algoritmos son capaces de generar multitud de trayectorias libres de obstáculos a través de curvas diferenciables, definidas en porciones mediante polinomios (*Splines*). Después, se lleva a cabo un proceso de selección para definir cuál de estas posibles trayectorias es óptima, tomando en cuenta la seguridad del vehículo, la rapidez de llegada a su punto final y la suavidad de la curva estudiada. Estos métodos se basan en la teoría fundacional de las curvas *Spline*, postulada por el matemático I. J. Schoenberg en la década de los 40 [40].

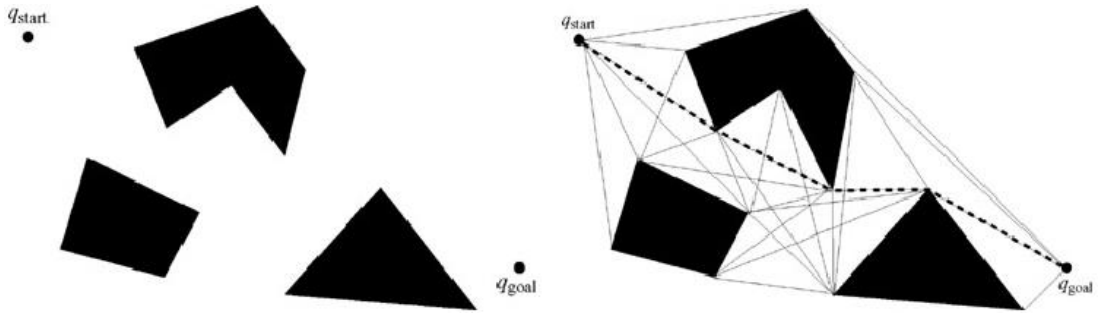


Figura 3.26. *Generación de trayectorias mediante grafo de visibilidad.*

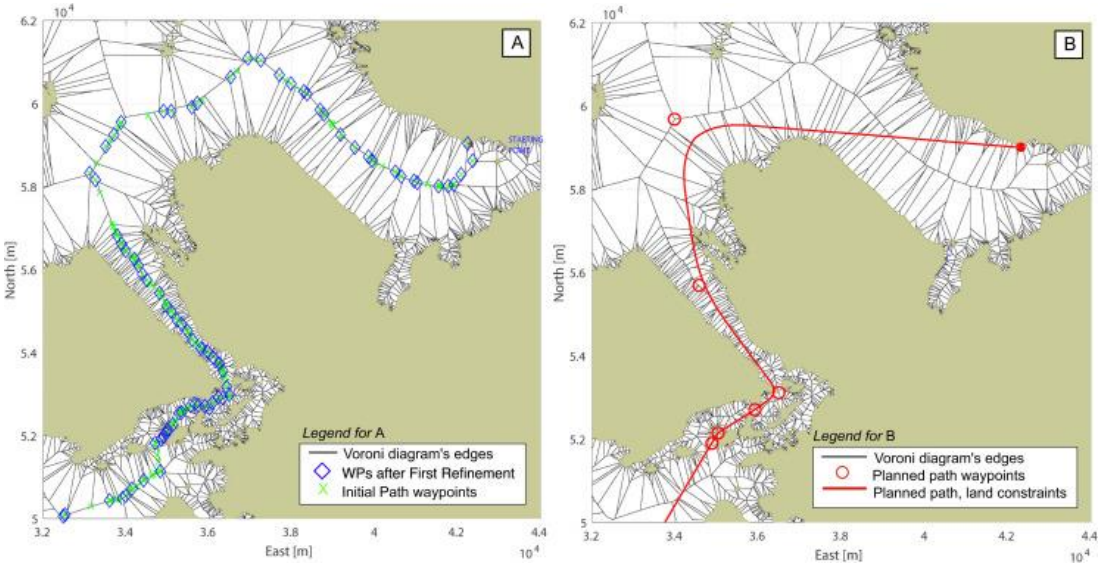


Figura 3.27. *Generación de trayectorias mediante diagramas de Voronoi.*

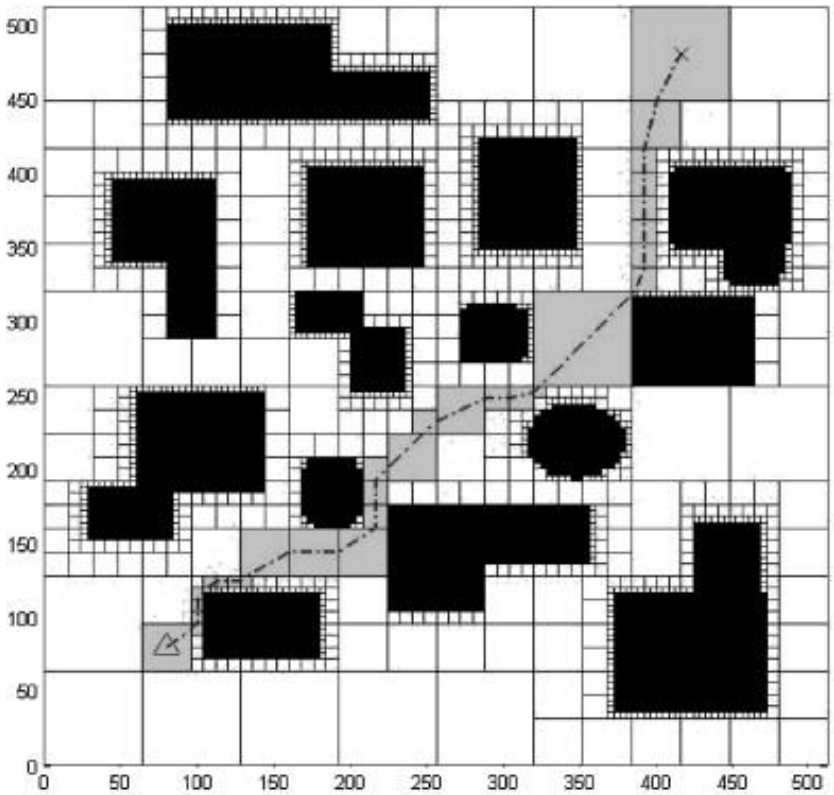


Figura 3.28. ***Generación de trayectorias mediante diagramas quadtree.***

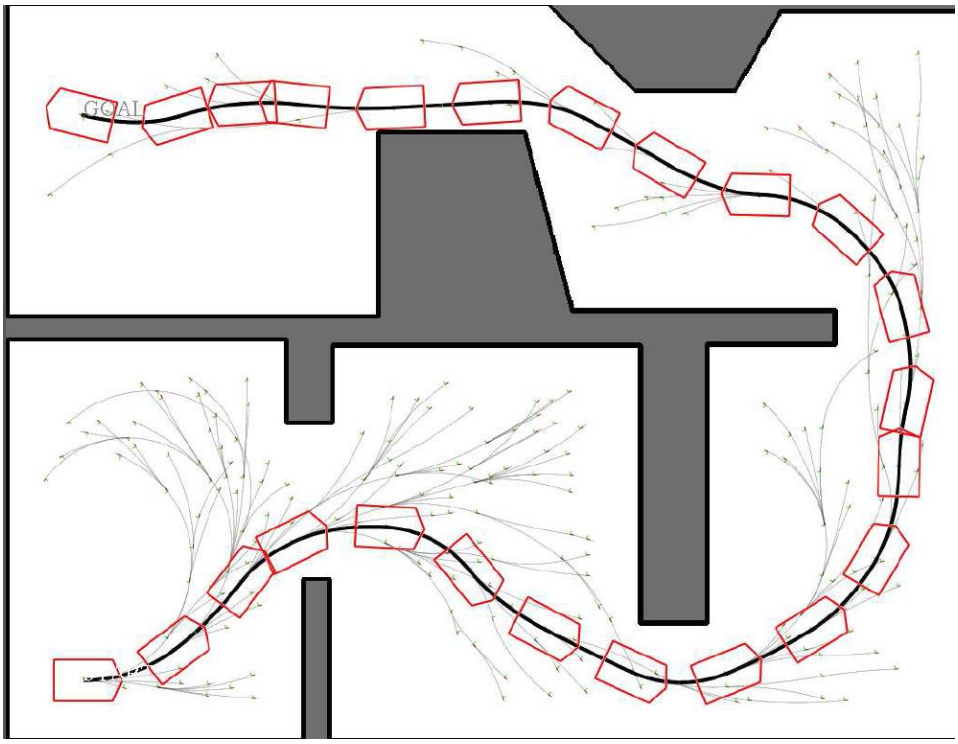


Figura 3.29. *Generación de trayectorias mediante el método probabilístico RRT.*

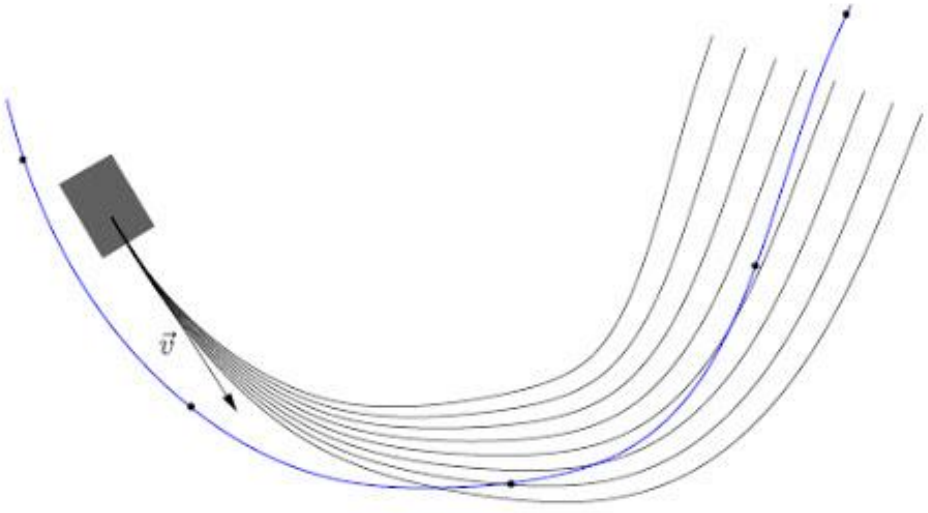


Figura 3.30. *Generación de trayectorias globales mediante Splines.*

Capítulo 4.

Automated Driving Toolbox™ como herramienta de simulación de escenarios de conducción autónoma

En este capítulo, se presentarán de forma básica los principales usos de la herramienta de simulación de vehículos autónomos Automated Driving Toolbox™, incluida en el entorno de programación de MATLAB®. Se enfocará el capítulo desde el punto de vista de la percepción y la simulación de sensores, tanto LiDAR como cámaras. Además, se realizará una introducción a determinados conceptos y aplicaciones que serán tratados en los capítulos posteriores.

4.1. Simulación de sistemas de conducción autónoma

El modelado y la simulación son herramientas consolidadas en el desarrollo de vehículos autónomos, tanto en sus etapas iniciales de diseño como en su etapa de implementación en la industria.

El método en V (definido como un proceso uniforme para llevar a cabo el desarrollo de productos en el ámbito de Tecnologías de la Información) y sus variantes, son los estándares dominantes en materia de conducción en la Unión Europea, y se componen de procesos de validación y pruebas en múltiples etapas [42], entre los que se incluyen las pruebas de simulación. De hecho, la norma ISO 26262 define una guía sobre el uso de la simulación, que incluye pruebas del tipo MIL (*Model-in-the-loop*), SIL (*Software-in-the-loop*) y HIL (*Hardware-in-the-loop*) [43].

Aunque el número de herramientas de simulación para el desarrollo de vehículos autónomos ha aumentado en los últimos años, es una tarea complicada seleccionar la adecuada para una tarea específica. Algunas son de código abierto, otras, multiplataforma, y solo unas pocas ofrecen una gran personalización del problema a tratar o una cantidad considerable de documentación que sirva de apoyo al trabajo de los ingenieros.

La simplificación también es importante en estos entornos de simulación, debido a la naturaleza compleja de los problemas que surgen en el desarrollo de vehículos autónomos y a la posible limitada capacidad de computación de los sistemas utilizados para implementar los algoritmos de fusión sensorial, planificación de trayectorias, y toma de decisiones en tiempo real. Es por esto, que determinados entornos de simulación poseen herramientas de programación sencillas e interactivas.

Cabe destacar que existen herramientas de simulación compatibles con casi cualquier lenguaje de programación, como C, C++, Python, Java, LabView o MATLAB®.

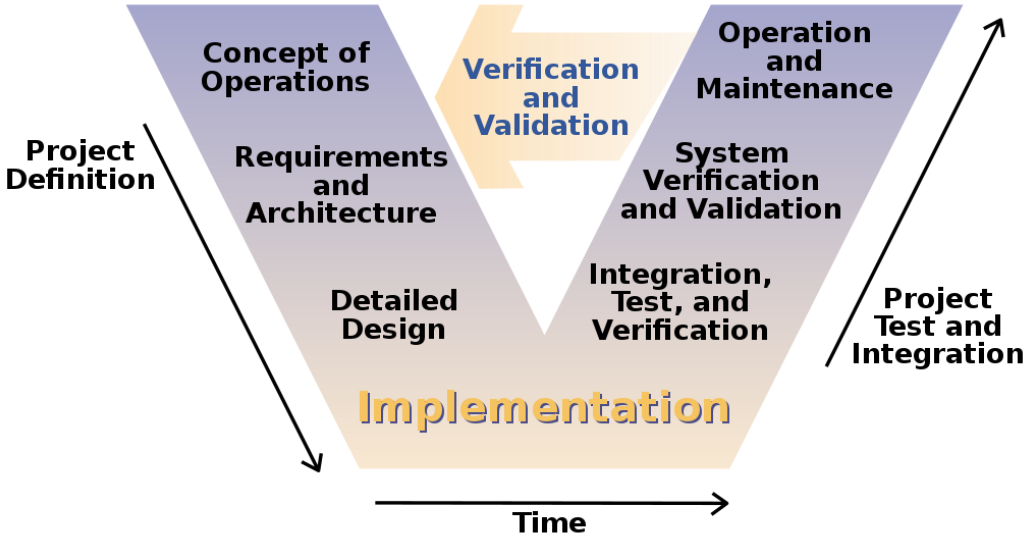


Figura 4.1. *Método en V de desarrollo de sistemas.*

De esta forma, se puede llegar a la conclusión de que para lograr una conducción autónoma eficaz, precisa y segura, se requiere probar todas las características del vehículo de forma virtual, ya que las pruebas en campo son usualmente costosas o imposibles de realizar.

En la siguiente tabla se incluyen diversas soluciones de simulación de entornos de conducción autónoma, que permiten llevar a cabo las pruebas virtuales necesarias para la puesta a punto de este tipo de vehículos [44].

Tabla 4.1. Principales herramientas de simulación de sistemas de CA.

Simulador	Licencia	Cumplimentación de la ISO 26262
PaTAVTT	GPL	-
Automated Driving Toolbox & Simulink	Comercial	Sí
DSpace GmbH	Comercial	Sí
LabView	Comercial	Sí
CarSim	Comercial	Sí
CAT Vehicle	GPL	-

Fuente: *Rosique, F; Navarro, Pedro J; Fernández, C; Padilla, A. (2016) "A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research."*

4.2. Introducción a Automated Driving Toolbox™

Automated Driving Toolbox™ es un paquete de herramientas perteneciente al entorno de programación de MATLAB®, que ofrece algoritmos para el desarrollo, simulación y puesta a punto de ADAS y sistemas de conducción autónoma.

Mediante este paquete de herramientas, es posible diseñar y probar sistemas de percepción para vehículos autónomos basados en la fusión de sensores de imagen, LiDAR y radar, así como implementar en las simulaciones de conducción algoritmos de planificación de rutas, controladores de vehículos, y mapeado del entorno.

Automated Driving Toolbox™ incluye herramientas de visualización en tiempo real, tanto de vista de pájaro como en tercera persona, que permiten un mejor entendimiento del alcance de los sensores y sus capacidades de percepción. Además, incluye opciones para importar y trabajar con datos *HERE HD Live Map* y redes de carreteras *OpenDrive*®.

El paquete de herramientas también incluye diferentes aplicaciones que mejoran la experiencia de usuario, como la aplicación de etiquetado virtual (*Ground Truth Labeler*) mediante la cual es posible automatizar el reconocimiento de elementos del entorno de conducción para entrenar y evaluar algoritmos de percepción, o el diseñador de escenarios (*Driving Scenario Designer*), con el que es posible crear simulaciones de conducción avanzadas con la ayuda de MATLAB® y Simulink® [45].

4.2.1. Nociones básicas

El principal objetivo de la creación de un escenario de simulación es la generación de pruebas de uso de sensores y algoritmos en vehículos autónomos. El sistema de coordenadas se moverá con el vehículo principal, debido a que los sensores usualmente se encuentran instalados sobre el mismo.

En primer lugar, debemos definir algunos conceptos que nos ayudarán a comprender cómo utilizar esta herramienta de un modo adecuado:

- La programación mediante Automated Driving Toolbox™ hace uso de un lenguaje de alto nivel del tipo “matriz/array”. Cada una de estas matrices organizan el código en unidades, propias del lenguaje orientado a objetos, que se relacionan entre sí para conseguir el objetivo de la aplicación [46].
- Los escenarios de simulación están constituidos de elementos denominados “actores”, de diferentes clases. Estos pueden ser utilizados para modelar carreteras, peatones, parquímetros, bocas de incendio y otros objetos, que interactuarán con los sensores de nuestro vehículo.
- Los actores son representados como cuboides con unas determinadas características: largo, ancho, alto y una sección transversal denominada RCS (*radar cross-section*).
- Estos actores se encuentran posicionados y orientados sobre su cara inferior.

- Un tipo especial de actor es el vehículo, el cual se mueve sobre ruedas, y sobre el cual pueden instalarse sensores.
- Todos los actores pueden colocarse en cualquier lugar dentro del escenario especificando sus respectivas propiedades.

4.3. Aplicaciones de Automated Driving Toolbox™

4.3.1. Driving Scenario Designer

La simulación de escenarios y la fusión de sensores es una parte crucial en la prueba de algoritmos de conducción autónoma. Automated Driving Toolbox™ utiliza dos entornos de simulación de escenarios distintos [47]:

- Driving Scenario Designer (DSD): el diseñador de escenarios implementado en MATLAB® permite la representación de vehículos autónomos, peatones, y elementos del entorno de forma sencilla, como polígonos en un entorno tridimensional.
- *Unreal Engine*®: mediante este motor gráfico, MATLAB® permite renderizar escenarios con gráficos más realistas, además de generar datos de alta calidad procedentes de los sensores LiDAR, radares y cámaras.

La interfaz del diseñador de escenarios de conducción, o DSD (Driving Scenario Designer) es sencilla, permitiendo la programación y definición de actores de cualquier índole, desde vehículos, hasta peatones, bicicletas, barreras y otros elementos. Además, permite la implementación y el testeo de sensores, mediante los gráficos asociados de vista de pájaro y vista en tercera persona.

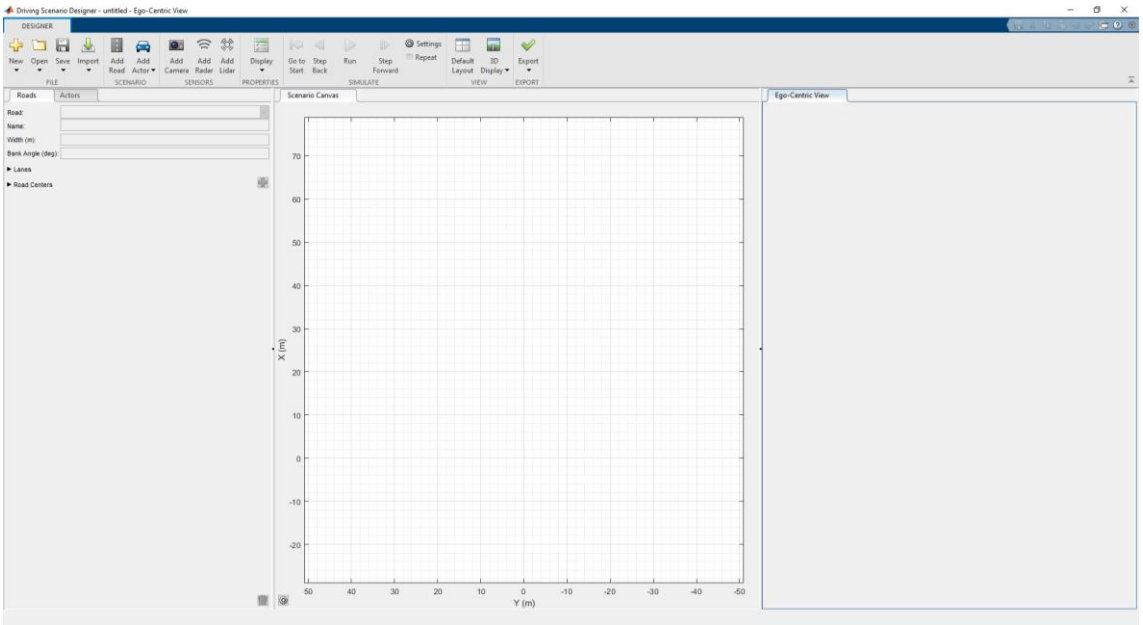


Figura 4.2. *Interfaz de la aplicación Driving Scenario Designer.*

En el menú "*File*", es posible cargar archivos de MATLAB® previamente diseñados, importar mapas provenientes de *OpenStreetMap*, y guardar escenarios creados a mano.

En el menú "*Scenario*", podemos añadir carreteras y actores a la simulación (tanto vehículos utilitarios como camiones, bicicletas, peatones, barreras, señales y definir clases de actores personalizadas). Desde el menú "*Sensors*", es posible abrir el gráfico de sensores del vehículo autónomo, y añadir cámaras, radares y sensores LiDAR, así como modificar sus características básicas. En el menú "*Simulate*" se encuentran los controles básicos de la simulación, como configurar el tiempo de muestreo, reproducir, pausar, volver al inicio y avanzar por pasos. Por último, en el "*Export*", es posible exportar el escenario al entorno de programación de MATLAB®, exportar los datos obtenidos mediante los sensores del vehículo autónomo, o construir un modelo en Simulink® del escenario creado.

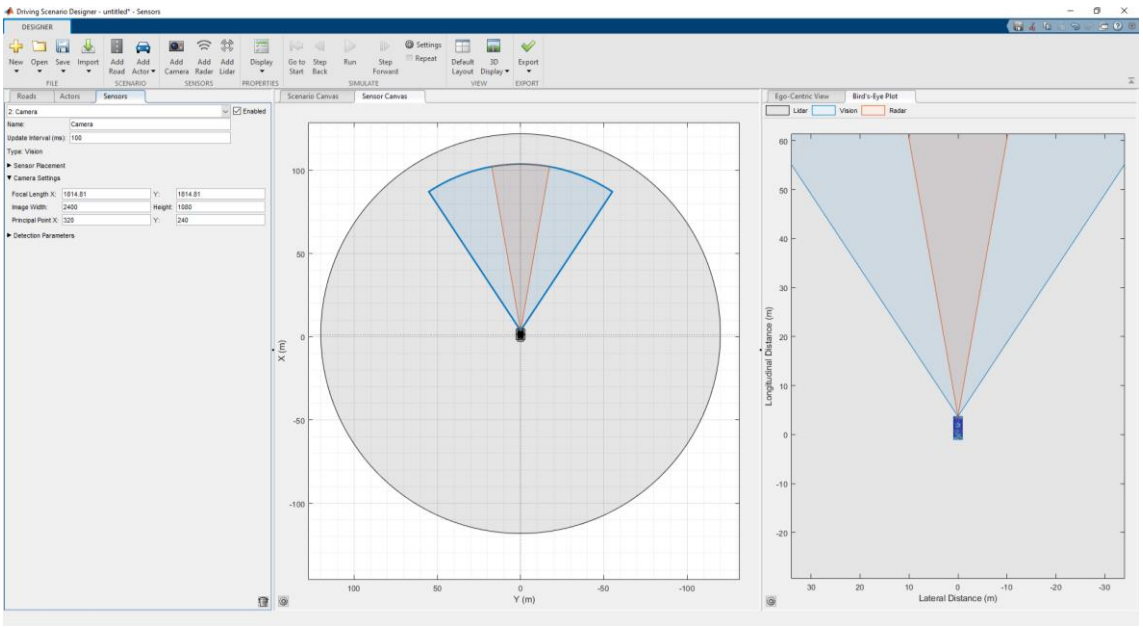


Figura 4.3. *Fusión de sensores en DSD.*

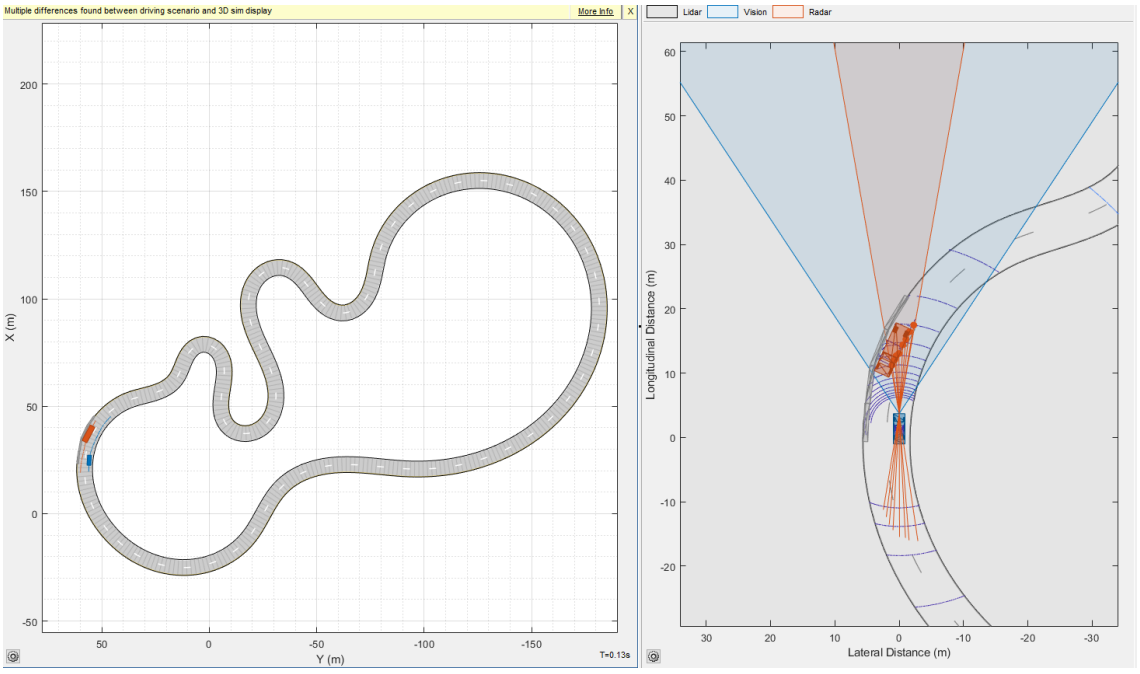


Figura 4.4. *Ejemplo de escenario de conducción.*

Figura 4.5. *Simulación mediante Unreal Engine®.*

4.3.2. Ground Truth Labeler

Mediante la aplicación de etiquetado de verdad (*Ground Truth Labeler*), es posible automatizar la clasificación de objetos y elementos de la conducción utilizando las señales de los sensores del vehículo o de escenas previamente cargadas en el programa (como secuencias de imágenes, video, nubes de puntos obtenidas mediante LiDAR, etc.).

Esta aplicación es realmente útil para obtener datos con los que llevar a cabo un entrenamiento profundo de inteligencia artificial, mediante técnicas de *Machine Learning*, que permitan al vehículo clasificar los elementos detectados mediante sus dispositivos de percepción [48].

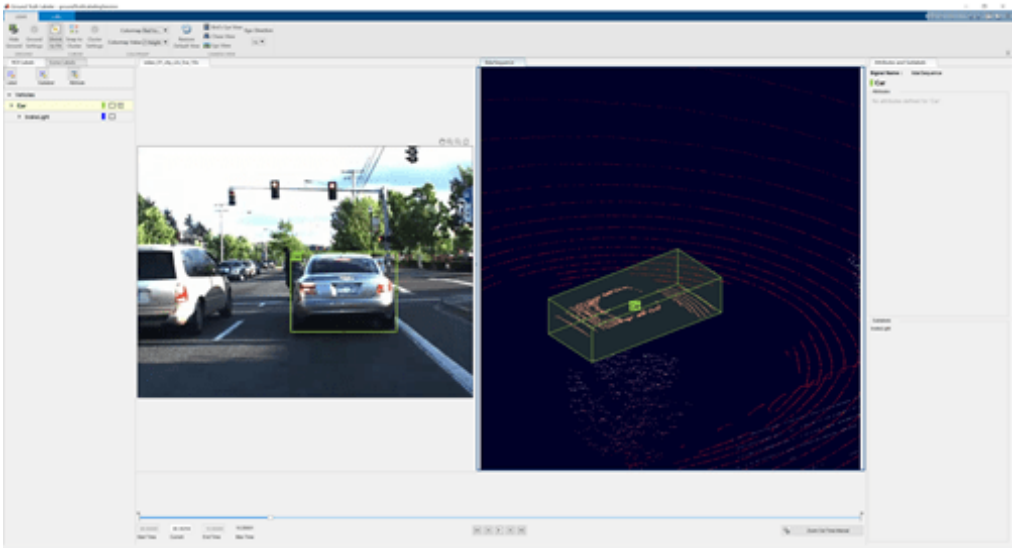


Figura 4.6. *Etiquetado Virtual mediante GTL.*

GTL permite automatizar el etiquetado de elementos mediante algoritmos prediseñados o propios, además de proporcionar APIs utilizadas para mostrar señales adicionales sincronizadas en el tiempo, y cargar fuentes de datos personalizadas.

4.3.3. Detección y seguimiento de objetivos

Los algoritmos de percepción de este paquete de herramientas utilizan imágenes de cámaras de visión artificial y datos obtenidos mediante sensores LiDAR para llevar a cabo la simulación de detección y rastreo de objetos de interés. Mediante estos algoritmos, es posible desarrollar ADAS y aplicaciones a la conducción autónoma, como sistemas de frenado automático o control de la dirección del vehículo [49].

Configuración de los sensores de imagen

Al desarrollar algoritmos de detección de objetivos, es importante una configuración de cámara precisa. Automated Driving Toolbox™ permite modificar los valores de configuración de la cámara del vehículo.

Utilizando el objeto `monoCamera`, es posible definir los parámetros intrínsecos y extrínsecos de la misma. Además, mediante la aplicación Camera Calibrator es posible transformar las coordenadas de la imagen en coordenadas del vehículo modelado, con el objetivo de medir distancias y crear gráficos de vista de pájaro del entorno [50].

Percepción visual

Otra característica dentro del ámbito de detección y seguimiento de objetivos mediante Automated Driving Toolbox™ es la capacidad de detectar objetos utilizando técnicas de *Machine Learning* y *Deep Learning*, aplicables desde el entorno de programación.

Mediante este tipo de técnicas, es posible segmentar, detectar y modelar límites de carril parabólicos mediante el algoritmo conocido como RANSAC (*Random Sample Consensus Algorithm*), y otros objetos del entorno del vehículo [51].

Procesamiento de datos LiDAR

Los sistemas avanzados de asistencia a la conducción (ADAS) utilizan habitualmente nubes de puntos en tres dimensiones obtenidas mediante sensores LiDAR para medir superficies. El procesamiento de estas nubes de puntos permite eliminar ruido, además de segmentar ciertos grupos de puntos haciendo posible su clasificación.

Mediante esta aplicación, es posible leer, escribir, almacenar, visualizar y comparar distintas nubes de puntos obtenidas mediante LiDAR, con aplicación directa sobre los sensores manufacturados por Velodyne, quien proporciona un paquete de herramientas adicional para esta tarea (PCAP) [52].

Seguimiento y fusión de sensores

Utilizando Automated Driving Toolbox™ es posible crear *tracking* de objetivos múltiples mediante la fusión de sensores de radar y cámaras de vídeo. Estos rastreadores (*trackers*), utilizan filtros de Kalman, tal y como se ha explicado en el capítulo anterior,

para estimar el estado del movimiento del objeto detectado. Pueden utilizarse las mediciones de los sensores para rastrear objetos en movimiento, mediante modelos de velocidad o aceleración constante [53].

4.3.4. Localización y mapeado del entorno

En todas las aplicaciones de conducción autónoma, es necesaria una buena localización del vehículo en todo momento. Los algoritmos de localización utilizan datos de mapas en la nube para estimar la posición y orientación del vehículo, en función de las lecturas obtenidas por los sensores. De esta forma, es posible utilizar estos datos para planificar rutas de conducción.

El mapeado del entorno, tal y como se ha planteado en el capítulo anterior, es el proceso de generación del modelo matemático de los alrededores del vehículo. Automated Driving Toolbox™ propone determinados algoritmos de SLAM para construir el mapa y estimar la ubicación del vehículo de forma simultánea [54].

4.3.5. Planificación y control

Mediante esta aplicación es posible planificar rutas de conducción, utilizando mapas reales importados directamente desde *OpenStreetMap* o *HERE HD Live Map*. Es posible utilizar y verificar la rapidez de algoritmos del tipo RRT, además de llevar a cabo una mejora continua de las trayectorias generadas. Además, esta aplicación permite diseñar sistemas de control de vehículos, utilizando controladores de los movimientos laterales y longitudinales del automóvil. Las opciones de control son implementadas mediante Simulink® [55].

Capítulo 5.

Estudio básico y modelado de un vehículo autónomo real

En los últimos años, los avances en vehículos autónomos han tomado un nuevo rumbo. La utilización de datos en la nube ha abierto un mundo de nuevas posibilidades a los desarrolladores de este tipo de vehículos, haciendo posibles nuevos tipos de pruebas de simulación de algoritmos, entrenamiento de modelos de aprendizaje profundo (*Deep Learning*) y generación de mapas de alta definición [56].

Los nuevos sistemas en la nube de conducción autónoma incluyen almacenamiento y computación distribuidos, así como sistemas de computación heterogénea. Cuando los algoritmos y las estrategias de conducción están listos, deben implementarse en sistemas de control reales con fines de ajuste y prueba antes de poner el automóvil en condiciones de tráfico reales. Esto implica contar con una plataforma de desarrollo hardware-software, con fines de control, que se pueda adaptar de forma sencilla y eficaz a cualquier automóvil, permitiendo a los desarrolladores concentrarse exclusivamente en los aspectos sensoriales y lógicos del prototipo [7].

De esta forma, en el Laboratorio de Vehículos Inteligentes de la Universidad Politécnica de Cartagena se ha desarrollado una plataforma, conocida como Cloud Incubator Car, que permite la experimentación de aspectos relacionados con la integración de sensores de terceros, control de dispositivos y prueba de algoritmos de conducción autónoma. En este capítulo se realizará una descripción del vehículo, y se realizará un modelado básico del mismo utilizando Automated Driving Toolbox™.

5.1. Diseño y estudio del Cloud Incubator Car

El Cloud Incubator Car (de ahora en adelante nombrado como CIC), es una iniciativa creada por la Universidad Politécnica de Cartagena para llevar a cabo pruebas de campo de sensores y algoritmos destinados a la conducción autónoma.

Esta plataforma está basada en un vehículo eléctrico comercial, modelo “Renault Twizy” (v*er Figura 5.1*), que ha sido objeto de diversas modificaciones: automatización de elementos principales de conducción, como la dirección, la caja de cambios, los frenos y el sistema de aceleración, instalación de soportes externos para sensores y acomodación del interior del vehículo para la instalación de sistemas de control, procesamiento de información y soporte de energía adicional.

Figura 5.1. *Izquierda, Cloud Incubator Car. Derecha, Renault Twizy.*

5.1.1. Modificación del sistema de dirección

Las modificaciones del sistema de dirección del vehículo son de tipo mecánico, tal y como se muestra en la *Figura 5.2.* Su principio de funcionamiento se basa en la transmisión de fuerzas sobre el eje principal. Un conjunto motor es fijado sobre la columna de dirección, y transmite el movimiento mediante engranajes. El eje original no ha sido modificado, por lo que el vehículo permite al conductor activar la dirección manual en todo momento. La implementación mecánica de la automatización en la dirección cumple los requisitos principales de precisión en la posición angular y velocidad máxima de movimiento del eje (60 rpm) [7].

El actuador del sistema de dirección está formado por un *driver* electrónico EPOS2 y un motor manufacturado por Maxon Motors (EC60fl-GP52C-1024IMP-100W).

5.1.2. Modificación del sistema de frenado

Las modificaciones sobre el sistema de frenado se han realizado de forma similar que en el sistema de dirección del vehículo. Un conjunto mecánico motor-engranaje acciona una leva que presiona el pedal de freno mediante un leve desplazamiento, de unos 15º, con control sobre la fuerza aplicada (ver *figura 5.2*). Tras las modificaciones realizadas, el sistema de frenos también puede ser activado de forma manual tal como ocurría con el sistema de dirección [7].

El actuador del sistema de frenado consiste en un controlador EPOS2 y un motor manufacturado por Maxon Motors (EC60fl-GP52C-1024IMP-100W).

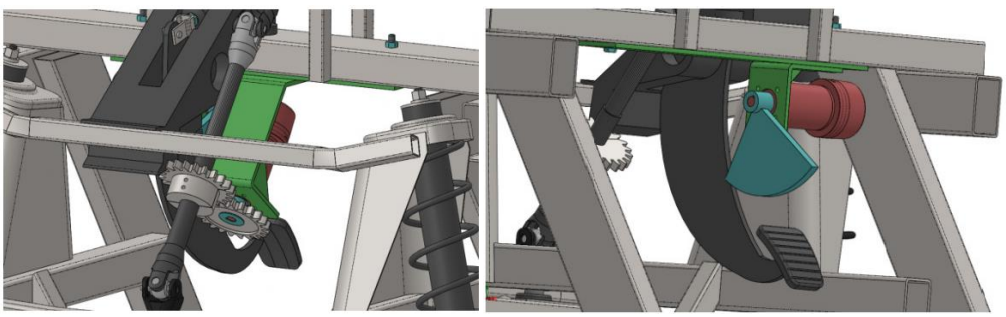


Figura 5.2. *Modificaciones realizadas sobre el sistema de dirección y sobre el sistema de frenado.*

5.1.3. Modificaciones sobre el acelerador y la caja de cambios

Al contrario que en las modificaciones presentadas anteriormente, las realizadas sobre el acelerador y la caja de cambios son de naturaleza electrónica. Después de escanear las señales de control de estos dispositivos, se intercambian datos con la ECU (Unidad de control electrónico) que permite seleccionar un estado PDNR en la caja de cambios (*Park*, estacionamiento, *Driver*, conductor, *Neutral*, *o Reverse*, revertido). Esto se realiza mediante un módulo software basado en una tarjeta analógica. Al igual que anteriormente, las modificaciones permiten el funcionamiento en paralelo del sistema original [7].

5.1.4. Soportes para sensores

Se han instalado en el CIC diferentes soportes con la capacidad de sostener distintos tipos de sensores, como cámaras de tiempo de vuelo, cámaras 3D, un sensor LiDAR, Unidad de Medición Inercial y GPS.

5.1.5. Arquitectura de comunicaciones del vehículo

Arquitectura a bajo nivel

Se ha modificado el interior del CIC para permitir el soporte eléctrico para sensores y comunicaciones. Esta infraestructura está compuesta por:

- Un bus de comunicaciones tipo CAN (*Controller Area Network*).
- Una red Ethernet Gigabyte, que permite la implementación de protocolos TCP y UDP.
- Un set de comunicaciones punto a punto, RS232 y USB.

La siguiente figura muestra los componentes que componen la arquitectura del vehículo autónomo [7]:

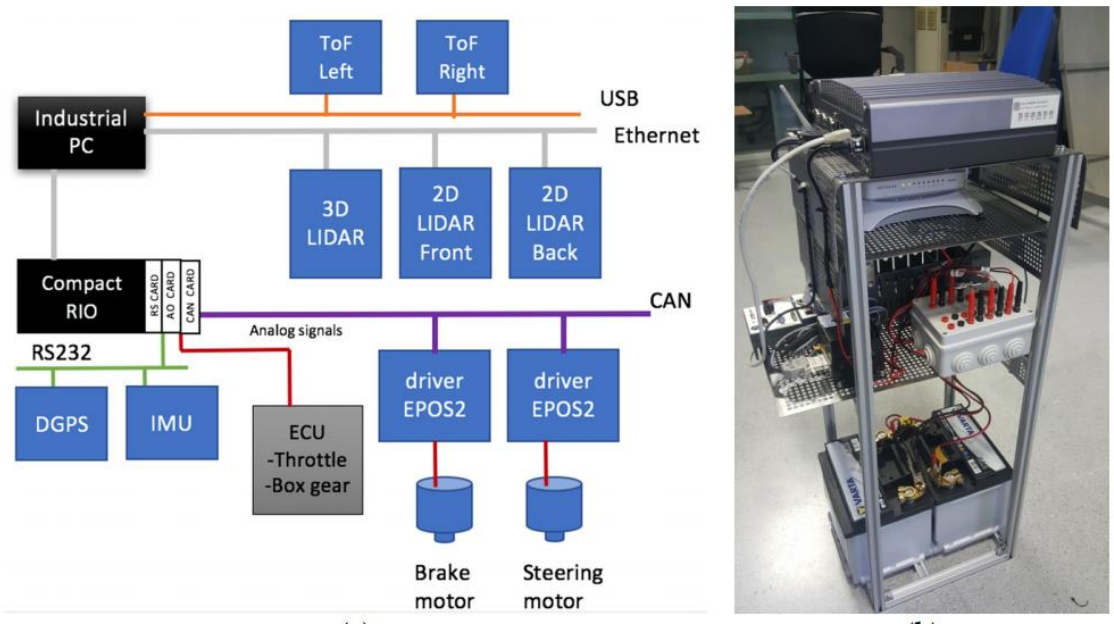


Figura 5.3. *Arquitectura de comunicaciones del CIC.*

Arquitectura a alto nivel

Desde el punto de vista de la arquitectura a un mayor nivel, el CIC se compone de diversos sistemas:

- Sistema de control.
- Sistema de percepción.
- Sistema de toma de decisiones.

La siguiente figura muestra el esquema de arquitectura a alto nivel del vehículo [7]:

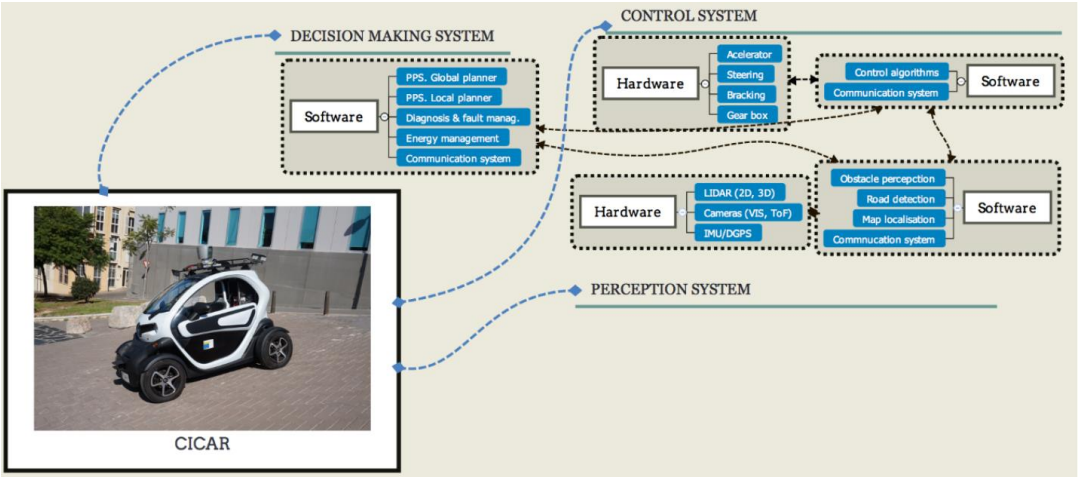


Figura 5.4. *Arquitectura del CIC a alto nivel.*

5.1.6. Sistema de control

El sistema de control del vehículo está compuesto de una unidad de procesamiento compactRIO 9082, manufacturada por National Instruments. Esta unidad dispone de dos microprocesadores: un Intel i7 de octava generación y un Xilinx FPGA, que confieren al vehículo de una alta capacidad de computación. El procesador Intel se hace cargo de las tareas que requieren poca carga de procesamiento, mientras que el FPGA utiliza sus altas frecuencias de reloj para asumir las tareas más críticas.

Esta unidad de procesamiento está gobernada por un sistema de operaciones en tiempo real (VxWorks, manufacturado por WindRiver) [57].

5.1.7. Elementos de percepción del entorno

El sistema de percepción que ha sido instalado en el vehículo consta de dos subsistemas: uno de largo alcance (LRS) y otro de corto alcance (SRS).

El sistema de percepción de corto alcance permite detectar elementos del entorno hasta una distancia de 10 m desde las partes delantera y trasera del vehículo, mediante un sensor LiDAR 2D. A los lados del vehículo, cámaras de tiempo de vuelo permiten la detección de objetos hasta unos 3 m de distancia. Este tipo de cámaras se ven menos afectadas por las condiciones lumínicas y permiten una alta velocidad de fotogramas.

El sistema de percepción de largo alcance está basado en un LiDAR 3D de alta definición, modelo Velodyne HDL64SE [58]. Este sensor gira a una velocidad de 800 rpm, y sus rayos láser pueden detectar objetos del entorno hasta una distancia de aproximadamente 100 m, con precisiones menores a los 2 cm. El sensor LiDAR permite el establecimiento de trayectorias, seguimiento de objetivos y clasificación de objetos. Además, este sistema está fusionado con una cámara de visión artificial instalada en el frontal del vehículo, que permite la detección de líneas de carretera, carriles, señales de tráfico y otros vehículos.

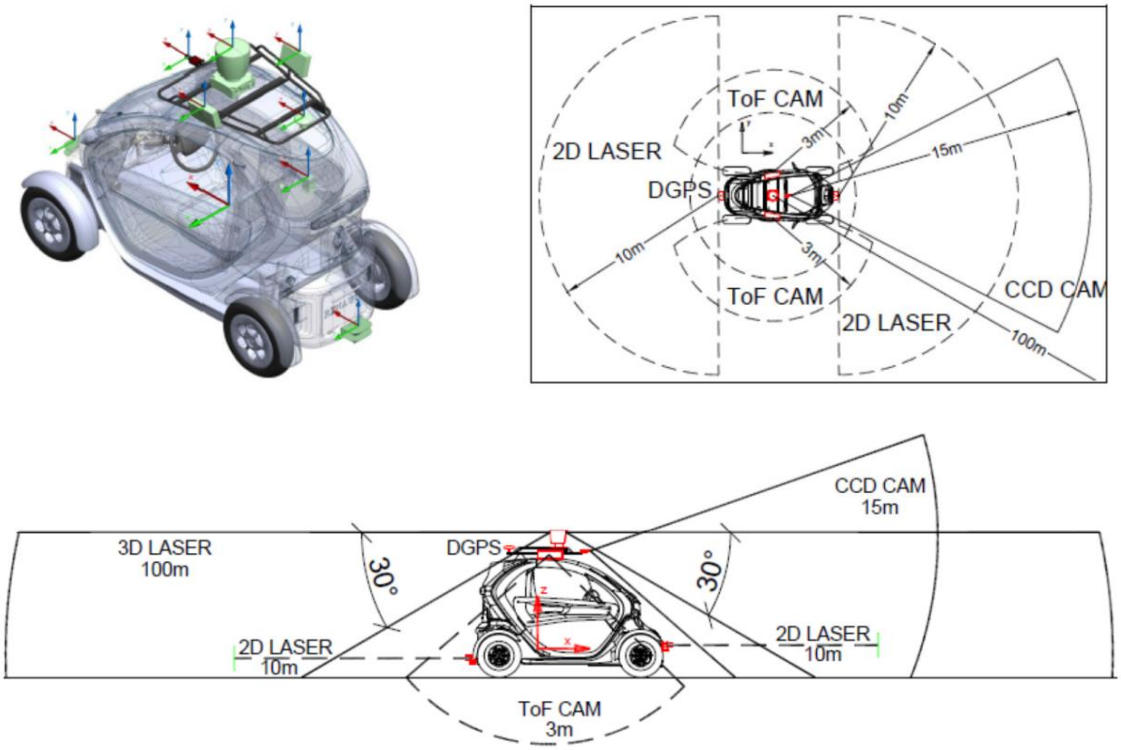


Figura 5.5. *Sistemas de percepción del CIC.*

Asimismo, el CIC incluye sistemas de localización, basados en satélites de posicionamiento global (GPS) y una Unidad de Medición Inercial. Un sistema RTK GNSS ha sido instalado en el vehículo, suministrando posición con una precisión de entre 20 y 1 cm [59], [60].

En la *Tabla 5.1.* se muestra un resumen de los sensores utilizados en el CIC:

Tabla 5.1. Sensores instalados en el CIC y su ámbito de aplicación.

Sensor	Características y configuración
SICK LASER Scanner 2D TIM551	Rango de medida: 0,05 m hasta 4 m - Ángulo de apertura: 270º
LiDAR Velodyne HDL64 Scanner 3D	Rango de medida: 120 m - 1,3 millones de puntos por segundo - Campo de visión: 26,9º
Prosilica GT1290 Cam	Color - 1,2 megapíxeles Ethernet Gigabit - Conector Ethernet RJ45
DGPS	GPS asistido mediante AHRS
ToF Sentis 3D-M420Kit Cam	Rango de medida: 4 m - Campo de visión horizontal: 90º
IMU Moog Crossbow NAV440CA-202	Precisión de cabeceo y balanceo: <0,4º - Precisión de posición: <0,3 m
EMLID RTK GNSS Receiver	Precisión en posicionamiento: 7 mm

Fuente: *Borraz, Raúl; Navarro, Pedro J; Fernández, Carlos; Alcover, Pedro M. (2018) "Cloud Incubator Car: A Reliable Platform for Autonomous Driving".* ***Publicado por****: Applied Sciences.*

5.1.8. Sistema de generación de trayectorias y toma de decisiones

El sistema de toma de decisiones del vehículo, cuyos algoritmos han sido perfeccionados y optimizados por los ingenieros del Departamento de Tecnologías de la Información y las Comunicaciones, incluye un planificador de trayectorias basado en curvas de Bézier (*Splines*), que permite planificar las rutas globales y locales del vehículo.

Planificador de trayectorias global

El planificador de rutas globales del CIC permite al vehículo utilizar la generación de mapas para crear trayectorias que se adapten a las restricciones de tráfico pertinentes, pero al mismo tiempo actuar de forma rápida y eficaz.

El algoritmo desarrollado para esta tarea es conocido como SCP (*Search for Cross Points*), y está basado en mapas binarios generados por una matriz de píxeles, donde el valor "1" se asigna a las zonas libres de obstáculos o "navegables", y el valor "0" a las zonas "no navegables".

De esta forma, el algoritmo se ejecuta en dos etapas diferenciadas:

- En la primera etapa, se escoge una dirección de búsqueda entre los cuatro puntos cardinales, dependiendo de la posición del vehículo captada de forma conjunta por el sistema GPS y la IMU. Cabe destacar que esta dirección es elegida midiendo el ángulo que forma el vehículo con el eje global horizontal θ.

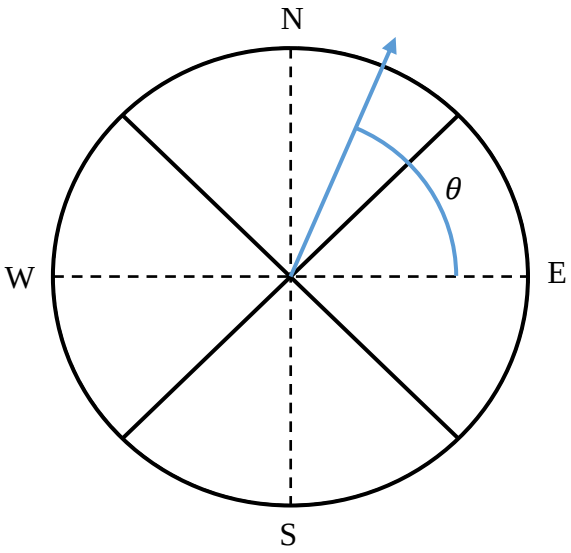


Figura 5.6. *Definición de la dirección inicial de búsqueda en el algoritmo SCP.*

$$Dirección\ de\ búsqueda: \begin{cases} Norte: -45° \leq \theta \leq -135° \\ Oeste: -135° \leq \theta \leq 135° \\ Este: 135° \leq \theta \leq 45° \\ Sur: 45° \leq \theta \leq -45° \end{cases} \qquad (5.1)$$

- Después de escoger la dirección de búsqueda de trayectorias, el algoritmo intentará encontrar una ruta entre el punto de inicio y el punto final la cual se encuentre compuestas de zonas navegables (definidas por el valor binario "1"). Para esto, utilizará los puntos cruzados o *cross points*, a partir de los cuales generará una curva *spline*. El primer punto hallado con el valor "0" será el primer punto de iteración del algoritmo (CP_0). Alternando el ángulo de dirección θ, mediante las ecuaciones siguientes:

$$x_i = x_{i-1} + d \cdot \cos(\theta) \qquad (5.2)$$

$$y_i = y_{i-1} + d \cdot \sin(\theta) \qquad (5.3)$$

El algoritmo encontrará el siguiente punto, CP_{i+1}. Actuando de forma iterativa, el sistema conseguirá llegar hasta el punto final de la trayectoria.

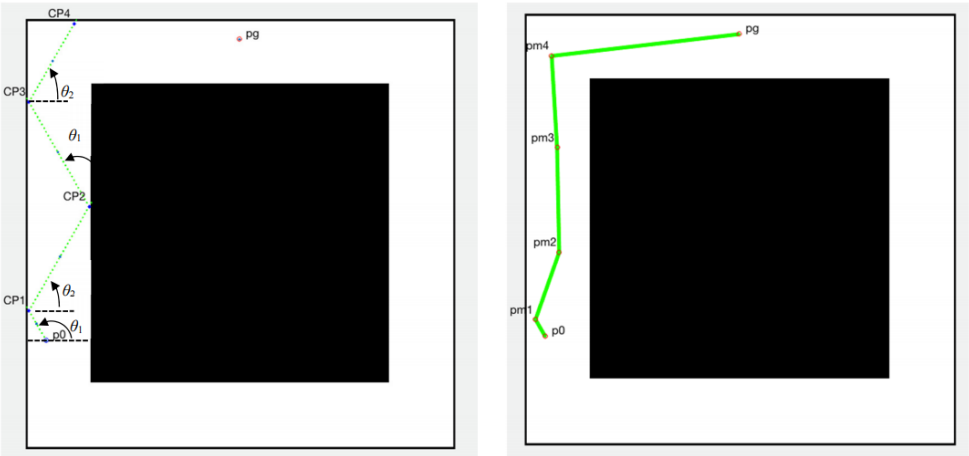


Figura 5.7. *Cálculo de trayectorias mediante el algoritmo SCP (I).*

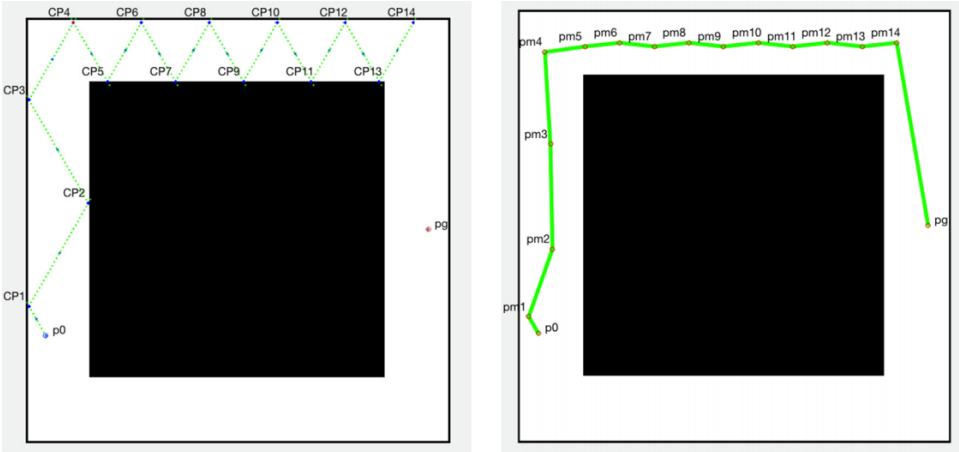


Figura 5.8. *Cálculo de trayectorias mediante el algoritmo SCP (II).*

Figura 5.9. *Algoritmo SCP aplicado a un mapa real (I).*

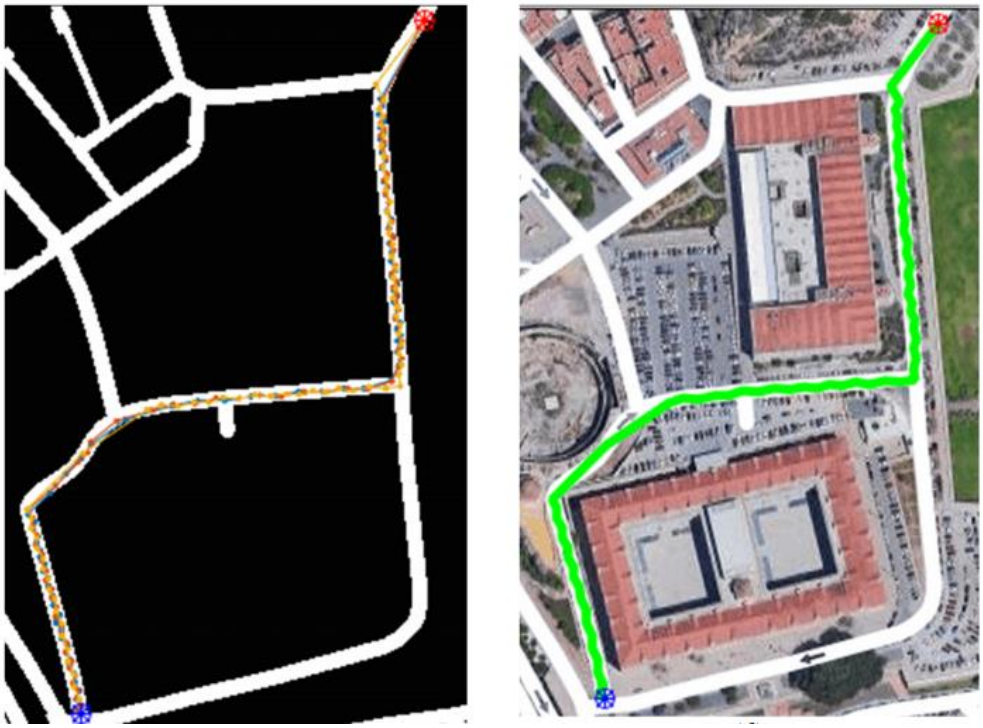


Figura 5.10. *Algoritmo SCP aplicado a un mapa real (II).*

Planificador de trayectorias local

El planificador de trayectorias local del CIC permite detectar obstáculos y sobrepasarlos, modificando la trayectoria global. Este algoritmo toma como punto de partida los puntos de ruta obtenidos mediante el algoritmo anterior, y las imágenes generadas por los sistemas de percepción del vehículo después de un proceso de filtrado.

El método utilizado para encontrar los bordes de los obstáculos en el entorno local del vehículo está basado en el algoritmo de Harris [61]. Este se centra únicamente en una zona limitada centrada en la dirección de avance del vehículo, conocida como "región de interés" (ROI). El flujograma del algoritmo se detalla en la *Figura 5.12.*

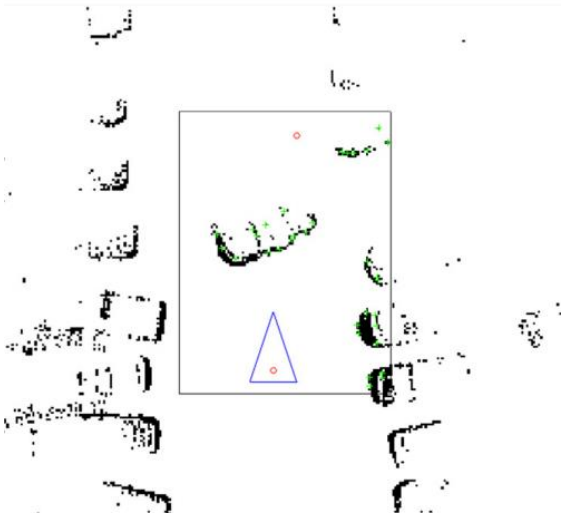


Figura 5.11. *Búsqueda de puntos clave sobre los obstáculos del entorno.*

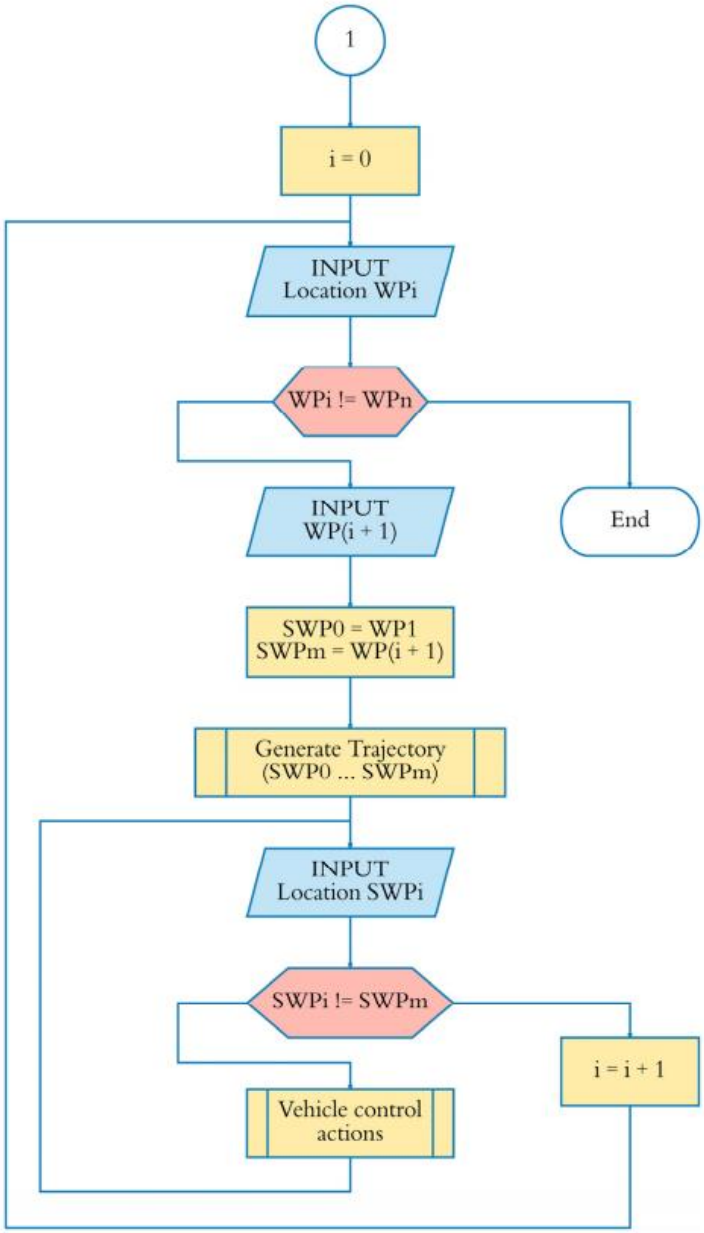


Figura 5.12. *Algoritmo de Harris para la búsqueda de puntos clave en la ROI del vehículo.*

En la *Figura 5.11*, el triángulo central representa el vehículo, y cada cruz de color verde es un conjunto de puntos obtenidos a partir de los límites del obstáculo detectado. Los círculos de color rojo, en cambio, son los puntos de inicio y fin de la trayectoria que debe crear el vehículo autónomo, obtenidos mediante el planificador global.

Sabiendo que una curva de Bézier puede ser generalizada mediante la ecuación:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} P_i (1-t)^{n-i} t^i, t \in [0,1] \qquad (5.4)$$

El algoritmo desarrollado calcula el número de trayectorias posibles e implementa curvas de Bézier basadas en polinomios de segundo grado:

```
VARIABLES:
Puntos:
P0(x0,y0), P1(x1,y1), P2(x2,y2), WPi, WP(i+1), KPi[] (siendo KPi[] la
matriz de puntos generados por el algoritmo de Harris).

ASIGNACIONES INICIALES:
P0 = WPi, P2 = WP(i+1)
P1:((WPi.x + WP(i+1).x) / 2, (WPi.y + WP(i+1).y) / 2)
P1.x = P1.x – ncurves * 3 / 2

PROCESO:
INICIO
WHILE (constant < ncurves)
{
      IF(P1 ∈ ROI)
      {
            B(t)=fB(P0, P1, P2)
            P1 = fP1(KP)
            constant = constant + 1
            IF((distance >= safety_distance) AND (B(t) ∈ ROI)
            {
                  Ti[] = B(t)
            }
      P1.x = P1.x + 3
      ncurves – ncurves + 1
}
FIN
```

Tras la ejecución del algoritmo, podemos obtener un conjunto de curvas que modifican la trayectoria local del vehículo [7]:

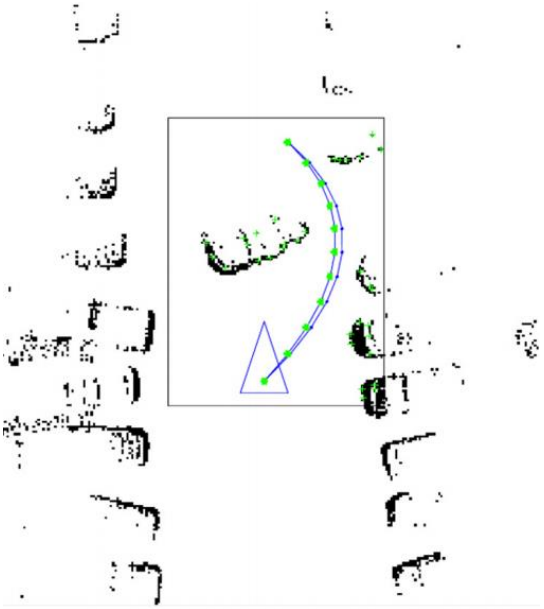


Figura 5.13. *Trayectorias finales generadas sobre el entorno del vehículo.*

5.2. Implementación del vehículo en el entorno de programación de MATLAB®

Para el modelado del vehículo en MATLAB® utilizaremos la herramienta Driving Scenario Designer, descrita anteriormente. Comenzaremos por definir un vehículo principal en el origen de coordenadas del espacio de programación interactiva de Automated Driving Toolbox™. Al comenzar modelando un vehículo en el entorno de simulación, el programa define sus valores por defectos como:

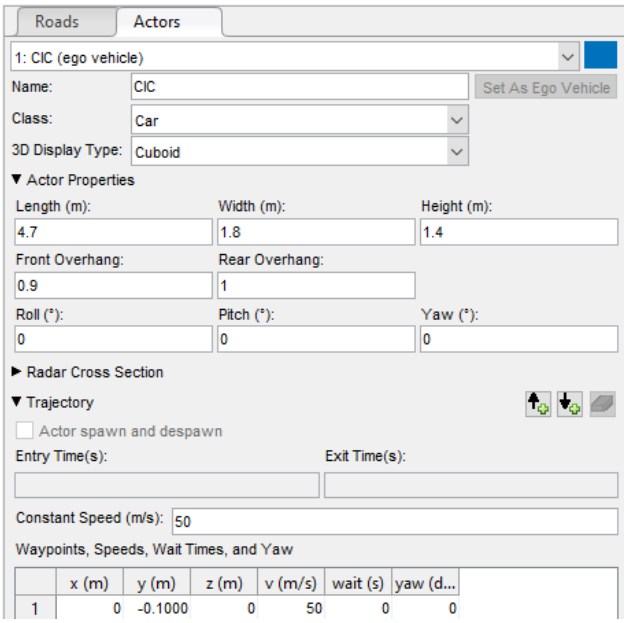


Figura 5.14. *Magnitudes físicas por defecto del vehículo definido en Driving Scenario Designer.*

5.2.1. Definición física del vehículo

Como se ha presentado anteriormente, el CIC es un vehículo de modelo Renault Twizy, cuyas medidas estándar en mm son las siguientes:

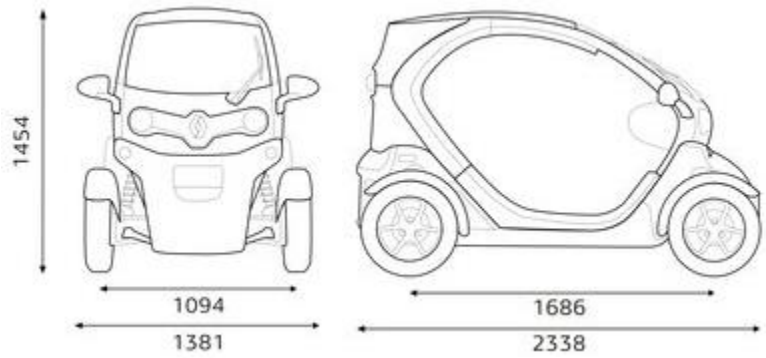


Figura 5.15. *Medidas del Renault Twizy.*

Sin embargo, no podemos introducir estas medidas exactas en el entorno de programación, debido a que se requiere, como se puede observar en la figura 5.1, las medidas adicionales del sensor LiDAR Velodyne HDL64E que sobresale por la parte superior del vehículo.

Según el fabricante [62], las medidas de este sensor son las siguientes:

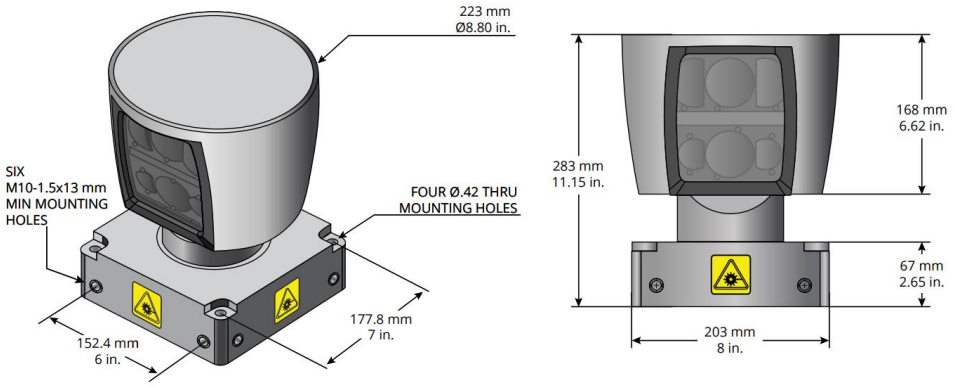


Figura 5.16. *Medidas del sensor LiDAR Velodyne HDL64E.*

De esta forma, el CIC será modelado físicamente como un cuboide de dimensiones 2,338 × 1,381 ×1,737 m.

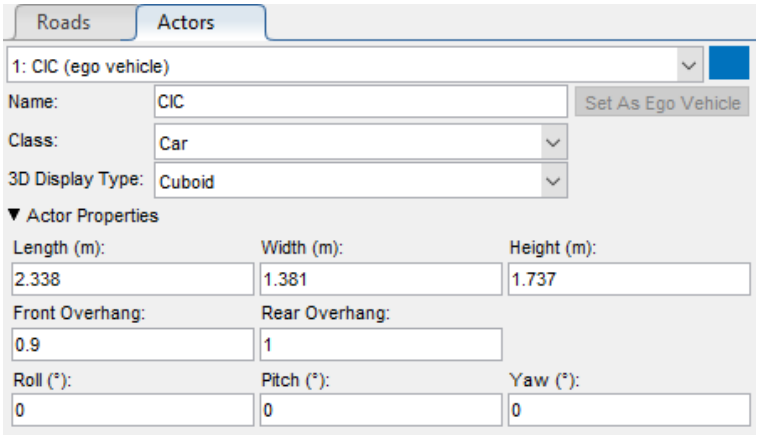


Figura 5.17. *Definición física del CIC mediante Driving Scenario Designer.*

5.2.2. Modelado de la fusión sensorial Cámara + LiDAR

Una vez definido el apartado físico del vehículo, es necesario modelar su sistema de sensores, presentado anteriormente. Para ello, utilizaremos los datos de la tabla 5.1:

- Un sensor LiDAR, con alcance de 120 m, y ángulo máximo de elevación de 26, 9º.
- Una cámara frontal de 33 fps (intervalo de actualización de 30 ms), con resolución 1280 × 960 píxeles.
- Dos cámaras ToF de 160 fps (intervalo de actualización de 6,25 ms), con una resolución de 120 × 160 píxeles.
- Dos sensores escáneres láser 2D de 10 m de rango de medida, modelados como radares en la parte frontal y trasera del vehículo, debido a su similar funcionamiento.

Mediante el *Driving Scenario Designer* podemos introducir fácilmente los parámetros de los sensores a modelar. En primer lugar, estableceremos una aproximación lo más fiel posible del sensor LiDAR Velodyne HDL64E, de la siguiente manera:

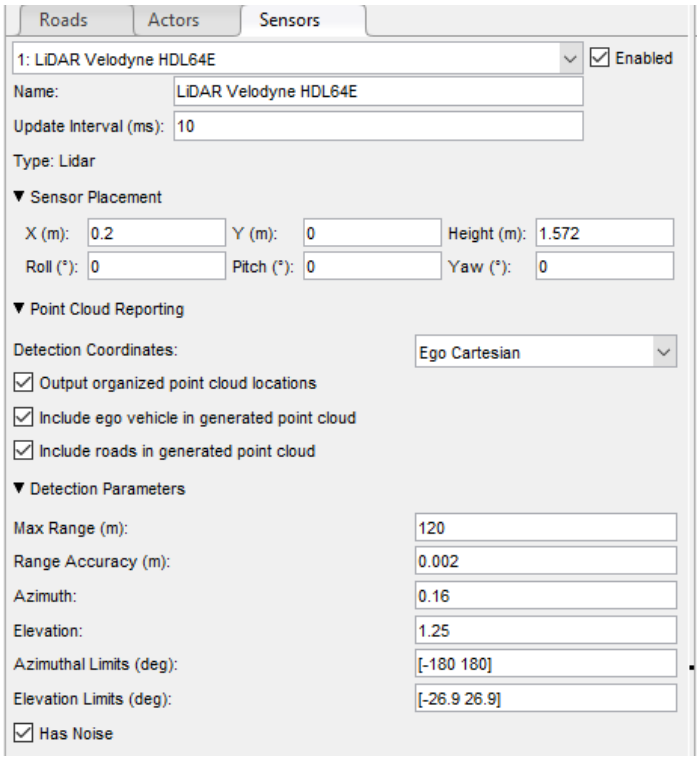


Figura 5.18. *Parámetros del sensor LiDAR en DSD.*

Cabe destacar que la frecuencia de intervalo del LiDAR es mucho menor que 10 ms, sin embargo, por motivos de falta de hardware, se establecerá una frecuencia de actualización de 10 ms, suficiente para obtener una nube de puntos correcta.

En segundo lugar, las cámaras frontales y laterales, serán introducidas en el entorno de simulación de la siguiente forma:

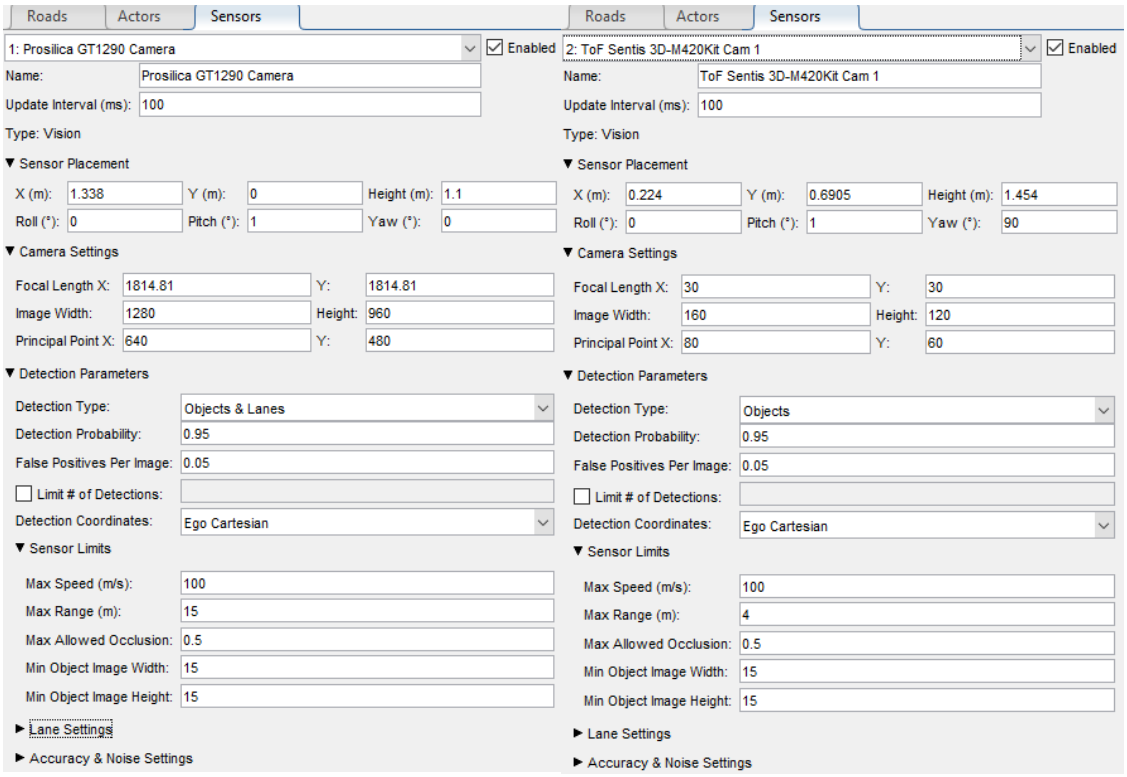


Figura 5.19. *Izquierda, parámetros de la cámara frontal en DSD. Derecha, parámetros de las cámaras ToF en DSD.*

Por último, los dos escáneres láser 2D, modelados como radares de rango 10 m, poseen los siguientes parámetros de configuración:

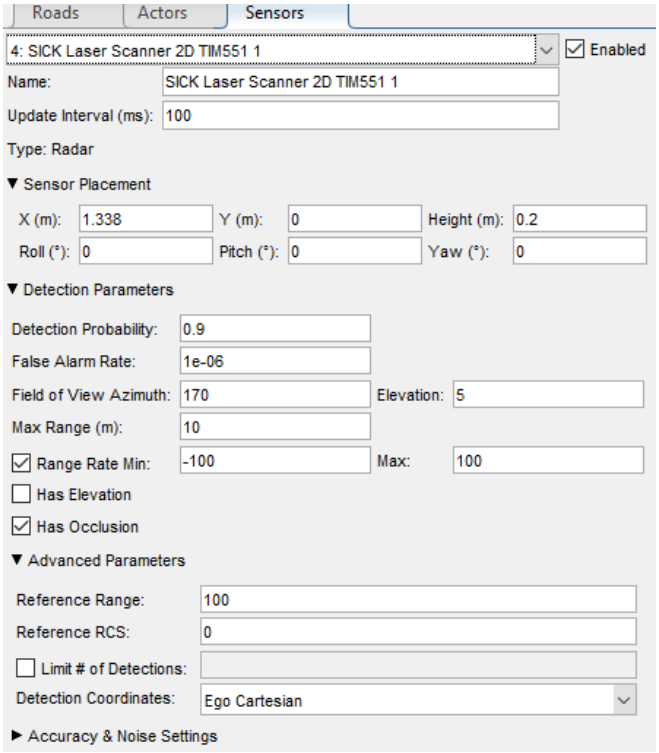


Figura 5.20. *Parámetros del escáner láser 2D en DSD.*

De esta forma, podemos obtener un modelo de vehículo autónomo representado de la siguiente forma en el entorno de programación:

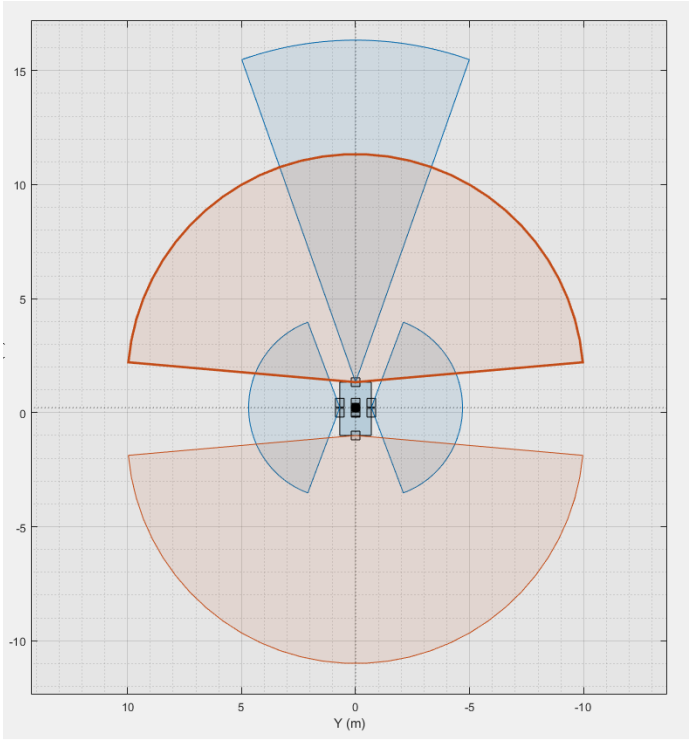


Figura 5.21. *Sensores principales del CIC y su rango de operación en DSD.*

Podemos realizar una simulación sencilla añadiendo puntos de ruta a nuestro vehículo directamente desde el DSD, para verificar que los sensores detectan las líneas viales y los obstáculos de forma correcta. Para ello, se ha implementado un escenario sencillo formado por una curva de carretera de doble sentido, con un camión circulando por el carril contrario y un ciclista que el CIC deberá adelantar.

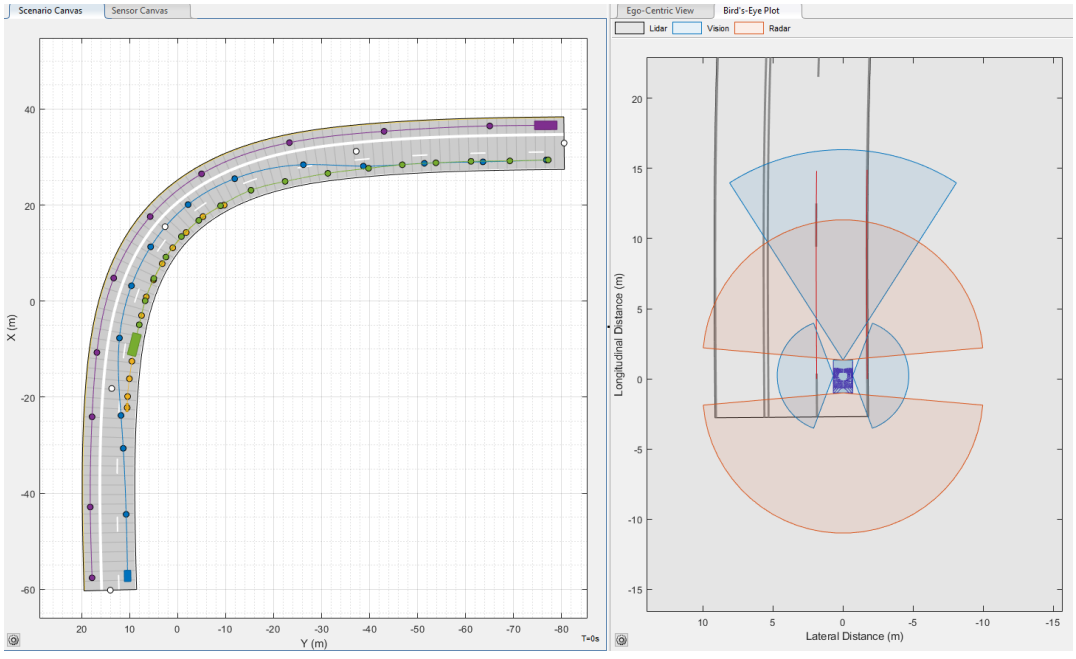


Figura 5.22. *Escenario de simulación.*

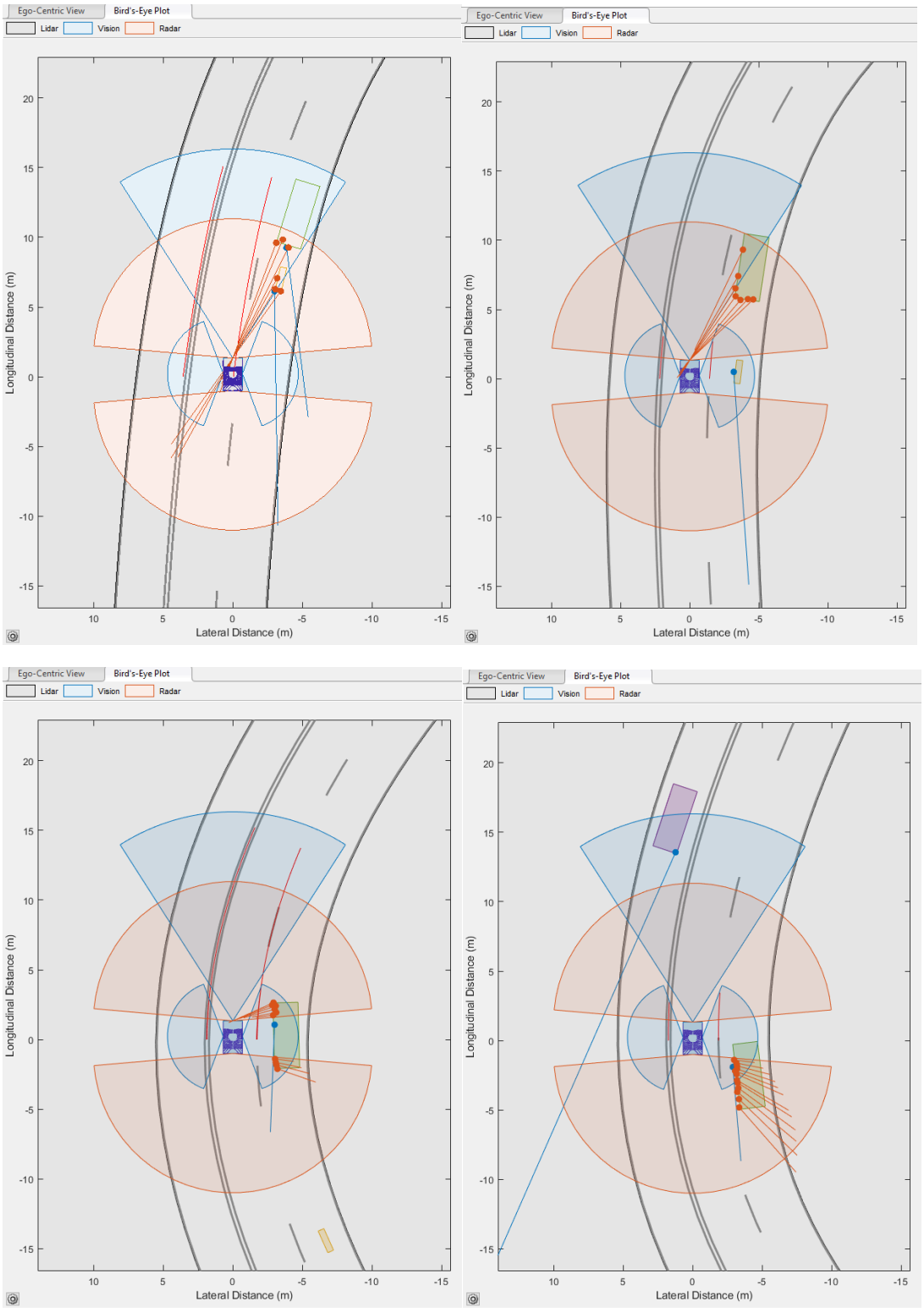


Figura 5.23. *Simulación de la fusión de sensores Cámara y LiDAR en el CIC.*

Capítulo 6.

Simulaciones de fusión sensorial mediante Automated Driving Toolbox™

Una vez planteados los conceptos principales en los que se basa la simulación de vehículos autónomos mediante MATLAB®, y haber realizado un repaso sobre las técnicas de percepción, fusión de sensores y toma de decisiones del Cloud Incubator Car (prototipo desarrollado por la Universidad Politécnica de Cartagena) se llevará a cabo una simulación de las herramientas de percepción principales de este vehículo, focalizando los esfuerzos en la tarea de seguimiento de múltiples objetivos mediante cámaras y LiDAR.

6.1. Creación del escenario de simulación

En primer lugar, se comenzará determinando el escenario de simulación. Se ha escogido el entorno de la Universidad Politécnica de Cartagena, tanto los alrededores de la Escuela Técnica Superior de Ingeniería Industrial como de la Escuela Técnica Superior de Ingeniería de Telecomunicaciones, por ser el principal terreno de pruebas del CIC, además de un entorno fácilmente reconocible.

Para crear el entorno de simulación, se ha accedido a la base de datos de OpenStreetMap, mediante la cual se ha acotado el mapa transferible a la aplicación. Seleccionando las coordenadas, obtenemos el mapa principal de la UPCT.

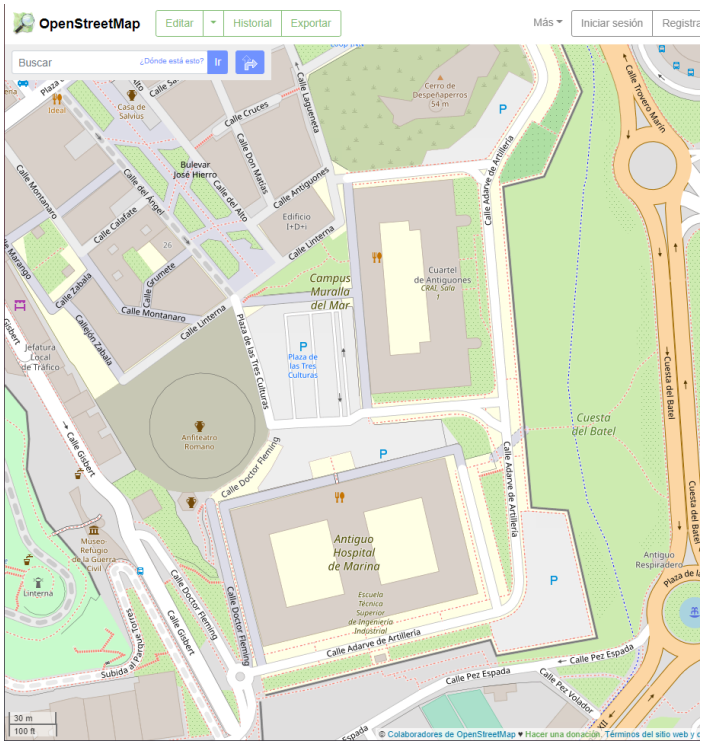


Figura 6.1. *Universidad Politécnica de Cartagena en OpenStreetMap.*

Tras la exportación del mapa a un archivo de extensión "*.osm*", es posible introducirlo en DSD y elegir las carreteras que formarán el entorno de simulación del vehículo.

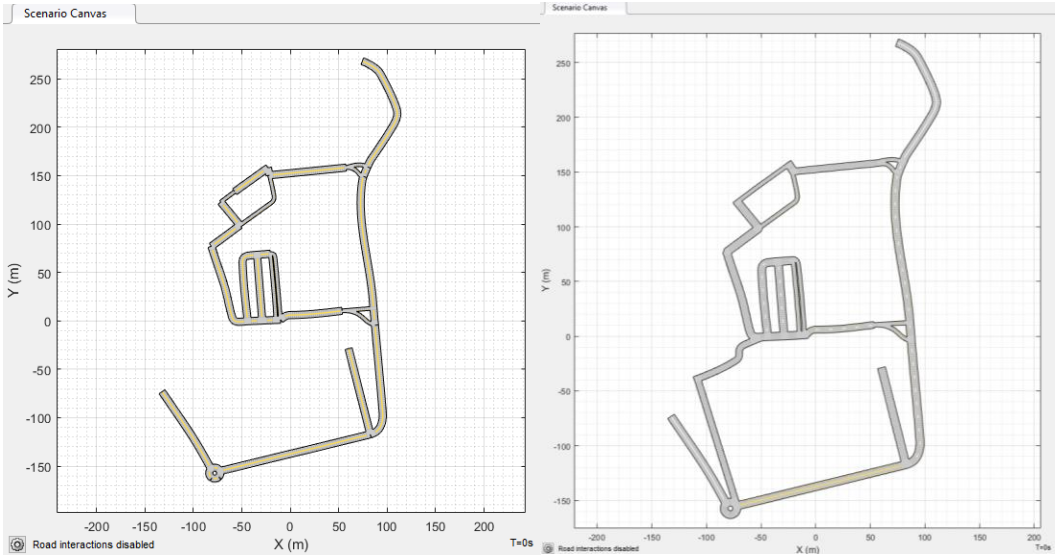


Figura 6.2. *Izquierda, mapa sin modificar. Derecha, mapa modificado final.*

Sin embargo, DSD genera carreteras por defecto, que deben ser actualizadas para obtener el mapa real del entorno de la UPCT. Se han añadido calles no importadas, modificado las líneas de carretera para que coincidan con las reales, y retocado ciertas uniones entre vías (*ver Figura 6.2*).

Una vez obtenemos el mapa de simulación, se añaden los actores. En primer lugar, se importó el modelo del CIC obtenido en el *Capítulo 5*. Se añadirá una trayectoria vial mediante curvas *Splines*, tal y como lo habría hecho el propio algoritmo de planificación de rutas. Aunque es posible programar esta característica mediante Automated Driving Toolbox™, se recuerda que no es objeto de estudio de este trabajo la simulación de algoritmos de planificación de rutas, sino la prueba de la fusión de sensores instalados en el vehículo.

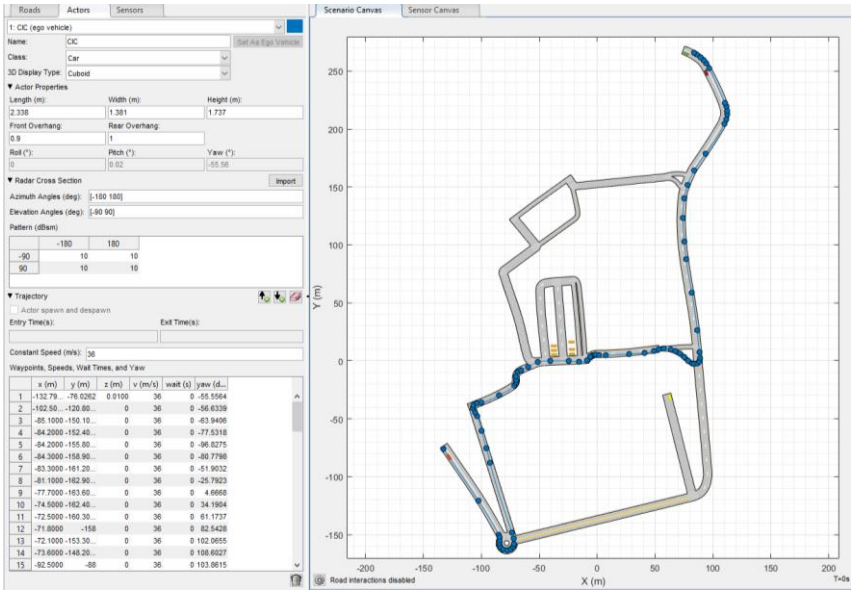


Figura 6.3. *Waypoints del actor principal (CIC).*

Tras varias pruebas realizadas, se ha creado un entorno de simulación con seis actores en movimiento, sin contar el propio vehículo principal. Cada uno de estos actores supondrá un reto al vehículo en su tarea de percepción y detección mediante la fusión de sensores Cámaras + LiDAR.

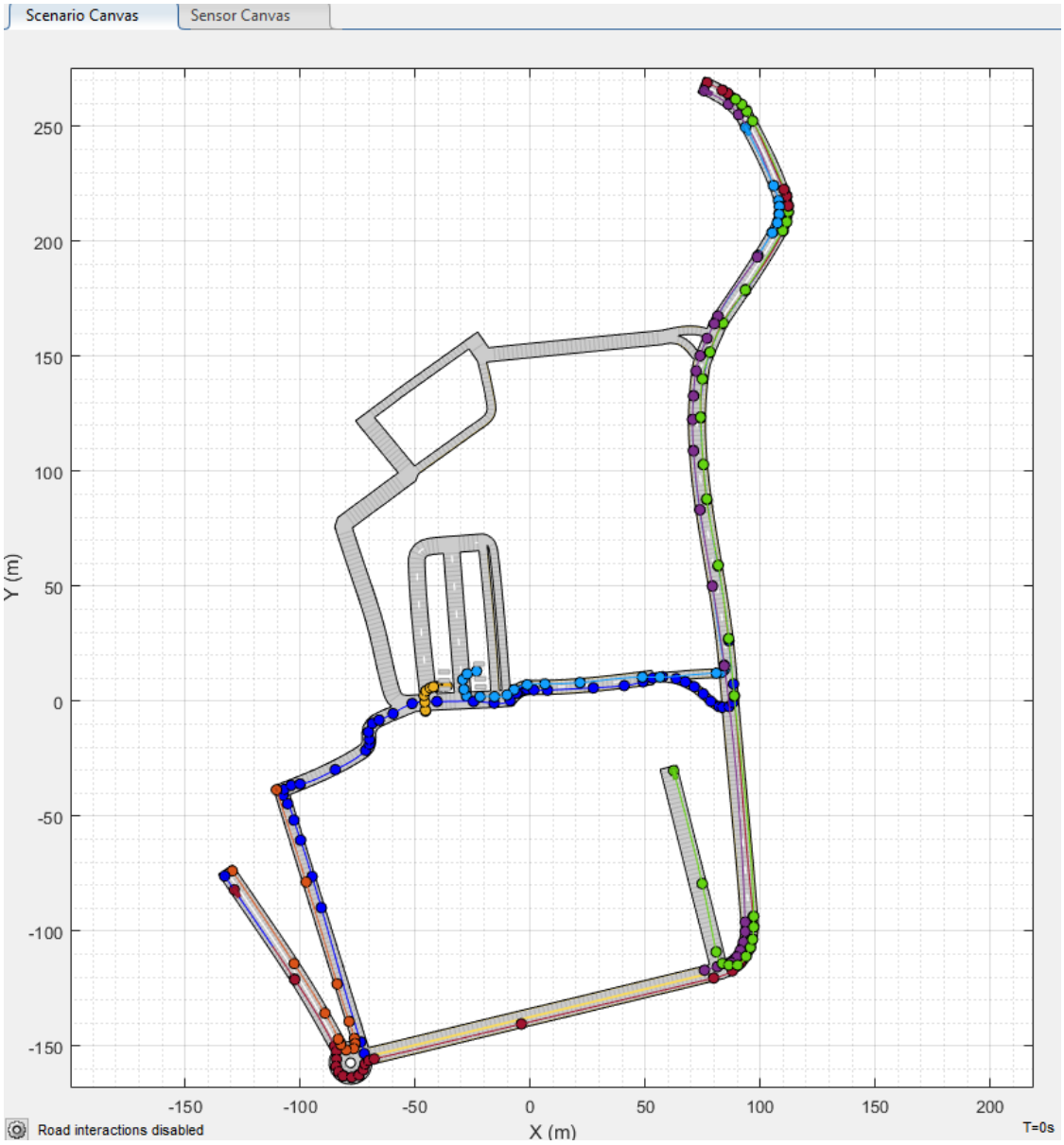


Figura 6.4. *Escenario final en DSD.*

El escenario de simulación se puede dividir en diferentes zonas o rutas fácilmente distinguibles:

- En el primer tramo de la simulación, el CIC circula manteniendo las distancias con un vehículo que se encuentra a unos metros de su parte delantera. Ambos vehículos se adentran en la rotonda, y posteriormente toman distintas direcciones de salida.
- En el segundo tramo, el CIC circula por una carretera estrecha, mientras se aproxima un ciclista en sentido contrario, y ambos vehículos se cruzan.

- En el tercer tramo, el CIC se detiene para dejar paso a un peatón que cruza el aparcamiento de la Universidad. Tras continuar su marcha, un vehículo que se dispone a entrar en el aparcamiento se encuentra con el CIC.
- En el último tramo de la simulación, el CIC toma la carretera principal y circula entre dos vehículos, manteniendo las distancias hasta el final de su ruta.

6.2. Seguimiento de objetos y vehículos mediante Cámaras y LiDAR 2D.

En este apartado, se mostrarán los avances realizados en una simulación de rastreo de vehículos en el entorno de simulación del CIC, mediante los sensores de imagen (cámaras) y sensores LiDAR.

6.2.1. Introducción

Mediante enfoques y algoritmos convencionales, como asociación conjunta de datos probabilísticos, la detección de un objeto mediante sensores genera una única señal de salida, es decir, un único punto de detección por sensor. Sin embargo, mediante escáneres LiDAR de alta definición y radares modernos, es posible obtener más de una detección por objeto, dependiendo de la forma y el tamaño de este [63], [64].

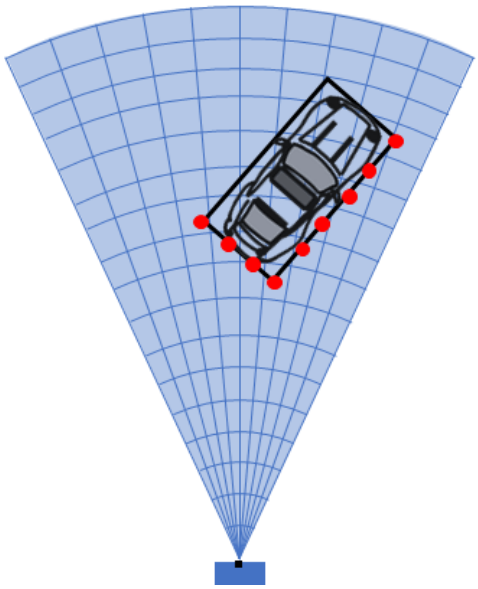
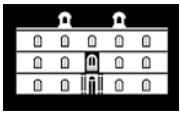


Figura 6.5. *Seguimiento extendido de objetos.*

El beneficio de este tipo de sensores HD es obtener más información sobre el objeto, es decir, dotar al vehículo autónomo de la capacidad de obtener dimensiones y orientación de los elementos de su entorno. Esta información adicional permite, en determinadas ocasiones, reducir el número de falsos positivos y mejorar la seguridad de la conducción.

Estos rastreadores de “objetos extendidos” (*Extended Objects Trackers*) como se les conoce dentro del mundo de la visión artificial, pueden estar configurados de diferentes



formas. En esta aproximación, se realizarán simulaciones mediante un rastreador convencional multi-objetivo, utilizando el objeto `multiObjectTracker` en MATLAB®.

6.2.2. Setup

En primer lugar, se importa el escenario de simulación creado anteriormente a la aplicación de MATLAB®. El código generado es el siguiente:

Definición del escenario

Creamos una instancia `drivingScenario`, con un tiempo de actualización de simulación de 10 ms, centrado en la referencia geográfica obtenida mediante *OpenStreetMap*.

```
scenario = drivingScenario('SampleTime', 0.01, ...
    'GeographicReference', [37.600675 -0.978985 0], ...
    'VerticalAxis', 'Y');
```

Creación de las carreteras que componen la simulación[1]

Para definir una vía de transporte, se deben especificar los centros de la carretera en cuestión, así como el tipo de líneas de marcado que se encuentran sobre la misma.

```
roadCenters = [-81.497 76.927 -0.00098531;
    -77.065 63.164 -0.00077869;
    -66.645 30.8 -0.00042234;
    -60.332 4.5057 -0.00029954;
    -59.33 1.9396 -0.00028437;
    -57.745 0.16722 -0.00026653;
    -55.578 -0.81129 -0.00024602;
    -52.83 -0.99596 -0.00022283;
    -43.968 -0.37725 -0.00015137];
marking = [laneMarking('Solid')
    laneMarking('Unmarked')
    laneMarking('Solid')];
laneSpecification = lanespec([1 1], 'Marking', marking);
road(scenario, roadCenters, 'Lanes', laneSpecification, 'Name', 'Plaza de las
Tres Culturas');
```

Programación del CIC y su trayectoria en un script de MATLAB®.

El vehículo principal, denominado `egoVehicle`, se define mediante sus características físicas. Después se debe definir su trayectoria mediante la función `trajectory(),` que requiere de al menos tres datos de entrada: el vehículo al que se le aplica la trayectoria, los puntos de ruta del vehículo sobre el escenario creado y su velocidad en cada uno de dichos puntos.

```
CIC = vehicle(scenario, ...
    'ClassID', 1, ...
    'Length', 2.338, ...
    'Width', 1.381, ...
    'Height', 1.7, ...
    'Position', [-132.796403878612 -76.0261585944371 0.01], ...
```

[1] El código de esta sección del programa se compone de 37 vías de transporte distintas, definidas de forma similar a la especificada en el ejemplo. No se ha incluido el resto de vías para no extender innecesariamente este documento.

```
    'Mesh', driving.scenario.carMesh, ...
    'PlotColor', [0 0 255] / 255, ...
    'Name', 'CIC');
waypoints = [];
speed = []
waittime = []
trajectory(egoVehicle, waypoints, speed, waittime);
```

Mediante las matrices `waypoints{}`, `speed{}` y `waittime{}`, se especifican respectivamente los puntos del terreno por los que debe moverse el vehículo, a qué velocidad y si debe esperar una cierta cantidad de tiempo.

Definición de los actores en el escenario de simulación[2]

Los actores se definen de la misma forma que el vehículo principal, definiendo sus características físicas y su trayectoria. La diferenciación entre tipos de actores se realiza, únicamente, mediante su definición de malla (`Mesh`), que define la forma física de los mismos.

```
bicycle = actor(scenario, ...
    'ClassID', 3, ...
    'Length', 1.7, ...
    'Width', 0.45, ...
    'Height', 1.7, ...
    'Position', [-110.3 -38.5 0], ...
    'Mesh', driving.scenario.bicycleMesh, ...
    'PlotColor', [217 83 25] / 255, ...
    'Name', 'Bicycle');
waypoints = [];
speed = [];
waittime = [];
trajectory(bicycle, waypoints, speed);
```

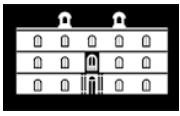
6.2.3. Programación de sensores

A continuación, se definirán los sensores del CIC, incluidos en una matriz denominada `sensors{}`, de 6 filas y 1 columna, para su posterior utilización.

```
sensors = cell(6,1);
profiles = actorProfiles(scenario);

% Cámara frontal
sensors{1} = visionDetectionGenerator('SensorIndex', 1, ...
    'UpdateInterval', 0.03, ...
    'SensorLocation', [1.338 0], ...
    'MaxRange', 15, ...
    'DetectionProbability', 0.95, ...
    'FalsePositivesPerImage', 0.05, ...
    'DetectorOutput', 'Objects only', ...
    'Intrinsics', cameraIntrinsics([1814.81, 1814.81],[640 480],[960 1280]),
    'ActorProfiles', profiles);
```

[2] En esta sección del programa desarrollado se ha incluido solo el código referente a uno de los 13 actores que componen el escenario de simulación. No se ha incluido el resto de actores para no extender innecesariamente este documento.



```
%Cámaras de tiempo de vuelo
sensors{2} = visionDetectionGenerator('SensorIndex', 2, ...
    'UpdateInterval', 0.01, ...
    'SensorLocation', [0.224 0.6905], ...
    'Height', 1.454, ...
    'Yaw', 90, ...
    'Roll', -10, ...
    'MaxRange', 4, ...
    'DetectionProbability', 0.95, ...
    'FalsePositivesPerImage', 0.05, ...
    'DetectorOutput', 'Objects only', ...
    'Intrinsics', cameraIntrinsics([30 30],[80 60],[120 160]), ...
    'ActorProfiles', profiles);
sensors{3} = visionDetectionGenerator('SensorIndex', 3, ...
    'UpdateInterval', 0.01, ...
    'SensorLocation', [0.224 -0.6905], ...
    'Height', 1.454, ...
    'Yaw', -90, ...
    'Roll', -10, ...
    'MaxRange', 4, ...
    'DetectionProbability', 0.95, ...
    'FalsePositivesPerImage', 0.05, ...
    'DetectorOutput', 'Objects only', ...
    'Intrinsics', cameraIntrinsics([30 30],[80 60],[120 160]), ...
    'ActorProfiles', profiles);

%Escáneres láser 2D
sensors{4} = radarDetectionGenerator('SensorIndex', 4, ...
    'UpdateInterval', 0.01, ...
    'SensorLocation', [1.338 0], ...
    'MaxRange', 10, ...
    'FieldOfView', [170 5], ...
    'ActorProfiles', profiles);
sensors{5} = radarDetectionGenerator('SensorIndex', 5, ...
    'UpdateInterval', 0.01, ...
    'SensorLocation', [-1 0], ...
    'Yaw', -180, ...
    'MaxRange', 10, ...
    'FieldOfView', [170 5], ...
    'ActorProfiles', profiles);

%Escáner LiDAR
sensors{6} = lidarPointCloudGenerator('SensorIndex', 6, ...
    'UpdateInterval', 0.01, ...
    'SensorLocation', [0.219 1.454], ...
    'ActorProfiles', profiles);
```

Como se puede observar, mediante la programación de sensores definida es posible caracterizarlos mediante su intervalo de actualización, su localización sobre el vehículo principal, su rango, y su campo de visión. Cada tipo de objeto sensor posee además características específicas.

6.2.4. Rastreador de vehículos y objetos

Para llevar a cabo el rastreo de objetivos y simular la percepción del vehículo, creamos un objeto `multiObjectTracker`, al que asignamos una función de inicialización y un umbral de tipo "*Threshold*" sencillo [65].

```
tracker = multiObjectTracker('FilterInitializationFcn', @initSimFilter, ...
    'AssignmentThreshold', 30, 'ConfirmationThreshold', [4 5]);
pS = [1 0 0 0; 0 0 1 0]; % Selector de posición
vS = [0 1 0 0; 0 0 0 1]; % Selector de velocidad
```

La función de inicialización del rastreador se define como:

```
function KF = initSimFilter(detection)
H = [1 0 0 0; 0 0 1 0; 0 1 0 0; 0 0 0 1];
KF = trackingKF('MotionModel', '2D Constant Velocity', ...
    'State', H' * detection.Measurement, ...
    'MeasurementModel', H, ...
    'StateCovariance', H' * detection.MeasurementNoise * H, ...
    'MeasurementNoise', detection.MeasurementNoise);
end
```

Se utiliza un modelo de velocidad constante en dos dimensiones para inicializar un filtro de Kalman lineal de seguimiento. Aunque ADT™ permite implementar otro tipo de filtros [53], se utilizará un KF por su sencillez de aplicación.

Aunque el seguimiento de objetivos se realiza normalmente en las tres dimensiones, el movimiento de los vehículos se limita al plano horizontal, por lo que se ha realizado una aproximación al entorno 2D [66].

6.2.5. Display de rastreo de objetivos

Para comprobar visualmente el rastreo de objetivos mediante cámaras y escáneres láser 2D, se creará una ventana de aplicación con tres *displays*, que serán actualizados según avanza la simulación.

El primer *display*, de vista de pájaro, permite visualizar las carreteras y los vehículos desde el aire para controlar la simulación. El segundo, sigue al CIC desde una vista en tercera persona. Por último, el display de sensores mostrará la actualización de cada uno de estos, identificando la distancia al objetivo y su dirección de avance respecto al vehículo principal.

Para crear la ventana de aplicación se ha creado una función, que permite organizar el código. En el *script* principal creamos un objeto *display* llamando a la función:

```
SensorPlot = createDisplay(CIC, sensors);
```

Siendo la función *createDisplay*:

```
function SensorPlot = createDisplay(CIC, sensors)
    % Creamos una ventana
    Window = figure('Position', [0, 0, 1200, 640], 'Name', 'Seguimiento ...
    de objetivos mediante fusión de sensores');
    movegui(Window, [10 -25]); % Genera la ventana arriba a la izquierda

    % Añadimos un gráfico que sigue al vehículo principal en tercera persona
    TPViewPanel = uipanel(Window, 'Position', [0 0 0.5 0.5], 'Title', ...
    'Seguimiento del CIC');
    CarPlot = axes(TPViewPanel);
    chasePlot(CIC, 'Parent', CarPlot);
```

```
% Añadimos un gráfico que sigue al vehículo desde arriba
    TViewPanel = uipanel(Window, 'Position', [0 0.5 0.5 0.5], 'Title', ...
    'Vista aérea');
    CarPlot = axes(TViewPanel);
    chasePlot(CIC, 'Parent', CarPlot, 'ViewHeight', 100, 'ViewLocation', ...
    [0 0], 'ViewPitch', 90);

    % Añadimos el display de sensores de vista de pájaro
    BEViewPanel = uipanel(Window, 'Position', [0.5 0 0.5 1], 'Title', ...
    Trackeo mediante cámaras y 2D laser scanner');

    % Creamos el gráfico y añadimos el rango de los sensores
    BEViewPlot = axes(BEViewPanel);
    SensorPlot = birdsEyePlot('Parent', BEViewPlot, 'Xlimits', [-65 65], ...
    'Ylimits', [-35 35]);

    % Graficamos el área que cubren las cámaras
    for i = 1:3
        cap = coverageAreaPlotter(SensorPlot,'FaceColor', ...
        'blue','EdgeColor','blue');
        if isa(sensors{i},'drivingRadarDataGenerator')
            plotCoverageArea(cap, sensors{i}.MountingLocation(1:2),...
                sensors{i}.RangeLimits(2), sensors{i}.MountingAngles(1), 45);
        else
            plotCoverageArea(cap, sensors{i}.SensorLocation,...
                sensors{i}.MaxRange, sensors{i}.Yaw, 45);
        end
    end

    % Graficamos el área que cubren los escáneres láser 2D
    for i = 4:5
        cap = coverageAreaPlotter(SensorPlot,'FaceColor','red', ...
        'EdgeColor','red');
        if isa(sensors{i},'drivingRadarDataGenerator')
            plotCoverageArea(cap, sensors{i}.MountingLocation(1:2),...
            sensors{i}.RangeLimits(2), sensors{i}.MountingAngles(1), ...
            sensors{i}.FieldOfView(1));
        else
            plotCoverageArea(cap, sensors{i}.SensorLocation,...
            sensors{i}.MaxRange, sensors{i}.Yaw, sensors{i}.FieldOfView(1));
        end
    end

    % Creamos el display de detección mediante visión
    detectionPlotter(SensorPlot, 'DisplayName','Visión', ...
    'MarkerEdgeColor','blue', 'Marker','^');

    % Se combinan las detecciones del escáner láser y se crea el display de
    % detección.
    detectionPlotter(SensorPlot, 'DisplayName','Escáner Láser 2D', ...
    'MarkerEdgeColor','red');

    % Se añaden los bordes de la carretera al gráfico
    laneMarkingPlotter(SensorPlot, 'DisplayName','Líneas');

    % Se añade el trackeo de objetivos al BEP
    trackPlotter(SensorPlot, 'DisplayName','Tracking', 'HistoryDepth',10);
```

```
    axis(SensorPlot.Parent, 'equal');
    xlim(SensorPlot.Parent, [-65 65]);
    ylim(SensorPlot.Parent, [-40 40]);

    % Se añade un trazador de contorno de objetivos
    outlinePlotter(SensorPlot, 'Tag', 'Ground truth');
end
```

6.2.6. Reproducción del escenario de simulación

Para hacer que el escenario de simulación avance, se necesita crear un bucle de actualización, definiendo en primer lugar las dos características principales de la simulación: la posición del CIC en todo momento, respecto al escenario en el que se encuentra situado, y el tiempo transcurrido de simulación.

En este bucle, se realiza una llamada a la simulación de sensores definida anteriormente, y se inicializa el seguimiento de objetivos. Cabe destacar que la frecuencia de actualización de los sensores y la del escenario pueden ser diferentes, sin embargo, la primera nunca debe ser menor que la segunda.

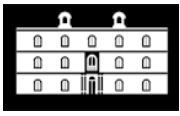
```
while advance(scenario)
    % Obtenemos el tiempo de simulación del escenario
    time = scenario.SimulationTime;

    % Obtenemos la posición del vehículo principal
    tp = targetPoses(CIC);

    % Simulamos los sensores
    detectionClusters = {};
    isValidTime = false(1,5);
    for i = 1:5
        [snsDet, numValidDet, isValidTime(i)] = sensors{i}(tp, time);
        if numValidDets
            for j = 1:numValidDet
                % Las detecciones mediante las cámaras no necesitan un
                % atributo SNR, por eso la definimos como NaN, ya que el
                % rastreador necesita que este objeto tenga la misma
                % estructura que la detección por radar.
                if ~isfield(snsDet{j}.ObjectAttributes{1}, 'SNR')
                    snsDet{j}.ObjectAttributes{1}.SNR = NaN;
                end

                % Eliminamos la componente Z de la medición (entorno 2D)
                snsDet{j}.Measurement = snsDet{j}.Measurement([1 2 4 5]);
                snsDet{j}.MeasurementNoise = ...
                snsDet{j}.MeasurementNoise([1 2 4 5],[1 2 4 5]);
            end
            detectionClusters = [detectionClusters; snsDet];
        end
    end

    % Actualizamos el rastreador si encontramos nuevas detecciones
    if any(isValidTime)
        if isa(sensors{1},'drivingRadarDataGenerator')
            vehicleLength = sensors{1}.Profiles.Length;
        else
            vehicleLength = sensors{1}.ActorProfiles.Length;
```



```
        end
        confirmedTracks = updateTracks(tracker, detectionClusters, time);

        % Actualizamos el gráfico de vista de pájaro
        updateSensorPlot(SensorPLot, CIC, detectionClusters, confirmedTracks,
        pS, vS);
    end
end
```

Se ha utilizado la función `updateSensorPlot` para realizar la actualización del gráfico de sensores:

```
function updateSensorPlot(SensorPlot, CIC, detections, confirmedTracks, pSel,
vSel)
    % Actualizamos las líneas de carretera en el gráfico SensorPlot
    [lmv, lmf] = laneMarkingVertices(CIC);
    plotLaneMarking(findPlotter(SensorPlot,'DisplayName','Líneas'),lmv,lmf);

    % Actualizamos el etiquetado de verdad del objetivo rastreado
    [position, yaw, length, width, originOffset, color] = ...
    targetOutlines(egoVehicle);
    plotOutline(findPlotter(SensorPlot,'Tag','Ground truth'), position, yaw,
    length, width, 'OriginOffset', originOffset, 'Color', color);

    % Se preparan y actualizan las detecciones en el display
    N = numel(detections);
    detPos = zeros(N,2);
    isRadar = true(N,1);
    for i = 1:N
        detPos(i,:) = detections{i}.Measurement(1:2)';
        if detections{i}.SensorIndex > 6 % Detecciones mediante visión
            isRadar(i) = false;
        end
    end
    plotDetection(findPlotter(SensorPlot,'DisplayName','Visión'), ...
    detPos(~isRadar,:));
    plotDetection(findPlotter(SensorPlot,'DisplayName','Escáner Láser 2D'),..
    detPos(isRadar,:));

    % Aquí se eliminan todas las detecciones de objetos que no han sido
    % identificados mediante visión antes de actualizar el display de
    % seguimeinto de objetivos (estos objetos pueden ser barreras u otros
    % obstáculos).
    isNotBarrier = arrayfun(@(t)t.ObjectClassID,confirmedTracks)>0;
    confirmedTracks = confirmedTracks(isNotBarrier);

    % Se prepara y se actualiza el display de seguimiento de objetivos
    trackIDs = {confirmedTracks.TrackID};
    labels = cellfun(@num2str, trackIDs, 'UniformOutput', false);
    [tracksPos, tracksCov] = getTrackPositions(confirmedTracks, psel);
    tracksVel = getTrackVelocities(confirmedTracks, vsel);
    plotTrack(findPlotter(SensorPlot,'DisplayName','Tracking'), tracksPos,
    tracksVel, tracksCov, labels);
end
```

Una vez obtenido el programa completo, podemos ejecutar la simulación y obtener los siguientes resultados.

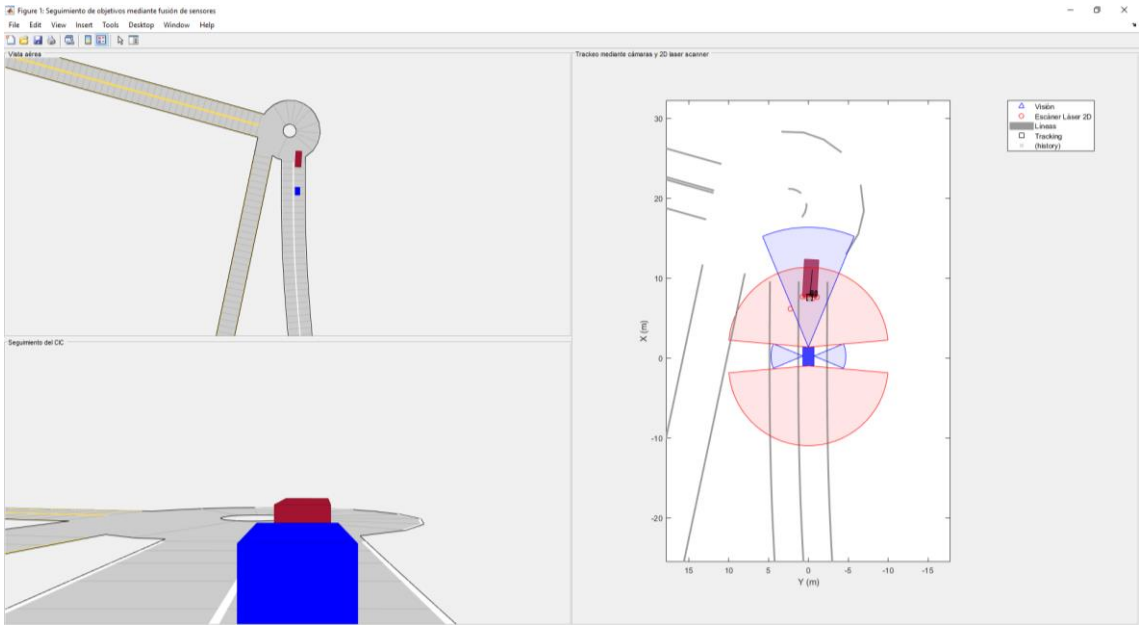


Figura 6.6. *Visualización global de la simulación de sensores.*

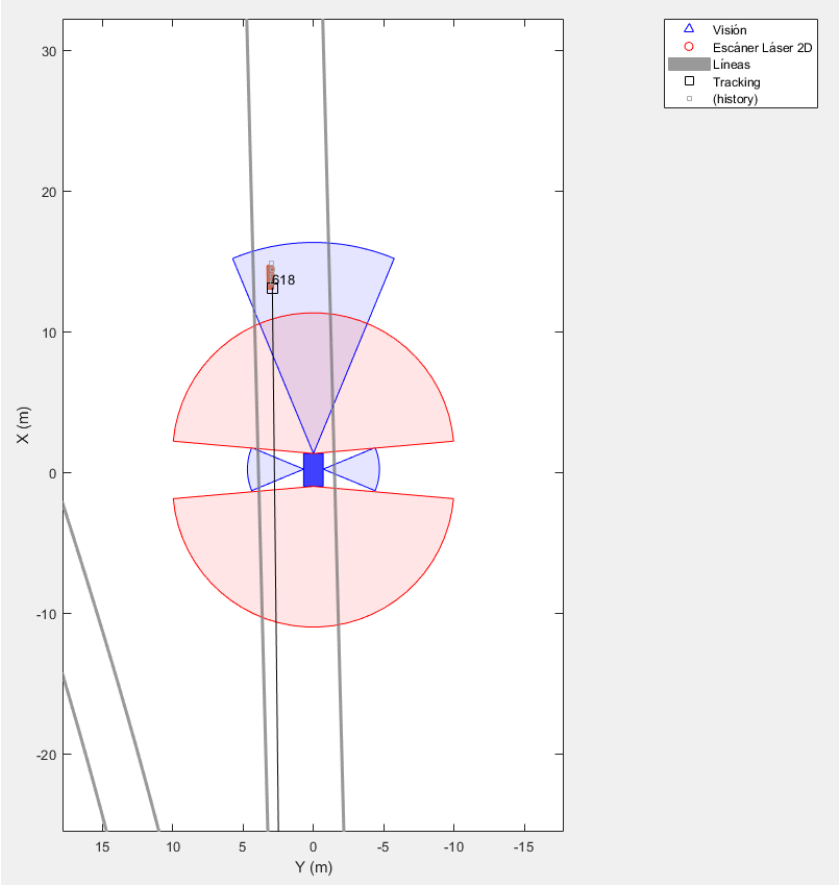


Figura 6.7. *Simulación de sensores. Seguimiento de un objetivo (bicicleta). Se indica su dirección, y distancia al vehículo principal.*

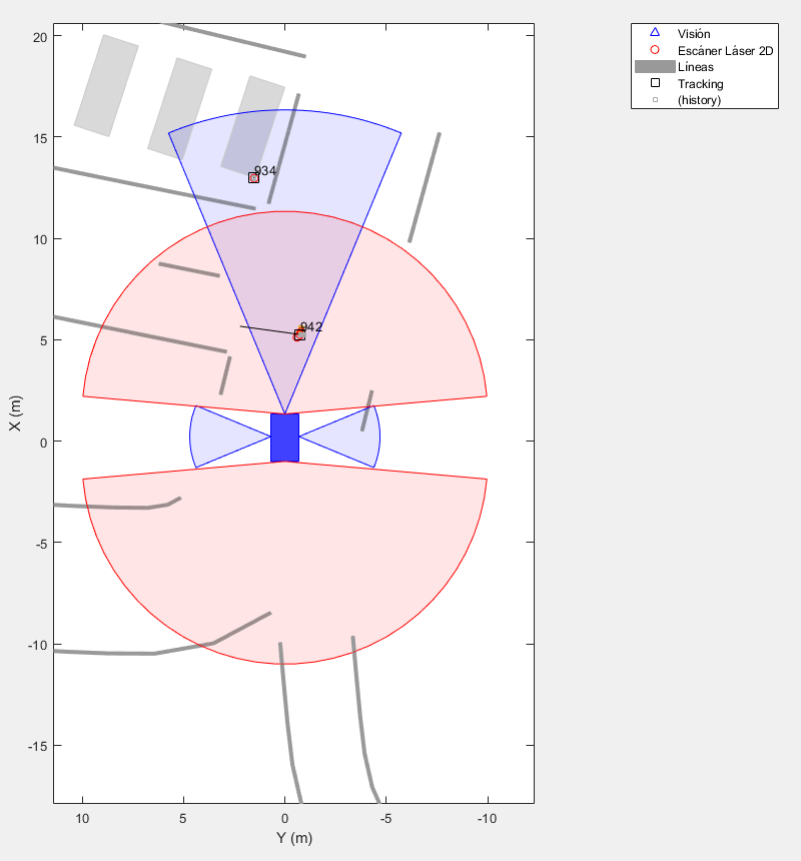


Figura 6.8. *Simulación de sensores. Detección y seguimiento conjunto de un peatón cruzando la vía, y un vehículo aparcado.*

6.3. Simulación del sensor LiDAR 3D

Como se ha especificado anteriormente, el CIC posee un sensor LiDAR 3D instalado en su parte superior, que permite llevar a cabo la tarea de mapeado del entorno y detección de objetivos en tres dimensiones. MATLAB® posee un paquete de herramientas con el que se puede hacer uso de este tipo de sensores y visualizar el entorno percibido (LiDAR Toolbox™). Mediante esta aproximación, se utilizará el programa desarrollado anteriormente para incluir la percepción del LiDAR en esta simulación.

6.3.1. Visualización de datos LiDAR 3D

La visualización de nubes de puntos LiDAR 3D no se puede llevar a cabo mediante simples gráficos en dos dimensiones. Para ello, se ha de definir una nueva ventana de programa, que incluya un reproductor de vídeo a tiempo real de tipo `pcplayer`.

En primer lugar, se definirán los límites del gráfico del reproductor y se creará el objeto que lo contendrá:

```
% Definimos los límites del gráfico
xlimits = [-20 20]; % en metros
ylimits = [-20 20];
zlimits = [0 10];
```

```
% Creamos el reproductor de vídeo
lidarPlayer = pcplayer(xlimits, ylimits, zlimits);
% Personalizamos las etiquetas de cada eje
xlabel(lidarPlayer.Axes, 'X (m)')
ylabel(lidarPlayer.Axes, 'Y (m)')
zlabel(lidarPlayer.Axes, 'Z (m)')

title(lidarPlayer.Axes, 'Datos del Velodyne HDL64E)
```

Dentro del bucle de simulación de escenario, se ha de activar la detección mediante el sensor LiDAR, y captar la nube de puntos generada en un objeto. El reproductor se actualizará en tiempo real para mostrar la nube de puntos generada.

```
% Actualizamos el gráfico de nube de puntos
    mesh = roadMesh(CIC);
    [ptCloud, isValidTime] = sensors{6}(tp, mesh, time);
    if isValidTime
        view(lidarPlayer, ptCloud);
    end
```

Merece la pena destacar que, en un entorno real, cada vehículo u objeto posee una forma diferente, que será visualizada mediante el sensor LiDAR. Sin embargo, al trabajar con una simulación, se ha de definir la forma que tienen los actores a detectar por el vehículo mediante su malla de puntos físicos, accesible mediante la propiedad `Mesh`.

En este caso, la malla de puntos será obtenida desde el vehículo principal, tal y como se especifica en:

```
mesh = roadMesh(egoVehicle);
[ptCloud, isValidTime] = sensors{6}(ta, mesh, time);
```

Los puntos obtenidos serán almacenados en la variable `ptCloud`, que será el objeto de visualización del reproductor. Al reproducir la simulación, obtendremos la representación siguiente:

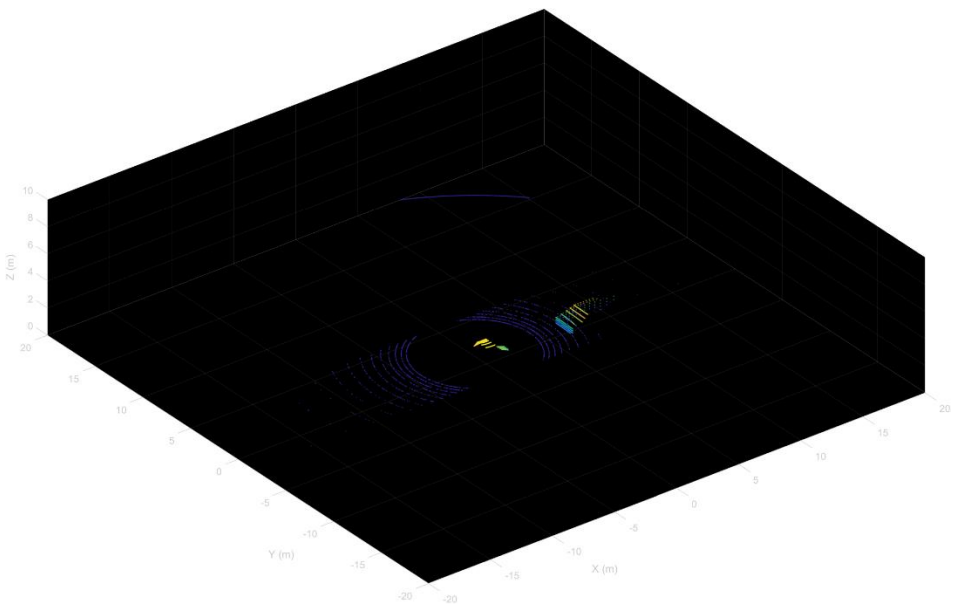


Figura 6.9. *Simulación del sensor LiDAR 3D (I).*

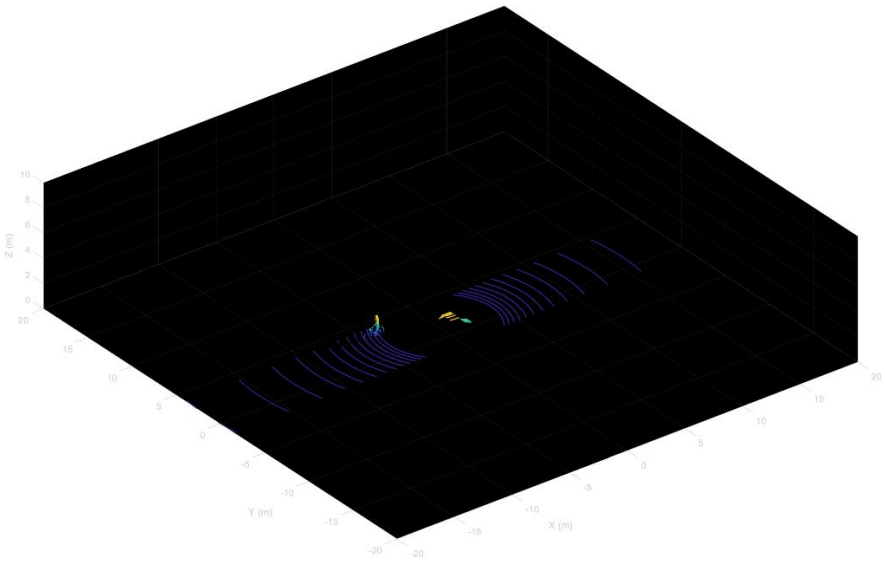


Figura 6.10. *Simulación del sensor LiDAR 3D (II).*

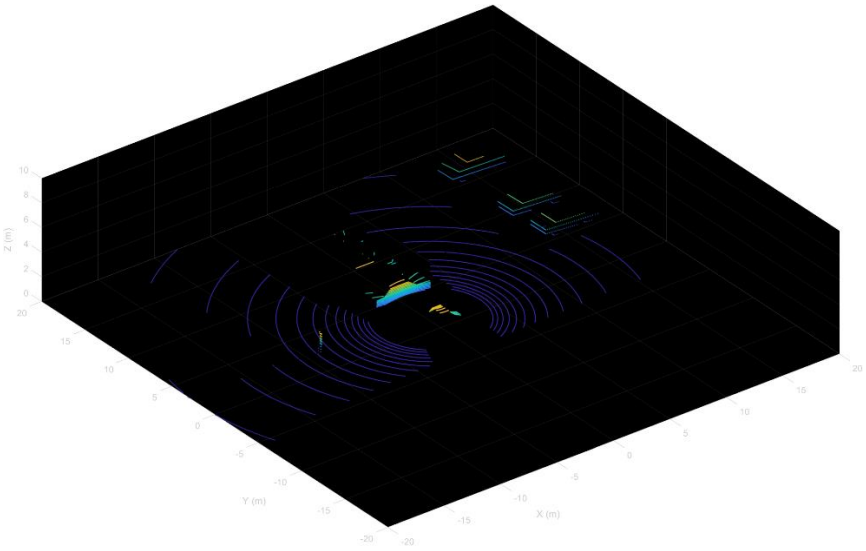


Figura 6.11. *Simulación del sensor LiDAR 3D (III).*

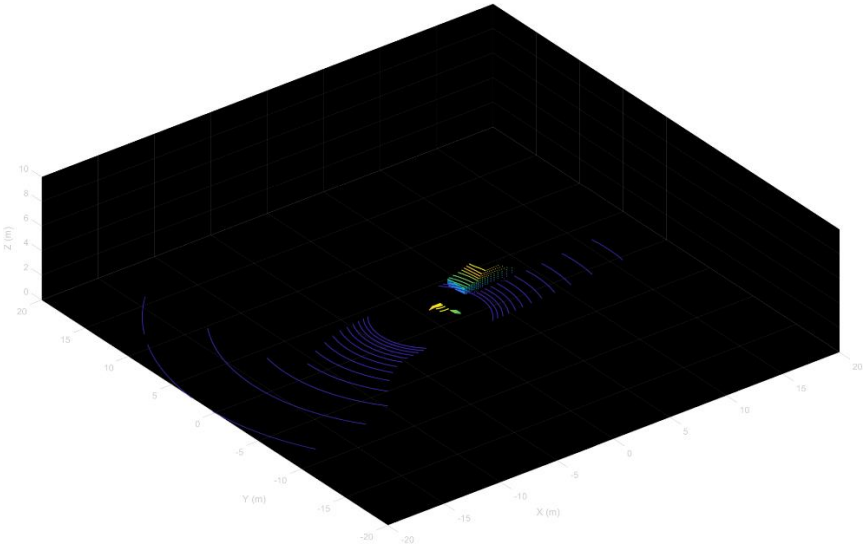


Figura 6.12. *Simulación del sensor LiDAR 3D (IV).*

Capítulo 7.

Conclusiones del trabajo

En este capítulo final, se valorarán los resultados obtenidos con el trabajo, tanto desde su parte bibliográfica y teórica, como desde su parte práctica, relacionada con la programación y la visión artificial. Además, se llevará a cabo una reflexión personal sobre los entornos de simulación de vehículos autónomos, el futuro de la industria en este ámbito, y se presentarán las posibles vías de avance en esta materia, tomando como punto de partida el presente documento.

7.1. Conclusiones

Las conclusiones obtenidas tras la resolución de este trabajo son las siguientes:

- Se ha realizado un estudio científico del estado del arte en materia de conducción autónoma. Este supone un recorrido por las principales técnicas de percepción, localización, construcción de mapas, y planificación de trayectorias usadas en vehículos autónomos actuales, y demuestra que este tipo de tecnologías, aunque algunas todavía se encuentran en desarrollo, supondrán un avance significativo no solo en la industria, sino también en el mundo del transporte.
- Se ha llegado a la conclusión de que, en la actualidad, la fusión de sensores es la única herramienta eficaz y disponible para obtener y procesar datos en bruto procedentes tanto del entorno, como del interior de los vehículos autónomos. Aunque existen multitud de algoritmos de fusión de datos, aquellos explicados en profundidad en este documento (Bayes y Filtros de Kalman, Dempster-Shafer, y técnicas avanzadas de inteligencia artificial y conjuntos difusos) [25], son los aplicados actualmente en materia de conducción autónoma, debido a su sencillez de uso y a sus buenos resultados.
- Tras explicar la necesidad de llevar a cabo simulaciones de entornos viales por parte de los desarrolladores de vehículos, se han presentado las características principales de Automated Driving Toolbox™, perteneciente a MATLAB®. Estas herramientas permiten el desarrollo de ADAS y la prueba de algoritmos relacionados con la conducción autónoma, y es una solución a tener en cuenta en la realización de simulaciones viales de un conjunto de sensores instalados en un vehículo.
- Se ha estudiado el diseño del Cloud Incubator Car, vehículo autónomo desarrollado por la UPCT, además de sus herramientas principales de percepción. Estas se han modelado en el entorno de programación de MATLAB® de forma sencilla, sirviendo de base para aquellas personas o empresas que deseen llevar a cabo simulaciones de su propio vehículo.

- Para finalizar, se ha llevado a cabo la implementación del CIC en un entorno vial simulado, tomando como base los alrededores de la Universidad Politécnica de Cartagena. Se han probado los sensores incluidos en el vehículo, realizando un programa que permite la detección de objetivos parados o en movimiento, mediante cámaras y sensores LiDAR. Cabe destacar que se requiere gran potencia de procesamiento para incluir un mayor número de actores en el escenario desarrollado, estando limitada así su similitud con la realidad. Merece la pena comentar que ADT™ tiene un campo de aplicación limitado en cuanto al estudio de simulaciones con metraje de cámaras y datos LiDAR reales, pero suficiente para el desarrollo de algoritmos de visión artificial y detección de objetivos. De esta forma, se recomienda el uso de este software durante la primera etapa de diseño del vehículo autónomo, haciendo más sencilla la tarea de creación de algoritmos y sistemas de conducción, pero no durante su etapa de pruebas en campo.

De esta forma, se puede asegurar que se han cumplido todos los objetivos propuestos para este TFG.

7.2. Vías de trabajo futuras

En este apartado se presentan una serie de líneas o proyectos que podrían tener como punto de partida el trabajo aquí desarrollado:

- Utilizar mayor potencia de procesamiento para aumentar el número de actores en el escenario y crear simulaciones de los sensores más fieles a la realidad. De la misma forma, se podría introducir el modelo del CIC en otro tipo de entornos, y llevar a cabo la simulación de los algoritmos desarrollados.
- Implementar el algoritmo de planificación de trayectorias mediante SCP en el entorno de programación de MATLAB®, llevando a cabo simulaciones no solo de la fusión sensorial del vehículo, sino también de otros sistemas lógicos.
- Utilizar la inteligencia artificial y estrategias de *Machine Learning*, para completar el algoritmo de detección desarrollado en este documento, permitiendo al CIC clasificar los objetos de su entorno.
- Utilizar metraje real obtenido con los sensores del CIC para programar los algoritmos de conducción autónoma, algo que no ha podido realizarse en este trabajo debido a las condiciones de la pandemia.
- Llevar a cabo una comparación técnica de la simulación del CIC mediante MATLAB® y ADT™, con otro tipo de simulaciones realizadas mediante otros entornos comerciales o de código abierto, como ROS (para sistemas operativos de tipo Unix, como Linux) o PaTAVTT.

7.3. Reflexiones personales

La conducción autónoma es, indudablemente, uno de los ámbitos de la industria que más acercan la actualidad al mundo del futuro. Aquello que parece ciencia-ficción está cada vez más cerca, y podemos observarlo cada día, no solo a través de grandes compañías, como Tesla o Google (las cuales dedican grandes cantidades de dinero a esta tarea), sino también a través de organismos como la Universidad Politécnica de Cartagena, que promueven el avance de la tecnología y la ingeniería a través de sus profesores y estudiantes.

De la misma forma, la fusión de sensores es una disciplina que deja muy atrás a otras tecnologías mono-sensor desarrolladas en el pasado. Sin embargo, desde un punto de vista más personal, será únicamente la punta del iceberg en el desarrollo de los vehículos del futuro. Los algoritmos de inteligencia artificial crecerán de manera exponencial, ofreciéndonos cada vez más formas de resolver los problemas que suponen la percepción del entorno y la replicación de las capacidades humanas al volante. Actualmente, existe una tendencia (observable en empresas como Tesla) a utilizar únicamente sistemas de percepción formados por cámaras y ayudados por redes neuronales [67].

Tal y como se comentó en el *Capítulo 4* del presente documento, la utilización de entornos de programación y simulación es una tarea necesaria para el desarrollo de vehículos autónomos en la actualidad; no basta con haber desarrollado un sistema que funcione correctamente; el uso de estos entornos de simulación es, principalmente, minimizar los posibles accidentes causados mediante las pruebas en campo del vehículo, además del perfeccionamiento de sus algoritmos implementados.

Figura 7.1. *Conducción autónoma, el futuro de la industria de la automoción.*

Referencias

[1] R. Lanctot, «Intel Newsroom,» Strategy Analytics, Junio 2017. [En línea]. Available: https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/05/passenger-economy.pdf. [Último acceso: Septiembre 2020].

[2] F. Richter, «Statista,» Statista, 29 Agosto 2017. [En línea]. Available: https://www.statista.com/chart/10879/autonomous-driving-patents/. [Último acceso: Septiembre 2020].

[3] E. Espinós, «Autofácil,» LUIKE, 9 Junio 2017. [En línea]. Available: https://www.autofacil.es/trafico/2017/06/09/71-91-accidentes-causados-factor-humano/38960.html. [Último acceso: Septiembre 2020].

[4] KM77, «Conducción autónoma: Niveles y Tecnología,» Ruedas de prensa S.L., 2018. [En línea]. Available: https://www.km77.com/reportajes/varios/conduccion-autonoma-niveles. [Último acceso: Septiembre 2020].

[5] MIT MediaLab, «MIT MediaLab,» Massachusetts Institute of Technology, 2015. [En línea]. Available: https://www.media.mit.edu/projects/ethics-of-autonomous-vehicles/overview/. [Último acceso: Septiembre 2020].

[6] R. M. García, «CEA: Comisariado Europeo del Automóvil,» CEA, 2020. [En línea]. Available: https://www.cea-online.es/blog/213-los-niveles-de-la-conduccion-autonoma. [Último acceso: Septiembre 2020].

[7] R. Borraz, P. J. Navarro, C. Fernández y P. M. Alcover, «Cloud Incubator Car: A reliable platform for autonomous driving,» Applied Sciences, Cartagena, 2018.

[8] Nueva Movilidad, «Nueva Movilidad,» Nueva Movilidad, 11 Agosto 2019. [En línea]. Available: https://www.nuevamovilidad.com/movilidad-compartida/historia-actualidad-y-futuro-de-los-coches-autonomos/#:~:text=Francis%20Houdina%2C%20un%20ingeniero%20eléctrico,El%20evento%20ocurrió%20en%20Manhattan.&text=El%20vehícuo%20Chandler%201926%20conducido%20a. [Último acceso: Septiembre 2020].

[9] J. M. López, «Hipertextual,» Hipertextual S.L., 15 Agosto 2020. [En línea]. Available: https://hipertextual.com/2020/08/origen-historia-vehiculo-autonomo. [Último acceso: Septiembre 2020].

[10] J. Delcker, «POLITICO,» POLITICO EU, 19 Julio 2018. [En línea]. Available: https://www.politico.eu/article/delf-driving-car-born-1986-ernst-dickmanns-mercedes/. [Último acceso: September 2020].

[11] European Conference of Transport Ministers, Resources for tomorrow's transport, Bruselas: ECMT, 1989.

[12] E. D. Dickmanns, Dynamic Vision for Perception and Control of Motion, Londres: Springer Verlag London, 2007.

[13] R. Behringer y D. M. Bailey, «The DARPA grand challenge: development of an autonomous vehicle,» IEEE, Parma, 2004.

[14] L. Kelion, «BBC News,» BBC, 21 Octubre 2014. [En línea]. Available: https://www.bbc.com/news/technology-29706473. [Último acceso: Septiembre 2020].

[15] LA NACIÓN Tecnología, «LA NACIÓN,» GDA, 15 Mayo 2015. [En línea]. Available: https://www.lanacion.com.ar/tecnologia/google-saca-a-la-calle-a-su-prototipo-de-vehiculo-sin-chofer-nid1793104/. [Último acceso: Septiembre 2020].

[16] D. Galán, «Motorpasión,» Webedia, 13 Octubre 2020. [En línea]. Available: https://www.motorpasion.com/tecnologia/asi-taxis-conductor-waymo-video-servicio-coches-autonomos-pedido-esta-marcha. [Último acceso: Octubre 2020].

[17] J. C. López, «Xataka,» Webedia, 9 Marzo 2018. [En línea]. Available: https://www.xataka.com/automovil/donde-esta-realmente-el-coche-autonomo-a-dia-de-hoy-y-que-han-prometido-las-marcas-en-el-salon-de-ginebra-para-el-futuro. [Último acceso: Octubre 2020].

[18] M. Martínez-Díaz y F. Soriguera, «Autonomous vehicles: theoretical and practical challenges,» de *Transportation Research Procedia*, Elsevier B.V., 2018.

[19] N. Suganuma y T. Uozumi, «Development of an autonomous vehicle-system overview of test ride vehicle in the Tokyo Motor Show 2011,» SICE Annual Conference, Tokio, 2012.

[20] L. Xiaoming, Q. Tian, C. Wanchun y Y. Xingliang, «Real time distance measurement using a modified camera,» de *IEEE Sensors Applications Symposium*, IEEE, 2010.

[21] M. S. Noraain, J. R. Hamid y M. H. Kamaruddin, Verification of 3D LiDAR data for use in Monte Carlo Ray Tracing Method, ISG, 2012.

[22] W. Rahiman y Z. Zaimal, «An Overview of Development GPS Navigation for Autonomous Car,» 2013.

[23] D. Hall y S. McMullen, «Mathematical Techniques in Multisensor Data Fusion,» BioMed Central, 2005.

[24] D. L. Hall y J. Llinas, AN Introduction to Multisensor Data Fusion, Nueva York: IEEE Xplore, 1997.

[25] D. Abeijón Monjas, Fusión de datos para la obtención de tiempos de viaje en carretera, Barcelona: Universidad Politécnica de Cataluña, 2007.

[26] I. Bloch y H. Maitre, Data fusion in 2D and 3D image processing: an overview, IEEE, 1996.

[27] J. Galindo, «Curso Introductorio de Conjuntos y Sistemas Difusos,» Universidad de Málaga, Málaga, 2008.

[28] H. F. Durrant-Whyte, Sensor Models and Multisensor Integration, The International Journal of Robotics Research, 1988.

[29] B. V. Dasarathy, Sensor fusion potential exploitation-innovative architectures and illustrative applications, Proceedings of IEEE, 1997.

[30] A. P. Dempster, Upper and lower probabilities induced by a multivalued mapping, The Annals of Mathematical Statistics, 1967.

[31] A. P. Dempster, «A generalization of Bayesian inference,» *Journal of the Royal Statistical Society*, 1968.

[32] R. E. Kalman, «A new approach to linear filtering and prediction problems,» *Journal of basic Engineering*, 1960.

[33] D. Poole, Computational Intelligence: A Logical Approach, Nueva York: Oxford University Press, 2018.

[34] J. M. Armingol, A. D. L. Escalera y M. Mata, «Traffic sign recognition and analysis for intelligent vehicles. Image & Vision Computing,» 2003.

[35] D. Frutos, «Análisis de técnicas, herramientas y soluciones existentes para la integración de camiones autónomos en entornos industriales,» Universidad de Castilla-La Mancha, 2017.

[36] I. Nourbaksh, R. Siegwart y D. Scaramuzza, Introduction to Autonomous Mobile Robots, Massachusetts: The MIT Press, 2011.

[37] M. Csorba, Simultaneous localisation and map building. Master's thesis, Oxford: Robotic Recherche Group Department of Engineering, 1998.

[38] M. Self, R. Smith y P. Cheesman, Autonomous Robot Vehicles, Nueva York: Springer-Verlag, 1990.

[39] A. Elfes, «Sonar-Based Real-World Mapping and Navigation,» Springer, Nueva York, 1990.

[40] O. García, R. Lemos y J. Ferreira, «Local and Global Path Generation for Autonomous Vehicles Using Splines,» Ingeniería, 2016. [En línea]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-750X2016000200006&nrm=iso.

[41] J. Latombe, Robot Motion Planning, Springer Science & Business Media, 2012.

[42] B. Liu, H. Zhang y S. Zhu, «An Incremental V-Model Process for Automotive Development,» de *Asia-Pacific Software Engineering Conference*, Hamilton, New Zealand, 2016.

[43] P. Kafka, «The Automotive Standard ISO 26262, the Innovative Driver for Enhanced Safety Assesment & Technology for Motor Cars,» Procedia Engineering, 2012.

[44] F. Rosique, P. J. Navarro, C. Fernández y A. Padilla, «A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research,» *Sensors*, 2019.

[45] MathWorks, «Get Started with Automated Driving Toolbox,» 2020. [En línea]. Available: https://es.mathworks.com/help/driving/getting-started-with-automated-driving-toolbox.html.

[46] M. Á. Álvarez, «Qué es la programación orientada a objetos,» 11 Diciembre 2019. [En línea]. Available: https://desarrolloweb.com/articulos/499.php#:~:text=La%20programaci%C3%B3n%20Orientada%20a%20objetos%20se%20define%20como%20un%20paradigma,los%20objetivos%20de%20las%20aplicaciones..

[47] MathWorks, «Scenario Simulation,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/scenario-simulation.html.

[48] MathWorks, «Ground Truth Labeling,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/ground-truth-labeling.html.

[49] MathWorks, «Detection and Tracking,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/detection-and-tracking.html.

[50] MathWorks, «Camera Sensor Configuration,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/camera-sensor-configuration.html.

[51] MathWorks, «Visual Perception,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/visual-perception.html.

[52] MathWorks, «Lidar Processing,» MathWorks & Velodyne, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/lidar-processing.html.

[53] MathWorks, «Tracking and Sensor Fusion,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/tracking-and-sensor-fusion.html.

[54] MathWorks, «Localization and Mapping,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/localization-and-mapping.html.

[55] MathWorks, «Planning and Control,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/planning-and-control.html.

[56] S. Liu, J. Tang, C. Wang, Q. Wang y J. Gaudiot, «Implementing a Cloud Platform for Autonomous Driving,» arXiv, 2017.

[57] M. Samek, «Practical UML Statecharts in C/C++. Event-Driven Programming fro Embedded Systems,» Newnes/Elsevier, Massachusets, 2009.

[58] P. J. Navarro, R. Borraz, C. Fernández y A. Alonso, «Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data,» Sensors, Cartagena, 2016.

[59] C. Piñana-Díaz, R. Toledo Moreo, F. Toledo Moreo y A. Skarmeta, «A Two-Layers based approach of an Enhanced-Map for Urban Positioning Support,» Sensors, Cartagena, 2012.

[60] C. Piñana Díaz, R. Toledo Moreo, F. Toledo Moreo y A. Skarmeta, «GPS multipath detection and exclusion with elevation-enhanced maps,» IEEE, 2011.

[61] C. Harris y M. Stephens, «A combined corner and edge detector,» de *Fourth Alvey Vision Conference*, Manchester, 1988.

[62] V. L. Inc, «Velodyne LiDAR HDL-64E Specs». California Patente 408.465.2800, 2018.

[63] MathWorks, «Extended Object Tracking of Highway Vehicles with Radar and Camera,» MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/help/driving/ug/extended-object-tracking.html.

[64] K. Granstrom, M. Baum y S. Reuter, «Extended Object Tracking: Introduction, Overview and Applications,» Cornell University, 2016.

[65] R. C. González y R. E. Woods, Thresholding in Digital Image Processing, Pearson Education, 2002.

[66] MathWorks, «Sensor Fusion Using Synthetic Radar and Vision Data,» MathWorks, 2021. [En línea]. Available: https://www.mathworks.com/help/driving/ug/sensor-fusion-using-synthetic-radar-and-vision-data.html;jsessionid=b33ce039884d7d92ffe297292948.

[67] I. Martín y Ladera, «Tesla patenta un sistema de visión que puede hacer innecesario el radar en el futuro,» Foro Coches Eléctricos, 30 Marzo 2021. [En línea]. Available: https://forococheselectricos.com/2021/03/tesla-patenta-un-sistema-de-vision-que-puede-hacer-innecesarios-los-radar-en-el-futuro.html.

[68] EDS Robotics, «Cámaras industriales, tipos y funcionamiento,» EDS Robotics, 2020. [En línea]. Available: https://www.edsrobotics.com/blog/camaras-industriales/.

[69] E. Dijkstra, «A Note on Two Problems in Connection with Graphs,» *Numerische Mathematik*, vol. 271, pp. 269-271, 1959.

[70] P. Hart, N. Nilsson y B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE, 1968.

[71] R. Pallás Areny, «Sensores y acondicionadores de señal.,» Universidad Politécnica de Cataluña, 2003.