



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

**MÁSTER UNIVERSITARIO
EN INGENIERÍA INFORMÁTICA**

TRABAJO FIN DE MÁSTER

**Análisis de técnicas, herramientas y soluciones
existentes para la integración de camiones
autónomos en entornos industriales**

David Frutos Talavera

Diciembre, 2017

**ANÁLISIS DE TÉCNICAS, HERRAMIENTAS Y SOLUCIONES EXISTENTES
PARA LA INTEGRACIÓN DE CAMIONES AUTÓNOMOS EN ENTORNOS
INDUSTRIALES**



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA
Tecnologías y Sistemas de Información

TRABAJO FIN DE MÁSTER

**Análisis de técnicas, herramientas y soluciones
existentes para la integración de camiones
autónomos en entornos industriales**

Autor: David Frutos Talavera

Tutor: Dr. David Vallejo Fernández

Cotutor: Dr. Javier A. Albusac Jiménez

Diciembre, 2017



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA
Tecnologías y Sistemas de Información

TRABAJO FIN DE MÁSTER

**Análisis de técnicas, herramientas y soluciones
existentes para la integración de camiones
autónomos en entornos industriales**

Autor
Fdo.:

Tutor
Fdo.:

Cotutor
Fdo.:

Diciembre, 2017

David Frutos Talavera

Ciudad Real – España

E-mail: David.Frutos.Talavera@gmail.com

Teléfono: 686 133 022

© 2017 David Frutos Talavera

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se permite la copia, distribución y/o modificación de este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.3 o cualquier versión posterior publicada por la *Free Software Foundation*; sin secciones invariantes. Una copia de esta licencia esta incluida en el apéndice titulado «GNU Free Documentation License».

Muchos de los nombres usados por las compañías para diferenciar sus productos y servicios son reclamados como marcas registradas. Allí donde estos nombres aparezcan en este documento, y cuando el autor haya sido informado de esas marcas registradas, los nombres estarán escritos en mayúsculas o como nombres propios.

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

En este trabajo de fin de máster se plantea la realización de un estudio detallado de los dispositivos hardware de sensorización, prototipos de camiones autónomos existentes en la actualidad, y técnicas y algoritmos relativos a la percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias, toma de decisiones y control del movimiento. Este estudio surge tras el creciente interés por el desarrollo de vehículos autónomos y con la intención de mejorar la conducción y reducir el número de accidentes causados por errores humanos. Las soluciones basadas en sistemas de asistencia a la conducción o incluso las relativas a la conducción totalmente autónomo han cobrado una relevancia significativa. Empresas como Tesla producen coches capaces de conducirse por sí solos tanto en entornos urbanos e interurbanos garantizando una menor tasa de accidentes, conduciendo de manera más eficiente y reduciendo la cantidad de residuos generados.

En este contexto, recientemente también se están implantando soluciones de conducción autónoma en entornos industriales. Uno de los problemas más recurrentes se conoce como *Load, Haul and Dump*, el cual consiste en automatizar los procesos de carga, transporte y descarga de materiales por parte de un camión autónomo. Si este problema se focaliza en escenarios físicos donde la empresa tiene cierto control sobre el entorno, entonces es posible construir soluciones que, aunque son tremendamente complejas, pueden ser más acotadas.

Después de llevar a cabo un estudio detallado, en este proyecto se propone la solución conceptual de camión autónomo para la resolución de problemas *Load, Haul and Dump* a partir del análisis del mercado actual realizado. Esta solución seguirá un planteamiento escalable que considere la existencia de más entidades (otros camiones autónomos, operarios, vehículos no autónomos, etc) en el entorno. Dicha solución conceptual seguirá un esquema de trabajo definido en este documento que establezca los pasos recomendados desde que se comienza a diseñar una solución hasta que el camión autónomo se implanta realmente. Para ello este documento busca identificar los dispositivos y técnicas aplicadas a la conducción autónoma en general que pueden extrapolarse para la resolución de problemas *Load, Haul and Dump*.

Por estos motivos se espera que este documento sirva como potencial punto de partida para aquellas empresas interesadas en integrar soluciones autónomas escalables en lo relativo al problema *Load, Haul and Dump* en entornos industriales.

Abstract

This work of end of master is concerned with a detailed study of sensor system hardware of prototypes of autonomous lorries currently existing and techniques and algorithms relative to perception and environment modelling, location, maps construction, planning of trajectories, decision making and movement control. This study arises after the increased interest of the development of autonomous vehicles and with the purpose of improving driving and reducing the number of accidents caused by human errors. The solutions based in advanced driver assisted systems or even the ones related to completely autonomous driving, have developed significant relevance. Companies like Tesla produce cars able to self-drive in urban and inner-city environments guaranteeing a minor rate of accidents, driving in a more efficient manner and reducing the amount of pollution generated.

In this context, solutions of autonomous driving in industrial environments are being introduced recently as well. One of the more recurrent problems is known as Load, Haul and Dump, consisting in making automatic the process of load, haul and dump of materials by an autonomous lorry. If this problem is focused in physical scenarios where the company has certain control over the environment, then, it is possible to create solutions that, although tremendously complex, can be more refined.

After accomplishing a detailed study, this project sets out the conceptual solution of an autonomous lorry for the resolution of problems relating to Load, Haul and Dump, based on the analysis of the current market. This solution will follow a scalable proposal that considers the existence of more entities (other autonomous lorries, operators, non-autonomous vehicles, etc) in the environment. This conceptual solution will follow a scheme of work defined in this document that establish the recommended steps starting from the beginning of the designed solution and following through until the autonomous lorry is actually introduced. In order to achieve this, this document seeks to identify the devices and techniques applied to autonomous driving in general, that can be used to solve Load, Haul and Dump problems.

For these reasons it is expected that this document will serve as a potential starting point for those companies interested in integrating scalable autonomous solutions related to the problems found in the Load, Haul and Dump system in industrial environments.

Agradecimientos

Este documento está dedicado a mi hermano, que espero encuentre este documento entretenido y didáctico y a todas las personas que me han ayudado a lo largo de los años y en especial durante la elaboración de este proyecto.

Quiero darle las gracias, en especial, a toda mi familia y en especial a mis padres y a mi hermano por el apoyo dado, además de agradecerle a mi prima Yolanda y a su marido Robin toda la ayuda dada.

Y dado que si hubiese realizado este proyecto sin parar a relajarme en verano me hubiese vuelto loco, quiero agradecerle a Merche que me sacara de viaje y como no, su compañía, tiempo, dedicación y cariño. Espero que en un futuro no muy lejano los vehículos autónomos estén a la orden del día y no haga falta conducir hasta San Juan.

Gracias también a todos mis amigos por los ánimos ofrecidos y las ayudas dadas.

Por último, pero no menos importante, darle las gracias a mi tutor David Vallejo. Muchas gracias por las horas echadas y la ayuda ofrecida.

David Frutos Talavera

*«- Hey, Doc. No tenemos suficiente carretera para ir a 140 km.
- ¿Carretera? Adonde vamos, no necesitaremos carreteras.»
Back to the Future*

Índice general

Resumen	V
Abstract	VII
Agradecimientos	IX
Índice general	XIII
Índice de cuadros	XVII
Índice de figuras	XIX
Listado de acrónimos	XXIII
1 Introducción	1
1.1 Problemática y alcance	3
1.2 Contexto socio-económico	5
1.3 Competencias adquiridas	8
1.4 Estructura del documento	9
2 Objetivos	11
2.1 Objetivo general	11
2.2 Objetivos específicos	12
3 Método de Trabajo	13
3.1 Metodología	13
4 Revisión científica del estado del arte	17
4.1 Percepción y modelado del entorno	17
4.1.1 Introducción	17
4.1.2 Fusión de informacion	22
4.1.3 Modelado dinámico del entorno	37

4.2	Localización y construcción de mapas	46
4.2.1	Localización	46
4.2.2	Construcción de mapas	49
4.2.3	SLAM y el teorema de Bayes	50
4.3	Planificación de trayectorias	55
4.3.1	Introducción	55
4.3.2	Algoritmos	57
4.3.3	Planificación de trayectoria locales y globales mediante el uso de Splines	76
4.4	Toma de decisiones	84
4.4.1	Introducción	84
4.4.2	Toma de decisiones de vehículos autónomos en entornos urbanos .	86
4.4.3	Estructura de la toma de decisiones	87
4.5	Modelo cinemático y control del movimiento	95
4.5.1	Introducción	95
4.5.2	Modelo cinemático	96
4.5.3	Control de movimiento	102
5	Soluciones comerciales	113
5.1	Introducción	114
5.2	Camiones autónomos	117
5.3	Hardware de sensorización	127
5.4	Entornos de desarrollo	134
6	Propuesta de camión autónomo	141
6.1	Introducción	141
6.1.1	Esquema propuesto	148
6.2	Definición del entorno	149
6.2.1	Entorno propuesto	151
6.3	Requisitos funcionales	152
6.3.1	Requisitos funcionales propuesto	153
6.4	Integración de sensores	155
6.4.1	Integración de sensores propuestos	156
6.5	Análisis de costes	157
6.5.1	LiDAR	157
6.5.2	Radar	158

6.5.3	Cámaras infrarrojas	158
6.5.4	GPS e INS	158
6.5.5	Controlador	159
6.5.6	Solución completa	159
7	Conclusiones	161
7.1	Conclusiones	161
7.1.1	Solución conceptual de un camión autónomo para la resolución de problemas LHD	161
7.1.2	Objetivos parciales	162
7.1.3	Trabajo futuro	163
7.1.4	Reflexiones personales	163
	Referencias	165

Índice de cuadros

4.1	Ejemplo de la obtención de la longitud del arco s	79
4.2	Interpolación del <i>spline</i> con el conjunto de datos x e y	80
6.1	Esquema características del entorno según nivel propuestas para el vehículo	151
6.2	Esquema tareas DDT de nivel 1 propuestas para el vehículo	153
6.3	Esquema tareas DDT de nivel 2 propuestas para el vehículo	153
6.4	Esquema tareas DDT, DDT Fallback y OEDR de nivel 3 propuestas para el vehículo	154
6.5	Esquema tareas OEDR de nivel 4 propuestas para el vehículo	154
6.6	Esquema de sensores según nivel propuestos para el vehículo	157
6.7	Tabla de sensores con sus unidades requeridas y precio por unidad	159
6.8	Costes de los sensores de un vehículo de nivel 1 y 2	160
6.9	Costes de los sensores de un vehículo de nivel 3	160
6.10	Costes de los sensores de un vehículo de nivel 4	160

Índice de figuras

1.1	Grafica con el número de patentes relacionadas con <i>vehículos autónomos</i> agrupado por empresas	2
1.2	Reducción de gastos en el mercado de transportes gracias al uso de <i>vehículos autónomos</i>	3
1.3	Dilema de doble moral. ¿A quién atropellar?	5
1.4	¿Cómo se sienten los consumidores respecto a los <i>vehículos autónomos</i> ?	6
1.5	Dinero invertido en tecnología automotriz	7
1.6	Grafica del interés mediático sobre los <i>vehículos autónomos</i>	7
4.1	Fusión de datos: Terminología	22
4.2	Proceso de fusión de datos	24
4.3	Sistema multi-sensor	25
4.4	Clasificación según la relación de las fuentes	28
4.5	Clasificación según Dasarathy. Las cinco categorías	29
4.6	Clasificación basada en el tipo de arquitectura	31
4.7	Agrupación de datos asignada a un objetivo	32
4.8	Algoritmo recursivo del Filtro de Kalman	35
4.9	Proceso de fusión con la técnica de Bayes	36
4.10	Proceso de fusión con la técnica de Dempster-Shafer	36
4.11	Esquema general de un framework de percepción y modelado del entorno	37
4.12	Sistema con múltiples sensores y su uso	38
4.13	Atributos de los sensores LiDAR	41
4.14	Imágenes generadas a partir de la intensidad del pulso laser	42
4.15	Explicación de las múltiples recepciones del pulso laser y su distinta intensidad	43
4.16	Vista esquemática de un sistema de cámara estéreo 3D	44
4.17	Atributos de las cámaras de vídeo a color	45
4.18	Atributos de los radares	45
4.19	Atributos de los sensores de ultrasonido	46
4.20	Histograma de la frecuencia de error de localización en el tiempo	47

4.21	Resultados experimentales del OGM	52
4.22	Esquema cinemático de un vehiculo	56
4.23	Arquitectura de un vehículo autónomo	57
4.24	Ejemplo de grafo de visibilidad	60
4.25	Definición de segmentos tangentes y vértices cóncavos	60
4.26	Ejemplo de diagrama del grafo de Voronoi	61
4.27	Ejemplo de una descomposición vertical	63
4.28	Condición de la triangulación de Delaunay	63
4.29	Ejemplo de una descomposición de Delaunay	64
4.30	Ejemplo de una descomposición de Quadtree	64
4.31	Arbol quadtree	65
4.32	Implementación circular de <i>Fast Marching Method</i>	66
4.33	Propagación del método <i>Fast Marching Method</i> con transformación Voronoi	67
4.34	Ejemplo de un <i>campo de potencial</i>	68
4.35	Unión de los campos de atracción y repulsión	68
4.36	Ejemplo de grafo generado mediante PRM	71
4.37	Función Extiende del algoritmo RRT	73
4.38	Ejemplo de arbol RRT con Voronoi	74
4.39	Ejemplo de arbol RRT*	75
4.40	Distintas etapas del postprocesado de un árbol RRT	77
4.41	Ruta global junto a una serie de rutas candidatas generadas por el algoritmo	78
4.42	Distancia acumulativa d junto a la distancia del arco s	79
4.43	Localización del vehículo en relación a la ruta global	81
4.44	Sistema de coordenadas curvilíneas $s0q$ y el conjunto de trayectos candidatos representados en las coordenadas cartesianas	82
4.45	Dilema del tranvía	85
4.46	Visión general de los temas de investigación de los vehículos autónomos y el alcance de la toma de decisiones en tiempo real	85
4.47	Vista simplificada de la arquitectura de software de control y toma de decisiones del vehículo autónomo y su flujo de datos	88
4.48	Entradas y salidas del modelado del entorno	89
4.49	Ejemplo de autómata finito de una maniobra de conducción con múltiples estados de marcha	91
4.50	La unidad de toma de decisiones para la selección de maniobras de conducción factibles	93
4.51	Red de Petri para la selección de maniobras de conducción factibles	94
4.52	Rueda moviéndose en una sola dirección. Restricción holonómica	96

4.53	Rueda moviéndose en plano 2D. Restricción no-holonómica	96
4.54	Relación entre la cinemática directa e indirecta	97
4.55	Esquema de un robot articulado	97
4.56	Problema del codo arriba, codo abajo en un robot de brazos articulado	98
4.57	Modelo cinemática de un vehículo articulado compuesto de vehículo tractor y remolque	99
4.58	Vehículo articulado en estado de circulación con centro de circulación P . .	100
4.59	Vehículo articulado en estado de circulación donde la trayectoria RA inter- seca antes con la RC que con RB	101
4.60	Etapas del efecto Jackknife	101
4.61	Maniobra de acercamiento marcha atrás	104
4.62	Ejemplo de trayectoria que se interseca a sí misma	105
4.63	Esquema de los errores longitudinales y laterales	105
4.64	Esquema del cálculo del error de curvatura	107
4.65	Gráfico del controlador P con error de estado estable	109
4.66	Gráfico del controlador P con error de estado estable casi eliminado median- te la modificación del ángulo de referencia	109
4.67	Gráfico del controlador P sin error mediante la modificación del ángulo de referencia y el uso de integrales <i>anti-windup</i>	110
4.68	Gráfico del controlador de Lyapunov	111
5.1	Mapa del mercado actual de vehículos autónomos	113
5.2	Financiación del mercado automovilístico en 2017	114
5.3	Foto del European Truck Platooning Challenge 2016	118
5.4	Mejora de conducción tras instalar el sistema V2V	119
5.5	Sensores del sistema <i>Peloton</i>	119
5.6	Componentes del sistema <i>Peloton</i>	120
5.7	Volquete sin cabina de Komatsu, denominado Autonomous Haulage Vehicle (AHV)	121
5.8	Volquete de Hitachi con AHS	122
5.9	Sensores del modelo inicial de Otto	123
5.10	Nuevo conjunto de sensores de los camiones de <i>Uber</i>	123
5.11	Camión de la basura de Volvo	124
5.12	Volvo FMX totalmente autónomo realizando un recorrido de prueba en las profundidades de la mina Kristineberg	125
5.13	Volvo FMX. Solapamiento de los sensores LiDAR	126
5.14	Kit de automatización realizado por la empresa <i>ASI Robotics</i>	126

5.15	Sensores LiDAR de la empresa <i>Sick AG</i> montados sobre el vehículo Stanley	128
5.16	Distintos sensores LiDAR de la empresa <i>Velodyne</i>	129
5.17	Sensor <i>Velarray</i> de la empresa <i>Velodyne</i>	129
5.18	<i>Mobileye Serie 5</i> . Sistema de prevención de accidentes	130
5.19	Comparativa de distintos sensores en distintos entornos	132
5.20	Visión conjunta del autopilot del vehículo <i>Tesla</i>	133
5.21	Superposición de los atributos del radar, ultrasonidos y cámara de video	134
5.22	Interfaz gráfica de ROS	135
5.23	Interfaz gráfica de MOBIUS!	137
5.24	Detección de obstáculos del SDK <i>NVIDIA DriveWorks</i>	138
5.25	Localización y planificación de trayectorias del SDK <i>NVIDIA DriveWorks</i>	138
5.26	Interfaz gráfica de Ros MoveIt	139
6.1	Vista esquemática de la tarea de conducción que muestra la seccion de DDT	143
6.2	Diagrama que muestra los ejes de movimiento del vehículo	145
6.3	Diagrama de flujo lógico simplificado para asignar el nivel de automatización de conducción a una función	147
6.4	Ilustración de la importancia de ODD en relación con los niveles	150
6.5	Colocación propuesta de ocho LiDAR de estado sólido de 120 grados de apertura.	155

Listado de acrónimos

TFM	Trabajo fin de Master
SAE	Society of Automotive Engineers
ADAS	Advanced Driver Assistance Systems
SW	Software
HW	Hardware
IA	Inteligencia Artificial
RAE	Real Academia Española
LDR	Light-Dependent Resistors
CCD	Charge-Coupled Device
RTD	Resistance Temperature Detector
GPS	Global Positioning System
LiDAR	Light Detection and Ranging
ISIF	International Society of Information Fusion
JDL	Joint Directors of Laboratories
DFIG	Data Fusion Information Group
ICCC	Independiente del Contexto con Comportamiento Constante
ICCV	Independiente del Contexto con Comportamiento Variable
DC	Dependiente del Contexto
DAI-DAO	Data In-Data Out
DAI-FEO	Data In-Feature Out
FEI-FEO	Feature In-Feature Out
FEI-DEO	Feature In-Decision Out
DEI-DEO	Decision In-Decision Out
HCI	Human-Computer Interaction
NN	Nearest Neighbor
PDA	Probabilistic Data Association

JPDA	Joint Probabilistic Data Association
MHT	Multiple Hypothesis Test
JPDA-D	Distributed Joint Probabilistic Data Association
ML	Maximum Likelihood
MP	Maximum Posterior
INS	Inertial Navigation System
RGB	Red, Green and Blue
IMU	Inertial Measurement Unit
SLAM	Simultaneous Localization And Mapping
EKF	Filtro de Kalman Extendido
OGM	Occupancy Grid Mapping
JPDAF	Joint Probabilistic Data Association Filter
PHD	Probability Hypothesis Density Filter
DATMO	Detection and Tracking of Moving Objects
ROS	Robot Operating System
PCL	Point Cloud Library
RRT	Rapidly Exploring Random Trees
PRM	Probabilistic Roadmap
PDRRTs	Parti-game Directed RRTs
CL-RRT	Closed-loop Rapidly-exploring Random
RRG	Rapidly-exploring Random Graph
LHD	Load Haul Dump
PI	Proportional Integra
FSM	Finite State Machine
POMDP	Partially Observable Markov Decision Process
GPU	Graphics Processing Unit
QMDP	Point-based Markov Decision Process
MDP	Markov Decision Process
API	Application Programming Interface
RNDF	Route Network Definition File
DEDS	Discrete Event Dynamic System
MCDM	Multiple-Criteria Decision-Making

ADG	Autonomous Driving Group
REM	Road Experience Management
TRI	Toyota Research Institute
V2V	Vehicle-to-Vehicle
CACC	Cooperative Adaptive Cruise Control
ACC	Adaptive Cruise Control
NOC	Network Operations Center
AHV	Autonomous Haulage Vehicle
AHS	Autonomous Haulage System
SO	Sistema Operativo
BSD	Berkeley Software Distribution
SDK	Software Development Kit
GIE	GPU Inference Engine
ADS	Automated Driving System
ADS-DV	ADS-Dedicated Vehicle
DDT	Dynamic Driving Task
ODD	Operational Design Domain
OEDR	Object and Event Detection and Response
MRR	Mid-Range Radar
ESR	Electronically Scanning RADAR
AHRS	Attitude Heading Reference System
GNSS	Global Navigation Satellite System
ISO	International Organization for Standardization

Capítulo 1

Introducción

EN los últimos años se ha visto un gran avance de la tecnología relacionada con los **vehículos autónomos**. ¿Pero qué significa vehículo autónomo? La pagina *Wikipedia* lo define informalmente así: «*Un vehículo autónomo, también conocido como robótico, o informalmente como sin conductor o autoconducido, es un vehículo capaz de imitar las capacidades humanas de manejo y control. Como vehículo autónomo, es capaz de percibir el medio que le rodea y navegar en consecuencia. El conductor podrá elegir el destino, pero no se le requiere para activar ninguna operación mecánica del vehículo* »¹.

Si preguntamos a un neófito en la materia nos hablara de los vehículos *Tesla* y su piloto automático, esto es, en parte, gracias a la publicidad que ha dado la empresa **Tesla** a tecnología, aunque no por ello lideran el mercado de los *vehículos autónomos*. Actualmente existen multitud de empresas (como mínimo **44 grandes empresas** según *Cbinsights*²) que apuestan por esta nueva tecnología. Cada una abarca un ámbito de los *vehículos autónomos*, unas fabrican vehículos que incluyen tecnología **Advanced Driver Assistance Systems (ADAS)** de serie, otros desarrollan sensores que se utilizan en los vehículos, algunas investigan nuevos algoritmos, etc. Por ello este mercado mueve cada año más dinero y como cabe esperar existe una carrera para ver quién es la primera empresa en crear el vehículo totalmente autónomo con un coste aceptable y una seguridad irreprochable (véase Figura 1.1).

Actualmente la *autonomía* de un vehículo está muy reducida, sin entrar en detalles del nivel **SAE**, de los cuales se hablara más adelante, los vehículos actuales son capaces, como mucho, de realizar trayectos bajo supervisión. Por esto mismo se define como tecnología emergente dado que aún queda mucho que investigar y desarrollar.

Pero este documento no se centra en los vehículos de turismo o utilitarios, este documento se centra en el mercado de los vehículos de medio y gran tonelaje utilizados para transporte de mercancías, transporte industrial, etc, las denominadas tareas **Load Haul Dump (LHD)**. Recientemente se están implantando soluciones de conducción autónoma en entornos industriales centradas en solucionar el problema LHD. Estas tareas conllevan la mayor parte del tiempo de trabajo en entornos industriales y , en cierta parte debido a las grandes cargas que

¹https://es.wikipedia.org/wiki/Vehículo_automóvil

²<https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/>

Who Leads the Autonomous Driving Patent Race?

Number of worldwide patent filings related to autonomous driving (January 2010–July 2017)



Based on a total of 5,839 patent filings related to autonomous driving identified and analysed by the Cologne Institute for Economic Research
Sources: Cologne Institute for Economic Research, WIPO



statista

Figura 1.1: Grafica con el número de patentes relacionadas con *vehículos autónomos* agrupado por empresas.

se transportan, generan el mayor número de accidentes con vehículos en dichos entornos, por eso los esfuerzos se están centrando en automatizar los procesos de carga, transporte y descarga de materiales por parte de un camión autónomo. Actualmente las empresas del sector tienden a instalar soluciones autónomas que automatizan paso a paso los niveles SAE.

Las tecnologías de conducción autónoma están determinando el futuro de la industria del transporte por carretera. Un estudio de PricewaterhouseCoopers³ sugiere que para 2025, los propietarios de flotas podrían reducir sus costos totales en un 15 % en comparación con 2016 (véase Figura 1.2).

Con este documento se pretende realizar un análisis de las técnicas, herramientas y soluciones existentes para la integración de *vehículos autónomos* articulados en entornos industriales. A lo largo de este documento se plantearán una serie de técnicas utilizadas en los distintos procesos del vehículo autónomo, una serie de herramientas útiles para su desarrollo y un conjunto de soluciones existentes ya sean vehículos, kits de mejora, software de desarrollo o sensores existentes. Al final de este documento se presentará un vehículo, teórico, que englobe todos los puntos investigados en este documento.

³<https://www.pwc.es/>

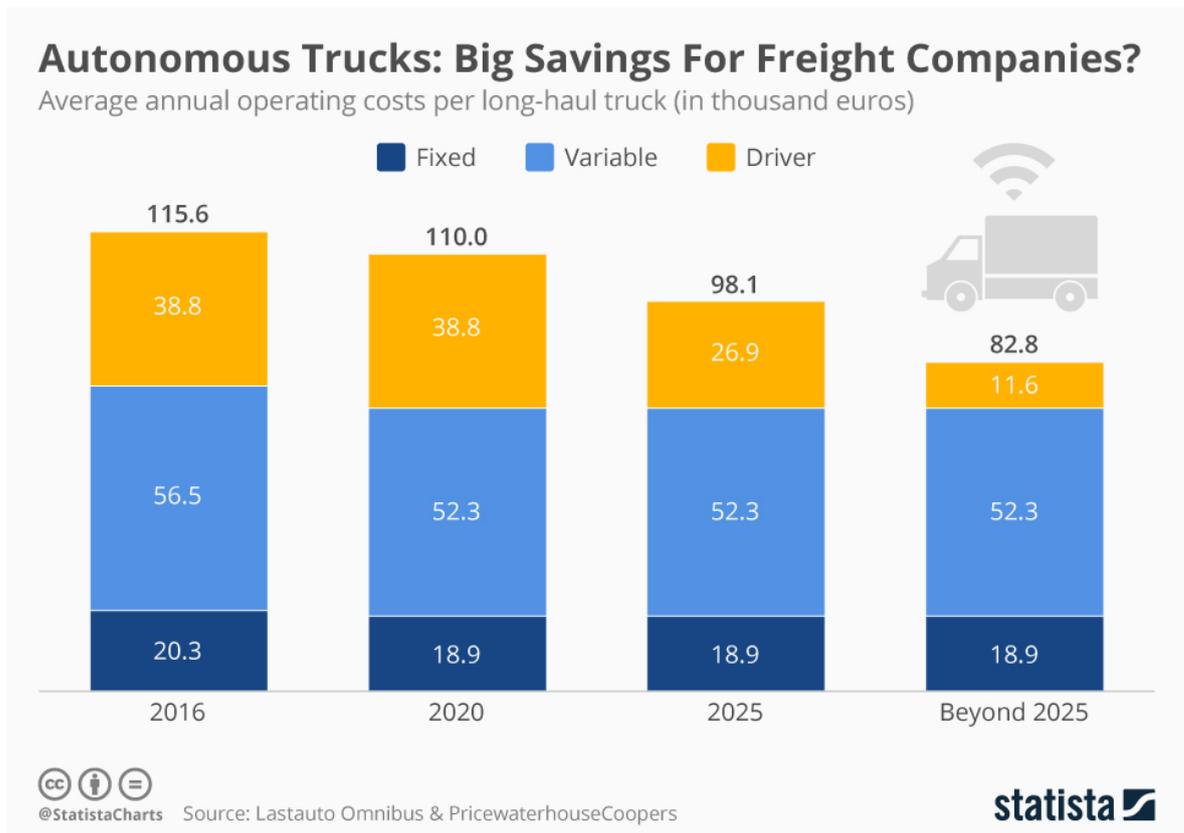


Figura 1.2: Reducción de gastos en el mercado de transportes gracias al uso de *vehículos autónomos*. En azul oscuro el coste fijo, en azul claro el variable y en amarillo el coste de los conductores.

1.1 Problemática y alcance

El 87.2 % de los accidentes de tráfico que involucran a vehículos de transporte son causados por un **error humano**. Un 65 % de este porcentaje ocurre en transportes de larga distancia. Un 41 % de los conductores alega, entre otras cosas, una fatiga leve, falta de atención o la presión del trabajo y un 13 % culpa a la fatiga. Un 7 % de los accidentes ocurren tras quedarse dormido el conductor⁴.

Como puede observarse las largas horas de conducción merman las capacidades de los conductores derivando en accidentes lo que supone, en el mejor de los casos, la pérdida del vehículo y de la carga y en el peor de los casos pérdidas humanas. Según el estudio *The sleep of long-haul truck drivers* [MML⁺97] la media de horas de sueño de un conductor de camiones es de 5 horas en vez de las 7-9 horas que requiere un adulto. Cabe destacar que la falta de sueño prolongado tiene los mismos efectos que un 0.10 % de alcohol en sangre.

Esto es debido a que los horarios laborales de los conductores son excesivos, se les permite realizar 11 horas consecutivas de conducción y 14 horas de trabajo. Tienen que realizar un

⁴<https://visual.ly/community/infographic/transportation/truck-driver-fatigue-statistics>

mínimo de 10 horas consecutivas entre conducción y trabajo y un máximo de 60 horas en los últimos 7 días o 70 horas en los últimos 8 días.

Todas estas horas de trabajo y conducción conllevan lo que se corrobora con datos. Los conductores tienen más accidentes por causa de las horas trabajadas. La causa de los accidentes en la industria minera no se distancia mucho de la industria del transporte. Según un estudio de la Departamento de Energía de EE.UU. el 93 % de los accidentes donde se ve involucrado un vehículo están ocasionados por un error humano [RB96]. Este porcentaje esta causado por las demandas inusuales que los vehículos ponen a sus operadores humanos. Entre las causas podemos encontrar el estado de las carreteras, las cargas de los vehículos, traslados largos dentro de la propia mina, falta de visibilidad por las características del vehículo, etc. Incluso el estudio hace hincapié en que la simple conducción del vehículo supone un foco de enfermedades (musculares y articulares) para el operario.

Por estos factores se quiere investigar y analizar la situación actual para integrar *vehículos autónomos* articulados en entornos industriales. Se prevé que el uso de los *vehículos autónomos* para transporte de mercancías en largas distancias o en entornos industriales pueda minimizar el número de accidentes ocasionados por el factor humano. Se propone crear un sistema evolutivo que siga los niveles definidos en SAE. SAE define seis niveles de autonomía para los vehículos que varían según el grado de atención humana y según la capacidad de toma de decisiones del vehículo. Por ello, a grandes rasgos, se definen los siguientes niveles:

- **Solo conductor:** el vehículo carece de automatismos. Atención total del conductor.
- **Asistente de conducción:** el vehículo incluye alertas de carril.
- **Semiautónoma:** el vehículo controla velocidad y dirección pero no controla los obstáculos. El conductor debe permanecer alerta.
- **Autonomía controlada:** el vehículo es capaz de seguir una trayectoria en un escenario concreto.
- **Alto nivel de autonomía:** puede circular sin supervisión siempre que tenga toda la información del entorno.
- **Autonomía total:** el conductor no es necesario.

Partiendo de un vehículo de nivel SAE 0 se pretende añadir componentes que vayan dotando de autonomía al vehículo hasta alcanzar el nivel máximo posible según el mercado actual. Cabe destacar que es posible que dada la tecnología actual no sea necesario empezar desde un nivel SAE 0.

1.2 Contexto socio-económico

En la actualidad existe **gran controversia** sobre los *vehículos autónomos*. Incluso se plantean preguntas filosóficas como ¿Debería atropellar un vehículo autónomo a unos peatones para salvar a sus pasajeros? (véase Figura 1.3). Esta pregunta, según los entendidos, no tiene respuesta alguna, pero aun así se sigue planteando. Según estudios, los *vehículos autónomos* podrían evitar un 90 % de los accidentes causados por errores humanos por lo que su uso estaría reduciendo el número de accidentes drásticamente frente a los posibles accidentes de doble moral que podrían causar. Se definen como accidentes de doble moral ya que los participantes de dichos estudios indicaron que: «*los coches autónomos deberían estar programados para proteger a los peatones (usando algoritmos utilitarios)* », pero cuando se les pregunto que preferirían comprar indicaron que: «*aquellos cuya programación les diera prioridad como pasajeros en cuanto a seguridad, sobre todo viajando con la familia.* »⁵. Por estas razones se define como accidente de doble moral. Cabe destacar que, por lo general, la mayoría de los entrevistados tuvo en cuenta el número de peatones que podrían ser salvados. Estos escenarios y más pueden ser encontrados en la web interactiva Moral Machine⁶ en la cual se pueden ver los posibles escenarios y las estadísticas de los resultados.

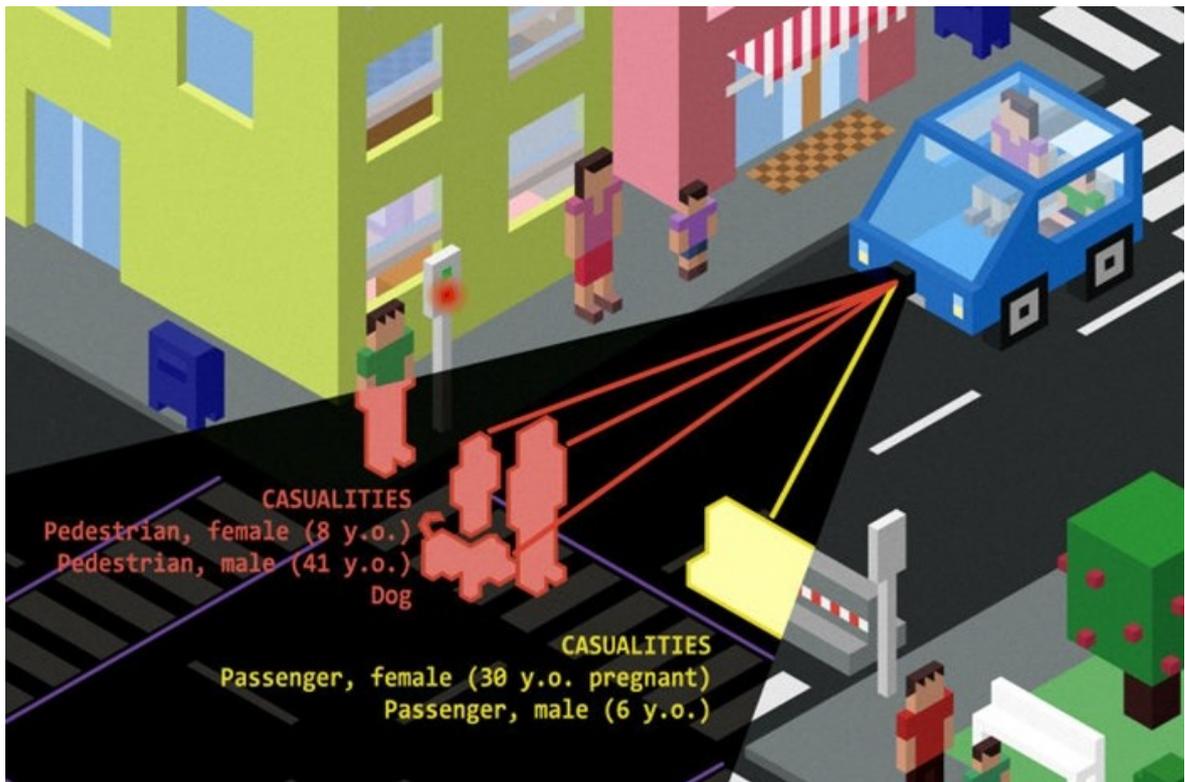


Figura 1.3: Dilema de doble moral. ¿A quién atropellar?

⁵<http://www.abc.es/motor/reportajes/abci-dilema-coche-autonomo-salvar/>

⁶<http://moralmachine.mit.edu/hl/es>

Aun así, otro estudio realizado demuestra que por lo general **la gente de a pie tiene buen concepto** de los *vehículos autónomos* y, que de ser asequibles, los utilizarían (véase Figura 1.4).

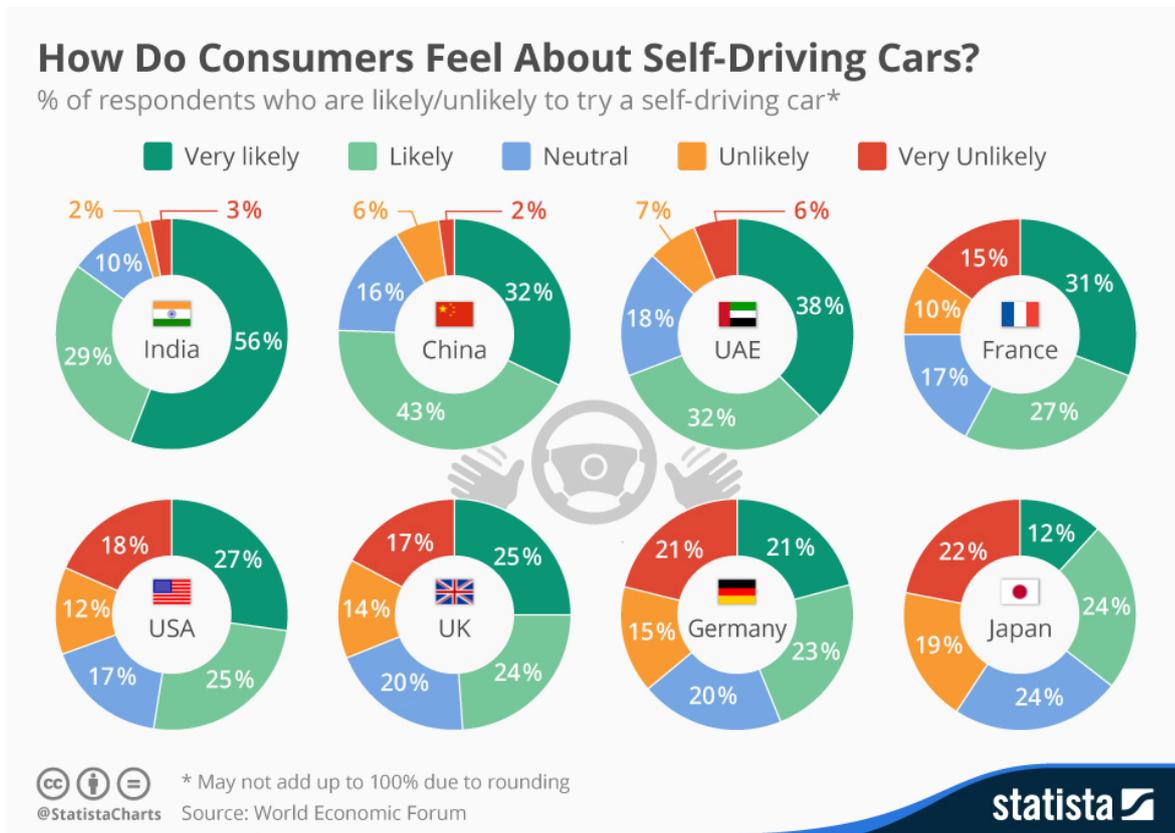


Figura 1.4: ¿Cómo se sienten los consumidores respecto a los *vehículos autónomos*?

Es por ello que el mercado de los *vehículos autónomos* ha recuperado en 2017 un 76 % de los casi \$4.000 millones invertidos en tecnología automotriz. Esta inversión es el triple que el año pasado. El número de acuerdos ha ido en aumento en los últimos años y solo en el ámbito de los *vehículos autónomos* se han realizado 77 grandes acuerdos que han movido unos \$3.000 millones. Es por ello que la tecnología automotriz sigue atrayendo a un gran grupo de patrocinadores, ya que la cantidad esta aumentado cada año desde 2012, con una cifra de 96 inversores y patrocinadores en 2017, casi duplicando las cifras del año anterior (véase Figura 1.5)⁷.

También el interés mediático ha crecido en los últimos años aunque se ha estancado este año lo que demuestra el interés de la sociedad por las nuevas tecnologías relacionadas con los *vehículos autónomos* (véase Figura 1.6).

⁷<https://www.cbinsights.com/research/auto-tech-startup-investment-trends/>

Auto tech annual global financing history
2013 - 2017 YTD (11/14/2017)

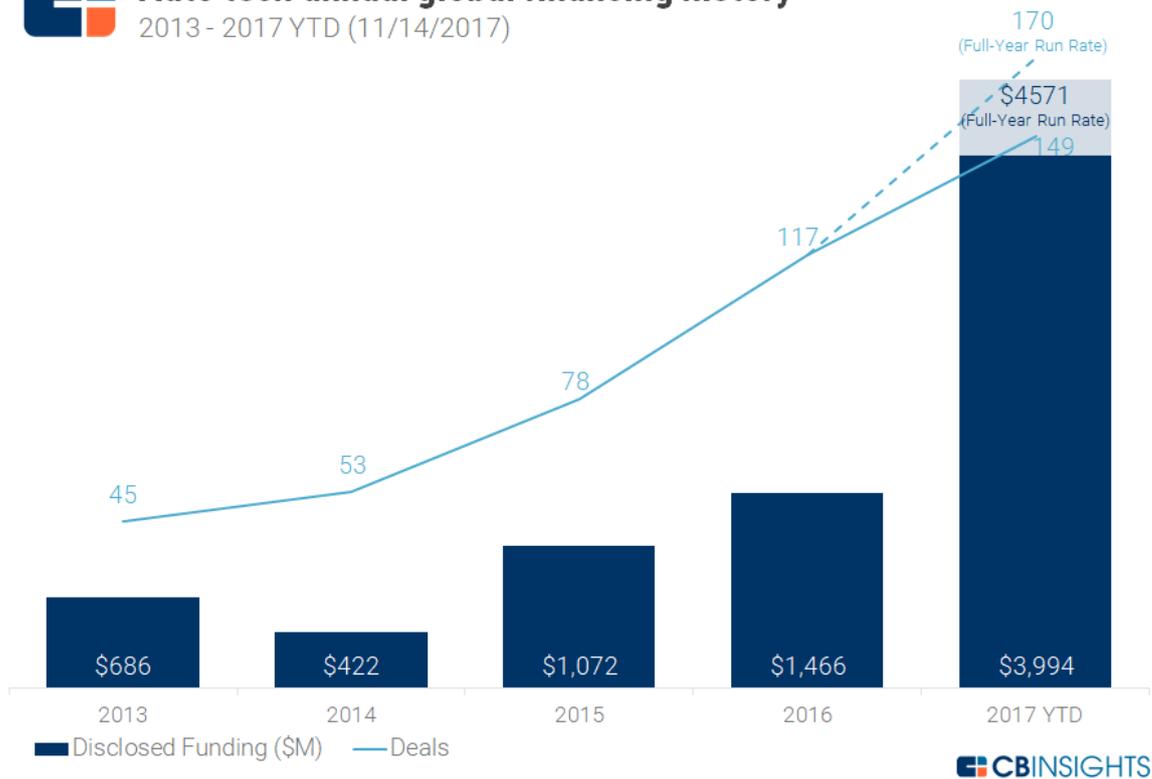


Figura 1.5: Dinero invertido en tecnología automotriz <https://www.cbinsights.com/research/auto-tech-startup-investment-trends/>.

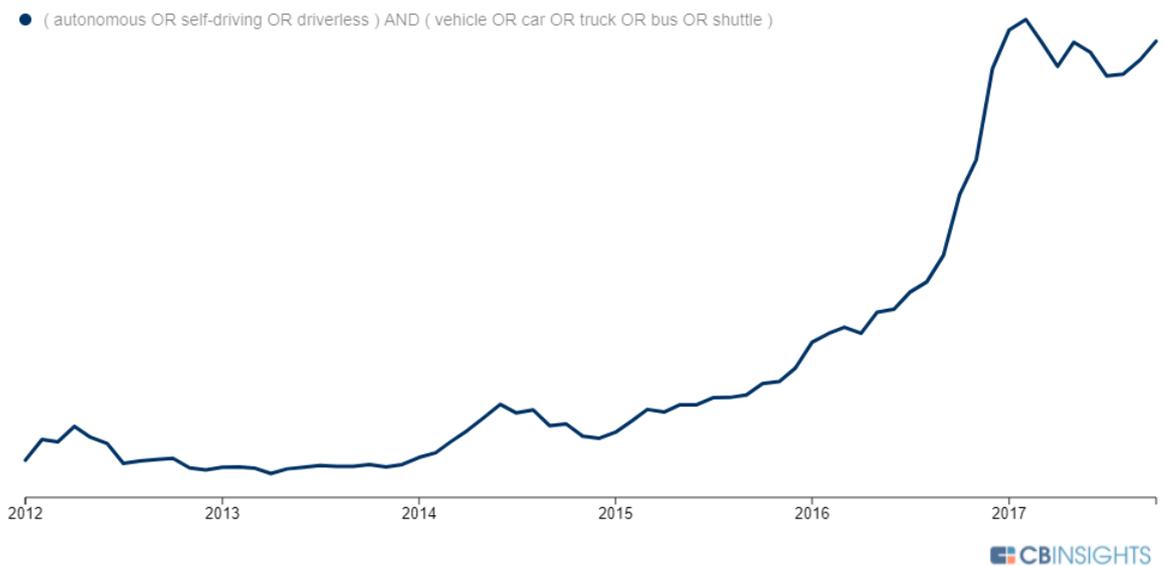


Figura 1.6: Grafica del interés mediático sobre los *vehículos autónomos* <https://www.cbinsights.com/research/auto-tech-startup-investment-trends/>.

Es por esto que actualmente es correcto investigar y analizar el mercado de los *vehículos autónomos* dado que está demostrado que es un mercado en auge y que como mínimo durara hasta 2020-2025, año en el que se prevé salga el primer vehículo totalmente autónomo.

1.3 Competencias adquiridas

Gracias al estudio realizado durante este documento se han desarrollado las siguientes competencias:

- **CE4** Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- **CE12** Capacidad para aplicar métodos matemáticos, estadísticos y de inteligencia artificial para modelar, diseñar y desarrollar aplicaciones, servicios, sistemas inteligentes y sistemas basados en el conocimiento.

Estos conocimientos han sido desarrollados a lo largo de este documento más en concreto durante la investigación de las tecnologías que cubren todos los aspectos relacionados con los *vehículos autónomos* haciendo especial hincapié en:

- **Niveles SAE:** dado que este documento tiene como intención realizar una propuesta de vehículo autónomo que evolucione a la par que se van realizando análisis y pruebas se ha propuesto seguir el estándar presentado por SAE y seguir sus niveles de autonomía como modelo a seguir.
- **Principales áreas de automoción:** se pretende analizar distintas áreas de trabajo relacionadas con los *vehículos autónomos*. Estas áreas están relacionadas con los distintos módulos que debe implementar un vehículo autónomo:
 - **Percepción y modelado del entorno:** de este área se adquirirán conocimientos de tecnologías actuales, sensores y métodos para fusionar la información y representarla.
 - **Localización y construcción de mapas:** se analizaran métodos existentes para construcción de mapas y métodos de localización.
 - **Planificación de trayectorias:** en este área se investigara y analizara distintos algoritmos de planificación de trayectorias.
 - **Toma de decisiones:** se analizaran distintos métodos para la toma de decisiones y se detallara una arquitectura base para su definición.
 - **Modelo cinemática y control de movimiento:** se tomaran conocimientos de modelos cinemáticos y se definirán sistemas de control de movimiento centrándose en los vehículos articulados.

- **Estado actual del mercado:** para realizar un buen análisis se investigara sobre el estado actual del mercado para obtener un conocimiento actual de él, centrándose en los vehículos articulados, sensores y software de desarrollo.

1.4 Estructura del documento

- **Capítulo 1: Introducción**

Se realiza una introducción al Trabajo fin de Master (TFM)

- **Capítulo 2: Objetivos**

Se definen los objetivos propuestos y que se esperan realizar durante la ejecución de este proyecto.

- **Capítulo 3: Método de trabajo**

En este capítulo se detalla la metodología de trabajo utilizada para la realización de este documento.

- **Capítulo 4: Revisión científica del estado del arte**

En este capítulo se realiza un breve repaso al estado del arte de las etapas de trabajo relacionadas con los *vehículos autónomos*. Se detallaran componentes HW útiles para la percepción y técnicas de modelado del entorno. Posteriormente se entra en detalles relacionados con métodos de localización y construcción de mapas para continuar con técnicas de planificación de trayectorias. Se continuara con algoritmos y métodos de toma de decisiones y por ultimo métodos y sistemas de control del movimiento.

- **Capítulo 5: Soluciones comerciales**

En este capítulo se detalla el estado actual del mercado y las soluciones comerciales, tanto de vehículos autónomos, como de sensores y software de desarrollo.

- **Capítulo 6: Propuesta de camión autónomo**

En este capítulo se realiza una definición del entorno junto a los requisitos funcionales del prototipo para al fin detallar como se ha realizado la integración de sensores. Por último se realizara un análisis de costes del proyecto.

- **Capítulo 7: Conclusiones**

En este capítulo se detallan las conclusiones de lo aprendido durante la elaboración de este documento y las conclusiones del prototipo presentado.

Capítulo 2

Objetivos

ESTE apartado está orientado a informar al lector de los detalles y objetivos del proyecto. A lo largo de este capítulo se describirán detalladamente los objetivos propuestos en este TFM por lo que inicialmente se describirá un objetivo global o general que será detallado minuciosamente y posteriormente se definirán una serie de puntos que desglosaran los objetivos parciales del proyecto.

2.1 Objetivo general

Como ya se ha avanzado en el capítulo de Introducción las soluciones basadas en sistemas de asistencia a la conducción, comúnmente denominadas ADAS¹, o incluso las relativas a la conducción totalmente autónoma han cobrado una relevancia significativa. Numerosas empresas están invirtiendo capital en sus equipos de I+D para investigar dicha tecnología, incluso algunas ya están desarrollando vehículos capaces de conducirse por sí solos tanto en entornos urbanos e interurbanos garantizando una menor tasa de accidentes², conduciendo de manera más eficiente y reduciendo la cantidad de residuos generados. En este contexto, y con la intención de reducir el número de accidentes debidos a errores humanos, se están implantando, recientemente, soluciones de conducción autónoma en entornos industriales. Una de las tareas más recurrentes en el entorno industrial es la denominada Load Haul Dump (LHD), consistente en el proceso que se realiza de carga, transporte y descarga de los materiales por parte de vehículos de gran tonelaje, y que en cierta medida es muy similar a la tarea que realizan los transportistas de mercancías. Si esta tarea se focaliza en escenarios físicos donde la empresa tiene cierto control sobre el entorno, entonces es posible construir soluciones que, aunque son tremendamente complejas, pueden ser más acotadas y por lo tanto más simples.

Por ello, el objetivo general de este TFM consiste en proponer una solución conceptual de camión autónomo para la resolución de problemas LHD. Dada la similitud entre tareas se analizara e investigara por igual vehículos articulados de gran tonelaje (vehículos mineros en entornos industriales) y vehículos de remolque de medio tonelaje (camiones de transporte). En

¹<https://www.escoladeconductores.es/ver/758/todo-sobre-el-sistema-adas.html>

²<https://www.seguridadvialenlaempresa.com/seguridad-empresas/actualidad/noticias/coches-con-tecnologia-adas-evitarian-50-mil-accidentes.jsp>

este documento se planteara la realización de un estudio detallado de los dispositivos hardware de sensorización, prototipos de camiones autónomos existentes en la actualidad, y técnicas y algoritmos relativos a la percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias, toma de decisiones y control del movimiento. Una vez finalizado dicho objetivo se pretende obtener un esquema de trabajo que defina:

- Una solución escalable que siga los niveles SAE.
- Pasos a seguir o recomendaciones para implementar los distintos niveles SAE.
- Conjunto de sensores HW que permita cubrir todas las necesidades de cada nivel SAE.
- SW a utilizar dependiendo de las necesidades de cada nivel SAE.

2.2 Objetivos específicos

El objetivo general anteriormente descrito puede ser dividido en los siguientes objetivos parciales que compondrán los hitos o bases para definir si el objetivo general se ha completado o no:

- **Identificación y definición** de dispositivos y técnicas aplicadas a la conducción autónoma en general que pueden extrapolarse para la resolución de problemas LHD. Una vez más se hace hincapié en que se tendrán en cuenta soluciones industriales y de transporte de mercancías.
- **Esquema.** Se busca definir un esquema de trabajo que establezca los pasos recomendados desde que se comienza a diseñar una solución hasta que el camión autónomo se implanta realmente según el nivel SAE correspondiente. Dicho esquema definirá una serie de pasos a seguir para definir las características del entorno, tareas a realizar por parte del vehículo y el gasto de implementar dicha solución.
- **Escalabilidad.** Se busca la realización de una solución escalable que considere, por último, la existencia de más entidades (otros camiones, vehículos, personas, etc) en el entorno. Se busca que la solución obtenga, como mínimo, un nivel SAE 4.
- **Estimación de costes** para una solución real de un camión autónomo para la resolución de problemas LHD que siga el esquema definido considerando la integración de dispositivos HW existentes y de soluciones SW ready-to-use.
- **Usabilidad.** Se quiere desarrollar el esquema lo más amigable posible y por ello se definirán los términos utilizados por el estándar SAE y que serán utilizados para definir el esquema con el fin de que dicho esquema pueda ser utilizado por empresa del sector como punto de partida para implementar sistemas de vehículos autónomos.

Método de Trabajo

EN el siguiente capítulo se detallarán los pasos seguidos para la elaboración de este estudio detallado de los dispositivos hardware de sensorización, prototipos de camiones autónomos existentes en la actualidad, y técnicas y algoritmos relativos a la percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias, toma de decisiones y control del movimiento. Este capítulo tiene como finalidad que el lector entienda como ha sido estructurado y planificado el estudio.

3.1 Metodología

Según la RAE, una **metodología** es, «*Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal*»¹. Por ello, en la siguiente sección se definirán las etapas que se han seguido de forma sistemática para la elaboración de este estudio.

A grandes rasgos, todo estudio empieza por el **planteamiento de un problema**. En este caso se pretende solucionar los problemas derivados de las operaciones LHD.

Para la elaboración de este documento se ha utilizado una variante del procedimiento básico para la solución de problemas que plantea **seis etapas bien definidas** que seguidas de forma sistemática desembocan en la solución del problema. A continuación se detallan las etapas, previamente adaptadas, que se seguirán en este documento:

- **Definir** el ¿Qué? y el ¿Cómo? del problema. En esta etapa se busca definir qué problema existe, a partir de una **definición del problema**, y el cómo se debe solucionar, a partir de la definición de una serie de objetivos que se desea cumplir para que dicho problema este solucionado. Dichos **objetivos** serán detallados de forma general para luego ser desglosados en pequeños **objetivos parciales** que marcaran pequeños hitos solventados.
- **Recoger** todos los datos. En este paso se debe recoger todos los datos relativos al problema para así poder describir el problema, y sus objetivos, de la manera más correcta y veraz. Para ello se puede realizar un **análisis de palabras clave**. Este análisis de palabras clave generara un conjunto de palabras que serán utilizadas como base del

¹<http://dle.rae.es/srv/search?m=30&w=metodología>

análisis.

- **Examinar** los datos recogidos. Tras la recopilación de los datos relevantes al problema a partir de las palabras claves obtenidas se debe realizar un **examen crítico de los datos recopilados**.
- **Considerar** las posibles soluciones que cumplan los objetivos. En esta consideración se tendrán en cuenta **soluciones existentes** que cubran dichos objetivos o posibles **soluciones planteadas** a partir de los datos obtenidos.
- **Aplicar** la solución. En este paso se aplicara la solución elegida en el paso anterior, para así **solventar el problema y alcanzar los objetivos**.
- **Mantener en observación** los resultados y los datos. Esta última etapa se realizada de forma periódica y buscara **actualizar los datos** recopilados, a partir de las palabras claves, para así mantener dicha solución lo más actualizada posible.

Con la intención de seguir estos pasos de la manera más sistemática posible y con la intención de documentarlo de la mejor manera se ha estructurado este documento siguiendo dichas etapas. A continuación se detalla cómo se han seguido dichas etapas en la elaboración de este documento.

Como se puede observar, en el capítulo *Introducción* (véase § 1) se ha presentado el problema a partir su alcance y contexto socio-económico para posteriormente, en el capítulo *Objetivos* (véase § 2 definir el objetivo principal que se quiere alcanzar tras solucionar dicho problema y los objetivos parciales que lo desglosan. Estos dos capítulos englobarían el paso de *definición* del problema.

El siguiente paso, la *recogida de datos*, se ha llevado a cabo en el capítulo *Revisión científica del estado del arte* (véase § 4). En paso es el más importante ya que la obtención de unos buenos datos da paso a una buena solución. Para ello se realizo un **análisis de palabras clave** relacionadas con las soluciones de vehículos autónomos y se obtuvo la siguiente lista de palabras relacionadas: sensorización, algoritmos, percepción, modelado, entorno, trayectoria, planificación, decisiones, control, movimiento, mapas, localización, estándares, empresas y desarrollo, entre otras. Estas palabras clave se obtuvieron de una serie de preguntas que surgieron a partir de los objetivos. A partir de dichas palabras se obtuvieron los siguientes **campos de investigación**: percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias, toma de decisiones y control de movimiento. Con los campos de investigación ya definidos se realizo una **búsqueda sistemática**, de cada campo, definida por los siguientes pasos:

- **Búsqueda general** del campo a investigar. En este paso se asienta la definición del concepto y se apuntan los términos relacionados.
- **Búsqueda de términos relacionados** relacionados con el campo anterior. En este paso se obtiene la información detallada de los términos que forman el campo a investigar.

- **Recopilación bibliográfica** a partir de la búsqueda anterior.

Una vez se ha realizado la búsqueda y se ha recopilado toda la información relevante al campo de investigación se realiza una **estructura y resumen** que permite organizar los términos encontrados y a partir de esta estructura se realiza la **documentación** del campo de investigación. Estos pasos se realizan de forma **iterativa** hasta que todos los campos han sido investigados y la recogida de datos ha sido terminada.

En el capítulo *Soluciones comerciales* (véase § 5), y siguiendo con la recogida de datos, aunque ligado con el paso cuatro de *considerar* posibles soluciones, se investigaron soluciones actuales de camiones autónomos, HW de sensorización y entornos de desarrollo ya que dicha investigación estaba estrechamente ligada a la búsqueda de una solución para nuestro problema.

El paso *aplicar* ha sido desarrollado en el capítulo *Propuesta de camión autónomo* (véase § 6) donde se han aplicado los datos obtenidos y la solución elegida para alcanzar el objetivo de la creación conceptual de camión autónomo para la resolución de problemas LHD.

El último paso queda reflejado en las tareas propuestas como trabajo futuro del capítulo *Conclusiones* (véase § 7).

En resumen, la elaboración de este documento y todo el análisis realizado ha sido llevado a cabo mediante este método de trabajo

Revisión científica del estado del arte

EN el siguiente apartado se realizara una revisión científica del estado del arte actual relacionado con el mundo de los vehículos autónomos con la intención de poner al lector en un contexto actual sobre las materias a tratar. Se detallaran los conceptos relacionados con la percepción y modelación del entorno, los métodos de localización, mecanismos para la construcción de mapas locales y globales, mecanismos de toma de decisiones y control del movimiento de vehículos autónomos. Una vez finalizado este capítulo se pretende que el lector tenga un conocimiento elevado y actual sobre las tecnologías involucradas en este proyecto.

4.1 Percepción y modelado del entorno

4.1.1 Introducción

Un dispositivo autónomo, por definición, debe ser capaz de obtener información del entorno y poder actuar consecuentemente. Para ello la **percepción y el modelado del entorno** tiene como principal responsabilidad la obtención de información.

Según la Real Academia Española (RAE) ¹ la percepción es: «*Sensación interior que resulta de una impresión material hecha en nuestros sentidos*».

En el ámbito de la investigación, a nivel general, la percepción se organiza según los sentidos clásicos: visión, audición, tacto, olfato y gusto. Aunque existen distintos tipos de percepción derivados de estos sentidos:

- **Percepción visual:** de los dos planos de la realidad externa. Dicha percepción está relacionada con la forma, color y movimiento, entre otras cosas.
- **Percepción espacial:** de las tres dimensiones. La inclusión de la tercera dimensión permite percibir la profundidad del entorno.
- **Percepción auditiva:** de los ruidos y sonidos.
- **Percepción térmica:** de las variaciones de temperatura. Este tipo de percepción permite identificar fuentes de calor y de frío en un entorno.

¹<http://dle.rae.es/srv/search?m=30&w=percepción>

- **Percepción del equilibrio:** que permite percibir la orientación local del individuo permitiendo así discernir entre arriba y abajo.
- **Percepción de la forma:** es el resultado de la percepción del contorno y del contraste de las cosas, fundamentalmente a través de la vista y del tacto.
- **Percepción del campo magnético:** permitiendo así percibir las variaciones del campo magnético y su orientación.

Aunque existen más tipos de percepción estos son los más relacionados con los dispositivos autónomos, dado que actualmente existen en el mercado dispositivos capaces de percibir dicha información del entorno. En la definición dada sobre percepción se habla de «*sentidos*», pero ¿Cómo podemos emular los sentidos humanos? En un vehículo autónomo estos sentidos son aportados por sensores, que son distribuidos a lo largo del vehículo para así poder obtener información del entorno.

Un **sensor** es un dispositivo eléctrico que mide una magnitud física y la transforma en un impulso eléctrico. Existen multitud de dispositivos o *sensores* que son clasificados según la magnitud física que son capaces de medir:

- **Luz** (con su gama de espectro: visible, infrarroja, ultravioleta)
- **Sonido y ultrasonido**
- **Gravedad** (inclinación, posición)
- **Temperatura**
- **Humedad**
- **Presión y/o fuerza**
- **Velocidad**
- **Magnetismo**
- **Ubicación**
- **Proximidad**
- **Distancia**

Dado que los *sensores* miden magnitudes y las transforman en impulsos eléctricos es lógico diferenciar entre dos tipos de sensores: sensores **digitales** y sensores **analógicos**.

Los *sensores digitales* son aquellos que transforman la información obtenida del entorno en impulsos eléctricos que comprenden valores discretos. Este tipo de sensores son utilizados para discernir entre dos tipos de valores, como por ejemplo, pulsado o no pulsado.

Cuando se requiere obtener un valor intermedio se utilizan los *sensores analógicos* capaces de transformar las magnitudes medidas en impulsos eléctricos que comprenden valores

continuos. Cabe destacar que este tipo de *sensores* necesitan de un **conversor analógico - digital** para que dicha información pueda ser interpretada por el vehículo autónomo.

Los *sensores*, ya sean *sensores digitales* o *sensores analógicos*, poseen una serie de características importantes que pueden ser utilizadas para decidir si, según la tarea a realizar, son adecuados o no.

Una de las principales características de un *sensor* es su **fiabilidad**. La *fiabilidad* de un sensor es la cualidad que posee dicho dispositivo para obtener datos del entorno lo más estables y semejantes a la realidad. Se identifica como una de las características más importantes dado que de nada sirve que las demás características del *sensor* sean las requeridas si la *fiabilidad* del dispositivo es nula y el valor obtenido por él no es fiel a la realidad.

Cabe destacar que la *fiabilidad* de un *sensor* puede ser compensada mediante redundancia de sensores y una programación adicional, pero estos métodos implican aumentar el tiempo de computación y elevar los costes del proyecto.

Otra de las características de un *sensor* es la **precisión** de los datos obtenidos. Esta característica define como de *preciso* es el valor continuo del *sensor*, o lo que es lo mismo, cual es la diferencia entre dos valores lo más próximos posibles. Es una característica importante siempre y cuando se quieran diferenciar unidades muy próximas entre sí.

Las dos características anteriores están muy ligadas al **rango de valores** del *sensor*. Este *rango de valores* define los valores máximos y mínimos que puede tomar el sensor. Por regla general, *sensores* con gran *rango de valor* poseen una *precisión* inferior a *sensores* con *rango de valores* inferior.

La **velocidad de muestreo** es otra de las características de los *sensores* midiendo así los intervalos entre toma de medidas por parte del sensor. Es posible desechar los datos obtenidos y así reducir dicha *velocidad de muestreo* en caso de que el procesador no sea capaz de gestionar la información obtenida del sensor. Esta posibilidad lo transforma en una característica con menor peso que las demás.

Existen otras características como tal que no están relacionadas con la información obtenida por parte del *sensor* si no con el propio *sensor* en sí: una es la capacidad de **calibración** que permite modificar la funcionalidad del *sensor* de forma rápida y sencilla y por último está el **coste** del sensor, que está estrechamente ligado a las características de la información obtenida mediante dicho sensor. Estas características son irrelevantes a nivel científico hablando desde el punto de vista teórico.

Por último, y para entrar más en detalle sobre los *sensores* actuales en el mercado, se desglosarán los tipos de sensores que existen actualmente:²

²http://robots-argentina.com.ar/Sensores_general.htm

■ Sensores de luz

- Elementos sensibles
 - Light-Dependent Resistors (LDR)s³ o Fotorresistores (resistores variables por la incidencia de la luz)
 - Foceldas o celdas fotovoltaicas
 - Fotodiodos
 - Fototransistores
 - Charge-Coupled Device (CCD)⁴
 - Cámaras de vídeo
- Módulos integrados
 - Reflectivo
 - De ranura

■ Sensores de presión y fuerza

- Elementos sensibles
 - Microinterruptores
 - Sensores de presión
 - Sensores de fuerza
- Sensores
 - Sensores de contacto (sandwich, bigotes, antenas)
 - Piel robótica

■ Sensores de sonido

- Elementos sensibles
 - Micrófonos
 - Captadores piezoeléctricos
- Módulos integrados
 - Rangars (medidores de distancia) ultrasónicos

■ Sensores para medición de distancia

- Módulos integrados
 - Medidores de distancia ultrasónicos
 - Medidores de distancia por haz infrarrojo

■ Sensores de gravedad (posición)

- Acelerómetros, sensores de vibración
- Sensores pendulares (Inclinómetros)

³<https://en.wikipedia.org/wiki/Photoresistor>

⁴<https://es.wikipedia.org/wiki/Dispositivo.de.carga.acoplada>

- Contactos de mercurio
- Giróscopos
- **Sensores de temperatura**
 - Termistores
 - Resistance Temperature Detector (RTD)s⁵ (Termorresistencias)
 - Termopares, Termocuplas
 - Diodos
 - Circuitos integrados
 - Piro sensores (a distancia)
- **Sensores de humedad**
 - Sensores capacitivos
 - Sensores resistivos
 - Módulos integrados
- **Sensores de velocidad**
 - Tacómetros
 - Codificadores (encoders)
- **Sensores de magnetismo**
 - Efecto Hall
 - Brújulas electrónicas
 - Interruptores magnéticos
- **Sensores de ubicación geográfica**
 - Global Positioning System (GPS)⁶
 - Receptores de radiobalizas
- **Sensores de proximidad**
 - Sensores capacitivos
 - Sensores inductivos

Este desglose coincide, en ciertos aspectos, con la distinción que se ha realizado de los tipos de percepción.

Cualquier sistema que necesite *percibir* u obtener información del entorno necesita utilizar uno o varios de estos *sensores*.

En el ámbito que nos atañe, el de los vehículos autónomos, y más concreto en, los camiones autónomos, estos *sensores* o dispositivos son de distinta naturaleza, siendo los más

⁵<https://es.wikipedia.org/wiki/RTD>

⁶https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global

utilizados los sistemas GPS, los dispositivos Light Detection and Ranging (LiDAR)⁷ o de medición infrarroja, las cámaras de visión estéreo y los radares. Cabe destacar que, por lo general, es bastante frecuente que los vehículos autónomos posean varios dispositivos de un mismo tipo, por ejemplo dispositivos LiDAR, para garantizar, la obtención de datos del entorno desde una perspectiva completa, es decir, cubriendo la práctica totalidad del entorno físico que rodea al vehículo. También, gracias a la redundancia de dispositivos se aumenta el grado de seguridad de los datos obtenidos, aunque estos posean un grado elevado de *fiabilidad*.

4.1.2 Fusión de informacion

Debido a la gran variedad y al número elevado de *sensores* y a las distintas magnitudes que estos pueden captar, es inevitable requerir un mecanismo, o técnica, que sea capaz de gestionar e interpretar tal cantidad de información. Esta información, en su origen, es información en **bruto**, sin tratar, y necesita ser procesada, gestionada e integrada en el sistema mediante técnicas de **fusión de información** [XS02].

La fusión de información es un subconjunto de la fusión de datos (véase Figura 4.1), siendo la fusión de datos un proceso de integración de datos generados por múltiples fuentes de datos para así producir información más consistente, fiel y útil que la producida individualmente por las fuentes de datos [HL97]⁸.

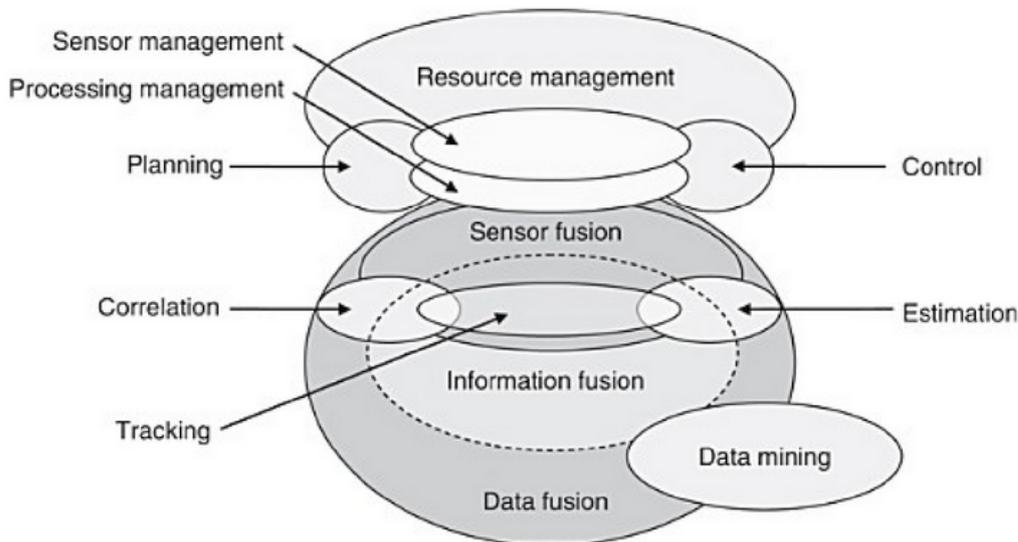


Figura 4.1: Fusión de datos: Terminología.

⁷<https://es.wikipedia.org/wiki/LIDAR>

⁸D. L. Hall and J. Llinas definieron: «*data fusion techniques combine data from multiple sensors and related information from associated databases to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone.*»

Los procesos de *fusión de datos* son normalmente categorizados como nivel bajo, intermedio y alto, dependiendo de en qué momento del proceso tenga lugar la fusión de datos [Kle04].

La *fusión de datos* a bajo nivel combina múltiples fuentes de datos en bruto para así producir un nuevo dato en bruto. Se espera que la fusión de datos genere datos más sintéticos⁹ e informativos que los originales.

A mediados de los años 80, el **Joint Directors of Laboratories (JDL)** formó el **Data Fusion Subpanel** que posteriormente sería conocido como el **Data Fusion Information Group (DFIG)**. La JDL/DFIG introdujo un modelo de *fusión de datos* que estaba dividido en varios procesos o niveles. Actualmente, los seis niveles del modelo DFIG son:

- Nivel 0: Pre-proceso de las fuentes.
- Nivel 1: Evaluación del objeto.
- Nivel 2: Evaluación de la situación.
- Nivel 3: Evaluación del futuro.
- Nivel 4: Refinamiento de Procesos.
- Nivel 5: Refinamiento del Usuario.

En el **nivel 0** del modelo se obtienen los datos de los *sensores* y se pre-procesan independientemente los datos realizando operaciones (filtrado de datos, ordenación, etc. . .) sobre los datos obtenidos a partir del SW del *sensor*. Es en el **nivel 1** donde se combinan los datos de los sensores para así, en el ámbito de los vehículos autónomos, obtener características del vehículo. Las operaciones más utilizados a este nivel son: **alineación de datos, correlación de datos, obtención de atributos** y dependiendo de los sensores, **estimación de la identidad** de obstáculos u objetos. En el **nivel 2** del modelo se busca interpretar los resultados del nivel anterior. Las técnicas más utilizadas en este nivel son el uso de la Inteligencia Artificial (IA) y el razonamiento automático. Los procesos que se llevan a cabo en este nivel son: **agregación de objetos, interpretación contextual y evaluación multiperspectiva**. En el **nivel 3** se busca prever el futuro y calcular la proyección futura de la entidad a partir de la situación actual. Las técnicas utilizadas en este nivel están relacionadas con la IA, los modelos predictivos, razonamiento automático y la estimación estadística. Los procesos realizados en este nivel son: la **estimación de capacidad de fuerza, estimación de implicaciones** y como en el nivel anterior, la *evaluación multiperspectiva*. En el **nivel 4** se realiza un meta-proceso que mejora el rendimiento del proceso de *fusión de datos* en tiempo real. Este nivel se considera parcialmente dentro del proceso de *fusión de datos* dado que es vital para su evolución y su correcto funcionamiento. Los procesos que se llevan a cabo en este nivel son:

⁹Según el Diccionario McGraw-Hill de Términos Científicos y Técnicos «*Any production data applicable to a given situation that are not obtained by direct measurement*».

gestión de la misión, predicción de entidades, requerimiento de las fuentes, textbfmodelización del rendimiento del sistema y por último el **control del sistema**. Por último, en el **nivel 5** del modelo DFIG, se busca mejorar el modelo de *fusión de datos* mediante la relación sensor-procesamiento-persona. En este nivel se **representan los resultados** para permitir al operador **corregir errores** (véase Figura 4.2) [Abe07].

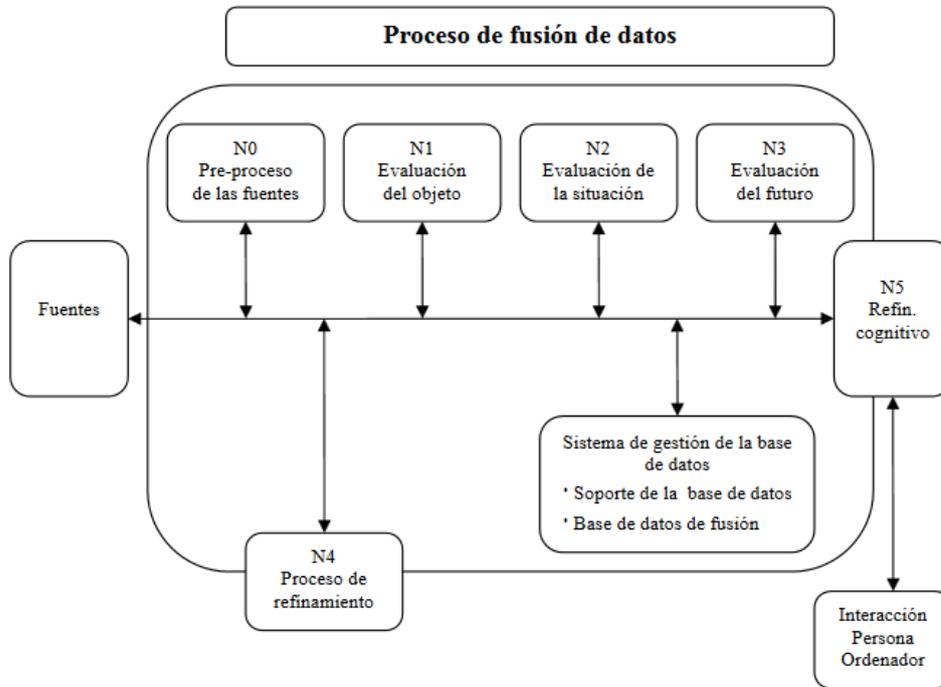


Figura 4.2: Proceso de fusión de datos.

Aunque el modelo JDL (Niveles 1-4) aun está en uso, es comúnmente criticado dado que los niveles deben producirse necesariamente en un orden y no tiene en cuenta al ser humano en el proceso. El modelo DFIG (Niveles 0-5) corrige los problemas del modelo JDL y añade a los procesos la concienciación de la situación, el refinamiento del usuario y la gestión de la misión [EBL12]. A pesar de estas deficiencias, los modelos JDL/DFIG son útiles para visualizar el proceso de fusión de datos, lo que facilita la discusión y el entendimiento común [MLL12], además de ser importantes para el diseño de la *fusión de la información* a nivel de sistemas [EBL12].

Otro termino a tener en cuenta es **la integración de información** que es la unión de información de distintas fuentes heterogéneas con diferentes representaciones conceptuales, contextuales y tipográficas. Esta técnica es usada en minería de datos y en consolidación de datos de recursos no estructurados o semi-estructurados.

La *fusión de información*, es un término relacionado con la *integración de la información* e implica la combinación de información en un nuevo conjunto de información para reducir la redundancia y la incertidumbre.

Ejemplos de tecnologías disponibles para la *integración de la información* son la deduplicación¹⁰ y las métricas de cadena¹¹ que permiten la detección de texto similar en diferentes fuentes de datos por coincidencia difusa. Existen una gran cantidad de métodos para estas áreas de investigación propuestos por el International Society of Information Fusion (ISIF).¹²

La **fusión de sensores** también conocida como *fusión de datos multi-sensor* es un subconjunto de la fusión de información.

La técnica *fusión de sensores* es una técnica que está estrechamente vinculada con la gestión y procesamiento sinérgico de la información proveniente de diferentes *sensores*. Esta técnica es ampliamente utilizada en cualquier ámbito que requiera combinar, agrupar y fusionar una cantidad elevada de datos para su posterior tratamiento, asegurando así la apropiada calidad e integridad de las decisiones a tomar (véase Figura 4.3).

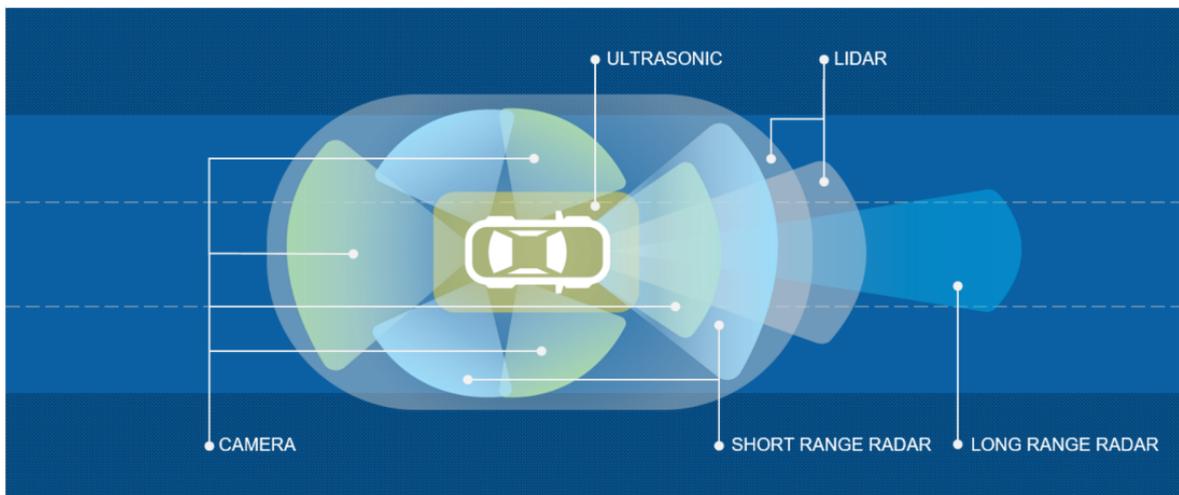


Figura 4.3: Sistema multi-sensor.

Aunque el procesamiento de estos datos gracias a las técnicas de *fusión de información* puede ser costoso a nivel computacional, sobrepasa las limitaciones fundamentales de los sistemas *mono-sensor*:

- El nivel de **incertidumbre** generado en los sistemas *mono-sensor* es invariable, dado que las mediciones realizadas por los sensores, que, aunque tengan un grado de *fiabilidad* elevado, pueden llegar a ser inciertas y ocasionalmente incorrectas.

¹⁰En computación, la deduplicación de datos es una técnica especializada de compresión de datos para eliminar copias duplicadas de datos repetitivos

¹¹En matemáticas y ciencias de la computación, una métrica de cadena (también conocida como una métrica de similitud de cadena o función de distancia de cadena) es una métrica que mide la distancia entre dos cadenas de texto para la comparación aproximada de cadenas o en la búsqueda de cadenas difusas

¹²<http://isif.org/>

- La mayoría de los *sensores* están centrados en una única magnitud física, por lo que los sistemas *mono-sensor* no pueden *percibir* en su totalidad el entorno, llegando sólo a obtener una **imagen parcial** de él.
- Sin redundancia, ni sensores de respaldo, la falla del único sensor equivale a la **falla total del sistema**.

Aparte de las desventajas de utilizar un sistema *mono-sensor*, los propios sistemas *multi-sensores*, utilizados conjuntamente con técnicas de *fusión de información*, poseen ciertas ventajas:

- La **redundancia** de dispositivos o *sensores* reduce la *incertidumbre* e incrementa la precisión de los datos *percibidos* aumentando así la *fiabilidad* intrínseca de los *sensores*.
- Dispositivos o *sensores* de naturaleza diferente aumentan la *percepción* del entorno, llegando así a obtener una **imagen fiel** de él.

Las técnicas de *fusión de información* están presentes en aéreas de robótica, sistemas bio-médicos, sistemas militares, equipo de monitoreo, medición remota, **sistemas de transporte**, control de procesos y sistemas de información. Dado que dicha técnica está muy centrada en el uso de *sensores* es necesario hacer una diferencia sobre ellos, por ello, los *sensores* pueden ser clasificados como **activos** o *pasivos* dependiendo de su manera de capturar información del entorno. Así un dispositivo LIDAR será clasificado como un *sensor activo* dado que **irradian las señales** infrarrojas para **capturarlas posteriormente** y así obtener información del entorno, también son tratados como *sensores activos* los radares y los *sensores* de ultrasonido y de radio. Por lo contrario, los *sensores pasivos*, como los sensores de infrarrojos y cámaras de visión, **capturan las emisiones de radiación** emitida por el entorno y la transforman en información. Cabe destacar que los *sensores activos* son *pasivos* a su vez.

Como ya se ha mencionado con anterioridad, los *sensores*, poseen ciertas características intrínsecas a la calidad del dispositivo, además del tipo de magnitud que pueden *percibir*, por eso a la hora de fusionar información es necesario considerar que cada sensor tiene sus propias características, un rango de actuación y que funcionan mejor o peor en base a las condiciones meteorológicas basándonos en su *fiabilidad*.

Al igual que las técnicas de *fusión de datos* son clasificadas según su tipo, las técnicas de *fusión de información* actuales pueden ser clasificadas como **bajo nivel**, **alto nivel** y **fusión híbrida** [THS08] [MWS05]

4.1.2.1. Técnicas de fusión de datos

Existen distintas técnicas de *fusión de datos* y existen multitud de clasificaciones dependiendo de quién las presente.

Según I. Bloch pueden ser clasificadas según su *comportamiento*. Esta técnicas fusionan los datos obtenidos basándose en el grado de credibilidad que tiene la fuente, en el intervalo

de valores y en una operación de combinación. La clasificación se realiza basándose en dos niveles. El primer nivel o primer grado de clasificación se centra en su dependencia del contexto y el segundo nivel o grado se centra en tipo de operación o función: **severa**, **prudente** o **indulgente**. Un operador *severo* implica que se le da menos peso a las fuentes con menos *fiabilidad*. El operador *prudente* tiene en cuenta a todas las fuentes y provee una información global. Por último, el operador *indulgente* incrementa el valor de *fiabilidad* de las fuentes teniendo en cuenta así la información menos incierta [Blo96].

A continuación se listan los distintos tipos de operadores:

- **Independiente del Contexto con Comportamiento Constante (ICCC):** Engloba todos los operadores que computan la información sin tener en cuenta los datos del exterior o del contexto, por lo que como su nombre indica son constantes. Las técnicas más conocidas son *inferencia Bayesiana*, *Dempster-Shafer* y ciertas clases de *lógica difusa*.
- **Independiente del Contexto con Comportamiento Variable (ICCV):** Estos operadores son independientes del contexto pero no tienen un comportamiento constante y tienen en cuenta el valor de credibilidad de las fuentes. Las técnicas utilizadas son ciertas clases de *lógica difusa* e IA, como los *sistemas expertos*.
- **Dependiente del Contexto (DC):** Este conjunto de operadores depende del contexto al igual que de la credibilidad de las fuentes. Este conjunto utiliza algunas técnicas de *conjuntos difusos* y IA como las *híbridas*.

Otra manera de clasificar las técnicas utilizadas es según la *lógica matemática* que éstas utilizan para incorporar la incertidumbre [HM04], esta clasificación es defendida por D.L. Hall y S.A.H McMullen. Este parámetro, la incertidumbre, es lo que diferencia las distintas técnicas utilizadas: *Bayes*, *Dempster-Shafer* y *Conjuntos difusos* por lo contrario, las técnicas de IA utilizan el factor del aprendizaje.

Existen distintas formas de obtener dicho valor de incertidumbre: **probabilidad**, **lógica evidencial** y **lógica difusa**. Utilizar métodos *probabilísticos* es la forma más difundida y por lo tanto la que tiene una base teórica más desarrollada y, aunque es la menos correcta para problemas complejos, es la más simple a nivel computacional. El método de *Bayes* y las *Leyes clásicas de probabilidad* son las técnicas más utilizadas.

La *lógica evidencial* que, a diferencia de la *probabilística*, propone aumentar la lógica proposicional añadiendo un operador que define el conocimiento que se tiene del entorno. Los métodos más utilizados son: *Dempster-Shafer* y la *Teoría evidencial generalizada*.

La *lógica difusa* que permite representar de forma matemática conceptos o conjuntos imprecisos. Los métodos más utilizados son: *conjuntos difusos* y algunos operadores *Híbridos de IA*.

Dado que es un ámbito multidisciplinar existen múltiples clasificaciones. Otros tipos de clasificaciones son [Cas13]:

- H.F. Durrant-Whyte [DW88] propuso clasificarlos según la **relación de las fuentes** de datos de entrada. Están clasificadas como: **complementarias**, **redundantes** o **cooperativas**.
- Dasarathy [Das97] propuso una clasificación basada en los tipos de datos de entrada y salida y su naturaleza.
- Dependiendo del **tipo de abstracción** de los datos: **medición en bruto**, **señales** y **características o decisiones**.
- Clasificación basada en los niveles definidos por el modelo JDL.
- Según el **tipo de arquitectura**: **centralizada**, **descentralizada** o **distribuida**.

Más en detalle, la clasificación definida por H.F. Durrant-Whyte [Cas13] define (véase Figura 4.4):

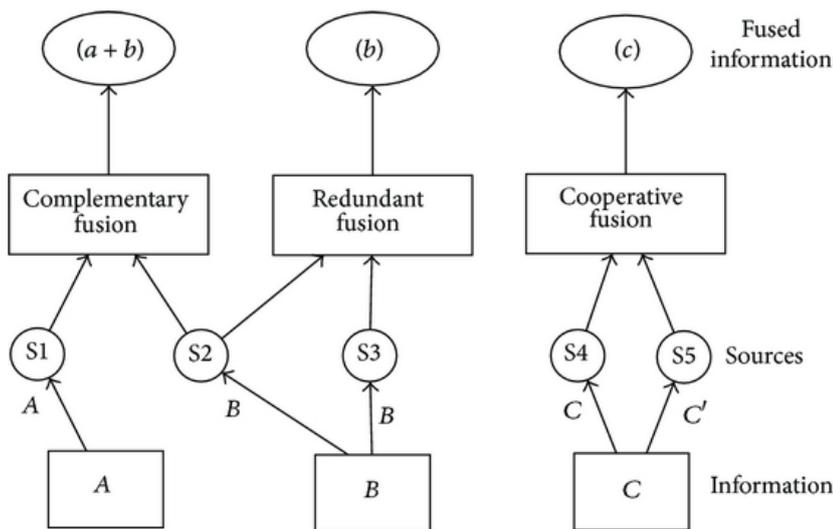


Figura 4.4: Clasificación según la relación de las fuentes.

- Complementarias: este grupo engloba aquellos casos donde las fuentes de entrada obtienen información del entorno desde diferentes puntos y por consiguiente dicha información de complementa posibilitando así la obtención de una visión global del entorno.
- Redundantes: cuando la información obtenida se solapa obteniendo así dicha información *redundantemente* llevando a un aumento del nivel de confianza de dicha información.

- Cooperativas: este grupo incluye los casos donde la información se fusiona o combina para obtener información más compleja del entorno.

Una de las clasificaciones más conocidas es la definida por Dasarathy [Das97] donde define cinco categorías (véase Figura 4.5) [Cas13]:

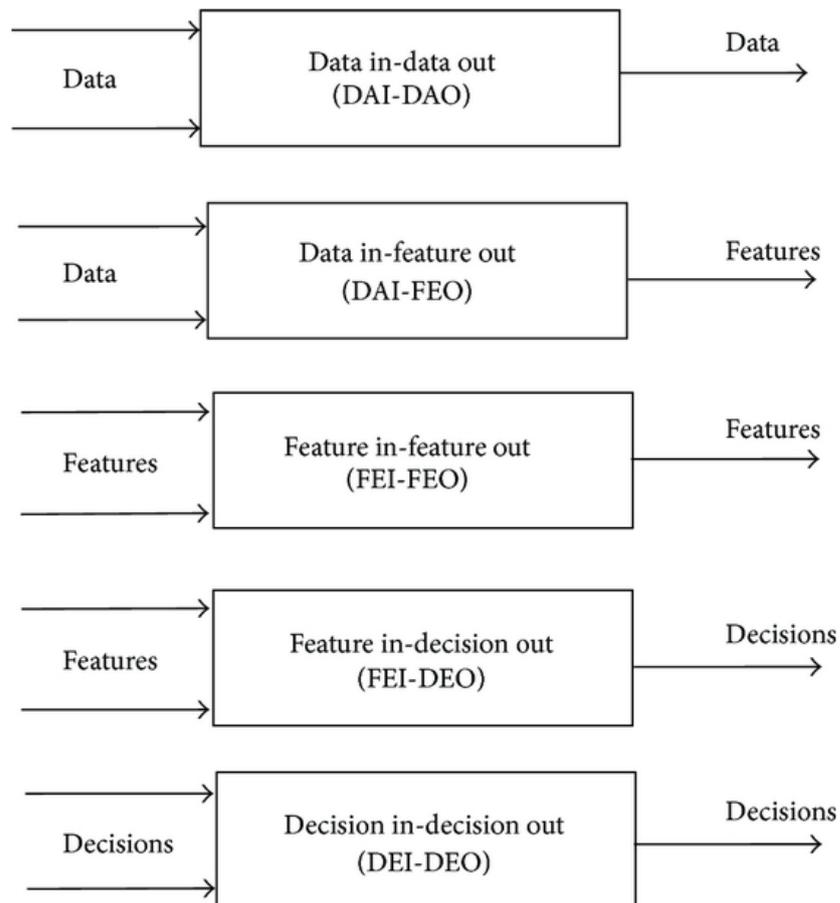


Figura 4.5: Clasificación según Dasarathy. Las cinco categorías.

- **Data In-Data Out (DAI-DAO):** este método emite los datos sin procesar, una vez recibidos. Los algoritmos empleados en este método se basan en algoritmos de procesamiento de señales e imágenes.
- **Data In-Feature Out (DAI-FEO):** con este método se extraen, de los datos en bruto, características o rasgos que describen una entidad del entorno.
- **Feature In-Feature Out (FEI-FEO):** este método procesa características o rasgos para así mejorar las actuales, obtener nuevas características, o refinarlas. También es conocido como fusión de características, fusión simbólica, fusión de información o fusión de nivel intermedio.

- **Feature In-Decision Out (FEI-DEO):** este método proporciona una decisión basándose en una serie de características que recibe.
- **Decision In-Decision Out (DEI-DEO):** como pasaba con el método FEI-FEO, este método procesa decisiones para así mejorarlas u obtener nuevas decisiones.

La clasificación basada en los niveles de abstracción de los tipos de datos fue definida por R. C. Luo, C.-C. Yih y K. L. Su [RCLS02]. Estos definieron cuatro niveles de abstracción:

- **Nivel de señales:** trata directamente las señales obtenidas de los sensores.
- **Nivel de pixel:** opera a nivel de imágenes siendo utilizado para mejorar imágenes.
- **Características:** se basa en los rasgos que extrae de las imágenes o sensores (velocidad, forma, etc. . .).
- **Símbolos:** en este nivel la información se transforma en símbolos. Este nivel también es conocido como el **nivel de decisiones**.

La fusión de la información normalmente se ocupa de tres niveles de abstracción: *mediciones, características y decisiones*. Otra posible clasificación basada en niveles de abstracción es:

- **Nivel bajo:** los datos en bruto son tratados directamente.
- **Nivel medio:** se tratan las características o los rasgos (forma, textura, posición, etc. . .) y se combinan para obtener rasgos nuevos.
- **Nivel alto:** conocido como el nivel de toma de decisiones, toma la representación simbólica y las combina para tomar una decisión.
- **Fusión de múltiples niveles:** utiliza los datos obtenidos en diferentes niveles de abstracción.

La clasificación JDL es el modelo más conocido relacionado con la fusión de datos. Como ya se ha comentado define cinco niveles de procesamiento. Estos cinco niveles pueden ser agrupados en dos grupos, **fusión de bajo nivel** y **fusión de alto nivel** que comprenden los siguientes componentes:

- **Fuentes:** las fuentes son las encargadas de obtener los datos.
- **Human-Computer Interaction (HCI):** es la interfaz encargada de aceptar datos de los operadores hacia el sistema y de producir una salía hacia los operadores.
- **Sistema de gestión de base de datos:** es el encargado de almacenar los resultados y la información utilizada.

Por último, la clasificación basada en el tipo de arquitectura está centrada en la ubicación de los datos, donde se generan y donde se procesan. Esta clasificación define cuatro tipos de arquitecturas (véase Figura 4.6) [Cas13]:

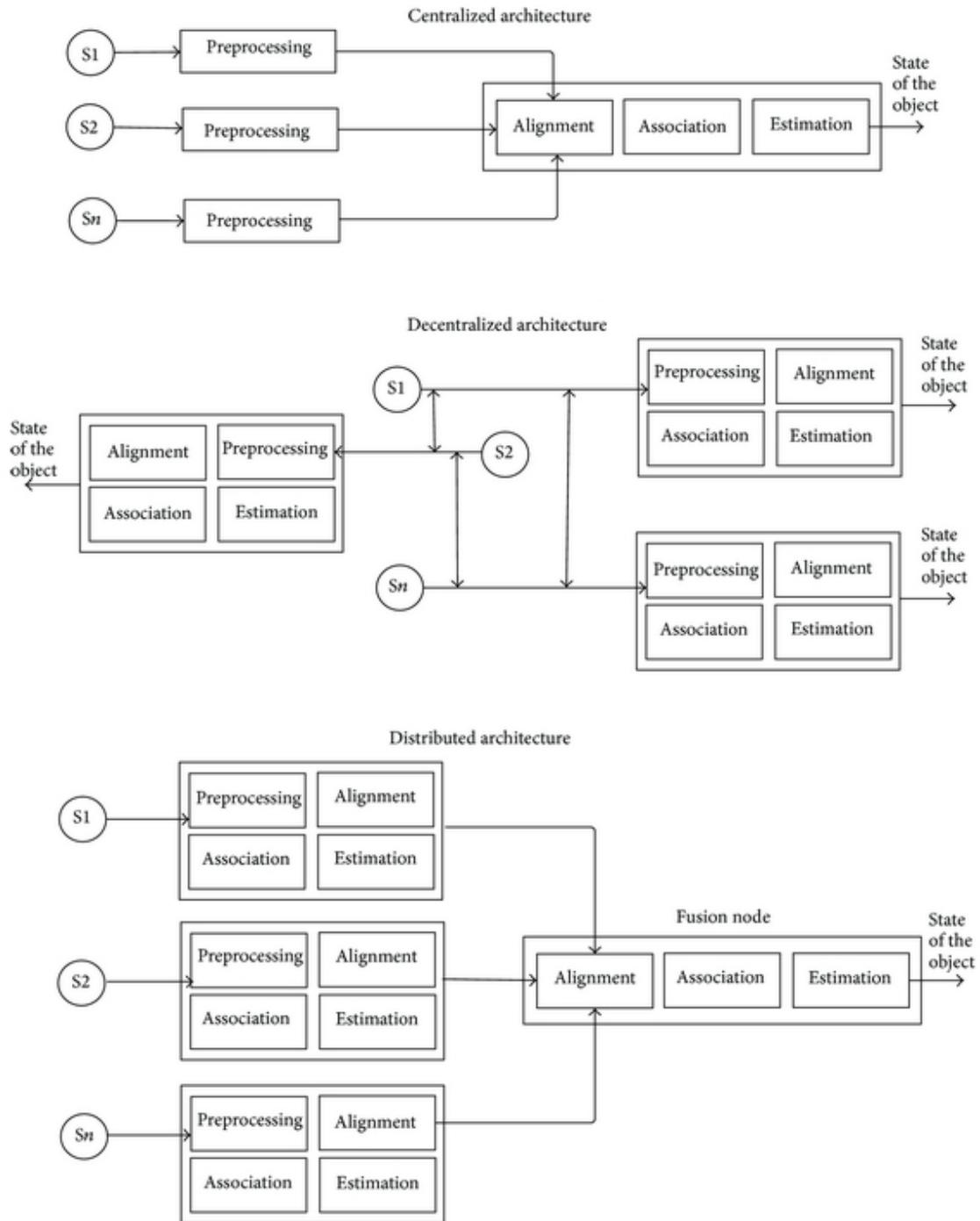


Figura 4.6: Clasificación basada en el tipo de arquitectura.

- *Centralizada:* el núcleo de procesamiento reside en el procesador central que recibe toda la información a tratar.
- *Descentralizada:* está compuesta por una red de nodos que procesan sus datos localmente. No existe un único punto de fusión de datos.

- *Distribuida*: cada nodo de la red procesa sus datos antes de enviarlos al nodo central o nodo de fusión.
- *Jerárquica*: combina arquitecturas descentralizadas y distribuidas generando así un esquema jerárquico donde los datos se procesan dependiendo del nivel.

Otros estudiosos defienden una clasificación de las técnicas de fusión de datos basándose en tres tipos no excluyentes: **asociación de datos**, **estimación del estado** y *toma de decisiones* [Cas13].

Las técnicas relacionadas con la *asociación de datos* se centran en determinar a qué objetivo corresponde la información obtenida (véase Figura 4.6) [Cas13].

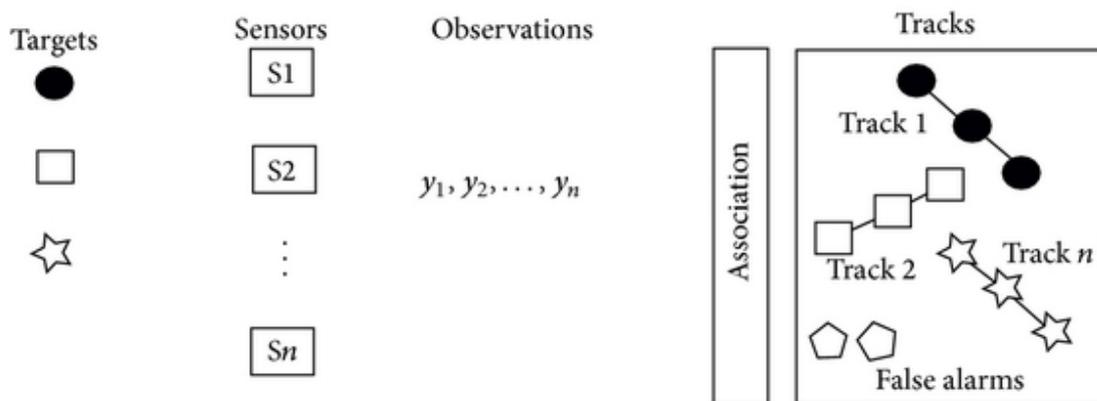


Figura 4.7: Agrupación de datos asignada a un objetivo.

Existen una serie de problemas relacionados con la asociación de datos en entornos con múltiples objetivos cercanos entre sí:

- Cada *sensor* tiene una *velocidad de muestreo* propia.
- El núcleo de procesamiento de datos recibe la información en un intervalo de tiempo discreto.
- Alguna de las observaciones obtenidas por los *sensores* son ruido y otras corresponden a datos del objetivo.
- No se conoce el número de muestras que serán tomadas del objetivo.

La finalidad de la *asociación de datos* es estabilizar el conjunto de observaciones o mediciones de un objetivo en el tiempo. Según D. L. Hall and J. Llinas [HL97] la *asociación de datos* es: «*The process of assign and compute the weights that relates the observations or from one set to the observation of tracks of another set.* ».

Esta fase es clave para el procesamiento de los datos fases posteriores. Cabe destacar que esta fase puede ser realizada durante las otras fases (estimación del estado y toma de decisiones) para pre-procesar los datos de entrada.

Las técnicas más utilizadas para la *asociación de datos* son:

- **Nearest Neighbor (NN) y K-means:** estos algoritmos de clustering definen agrupan los datos y definen una serie de grupos en relación a la distancia entre a ellos. La distancia entre los datos puede ser medida utilizando: **distancia absoluta**, **distancia Euclídea** y **una función estática** para el cálculo de la distancia.
- **Probabilistic Data Association (PDA):** este algoritmo, o método, asigna una probabilidad de asociación a cada hipótesis a partir de una muestra validad del objetivo. Como muestra valida se tiene una muestra que está dentro del área de mediciones previstas para dicho objetivo.
- **Joint Probabilistic Data Association (JPDA) [TEFS80]:** está enfocado en el seguimiento de múltiples objetivos. Es similar a PDA, con la diferencia de que las probabilidades asociadas se calculan utilizando todas las observaciones de todos los objetivos. Así, a diferencia de PDA, JPDA considera diversas hipótesis juntas y las combina.
- **Multiple Hypothesis Test (MHT):** está basado en utilizar dos muestras consecutivas para así obtener una *asociación de datos* con menos error a diferencia de otros algoritmos que obtienen una peor *asociación de datos*. Al contrario que PDA y JPDA, MHT calcula todas las posibles hipótesis y crea posibles hipótesis en cada iteración del algoritmo.
- **Distributed Joint Probabilistic Data Association (JPDA-D) [KCCBS86]:** este método calcula el estado estimado de dos sensores basándose en su conjunto de muestras y sus hipótesis.
- **Modelos gráficos:** este método representa las probabilidades e independencias de los datos representando así la descomposición condicional de la probabilidad conjunta. Puede ser representado mediante una gráfica donde los nodos denotan variables aleatorias, los bordes denotan posibles dependencias entre variables. Existen dos principales modelos gráficos: **redes Bayesianas** y **los campos aleatorios de Markov**.

La siguiente agrupación de técnicas, la de **estimación del estado**, se centra en determinar el estado del objetivo en movimiento dadas unas observaciones o muestras. Estas técnicas también son conocidas como técnicas de rastreo. Son mayormente utilizadas durante la fase de estimación del estado. Esta fase es una fase conocida en los algoritmos de fusión de datos ya que las observaciones de un objetivo pueden venir desde distintos sensores y la meta final es obtener el estado final de dicho objetivo.

La estimación se centra en encontrar el valor del estado vectorial (posición, velocidad, tamaño, etc. . .) que se ajuste lo más posible a las muestras obtenidas. Desde el punto de vista

matemático se busca obtener un conjunto de parámetros que proporcione el mejor ajuste a partir del conjunto de muestras redundantes que se posee.

Estos métodos están englobados en el *nivel 1* del modelo JDL y son divididos en dos grupos:

- **Dinámica lineal y muestreo:** en este caso la estimación tiene una solución estandar. Específicamente, cuando el muestreo y las ecuaciones son lineales entonces el ruido sigue una **distribución gaussiana** y se recomienda utilizar el *filtro de Kalman* como solución.
- **Dinámica no lineal:** no existe una solución analítica para resolver este problema de una manera general. Actualmente no existen técnicas o algoritmos prácticos para resolver este problema.

La mayoría de los métodos de estimación están basados en la teoría del control y utilizan las leyes de la probabilidad para calcular así el estado vectorial a partir de las muestras obtenidas. Los principales y más importante métodos son:

- **Maximum Likelihood (ML) y Maximum Posterior (MP):** la técnica ML está basado en la **teoría probabilística**. Estos métodos son apropiados cuando la variable del estado sigue una distribución de probabilidad desconocida [CBS92]. MP está basado en la teoría Bayesiana y es empleado cuando el parámetro a estimar se obtiene mediante una función de densidad de probabilidad conocida.
- **El filtro Kalman:** es una técnica de estimación creada por R. E. Kalman [Kal60]. Esta técnica estima el estado de un proceso a partir de la formula 4.1 siendo z en el momento k del estado x representado por la formula 4.2 donde $\Phi(k)$ es la matriz de transición de estados, $G(k)$ es la matriz de transición de entrada, $u(k)$ es el vector de entrada, $H(k)$ es la matriz de muestreo y ω y v son las variables Gaussianas aleatorias con matrices de ceros y de covarianza de $Q(k)$ y $R(k)$, respectivamente.

$$x(k + 1) = \Phi(k)x(k) + G(k)u(k) + \omega(k) \quad (4.1)$$

$$z(k) = H(k)x(k) + v(k) \quad (4.2)$$

Esta técnica es utilizada comúnmente para fusionar datos de bajo nivel. Cabe destacar que esta técnica obtiene estimaciones óptimas en sistemas de modelos lineales con ruido Gaussiano.

Otras técnicas y métodos son: el filtrado de partículas distribuida y no distribuida, el filtro de Kalman distribuido (véase Figura 4.8) y métodos basados en la covarianza.

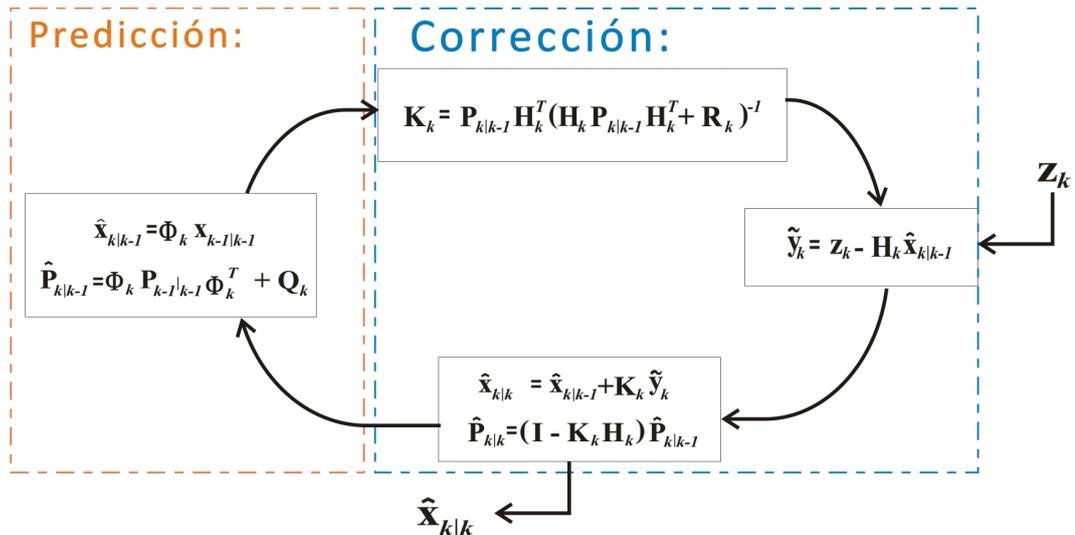


Figura 4.8: Algoritmo recursivo del Filtro de Kalman.

La última de las clasificaciones es la que engloba las técnicas relacionadas con la **toma de decisiones**. Las decisiones suelen ser tomadas en base a un conocimiento previo o en base a la información percibida. Son clasificadas como técnicas de alto nivel y están enfocadas en el nivel 2 y 4 del modelo JDL. Estas técnicas tienen como objetivo tomar decisiones sobre los eventos o actividades que se producen en el entorno basándose en los datos que obtienen de los objetivos detectados. Algunas de las técnicas englobadas en este grupo han sido ya previamente nombradas en este documento, aunque no han sido explicadas, como pueden ser: **métodos Bayesianos** y la **inferencia de Dempster-Shafer**. A continuación se detallarán las técnicas que componen este grupo:

- **Métodos Bayesianos:** esta técnica de inferencia, utiliza lógica probabilística y basa su razonamiento en el conocimiento previo que se tiene de la situación. Este método calcula la variedad de probabilidad de una hipótesis dada (véase Figura 4.9) [Abe07].
- **Inferencia de Dempster-Shafer:** esta técnica es utilizada cuando la información obtenida por todas las fuentes es insuficiente pudiendo así formalizar un conocimiento incompleto o una combinación de evidencias que nos permiten representar la incertidumbre explícitamente (véase Figura 4.10) [Abe07].
- **Razonamiento abductivo:** es un método que elige una hipótesis basándose en el hecho de que, en caso de que sea la hipótesis correcta, definirá la situación observada con mayor precisión. Esta técnica encuentra el ML del sistema dado un conjunto de variables observadas. Esta técnica es más un razonamiento que una técnica en sí por lo que diferentes técnicas como NN o *lógica difusa* pueden ser utilizadas
- **Métodos semánticos:** el uso de métodos semánticos en las técnicas de fusión de decisiones hace que los resultados sean más precisos. La *fusión de información* semántica

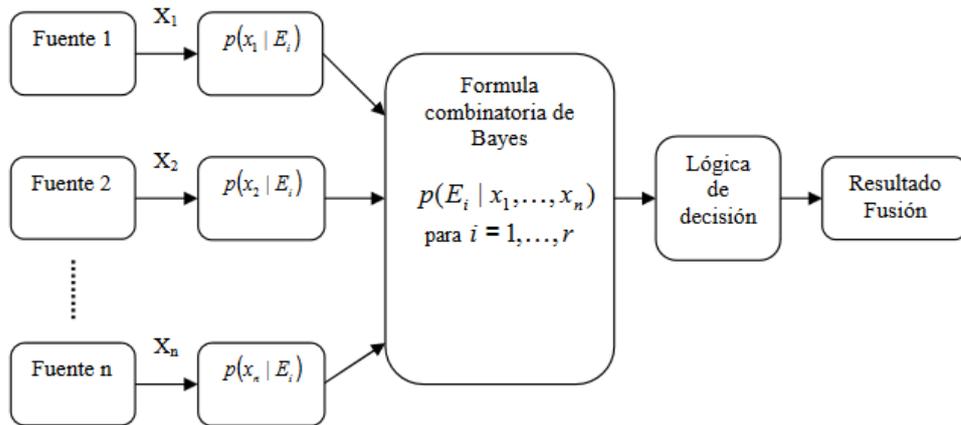


Figura 4.9: Proceso de fusión con la técnica de Bayes.

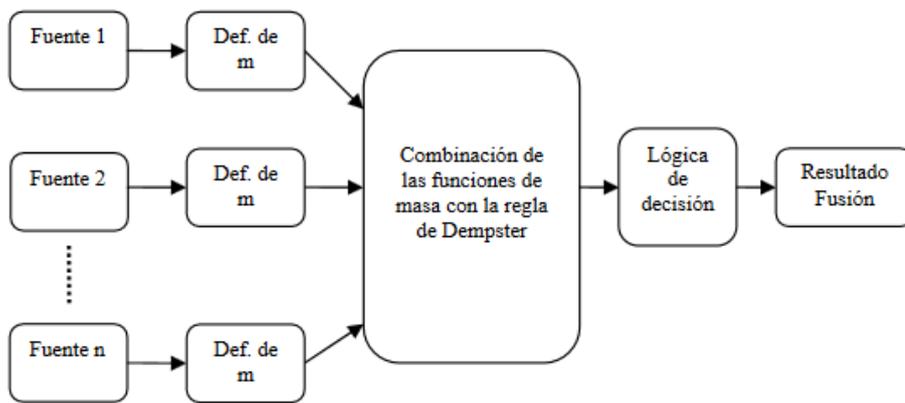


Figura 4.10: Proceso de fusión con la técnica de Dempster-Shafer.

es un esquema en el cual los datos de los sensores en bruto son procesados de tal manera que los nodos sólo intercambian los datos semánticos una vez procesados. Estas técnicas abarcan dos fases: **construcción del conocimiento**, donde se incorpora el conocimiento apropiado a la información y **concordancia de patrones**, donde se fusionan los datos relevantes y se proporciona una interpretación semántica de los datos del sensor.

Hasta aquí un pequeño resumen de todos los tipos de agrupaciones y clasificaciones de técnicas de fusión de información.

Cabe destacar que la obtención de unos datos fiables y útiles es de suma importancia para una buena *percepción* del entorno por parte de los vehículos autónomos, los cuales, llevarán a cabo las tareas de forma más eficiente y segura cuanto mejor entiendan el entorno.

4.1.3 Modelado dinámico del entorno

En la Figura 4.11 se resumen los niveles involucrados en la *percepción y modelado del entorno*. El primer paso a tomar es la obtención, gracias a los *sensores* del vehículo, de los datos del entorno que una vez procesados, mediante técnicas de *fusión de datos*, permiten tener un conocimiento local de lo que rodea al vehículo. Una vez procesados estos datos se utilizan técnicas basadas en el **modelado dinámico del entorno** [LW10] para así obtener distinta información (color, aristas, contornos, etc. . .) de dicho entorno. La transición entre estos dos niveles se denomina **navegación** ya que se centra en definir el entorno de manera que el vehículo pueda moverse sin problemas. Con las características del entorno ya definidas es posible utilizar técnicas que permiten obtener los objetos que **interactúan** con el entorno (señales, peatones, vehículos, etc. . .). Por último y tras un **razonamiento** de los datos obtenidos en los anteriores niveles es posible tomar decisiones y así actuar en consecuencia.

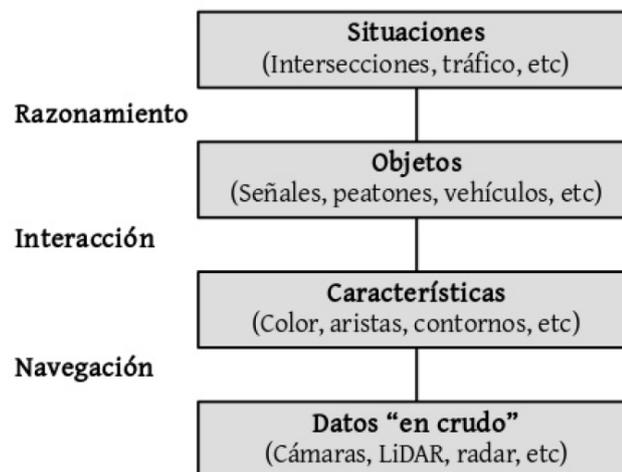


Figura 4.11: Esquema general de un framework de percepción y modelado del entorno.

El *modelado dinámico del entorno* busca definir un modelo que describa de forma matemática el entorno del vehículo autónomo. Existen multitud de técnicas de modelado aunque actualmente una de las más utilizadas es la basada en *sensores* LiDAR. Que sean técnicas basadas en *sensores* LiDAR no significa que no utilicen otros sensores.

Por ejemplo, los sistemas basados únicamente en cámaras de vigilancia añaden complejidad al sistema debido a que es necesario considerar el cambio continuo en las condiciones lumínicas y cuestiones como el motion blur¹³. En este contexto no es posible emplear, por lo tanto, técnicas que tradicionalmente se han usado en vídeo vigilancia [BLG99]. Aun-

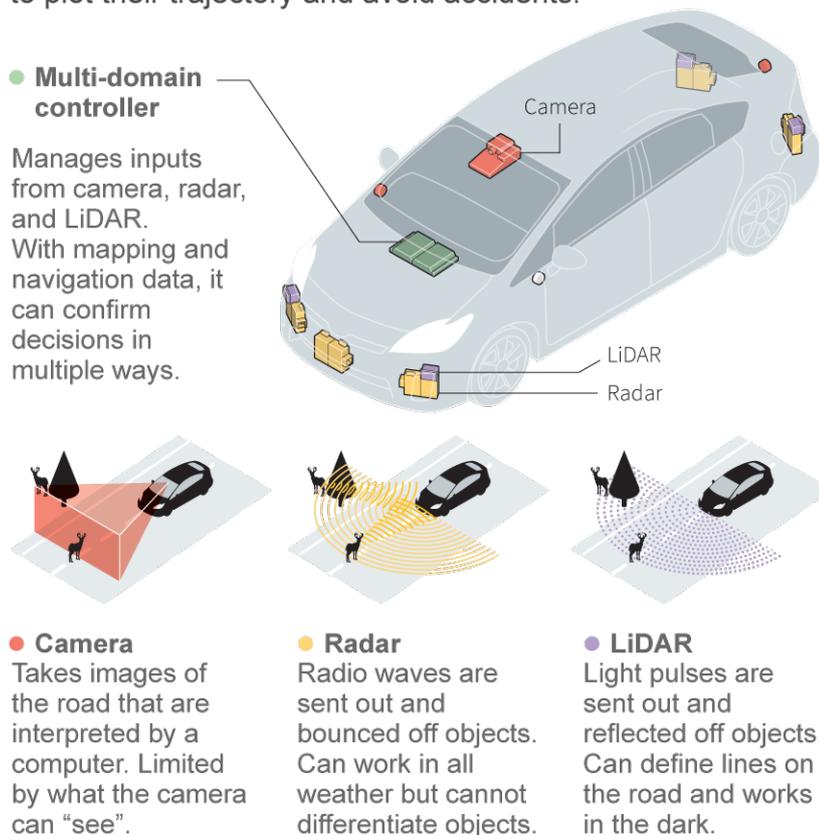
¹³«El desenfoque de movimiento (del inglés motion blur) es el rastro dejado por los objetos en movimiento en una fotografía o en una secuencia de imágenes como una película o una animación. Aparece cuando el objeto siendo grabado cambia su posición durante la captura de un fotograma debido a su velocidad o al movimiento de la cámara.» https://es.wikipedia.org/wiki/Desenfoque_de_movimiento

que en los últimos años se han desarrollado propuestas para abordar este reto tan complejo [MBD00] [BLG07], pero a día de hoy sigue representando una cuestión difícil de resolver en función de la problemática subyacente.

Por eso, las técnicas de modelado del entorno usan sistemas híbridos de *sensores* para obtener redundancia y aumentar así la *fiabilidad* de los datos. Cabe destacar, de nuevo, que no todos los sensores valen para todo tipo de condiciones y necesidades (véase Figura 4.12).

How self-driving cars see the road

Autonomous vehicles rely on a host of sensors to plot their trajectory and avoid accidents.



Source: Delphi

Figura 4.12: Sistema con múltiples sensores y su uso. <http://www.reuters.com/investigates/special-report/autos-driverless/>

Un sensor LiDAR utiliza una técnica de teledetección óptica que se basa un dispositivo laser para así calcular la distancia entre el dispositivo y un objeto o superficie utilizando dicho haz laser. Esta distancia se determina midiendo el tiempo de retraso entre la emisión del pulso y la detección del reflejo de la señal. A nivel computacional es un sistema sencillo dado que los cálculos matemáticos implicados en la medición tienen una complejidad baja a nivel de computo.

Los sistemas que utilizan sensores LiDAR son, por lo general, sistemas *híbridos* e implican distintos dispositivos como: cámaras, GPS, Inertial Navigation System (INS)¹⁴ y sensores LiDAR.

Dependiendo de donde este montado el sistema LiDAR este puede ser clasificado en dos tipos de sistemas. Por un lado tenemos los **aerotransportados** que son sistemas LiDAR instalados en dispositivos capaces de sobrevolar un área. Dado que estos sistemas son utilizados desde gran altitud los pulsos son emitidos hacia el suelo y son capturados por el sistema en movimiento. Este tipo de sistemas está a su vez dividido en dos tipos: **topográficos** y **batimétricos**.

Los sistemas **topográficos** son utilizados para obtener modelos de la superficie terrestre que son utilizados en distintas aplicaciones como: silvicultura, hidrología, geomorfología, planificación urbana, ecología del paisaje, ingeniería costera, evaluaciones de relevamiento topográfico y cálculos volumétricos.

El sistema **batimétrico**, al igual que el *topográfico*, es utilizado para obtener modelos tridimensionales de una superficie, pero a diferencia del *topográfico* este sistema es capaz de penetrar superficies líquidas y es capaz de obtener un modelo completo de una superficie líquida ya que tiene la capacidad de recopilar simultáneamente la profundidad del líquido y la distancia a la superficie. Este sistema utiliza dos dispositivos laser, uno de ellos de luz infrarroja, mide la distancia a la superficie mientras que el otro dispositivo es capaz de determinar la profundidad del líquido. Estos sistemas *batimétricos* son utilizados para obtener información de las líneas costeras, puertos y playas, también son utilizados para localizar y posicionar objetos en el fondo marino.

Por otro lado tenemos los sistemas LiDAR **terrestres** siendo estos agrupados en sistemas **móviles** o **estáticos**. Los sistemas *terrestres* son más precisos que los *aerotransportados* y por lo general son utilizados en aplicaciones de relevamiento topográfico de carreteras y vías férreas, e incluso crear modelos de ciudades en 3D para espacios en el exterior y en el interior, son algunos de los ejemplos.

Los sistemas **móviles** son capaces de generar un conjunto de puntos llamado **nube de puntos** desde una plataforma en movimiento. Estos sistemas pueden ser montados en vehículos, trenes e incluso barcos. Los datos obtenidos pueden ser utilizados para analizar carreteras, ubicar alambres aerotransportados que se superpongan, postes de luz y rótulos de carretera cerca de carreteras o vías férreas, entre otras muchas aplicaciones.

Entrando más en detalle, en el caso de un sistema *movil* montado en un coche los sensores LiDAR están posicionados sobre la zona superior permitiendo así al sistema obtener una visión total, de 360 grados alrededor del vehículo, del entorno. Los sensores LiDAR recogen distintas muestras de las distintas profundidades a las que rebota el impulso laser. Detallando

¹⁴https://es.wikipedia.org/wiki/Sistema_de_navegación_inercial

más el proceso, los sensores LiDAR envían un pulso laser que se ve reflejado por distintas superficies a lo largo de su trayectoria por lo que es comprensible que este pulso laser pueda regresar al sensor repetidas veces. Cuando el pulso laser atraviesa la superficie y continua su camino es cuando se obtienen distintas mediciones. Cabe mencionar que la primera recepción del pulso laser es la más importante para el sistema ya que determina el primer objeto en la trayectoria siendo las posteriores medidas no menos significativas ya que pueden significar estructuras de vegetación o superficies translucidas. En el ámbito de los sistemas *móviles* tanto la primera medida como la última son importantes para el sistema ya que determinan una zona de incertidumbre que tienen que ser tomada en cuenta. Continuando con los sistemas montados sobre vehículos, estos son capaces de obtener las distintas medidas y, mediante el ordenador de a bordo, pueden procesarlas junto a la información obtenida mediante otros sensores para así generar modelos 3D del entorno del vehículo. Estos sistemas son capaces de obtener 44 muestras por cada impulso aunque la media ronda las 20 o 30 muestras, cuanto más muestreo mejor resolución del entorno.

Los sistemas **estáticos** obtienen su *nube de puntos* desde una posición estática ya que suelen estar montados sobre trípodes. Por lo general estos sistemas son totalmente portátiles y están basados en un sensor *LiDAR* y en un sistema de imágenes. Este tipo de sistemas tienen la capacidad de obtener mediciones tanto en exteriores como en interiores. Las aplicaciones más comunes para este tipo de sistemas son la ingeniería, minería, topografía y la arqueología.

Los sensores LiDAR tienen un alcance medio de unos 70 a 100 metros, aunque algunos sensores son capaces de alcanzar los 200 metros. Estos sensores tienen una precisión del orden de ± 2 cm lo que hace que el sensor no reconozca pequeños baches o piedras en el entorno. Cabe destacar que debido a su posicionamiento en los vehículos los sensores LiDAR son incapaces de medir obstáculos que estén próximos al vehículo. La precisión de los sensores LiDAR es independiente de las condiciones de luz del entorno. Un punto negativo de los sensores LiDAR es su elevado coste, por ejemplo, los sensores utilizados por la compañía **Google** en sus vehículos rondan los 80kg de peso con un coste de unos 70.000\$, aunque la empresa **Quanergy**¹⁵ actualmente tiene un sensor con un rango de ~ 150 metros a un precio de \$250 con una precisión aceptable (véase Figura 4.13).

Como ya se ha mencionado, estos dispositivos son capaces de producir mediciones exactas que pueden ser posicionadas en un entorno 3D (x, z, y). Cabe destacar que el barrido con un dispositivo LiDAR genera un conjunto de puntos, o una *nube de puntos*, que puede ser visualizado gracias a un SW determinado.

Los puntos que forman la *nube* poseen más información aparte de sus coordenadas tridimensionales. Cada muestra generada por el LiDAR puede registrar a su vez: intensidad, número de recepción, cantidad total de recepciones, valores de clasificación de punto, valo-

¹⁵<http://quanergy.com/>

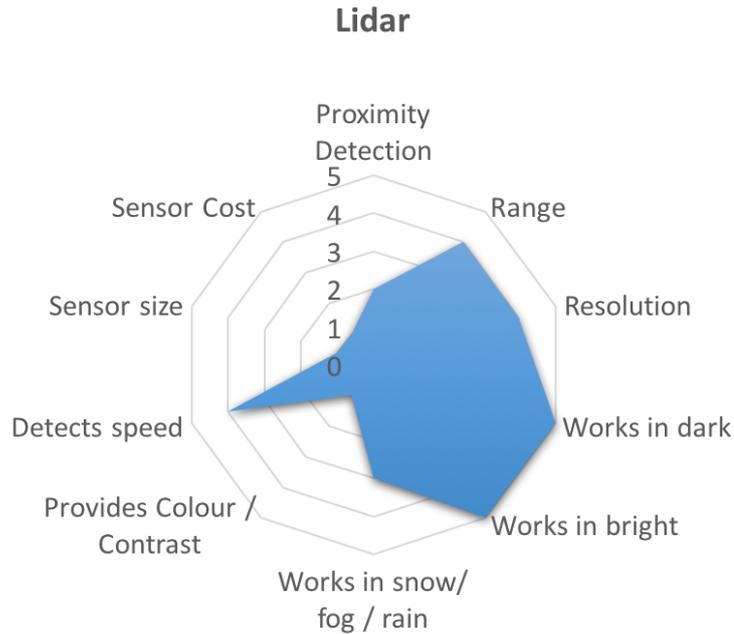


Figura 4.13: Atributos de los sensores LiDAR. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

res Red, Green and Blue (RGB), tiempo del GPS, ángulo de escaneo y dirección de escaneo. A continuación se detallaran algunos de los atributos:

- **Intensidad:** la fuerza del pulso recibido. Gracias a este dato podemos conocer la capacidad de reflexión y absorción del objeto. Cabe destacar que el valor de intensidad puede ser utilizado para generar una imagen en escala de grises en sistemas que no dispongan de camaras (véase Figura 4.14).
- **Número de recepción:** indica en que posición está el rebote recibido. Esta información determina el orden de los objetos.
- **Número de recepciones:** el número total de impulsos del mismo tipo recibidos por el sensor. Este atributo está muy ligado al *número de recepción* (véase Figura 4.15).
- **Clasificación de puntos:** los puntos LiDAR pueden ser clasificados gracias a sus atributos en varias categorías que incluyen suelo o terreno desnudo, agua, arboles, etc...
- **Tiempo del GPS:** la fecha y posición del GPS a la hora de tomar la medición.
- **Angulo de escaneo:** en alguno de emisión del pulso laser. Gracias a este atributo es posible posicionar de forma tridimensional el punto.

Gracias a estos atributos es posible obtener una *nube de puntos* que aporte información al sistema autónomo posibilitando así la creación de un modelo tridimensional del entorno que puede ser utilizado, entre otras cosas, para fijar puntos de referencia útiles para la nave-



Figura 4.14: Imágenes generadas a partir de la intensidad del pulso laser. <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>

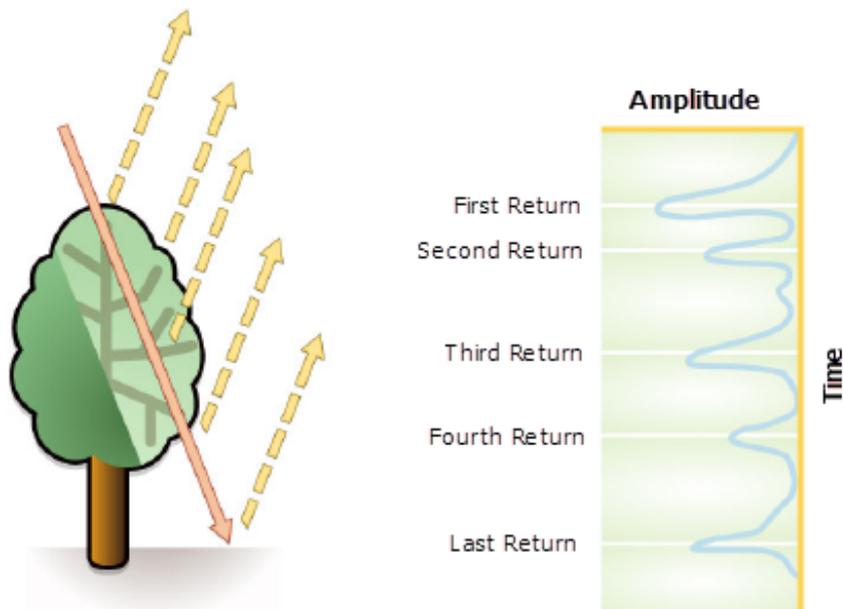


Figura 4.15: Explicación de las múltiples recepciones del pulso laser y su distinta intensidad. <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/what-is-intensity-data-.htm>

gación autónoma del vehículo [ADIEM03] o incluso detectar obstáculos en movimiento que pueden suponer un peligro para el vehículo, como por ejemplo un peatón que se cruza en la trayectoria del vehículo autónomo [JZea14].

Otro dispositivos o *sensores* importantes en los sistemas basados en LiDAR son las **cámaras de vídeo a color** que pueden ser utilizadas como complemento en los sistemas LiDAR o como sensor principal del sistema en caso de prescindir de los sensores LiDAR. Estos sistemas montan una serie de cámaras de distinto tipo en distintas localizaciones del vehículo para así capturar toda la información necesaria para recrear el entorno. Al contrario que los sensores LiDAR las *cámaras de vídeo a color* son capaces de reconocer colores por lo que son idóneas para reconocer señales de tráfico y las líneas de la carretera sin necesidad de un elevado coste del dispositivo dado que no se requiere una gran definición para reconocer dichos patrones, pero en caso de querer reconocer otros patrones como peatones, ciclistas o luces de freno o semáforos es necesario emplear cámaras de alta definición capaces de captar todos los matices, por último, en caso de querer recrear el entorno en un modelo tridimensional es necesario utilizar cámaras estereoscópicas capaces de capturar dos o tres imágenes simultáneas, dependiendo del modelo, desde distintos ángulos para así generar un modelo tridimensional simulando la visión del ser humano (véase Figura 4.16).

No todos los fabricantes de coches optan por usar sensores de tipo LiDAR y defienden el uso de *cámaras de vídeo*. Al contrario que *Google* y **Ford**, las empresas **Mercedes-Benz** y **Volvo** han optado por utilizar cámaras como sensor principal de sus sistemas.

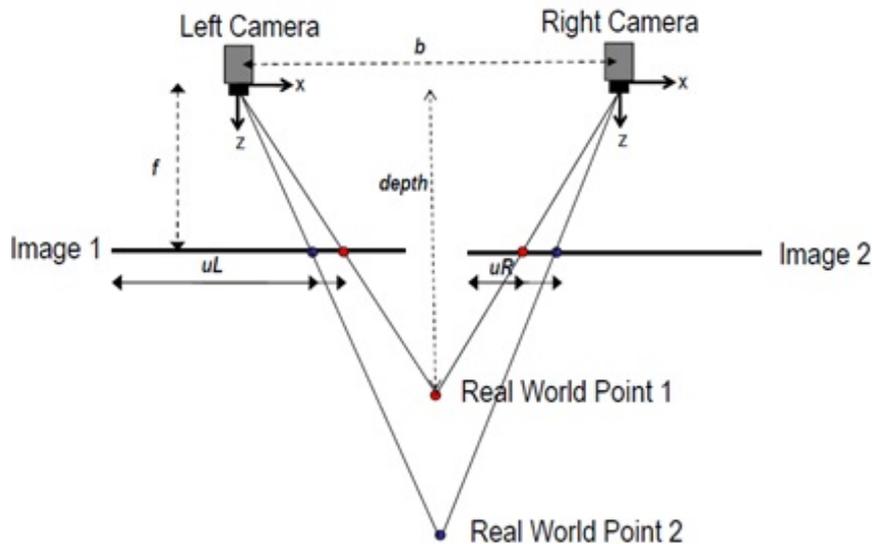


Figura 4.16: Vista esquemática de un sistema de cámara estéreo 3D. <http://www.depthbiomechanics.co.uk/?p=102>

Estas cámaras son montadas sobre el frontal del coche y en los laterales en caso de querer obtener una visión estereoscópica del entorno. En caso de no disponer de un sensor LIDAR que nos otorga información de los laterales del coche es necesario tres cámaras más, dos en los laterales del coche y otra en la parte trasera, para así obtener una visión de 360 grados del entorno. Las *cámaras de vídeo* son capaces de reconocer objetos a una distancia de 150 metros.

Debido a su naturaleza estos sensores pueden sufrir problemas de visibilidad según la climatología y la cantidad de luz disponible, aunque con el uso de un buen SW de reconocimiento de imágenes es posible detectar un mayor número de elementos al contrario que los sensores de tipo LIDAR (véase Figura 4.17).

Los **radars** son sensores activos que reciben la una antena del dispositivo una onda de radio emitida que rebota en un objeto, de modo que dependiendo del tiempo de retorno pueden medir así distancias. Este dispositivo, al igual que las *cámaras de vídeo*, está situado en el frontal del coche, para así, medir distancias hacia el frente del vehículo. También, y dependiendo del prototipo de vehículo autónomo, es posible que se disponga de dos radares más en las esquinas traseras a modo de retrovisores para localizar vehículos en los carriles contiguos o para comprobar si se tiene un coche detrás. Por último, y en caso de no tener un sensor LIDAR que otorgue un conocimiento del entorno de 360 grados, dos radares pueden ser montados en los laterales del vehículo.

Estos sensores tienen un alcance de 200 metros, no tienen problemas de visibilidad al contrario que las *cámaras de vídeo* aunque pueden dar mediciones poco fiables ante ciertos

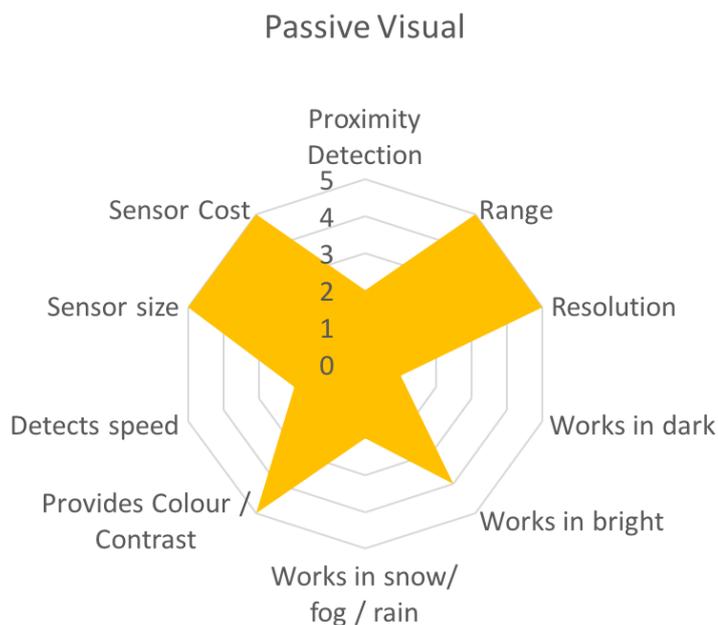


Figura 4.17: Atributos de las cámaras de vídeo a color. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

elementos del entorno como objetos con una superficie ínfima como son las rejillas o cuando coinciden direcciones diagonales como por ejemplo en una curva de radio pequeño, pues el radar sigue emitiendo señal en línea recta y puede confundirse (véase Figura 4.18).

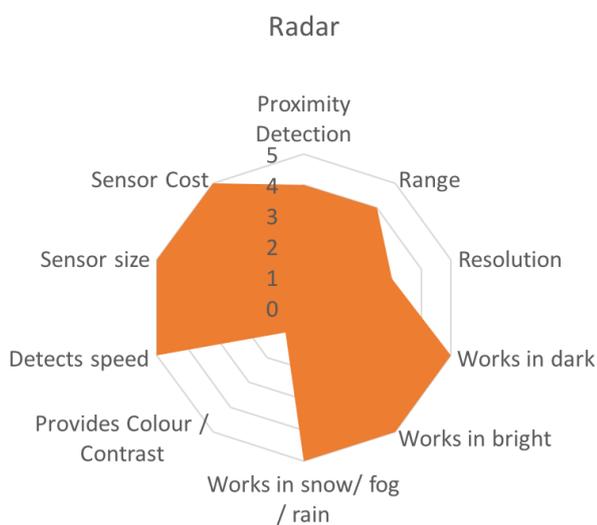


Figura 4.18: Atributos de los radares. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

Por último, los **sensores de ultrasonidos** tienen un rango de alcance entre los 30 cm y

los 5 metros y son utilizados para cortas distancias. Estos sensores suelen ir colocados en los paragolpes frontales, laterales y traseros de los vehículos y complementan a los *radars* (véase Figura 4.19).

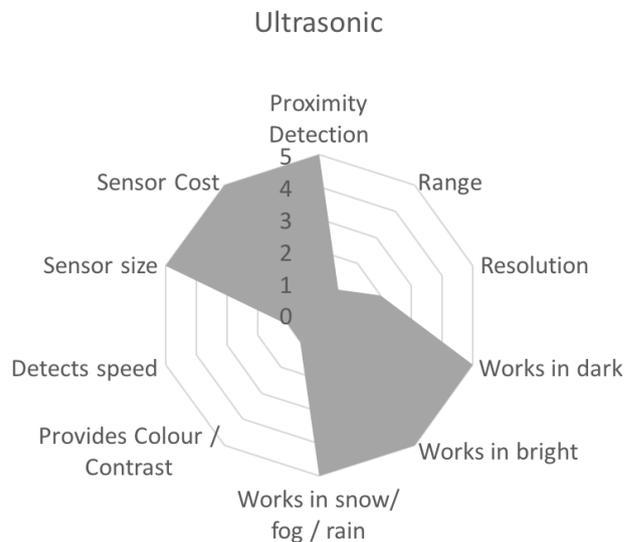


Figura 4.19: Atributos de los sensores de ultrasonido. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

4.2 Localización y construcción de mapas

4.2.1 Localización

Otro factor importante aparte del modelado del entorno mediante sensores es la obtención de la ubicación del vehículo autónomo en el entorno, lo que comúnmente se denomina *localización*.

Desde el punto de vista de la conducción autónoma, la localización de vehículos se suele traducir en la localización de una carretera por la que han de circular o en estimar la localización relativa del vehículo en base a referencias del mapa.

Si se pretende controlar un vehículo de forma autónoma es necesario que este conozca su **posición global** y su **posición local**, para así, poder desplazarse hacia su objetivo evitando obstáculos estáticos y dinámicos. Para este fin existen distintos tipos de dispositivos capaces de localizar al vehículo global y localmente. Tradicionalmente, los dispositivos GPS se utilizan para obtener la posición absoluta de un vehículo [AGU12] ya que cuando se habla de localización global no existe ningún dispositivo equiparable al **GPS**. Es, como su nombre indica, un *sistema de posicionamiento global*. Este dispositivo está basado en un receptor que recibe la señal de una serie de satélites que orbitan alrededor de la tierra y que mediante algoritmos de triangulación es capaz de obtener su posición global con cierto margen de error.

El sistema conocido como GPS fue desarrollado e instalado para uso propio del *Departamento de Defensa de los Estados Unidos*. El Departamento de Defensa desplegó 24 satélites que orbitan alrededor de la Tierra a 20.200 km de altura y que describen una trayectoria sincronizada para así cubrir la superficie de la Tierra en su totalidad. Este sistema fue posteriormente liberado para uso civil pero no sin incorporar un sistema de error, la denominada **disponibilidad selectiva**, sistema por el cual la posición adquirida por el dispositivo GPS podría variar entorno a unos 15 - 100 metros. Cabe destacar que dicha *disponibilidad selectiva* fue eliminada el 2 de Mayo del 2000, aunque no por ello los dispositivos GPS están exentos de tener un pequeño margen de error que, en el mejor de los casos¹⁶ puede obtenerse un margen de error inferior a 2,5 metros el 95 % del tiempo. Estos errores se suelen deber al propio bloqueo de la señal o incluso a la reflexión de la misma en edificios o árboles [RSS11] (véase Figura 4.21).

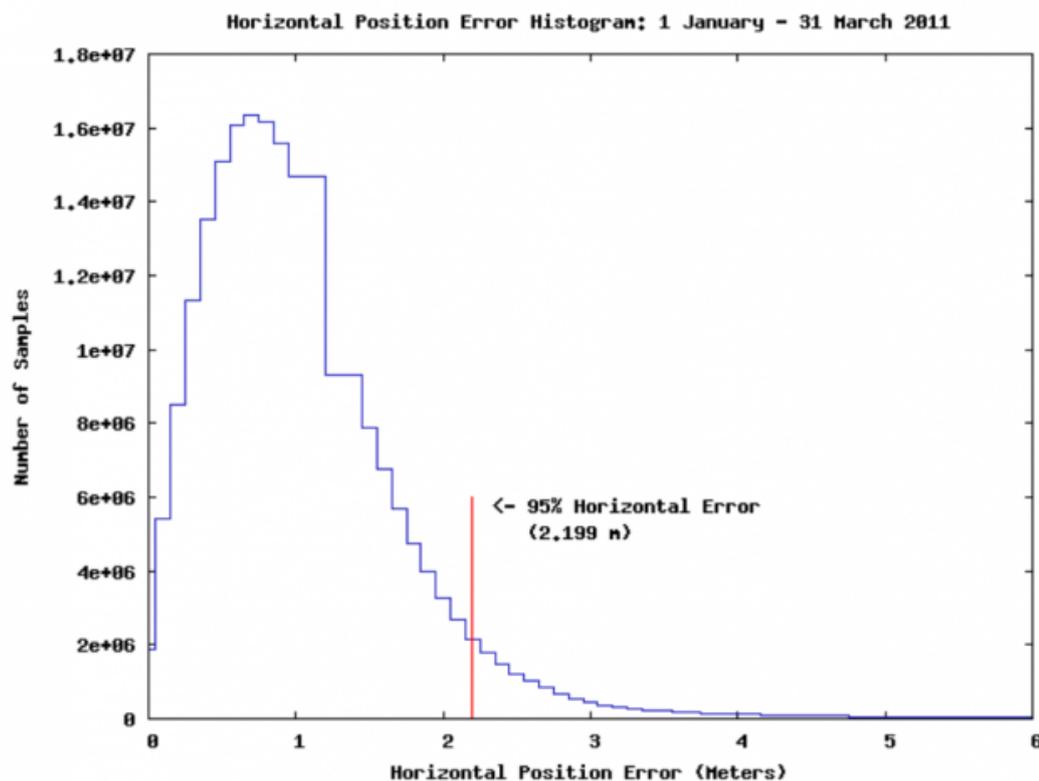


Figura 4.20: Histograma de la frecuencia de error de localización en el tiempo. *En vertical el número de muestras tomadas. En horizontal el margen de error horizontal del GPS. Muestras tomadas entre el 1 de Enero y el 31 de Marzo del 2011.* <http://www.gps.gov/>

Otro país en desarrollar un sistema similar fue la antigua Unión Soviética que desarrolló el programa GLONASS¹⁷. Actualmente está controlado por la Federación Rusa.

¹⁶Entre siete y nueve satélites disponibles en una geometría adecuada (dispersos)

¹⁷<https://es.wikipedia.org/wiki/GLONASS>

La Unión Europea también ha desarrollado un sistema similar llamado Galileo¹⁸. El programa entró en operatividad en Diciembre del 2016.

La República Popular China está implementando su propio sistema de navegación, el denominado Beidou¹⁹.

Actualmente, y debido a sus años de funcionamiento y fiabilidad, el GPS es el mejor sistema para un posicionamiento global de un vehículo autónomo, pero esto no quiere decir que sea infalible dado que su margen de error puede suponer la diferencia entre estar conduciendo por el carril derecho o el izquierdo en una carretera, por ello es necesario realizar correcciones. Cabe destacar que dicho margen de error puede verse incrementado en situaciones no optimas y que depende de la línea de visión que se tenga de los satélites por lo que, por ejemplo, un bosque frondoso o un túnel podrían obstaculizar la medición.

Para que un sistema de *localización* sea **optimo** es necesario que sea *eficiente, robusto* y en *tiempo real*²⁰.

Los sistemas mono-sensor basados en GPS carecen de estos atributos debido al margen de error que este tipo de dispositivos tiene ya que estan basados en una referencia externa, y para ello, como se ha visto con anterioridad, es necesario utilizar un sistema multi-sensor para así corregir los errores de posicionamiento del GPS.

Para que un vehículo autónomo sea capaz de desplazarse a grandes distancia utilizando un sistema multi-sensor son necesarios dos pasos: primero, localizar el vehículo globalmente gracias al dispositivo GPS y segundo, localizar al vehículo localmente para así corregir la localización del vehículo.

Actualmente existe un dispositivo capaz de obtener, mediante un conjunto de sensores, la aceleración, en cada uno de los tres ejes de movimiento X, Y y Z. Este dispositivo, o sistema, es el denominado **Inertial Navigation System (INS)** [RKB09].

El **INS** está formado por un conjunto de sensores de **movimiento** o *acelerómetros*, una serie de sensores de **rotación** o *giroscopios* y un pequeño computador capaz de computar la información para así obtener la posición y velocidad del vehículo. Este sistema es capaz de estimar, con un pequeño margen de error, la posición, orientación y velocidad del vehículo sin una referencia externa, lo que lo hace optimo para un buen sistema de *localización* aunque no es infalible por sí solo.

Estos sistemas utilizan como base la denominada **navegación por estima** que es un método matemático que utiliza como base formulas trigonométricas para así inferir la ubicación actual de un objeto realizando cálculos basados en el rumbo y la velocidad del vehículo a lo largo de un periodo de tiempo.

¹⁸<https://es.wikipedia.org/wiki/Sistema.de.navegación.Galileo>

¹⁹<https://es.wikipedia.org/wiki/Beidou>

²⁰<https://mappingignorance.org/2014/04/07/one-way-googles-cars-localize/>

Los INS tienen una gran desventaja ya que la parte desde una posición inicial y va calculando constantemente su posición a partir de los datos obtenidos por los sensores por lo que un pequeño error se irá acumulando (error acumulativo), incluyendo el error de Abbe²¹. Este error incrementa con el tiempo y hace que la incertidumbre entre la posición estimada y la posición real aumente por eso es necesario realizar correcciones en la navegación.

Estos INS suelen utilizarse en navegación marítima, aeronaves, misiles y naves espaciales además de en vehículos autónomos, ya que son capaces de detectar un cambio en la posición geográfica, cambios en la velocidad y cambios en la orientación del vehículo.

Un sistema de *localización* óptimo aunaría las fortalezas de los dispositivos GPS y INS para así paliar las debilidades de estos. Los sistemas conjuntos de GPS y INS están basados, a grandes rasgos, en la corrección de la *navegación por estima* obtenida por el INS gracias a los datos obtenidos por el GPS. Estos datos son tratados mediante un algoritmo de estimación de series temporales basado en la estadística bayesiana denominado **el filtro de Kalman** [WDK07] que es utilizado para combinar información con cierto grado de incertidumbre ya que es capaz de identificar datos ocultos o poco fiables en sistemas dinámicos. Este algoritmo es un algoritmo recursivo que permite procesar los datos a medida que llegan por lo que lo hace muy útil para procesar los datos del sistema de *localización* además de que tiene unos reducidos requerimientos computacionales.

Existen distintos proyectos o propuestas basados en la unión de los GPS y INS [Cas12] o en su defecto un dispositivo Inertial Measurement Unit (IMU) [KL16]. Por lo general, estas propuestas utilizan dos dispositivos GPS posicionados a un metro de distancia para obtener la posición global de forma redundante lo que añade información adicional al algoritmo para así poder estimar de forma más eficiente la posición del vehículo.

Gracias a estos dispositivos y algoritmos es posible localizar de manera eficiente el vehículo, pero incluso así sería toda una proeza alcanzar el destino fijado sin poner en riesgo al vehículo o a los peatones. Para conseguir llegar al destino correcto sin poner en peligro a nadie es necesario poder conocer el camino y reconocer cambios en el entorno.

4.2.2 Construcción de mapas

La *construcción de mapas* en base a los datos obtenidos por múltiples sensores es otra cuestión que se ha tratado por diversos autores en la literatura existente [DPDW98]. En este contexto es importante destacar que los vehículos autónomos pueden navegar considerando la existencia o inexistencia de mapas conocidos. Un ejemplo representativo de este último caso, en el que no se pueden obtener los mapas del terreno con antelación, es el de la exploración de Marte por parte de robots autónomos. Estos casos, donde es necesario abordar simultáneamente la localización del vehículo y el modelado del entorno, se encuadran bajo el término **Simultaneous Localization And Mapping (SLAM)** [Cso98] [GDB00].

²¹<http://www.e-medida.es/documentos/Numero-8/sabias-que-error-de-abbe>

SLAM busca resolver el problema que surge ante lo desconocido, ante una situación donde el vehículo no conoce su posición ni su entorno y que gracias al uso de esta técnica y a los sensores y actuadores del vehículo este sea capaz de generar un mapa de su entorno y así determinar su propia localización.

Esta tarea no es fácil ya que los sistemas sensoriales no son infalibles, los errores en las mediciones y los modelos empleados añaden ruido, a lo que se suma la complejidad de los entornos a los que se enfrenta el vehículo hacen que obtener la localización y el mapeado del entorno sea una tarea compleja.

Cabe destacar que los planteamientos o soluciones más exitosas de SLAM estén basadas en la utilización de técnicas probabilísticas. El uso de estas técnicas consigue hacer frente a todo ese ruido generado por los sensores y al nivel de incertidumbre que añaden al proceso. Este tipo de algoritmos, al igual que alguna de las técnicas de *fusión de datos*, tiene su base en el **teorema de Bayes**.

4.2.3 SLAM y el teorema de Bayes

Uno de los mayores obstáculos a la hora de modelar un entorno desconocido, aunque ya se ha mencionado anteriormente, es la incertidumbre que genera el uso de los sensores y los modelos empleados. En un mundo perfecto, con sensores y modelos perfectos se podría construir un modelo perfecto del entorno, pero actualmente este no es ese caso. Por eso, y para solucionar este problema, se utiliza un teorema probabilístico, en este caso el *teorema de Bayes*, tomando así la posición del robot y la de los elementos que lo rodean como variables aleatorias, por lo que los algoritmos existentes modelan de manera probabilista el entorno y la posición del vehículo y mediante métodos de inferencia determinan aquella configuración que se asemeja más a lo real teniendo en cuenta las mediciones que se van obteniendo.

Según el *teorema de Bayes*, para dos variables aleatorias x y d se establece la formula 4.3.

$$p(x|d) = \frac{p(x)d p(x)}{p(d)} \quad (4.3)$$

Suponiendo que se quiere obtener información de x basándonos en la información de otra variable d entonces, según la fórmula 4.3 sólo se tienen que multiplicar dos términos.

- El modelo generativo $p(d|x)$ que define la probabilidad de obtener la media d bajo la hipótesis expresada por el estado x .
- El grado de confianza que damos a que x sea precisamente el caso antes de recibir los datos, $p(x)$.

Se tiene que tener en cuenta de que el denominador de la ecuación no depende de la variable que se pretende estimar, x , por lo que $p(d)$ suele escribirse como un factor de nor-

malización.

Uno de los **algoritmos** más utilizados en SLAM es el **Filtro de Kalman Extendido (EKF)**²² ya que es uno de los algoritmos que mejores resultados proporciona en la práctica. Esta solución está categorizada como **SLAM-EKF** y fue introducida por *Randall Smith, Matthew Self* y *Peter Cheeseman* a finales de los años 80 [RSC90].

Este algoritmo requiere un mapa donde las entidades que lo componen sean fácilmente parametrizables, es decir, los elementos que lo componen deben poder ser descritos utilizando un conjunto de parámetros que puedan ser utilizados por el sistema. Por su propia naturaleza, el SLAM-EKF requiere disponer de un mapa en el cual las entidades que lo componen sean fácilmente parametrizables. Esto es, los elementos que componen el mapa deben poder ser descritos utilizando un conjunto de parámetros que encajen de forma sencilla en el vector de estado del sistema.

Las principales desventajas de utilizar el algoritmo EKF son²³:

- La suposición de una *distribución Gaussiana* para el estado del sistema puede no corresponderse con la realidad añadiendo así, con el tiempo, un error a los valores.
- Los problemas de consistencia en los datos lo hacen un algoritmo poco fiable haciendo que el nivel de confianza de la estimación no se corresponda con el auténtico error cometido.
- Computacionalmente está limitado al número de objetos que forman el mapa dado que su coste crece cuadráticamente con dicho número.
- No es tan sencillo describir los objetos a partir de primitivas que puedan ser asimiladas por el algoritmo.
- Se debe definir un método robusto capaz de asociar los datos observados con los elementos del mapa.

Aun así es uno de los algoritmos más utilizados en soluciones SLAM, dado que a pesar de sus inconvenientes presenta interesantes propiedades:

- La definición del entorno como entidades geométricas descriptibles se asemeja a nivel antropomórfico a cómo percibe el ser humano el mundo real.
- Su uso extendido implica que las ventajas e inconvenientes son ampliamente conocidos y que su estructura está bien definida. Se han planteando múltiples soluciones para paliar sus desventajas.
- Al mantenerse la matriz de covarianzas del sistema completa, es capaz de cerrar bucles exitosamente.

²²https://es.wikipedia.org/wiki/Filtro_de_Kalman_Extendido

²³[https://es.wikipedia.org/wiki/SLAM_\(robótica\)](https://es.wikipedia.org/wiki/SLAM_(robótica))

Otra técnica o algoritmo utilizado en SLAM es el uso de **Occupancy Grid Mapping (OGM)** introducidos por Hans Moravec y Alberto Elfes [Elf90] [ME85] a mediados de los años 80. Esta técnica se basa en discretizar el espacio haciendo «*celdillas*» para así tener un mejor control del entorno. Una vez se ha dividido el espacio en estas *celdas* entonces se clasifican como ocupadas o vacías a partir de un determinado nivel de confianza o probabilidad (véase Figura 4.21) [LR14].

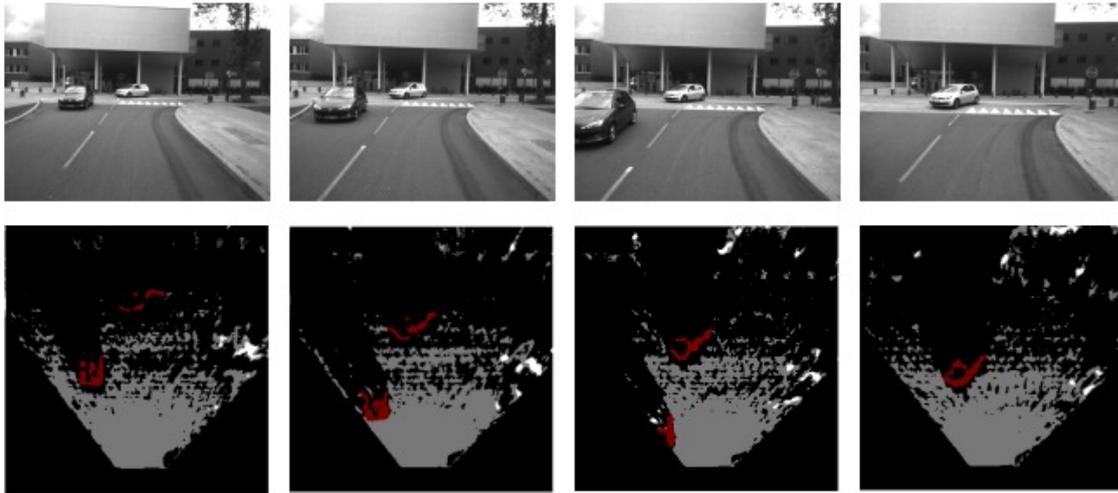


Figura 4.21: Resultados experimentales del OGM: objetos móviles independientes (rojos), áreas estáticas ocluidas (blancas), áreas libres (gris) y áreas no detectadas (negro). *Source: Occupancy Grid Mapping in Urban Environments from a Moving On-Board Stereo-Vision System*

Para utilizar esta técnica es necesario tener un sistema de localización que obtenga la posición del vehículo, ya que esta técnica parte de la hipótesis de que la posición del vehículo es conocida. La precisión del mapa depende del tamaño de la *celda* que se defina por lo que este tamaño se puede ajustar para permitir al sistema de localización tener cierto margen de error a lo largo del tiempo. Aun así, una de las mayores desventajas de este método es la pérdida de potencia que se deriva de no tener en cuenta la incertidumbre asociada a la posición del vehículo, lo cual origina que su capacidad para cerrar bucles correctamente se vea mermada.

Esta técnica posee ciertas ventajas:

- Es un algoritmo robusto y de fácil implementación.
- No está basado en la naturaleza geométrica de los elementos por lo que no hace suposiciones sobre ella.
- Al distinguir entre espacios ocupados y vacíos es muy popular en tareas de navegación dado que permite calcular trayectorias empleando métodos convencionales.

- Permite regular el tamaño de la *celda* por lo que la descripción del entorno puede ser más precisa a costa del rendimiento computacional.
- Permite una extensión conceptualmente simple al espacio tridimensional.

A lo largo de los años han aparecido variantes de este método que mejoran sustancialmente algunos aspectos. Por un lado tenemos los **coverage maps**, que definen una probabilidad de ocupación de la *celdilla* por lo que se ajustan más a casos en los que la *celda* está parcialmente ocupada. Otro caso es el **DP-SLAM** que utiliza el **filtro de partículas** para estimar cual de los mapas generados se ajusta más a la realidad.

Otra técnica utilizada está basada en la **solución factorizada del filtro de Bayes** como plantea Michael Montemerlo en su tesis [Mon03] donde realiza un detallado análisis de la formulación Bayesiana enfocado al problema SLAM llegando a las siguientes conclusiones: Si el recorrido del vehículo es plenamente conocido, al igual que se presupone con la técnica OGM, entonces la estimación del mapa se solucionaría casi de inmediato. Dado que no existiría la incertidumbre asociada a la posición del sensor entonces la estimación de los objetos que compondrían el mapa no estarían relacionados y sería independiente, por lo que, las correlaciones entre objetos serían nulas.

Además de estas técnicas, que son las más usadas, existen multitud de técnicas utilizadas en SLAM que pueden ser englobadas según en que estén centradas: *mapeado, sensores, modelos cinemáticas, objetos múltiples, objetos en movimiento, cierre del bucle y exploración*, aunque en este documento no se detallan todas las existentes.

Centrándonos en las técnicas de **mapeado** se pueden encontrar algoritmos centrados en la definición de mapas topológicos que se centran en una representación del entorno que captura la conectividad del entorno en vez de en crear un mapa geométricamente preciso. Los enfoques topológicos de SLAM se han utilizado para reforzar la consistencia global en algoritmos métricos SLAM [CN08]

Por otro lado existe la técnica OGM que, como se ha definido, se utiliza para crear mapas de cuadrícula.

Otro enfoque es la simplificación del problema de cartografía mediante el uso de mapas altamente detallados recogidos con antelación. Esta simplificación de la cartográfica simplifica también el problema SLAM reduciéndolo a una tarea sencilla de localización permitiendo que los objetos en movimiento actualicen el mapa en tiempo de ejecución.

Los nuevos **sensores** han impulsado la creación de nuevos algoritmos utilizados en SLAM ya que siempre ha estado muy ligado al uso de varios tipos diferentes de sensores [MB13]. Estos nuevos sensores dan pie a algoritmos más acordes para los datos que aportan. Por un lado, los sistemas basados en sensores de presión requieren de un fuerte modelo para compensar el uso de SLAM pero en contrapartida, los sistemas más visuales, que proporcionan muchos más detalles del entorno, hacen que a veces, la inferencia de SLAM sea innecesaria

ya que los datos aportados por los sensores pueden alinearse con facilidad y sin ambigüedad. Pero cabe destacar que la mayoría de los casos prácticos de SLAM se encuentran en un punto intermedio entre estos dos ejemplos.

A lo largo del tiempo han aparecido distintas variantes de SLAM debido al número de sensores existentes. En 2005 se empezó a investigar la vertiente denominada **VSLAM** o **Visual SLAM** que se centra en el uso de sensores visuales. Otras formas recientes de SLAM incluyen SLAM táctil [CFP12], radar SLAM [JMvH13] y wifi-SLAM.

Actualmente con la mejora de los dispositivos GPS, y para aplicaciones al aire libre, se ha eliminado, casi por completo, la necesidad del uso de SLAM debido a que algunos de estos dispositivos de localización tienen un margen de error tan mínimo que la dominan por completo la inferencia.

El **modelo cinemática** elegido es importante para una solución SLAM. Como parte de dicho modelo se añade la cinemática del vehículo, para mejorar así mejorar las estimaciones bajo condiciones de ruido. El modelo dinámico equilibra las mediciones del conjunto de sensores, los modelos de error parcial utilizados y finalmente realiza un mapa con la ubicación y el rumbo del vehículo. El mapa es la representación definitiva de dicho modelo siendo este la representación o abstracción del modelo.

Estos *modelos cinemáticas* se implementan con un ruido adicional en las acciones aunque existe un enfoque alternativo que ignora el término *cinemática* y lee los datos de *edometría* del vehículo después de cada acción.

Los problemas relacionados con la asociación de datos de **múltiples objetos** y la complejidad computacional se encuentran entre los problemas aún por resolver, por ejemplo la identificación de múltiples puntos de referencia confusos. Los últimos avances han llevado a reexaminar SLAM para que tenga en cuenta el filtrado Bayesiano de múltiples objetos con conjuntos finitos aleatorios que proporcionan un rendimiento superior a los principales algoritmos SLAM en escenarios con altas tasas de falsas alarmas y altas tasas perdidas de detección sin necesidad de asociación de datos [JMV11].

Las técnicas más populares para el manejo de objetos múltiples incluyen el Joint Probabilistic Data Association Filter (JPDAF)²⁴ y el Probability Hypothesis Density Filter (PHD).

Los entornos que contienen objetos no estáticos, **objetos en movimiento**, como vehículos o peatones, siguen representando un desafío. El uso de SLAM con Detection and Tracking of Moving Objects (DATMO) representa un modelo que realiza un seguimiento de los objetos en movimiento [CWH07].

El **cierre del bucle** es un problema relacionado con el reconocimiento de una ubicación previamente visitada y una posterior actualización de los datos estimados. Esto puede su-

²⁴https://en.wikipedia.org/wiki/Joint_Probabilistic_Data_Association_Filter

poner un problema ya que los errores del algoritmo o del modelo pueden asignar un valor inferior a la localización. Los métodos de *cierre del bucle* típicos aplican un segundo algoritmo para así obtener un nivel de similitud y reajustar los valores de la localización cuando detectan una similitud.

Active SLAM estudia el problema combinado del uso de SLAM con el problema de toma de decisiones que supone decidir a donde avanzar para construir el mapa de la manera más eficiente posible. La necesidad de una **exploración activa** es especialmente útil en situaciones de escasa sensibilidad como por ejemplo en situaciones de SLAM *tactil*. *Active SLAM* se realiza mediante una aproximación entrópica del mapa bajo acciones hipotéticas.

Existen otras vertientes de SLAM que tienen otros enfoques, como por ejemplo la centrada en la complejidad y también la que está inspirada en la biología. Estos enfoques y los detallados en este documento son totalmente válidos y pueden ser usados conjuntamente para solucionar un caso SLAM.

Muchos de estos algoritmos SLAM están implementados en la biblioteca de código libre de **Robot Operating System (ROS)**, a menudo son utilizados junto con la biblioteca **PCL** para mapas 3D o con las utilidades ofrecidas por **OpenCV** de tratamiento y reconocimiento de imágenes.

4.3 Planificación de trayectorias

4.3.1 Introducción

De nada sirve poseer un mapa y un modelo del entorno perfecto si el vehículo autónomo es incapaz de **planificar una trayectoria** de forma correcta evitando obstáculos estáticos y dinámicos .

Otro punto importante es, aun sabiendo la trayectoria que tiene que tomar, hacer que el vehículo realice dichos movimientos. Este punto está muy ligado al **control de trayectorias** que se basa en el modelo cinemático del vehículo para así, utilizando una serie de algoritmos, saber describir con el vehículo la trayectoria deseada.

Simplificándolo, un vehículo puede ser descrito como un cuerpo rígido que se mueve en un plano. Una vez descrito el vehículo a partir de una serie de variables que lo definen, como separación entre ejes (L), velocidad (s), ángulo de giro (ϕ) y radio de giro (p), se puede describir un modelo cinemático que comprende una serie de ecuaciones que lo forman (véase Figura 4.22).

Gracias a estos atributos y a la relación que existe entre ellos es posible escribir el modelo

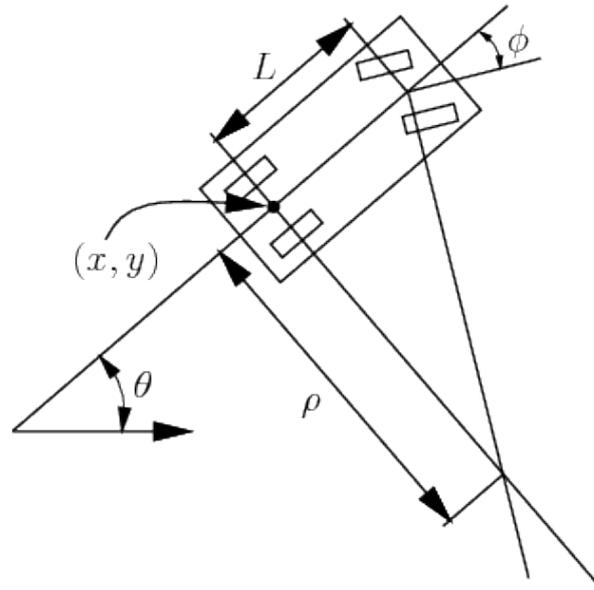


Figura 4.22: Esquema cinemático de un vehículo.

como $q = (x, y, \Theta)$ y definir las ecuaciones 4.4:

$$\begin{aligned}
 \dot{x} &= f_1(x, y, \Theta, s, \Phi) \\
 \dot{y} &= f_2(x, y, \Theta, s, \Phi) \\
 \dot{\Theta} &= f_3(x, y, \Theta, s, \Phi)
 \end{aligned}
 \tag{4.4}$$

Gracias a los modelos cinemáticos y a una serie de algoritmos y estrategias, como por ejemplo: *chasing the carrtos*, *pure pursuit*, *vector pursuit*, *vector field path following* y *look-ahead distance*, es posible controlar el vehículo de forma precisa. Pero el *control de la trayectoria*, aunque importante para el conjunto no es tan importante para la *planificación de trayectorias* ya que gracias a la estructura modular que se sigue es posible realizar una planificación sin tener en cuenta como se realizara.

Con el propósito de realizar un recorrido seguro y lo más óptimo posible, los vehículos tratan de estimar un recorrido en el espacio 2D o 3D del camino desde la posición inicial hasta la posición final evitando colisiones con los posibles obstáculos estáticos y dinámicos.

Al igual que está dividido el problema de la localización, el problema de la planificación de trayectorias está dividido en trayectorias globales y locales (véase Figura 4.23) [RLF16]. La planificación global implica el recorrido que debe tomar el vehículo para llegar desde la posición inicial hasta la posición final definida y la planificación local se centra en evitar los obstáculos que aparecen en la trayectoria previamente definida por el planificador global [JCS09] [Gie04].

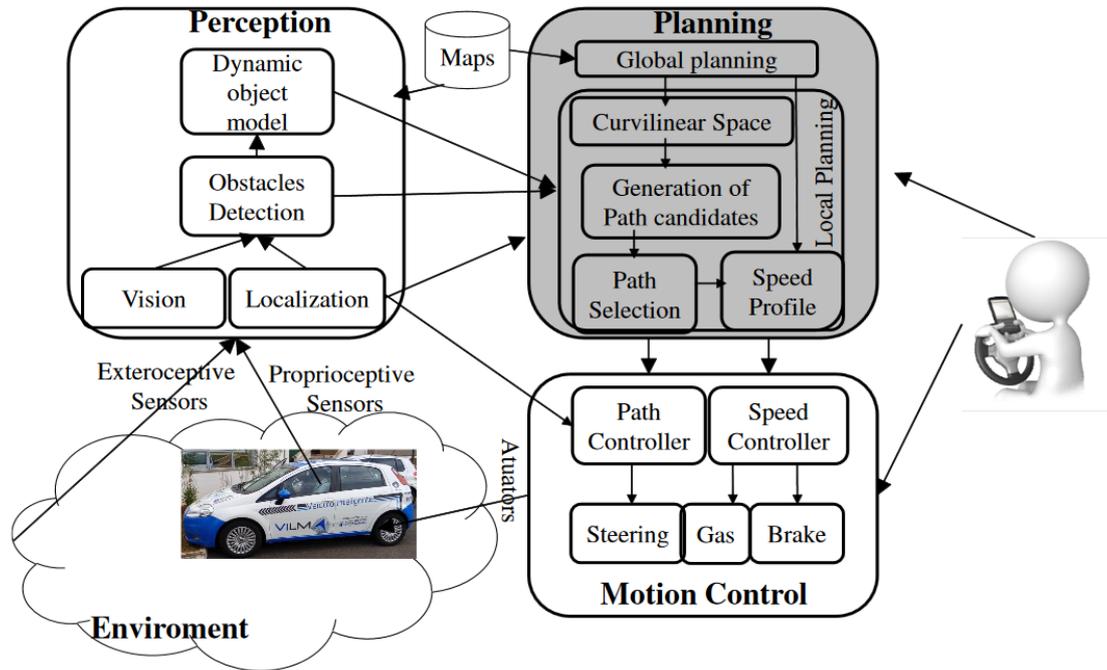


Figura 4.23: Arquitectura de un vehículo autónomo.

4.3.2 Algoritmos

Existen multitud de algoritmos o estrategias usadas en la planificación de trayectorias. No existe una estrategia predominante ya que todas poseen sus ventajas e inconvenientes, por lo que su uso depende de las características del problema a tratar. Por ejemplo, existen planificadores que requieren una fase de aprendizaje, otros que son muy óptimos en situaciones de pocos obstáculos, incluso algunos no se pueden utilizar dependiendo de las restricciones cinemáticas del vehículo, también existen algoritmos capaces de calcular distintas trayectorias óptimas en vez de una sola y por último, aquellos que tienen en cuenta obstáculos en movimiento. [Gar12]

La elección óptima de un algoritmo o estrategia depende de las características del planificador y de si estas cubren las características del problema a tratar. Entre todos los planificadores existen una serie de características comunes: [Lat12]

- **Configuración del vehículo:** conjunto de parámetros utilizados para identificar de forma inequívoca la posición y orientación de los componentes del vehículo.
- **Configuración origen:** la configuración en la posición inicial, o punto de partida, antes de realizar cualquier acción.
- **Configuración destino:** la configuración que se desea alcanzar desde la configuración origen.

- **Espacio de configuraciones:** es el conjunto de todas las configuraciones que pueden ser adoptadas por el vehículo. Cabe destacar que el espacio de configuraciones no coincide con el espacio físico, por lo que, en un vehículo autónomo, su configuración consiste en dos variables coincidentes con la posición cartesiana de su centro y el ángulo que define su orientación y se estaría hablando de un espacio de dimensión 3 cuando el vehículo se mueve en un plano.
- **Colisión:** se determina que un vehículo entra en colisión cuando su configuración dictamina que uno o varios elementos intersecan el espacio ocupado por un obstáculo o por los límites del escenario.
- **Obstáculo:** en planificación, un obstáculo se considera a un conjunto acotado de puntos del espacio físico que determinan el volumen que ocupa dicho elemento.
- **Obstáculo- C :** definimos CB_i como el obstáculo- C asociado a B_i , donde B_i es un obstáculo, como el conjunto de configuraciones cuyo espacio físico entra en colisión con el obstáculo B_i . El espacio físico del vehículo es definido como $A(q)$ cuando este adopta la configuración q , perteneciendo q al conjunto C , por lo que la colisión del vehículo con un obstáculo implica que el espacio físico $A(q)$ es compartido por el espacio que ocupa B_i , dicho de otra forma, y como puede verse en la ecuación 4.5, la intersección $A(q)$ y B_i no es nula ya que existen puntos de colisión.

$$CB_i := \{q \in C : A(q) \cap B_i \neq \emptyset\} \quad (4.5)$$

- **Región de obstáculos- C :** es definida como la unión de todos los obstáculos C . Se denota como CB (véase Ecuación 4.6).

$$CB := \bigcup_{i=1}^{i=n} CB_i \quad (4.6)$$

- **Espacio de configuraciones libres de colisión:** es el subconjunto del espacio de configuraciones cuyos elementos no presentan colisión. Se denota como C_{free} (véase Ecuación 4.7). Cabe denotar que no siempre el espacio de configuraciones libres de colisión coincide con el espacio físico libre de colisiones.

$$C_{free} := C \setminus CB \quad (4.7)$$

- **Camino o trayectoria:** es la sucesión continua de configuraciones adoptadas por el vehículo mientras este se traslada desde la *configuración de origen* hasta la *configuración destino*. La trayectoria se identifica como un subconjunto discreto y finito de la sucesión, en la que sus elementos distan entre sí un valor determinado.

Muchos de los métodos existentes simplifican, en una primera etapa, el modelo del vehículo llegando a utilizar una configuración básica que consiste en la posición cartesiana del mismo. Gracias a esta simplificación se hace coincidir el espacio de configuración con el espacio euclídeo y el espacio de configuraciones libres de colisión con el espacio libre de obstáculos. Para agrupar las estrategias existentes y así obtener una visión más estructurada de los sistemas de planificación, J.C. Latombe definió la siguiente cosificación: [Lat12]

- **Roadmap:** estos métodos describen el espacio de configuraciones libres de colisión a partir de una red de curvas unidimensionales que forman parte del espacio de soluciones. Los grandes representantes de este grupo son: los grafos de visibilidad y los diagramas de Voronoi.
- **Descomposición en celdas:** este método está basado en dividir el espacio de configuraciones libres de colisiones en una serie de regiones denominadas *celdas*, similares al método SLAM de OGM. Cada celda representa un nodo en el grafo de adyacencia, cuando dos celdas comparten un límite en común entonces se agrega un arco al grafo. El grafo generado es utilizado para hallar la solución más óptima. Entre los ejemplos de esta categoría tenemos: la descomposición vertical y las técnicas quadtree.
- **Campos de potencial:** está basado en la creación de una función escalar sobre el espacio de configuraciones de modo que su gradiente sea utilizado para trazar la trayectoria.
- **Probabilísticos:** son aquellos métodos que utilizan una generación de configuraciones aleatorias para así identificar el entorno. Entre estos métodos tenemos los Rapidly Exploring Random Trees (RRT) y los Probabilistic Roadmap (PRM)

Como se ha mencionado, entre los métodos *roadmap* existe el denominado **grafos de visibilidad**, este método, como su nombre indica, está basado en la definición de un grafo donde sus nodos representan configuraciones factibles del vehículo y sus arcos el coste de conexión directa entre ellos. Una vez obtenido el grafo, denominado grafo de conectividad, sólo es necesario recorrerlo desde el punto de origen hasta el punto de destino. Es definido como *grafo de visibilidad* ya que en este método se enlazan los vértices entre los que existe visión directa.

Para que este método pueda utilizarse es necesario una descripción poligonal del conjunto de obstáculos- C . Así pues, el entorno queda definido como un conjunto de polígonos que definen obstáculos y los límites del plano en el que se puede mover el vehículo.

El proceso implica la creación de una matriz cuadrada de tamaño igual a los vértices de los polígonos más el punto de origen y de destino. Posteriormente se procesan los puntos en parejas y se anota, si es posible, la longitud de la trayectoria en línea recta entre dichos puntos, en caso de que no exista colisión alguna, en caso de colisión también se anota dicha situación. Al finalizar el proceso, la matriz describirá un conjunto de puntos entre los cuales

existe un enlace directo, sin obstáculos (véase Figura 4.24). En este punto la obtención de la trayectoria óptima será reducida a un problema de caminos mínimos entre dos puntos, por lo que el algoritmo de Dijkstra es el más utilizado en las implementaciones de este método. Existen distintas mejoras para este método, como la simplificación de vértices mediante la eliminación de vértices cóncavos o la eliminación de los segmentos no tangentes (véase Figura 4.25).

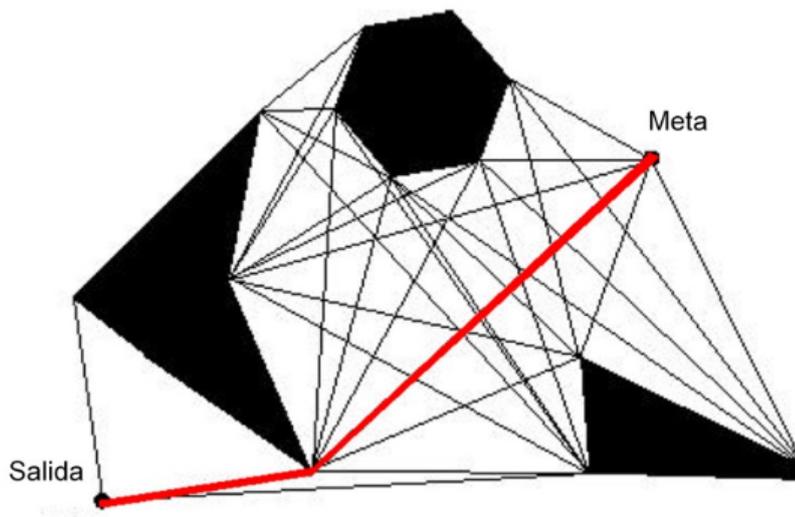


Figura 4.24: Ejemplo de grafo de visibilidad.

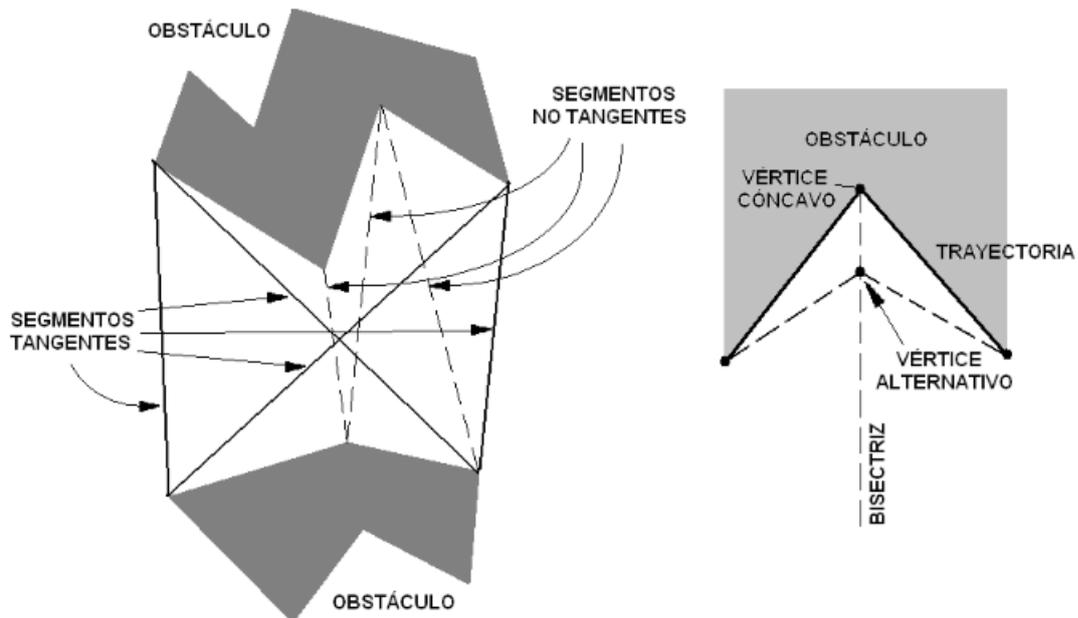


Figura 4.25: Definición de segmentos tangentes y vértices cóncavos.

Otro método son los **Diagramas de Voronoi**. Este método está basado en los grafos de los que coge el nombre. Una ventaja de este método es que maximiza la distancia entre el vehículo y los obstáculos. Este método puede ser utilizado directamente o como base para otros métodos. Este método, al igual que los grafos de visualización, necesita de una descripción poligonal de los obstáculos-*C*.

En un diagrama de Voronoi se representan los puntos que equidistan de los elementos vecinos, obstáculos o bordes del perímetro, entre sí. Este atributo de vecindad es utilizado para identificar que no existe otro elemento entre dos que sean vecinos.

Teniendo en cuenta que los puntos de un plano equidistantes de un punto y una recta define una parábola y que los puntos que equidistan entre dos rectas es otra recta, entonces, si los obstáculos están definidos como polígonos, se entiende que el diagrama estará compuesto por una sucesión de segmentos rectos y parábolas interconectados. Así pues, se puede obtener un grafo si se disponen las interconexiones entre segmentos como vértices y los segmentos rectos y parábolas como arcos.

Una vez está definido este grafo, se puede proceder a conectar los puntos de origen y destino al mismo. En el caso de que el punto de partida o de destino esté cercano a un vértice de un polígono se conectara al segmento más cercano mediante una recta que atraviese los dos puntos, en caso contrario, donde el punto esté más cercano a una arista del polígono, este se conectara, al segmento, mediante una recta perpendicular a la arista (véase Figura 4.26).

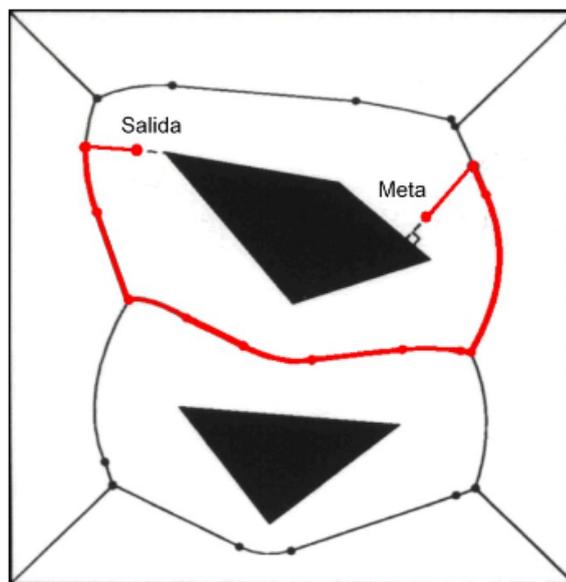


Figura 4.26: Ejemplo de diagrama del grafo de Voronoi.

Una vez completado el grafo, la solución es similar al método anterior de *grafos de visibilidad*, donde la solución está basada en aplicar el algoritmo de caminos mínimos de Dijkstra.

Cabe destacar que es necesario tener en cuenta que el vehículo es capaz de realizar ciertos cambios abruptos de dirección en su trayectoria, como punto en contra, aunque a favor está que el vehículo siempre mantendrá una distancia óptima del obstáculo.

Mediante el uso de estas tres ecuaciones y los valores obtenidos es posible elegir una de las posibles rutas de entre todas las disponibles. Se han elegido estos tres valores ya que representan cualidades de seguridad intrínsecas e extrínsecas del recorrido dado que se basan en su iteración con el exterior y en cómo se realiza dicha circulación.

Otro método utilizado es el denominado **descomposición de celdas**. Este método puede ser utilizado tanto para calcular trayectorias a globales o locales. La *descomposición de celdas* está basada, como su nombre indica, en la creación de unas *celdas* donde se cumpla la siguiente regla, debe ser posible unir dos puntos cualesquiera de la celda mediante una línea recta. Una vez se han definido dichas *celdas* se puede proceder a calcular la trayectoria, por ejemplo mediante un grafo de adyacencias, donde los nodos sean las celdas y las conexiones representen la unión entre dos celdas.

Entre las posibles estrategias a seguir para descomponer en celdas el mapa existen dos tipos: **exacta** o **aproximada**.

Se denomina **descomposición exacta** a aquella estrategia que genera un conjunto de celdas cuya unión coincide con el espacio de configuraciones libres de colisión. Dado que los obstáculos del escenario son descritos como figuras poligonales una de las posibles aproximaciones, definida como **Descomposición Vertical** [Cha85] a esta estrategia son trazar líneas verticales que crucen todos los vértices generando así celdas rectangulares/trapezoidales y triangulares (véase Figura 4.27). A partir de estas celdas se define un grafo de conectividad y se calcula la trayectoria óptima para llegar desde el origen al destino. Una vez obtenido dicho grafo es posible designar como puntos seguros o de paso a los puntos medios de las rectas y así trazar una trayectoria segura y fuera de obstáculos. Esta aproximación requiere de un pre-procesado de los datos para ordenar los vértices en el eje horizontal. El tiempo total de ejecución de esta estrategia es la suma del tiempo de pre-procesado de los datos, que implica un tiempo de computo de $O(n \log n)$ donde n es el número de vértices, la designación de celdas se realiza de forma concurrente e implica un tiempo de $O(n \log n)$ y por último, el cálculo de la trayectoria oscila entre $O(n)$ y $O(n \log n)$ dependiendo de si se desea hallar el camino más corto o no [Lat12].

Otra aproximación de la **descomposición exacta** es la denominada **Triangulación de Delaunay**²⁵. Se toma el conjunto de vértices de los obstáculos y se unen aquellos puntos que cumplen la condición de *Delaunay* (véase Figura 4.28). Una vez definidos estos puntos, se toman los centros de las circunferencias como puntos pertenecientes al diagrama de *Voronoi* resultante. Este método no resuelve el diagrama de *Voronoi* pero define la descomposición

²⁵https://es.wikipedia.org/wiki/Triangulación_de_Delaunay

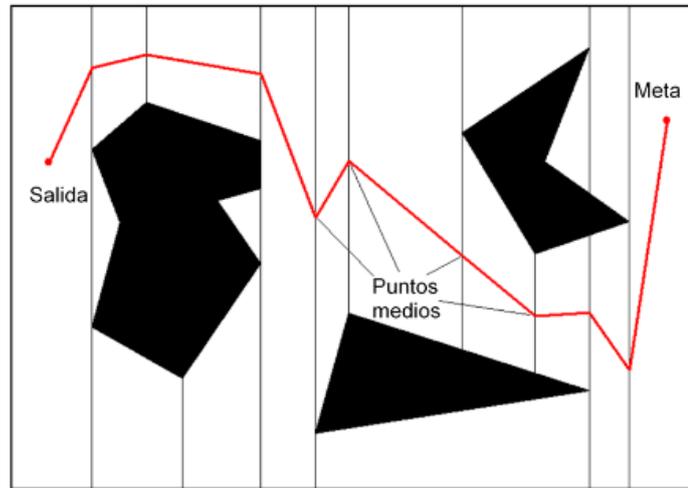


Figura 4.27: Ejemplo de una descomposición vertical.

de C_{free} en celdas triangulares delimitadas por los segmentos definidos. Una vez obtenidas las celdas se siguen los pasos descritos anteriormente para calcular la trayectoria. Esta aproximación, aunque más compleja, mantiene una distancia más óptima a los obstáculos que la *descomposición vertical* (véase Figura 4.29).

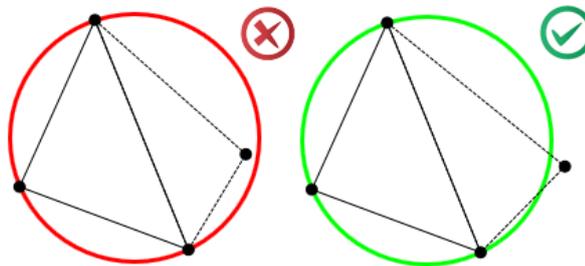


Figura 4.28: Condición de la triangulación de Delaunay.

La *descomposición aproximada* es aquella que utiliza celdas predefinidas, habitualmente rectangulares. La descomposición del escenario en celdas genera celdas que colisionan con algún obstáculo, dichas celdas son descartadas del conjunto y las restantes son utilizadas para calcular el grafo de adyacencia. La ventaja de esta aproximación respecto a las anteriores aproximaciones reside en el tiempo de computo. Entre los distintos métodos de descomposición aproximada podemos encontrar a los metodos **quadtree** y **fast marching method**.

El método *quadtree* inscribe el escenario en un rectángulo definiendo como obstáculos las partes que queden fuera. A continuación se subdivide el rectángulo en cuatro rectángulos del mismo tamaño y se repite el proceso siempre y cuando el rectángulo esté en un estado mixto o parcialmente ocupado, a su vez, son clasificados como C_{free} , celda libre, a aquellas celdas o rectángulos que no entra en colisión con algún obstáculo, CB , celdas llenas, a aquellas que entran en colisión total con un obstáculo. Cada etapa actualiza el árbol de grado

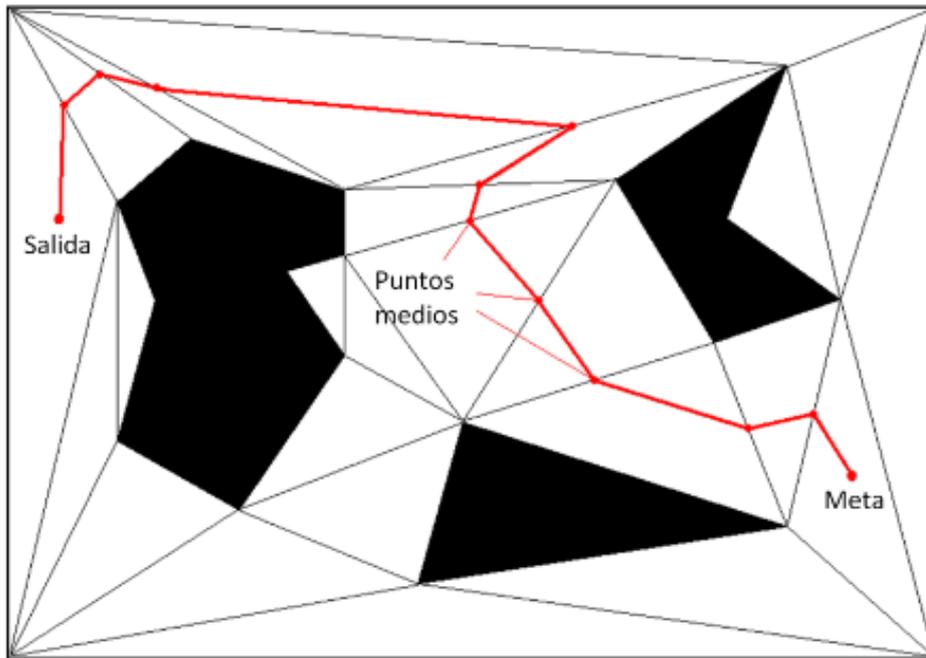


Figura 4.29: Ejemplo de una descomposición de Delaunay.

4 resultante donde el nodo raíz representa al rectángulo inicial que contiene al escenario. En cada iteración del algoritmo se comprueba si en dicho árbol existe una sucesión de celdas libres que permita llegar del origen al destino, en caso contrario se vuelve a iterar hasta que se llega al tamaño mínimo de celda definido (véase Figura 4.30 y Figura 4.31).

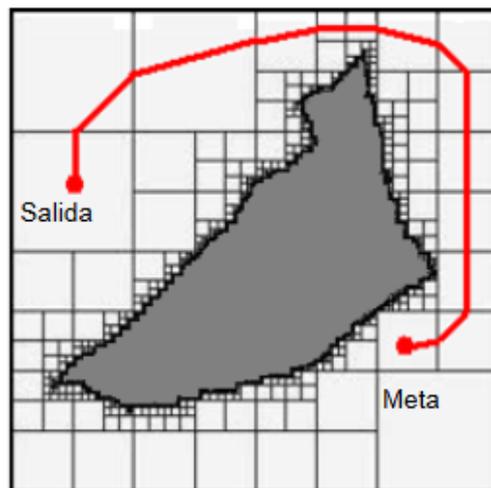


Figura 4.30: Ejemplo de una descomposición de Quadtree.

El uso de los quadtree surgió en el ámbito de la visión por computador y el modelado de sólidos [JT80] [Sam80] aunque posteriormente fue introducido en tareas de planificación [Hay86] [Her86], incluso existe una aproximación que utiliza el tiempo para calcular

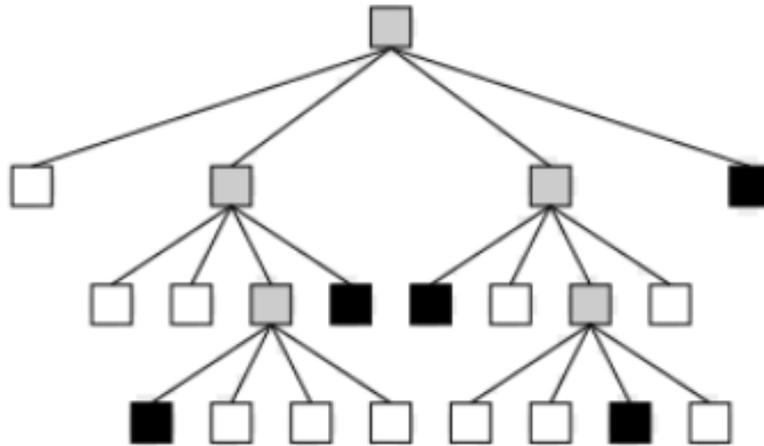


Figura 4.31: Arbol quadtree. Las hojas negras definen celdas llenas, las hojas blancas celdas vacías y los nodos grises celdas mixtas.

escenarios dinámicos [FS89].

Otra de las soluciones es el uso de celdas del mismo tamaño que utilizan una discretización regular del espacio de configuraciones. El método **fast marching method** fue definido por *Sethian* en 1996 [Set96] y es uno de los métodos más conocidos.

Este método está basado en la teoría de ondas y en la trayectoria que efectuaría un haz de luz desde el origen hasta el destino. Mediante la función $\Phi(x, y, t)$ se puede calcular un frente de onda que corresponde a la curva cerrada en el plano, es decir, si se originase un destello luminoso en el punto de origen dicha curva correspondería con un círculo centrado en el origen con un radio que aumenta en relación al tiempo, dicho aumento corresponde a una velocidad, en dirección normal a la curva, de $F(x, y)$.

Para ello Sethian definió la función de nivel $T(x, y)$ con la que se obtiene el instante t en el que el frente de onda (véase Ecuación 4.8) colisiona con el punto (x, y) . La ecuación del frente de onda es una expresión de la ecuación **eikonal**²⁶ y define la velocidad de propagación del frente de onda en inversamente proporcional al gradiente de la función de nivel. Sethian definió la ecuación 4.9 donde $T_{ij} = T(i, j)$, siendo i el elemento i -ésimo de la discretización del eje x , y j el correspondiente al eje y .

$$|\nabla T| F = 1 \tag{4.8}$$

²⁶https://en.wikipedia.org/wiki/Eikonal_equation

$$\begin{aligned}
D_{ij}^{+x}T &= (T_{i+1,j} - T_{i,j})/\Delta x; & D_{ij}^{-x}T &= (T_{i,j} - T_{i-1,j})/\Delta x; \\
D_{ij}^{+y}T &= (T_{i,j+1} - T_{i,j})/\Delta y; & D_{ij}^{-y}T &= (T_{i,j} - T_{i,j-1})/\Delta y;
\end{aligned} \tag{4.9}$$

$$\max(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T)) + \max(\max(D_{ij}^{-y}T), -\min(D_{ij}^{+y}T)) = \frac{1}{F_{ij}^2}$$

El método establece una *primera fase* de inicialización donde se definen tres listas: *Lista A*, con las celdas calculadas, *Lista B* con las celdas al frente y *Lista C* con las celdas restantes. En la segunda fase, *avance*, el algoritmo realiza las siguientes sub-etapas, de forma cíclica, hasta finalizar el algoritmo:

- Extracción de la *lista B* la celda con menor valor cel_{ij} y posterior agregación a la *lista A*.
- Cálculo de vecinos, es decir, obtención de las celdas adyacentes a la actual.
- Si alguno de los vecinos calculados anteriormente pertenece a la *lista C* se extrae y se añade a la *lista B*.
- Se recalculan los vecinos que estén en la *lista B* usando la ecuación 4.9.

Este algoritmo avanza en el tiempo y calcular T mediante el avance del frente de onda, celda a celda, una vez finalizado se halla T_{ij} para todas las celdas del escenario, aunque cabe destacar que en el caso que concierne a la planificación no es necesario calcular todas las celdas y una vez alcanzado el punto el algoritmo puede ser finalizado (véase Figura 4.32).

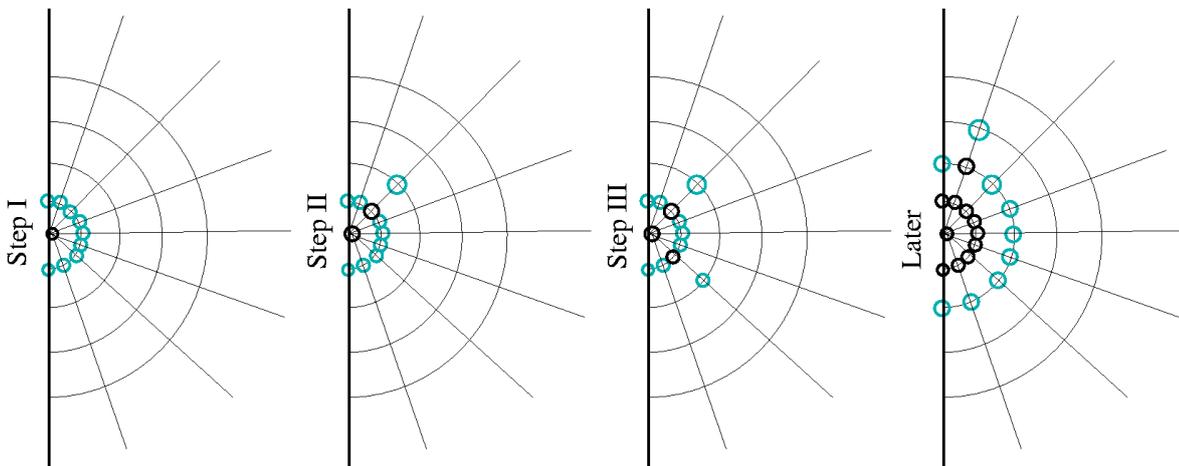


Figura 4.32: Implementación circular de *Fast Marching Method* http://sepwww.stanford.edu/public/docs/sep95/tariq5/paper_html/node4.html.

La solución obtenida con este algoritmo se aproxima a la que ofrecen los *grafos de visibilidad*. Uno de los puntos flojos de este algoritmo es que las trayectorias calculadas discurren

cerca de los obstáculos aunque existen métodos combinados de este algoritmo como la inclusión del método de Voronoi que asigna, durante el proceso, el valor de la distancia al obstáculo más cercano al valor de la celda para ser utilizados como valor del índice de refracción (inversa de la velocidad de propagación F) mientras se ejecuta el algoritmo (véase Figura 4.33) [SGMA07].

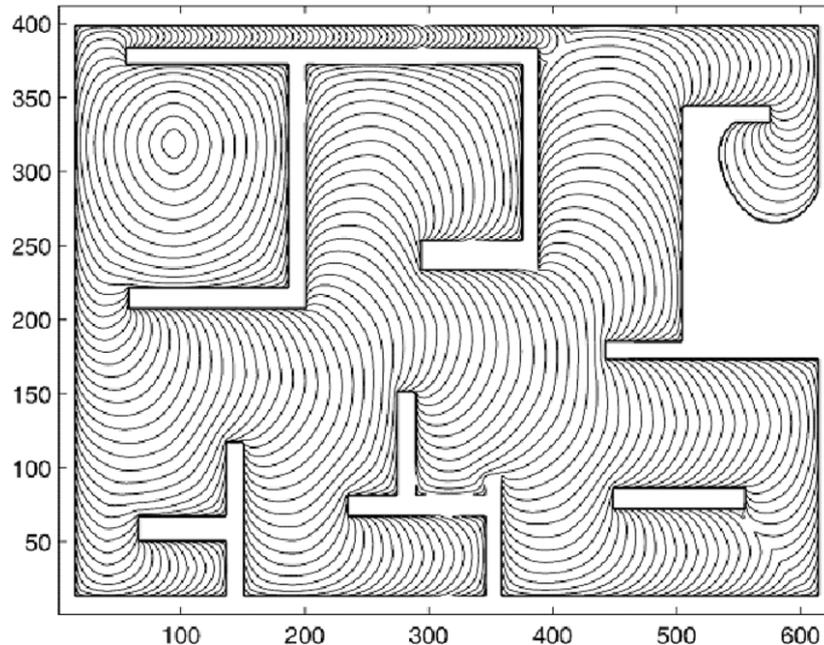


Figura 4.33: Propagación del método *Fast Marching Method* con transformación Voronoi.

Otro método de planificación es el denominado **campos de potencial**. Este método está basado en la representación del vehículo como un punto perteneciente a C_{free} que es a su vez tratado como una partícula sometida a diferentes fuerzas. Dichas fuerzas provienen de distintos puntos clave del escenario. Este método añade una fuerza de atracción en el punto de destino y trata a los obstáculos como fuerzas de repulsión. El escenario tiene asociado un campo de potencial cuyo valor depende de su proximidad a dichas fuentes (obstáculos y distancia al punto de destino)(véase Figura 4.34).

La trayectoria es generada en virtud de un vector gradiente derivado del campo potencial. El algoritmo calcula dicho vector en el punto de origen y añade a la trayectoria final un segmento de tamaño fijo con la misma dirección que el vector gradiente, a continuación vuelve a calcular el vector gradiente en el punto de origen nuevo, correspondiente al punto destino de la trayectoria actual. El proceso se repite iterativamente hasta que se alcanza el punto final o mínimo absoluto del campo potencial.

El campo potencial U puede ser construido de distintas formas, pero una de las más comunes es la unión de campos potenciales mediante la ecuación 4.10. Por un lado se calcula el campo potencial de atracción U_{atr} que genera el punto de destino y por otro lado se calcula el campo de potencial de repulsión U_{rep} que generan los obstáculos del escenario (véase

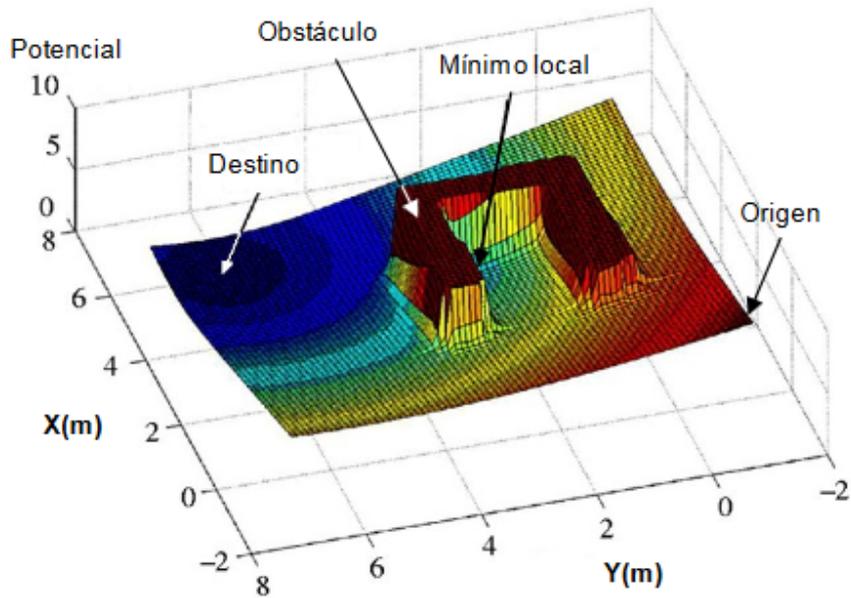


Figura 4.34: Ejemplo de un *campo de potencial*.

Figura 4.35)²⁷.

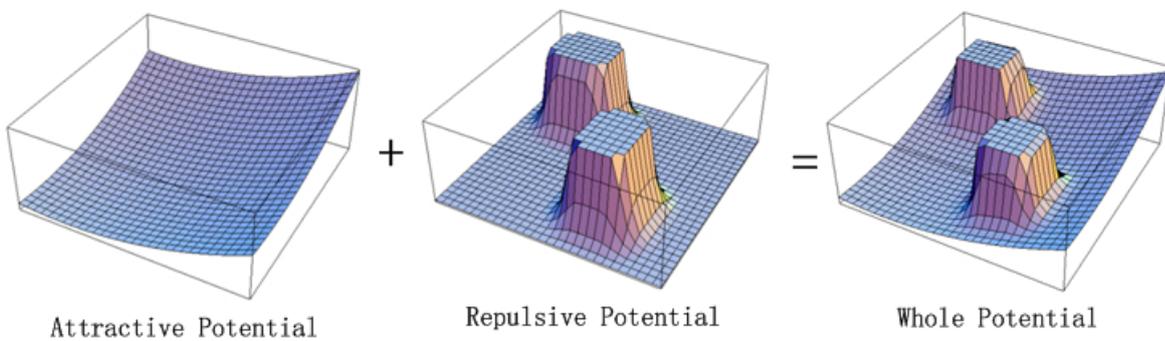


Figura 4.35: Unión de los campos de atracción y repulsión.

$$U = U_{atr} + U_{rep} \quad (4.10)$$

Mediante la ecuación 4.11, derivada de la ecuación 4.10 es posible calcular el campo de potencial de un punto q .

$$U(q) = U_{atr}(q) + U_{rep}(q) \quad (4.11)$$

²⁷<https://taylorwang.wordpress.com/2012/04/06/collision-free-path-planning-using-potential/>

El campo responsable de las fuerzas de atracción, o U_{atr} , es definido mediante la función parabólica 4.12, donde k representa un factor de escala, q_{goal} es la configuración destino y $\rho_{goal}(q)$ es la distancia euclídea $\|q - q_{goal}\|$. La ecuación 4.13 define la fuerza de atracción como resultado de derivar dicho campo de potencial.

$$U_{atr} = k\rho_{goal}^2(q) \quad (4.12)$$

$$\vec{F}_{atr} = -\vec{\nabla}U_{atr}(q) = -2k \cdot (q - q_{goal}) \quad (4.13)$$

Como resultado se obtiene un vector cuyo módulo es proporcional a la distancia entre dos puntos y que mantiene una dirección constante hacia la configuración de destino.

Para el cálculo del campo U_{rep} se define la ecuación 4.14 que es resultante de la suma de los campos generados por cada obstáculo.

$$U_{rep}(q) = \sum_{k=1}^r U_{CB_k}(q) \quad (4.14)$$

La ecuación 4.15 define un ejemplo de un campo potencial de repulsión asociado al obstáculo x , donde η corresponde a un factor de escala positivo, $\rho_k(q)$ es definido mediante la ecuación 4.16 que corresponde a la distancia de q al obstáculo- C k y por ultimo ρ_0 corresponde a la constante positiva denominada *distancia de influencia del obstáculo- C* .

$$U_{CB_k}(q) \begin{cases} \frac{1}{2}\eta \left(\frac{1}{\rho_k(q)} - \frac{1}{\rho_0} \right) & si \quad \rho_k(q) \leq \rho_0 \\ 0 & si \quad \rho_k(q) > \rho_0 \end{cases} \quad (4.15)$$

$$\rho_k(q) = \min_{q' \in CB_k} \|q - q'\| \quad (4.16)$$

La fuerza de repulsión asociada a cada obstáculo corresponde a la ecuación 4.17 y el conjunto de todas ellas es calculado mediante la ecuación 4.18.

$$\vec{F}_{CB_k}(q) = -\vec{\nabla}U_{CB_k}(q) = \begin{cases} \eta \left(\frac{1}{\rho_k(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho_k^2(q)} \vec{\nabla}\rho_k(q) & si \quad \rho_k(q) \leq \rho_0 \\ 0 & si \quad \rho_k(q) > \rho_0 \end{cases} \quad (4.17)$$

$$\vec{F}_{rep}(q) = \sum_{k=1}^r \vec{F}_{CB_k}(q) \quad (4.18)$$

Como resultado final, la fuerza resultante que seguiría el vehículo mediante este algoritmo estaría definido por 4.19.

$$\vec{F}_{res} = \vec{F}_{rep} + \vec{F}_{atr} \quad (4.19)$$

Por consiguiente el algoritmo del planificador seguirá una trayectoria propia de una partícula sometida a F_{res} . Un punto a tener en cuenta al utilizar dicho algoritmo son los **mínimos locales**, puntos donde se alcanza un *mínimo local* y donde todas las fuerzas son nulas, dicho *mínimo local* no corresponde con el *mínimo global* o punto de destino. Como solución a este problema se plantea la generación aleatoria de movimientos que permitan al sistema escapar del mínimo y así poder proseguir con el algoritmo, aunque este método implica a su vez una serie de problemas derivados del tiempo empleado para el cálculo de movimientos aleatorios y en cómo se generan estos.

Una mejora de este método implica el uso de una rejilla o *grid* para descomponer el escenario en celdas que facilitan la implementación del método [BL91] ya que cada celda define un valor de campo de potencial. Con esta modificación no es necesario elegir gradiente y segmento de avance ya que basta con seleccionar la celda que posee menor valor de entre las vecinas por lo que el algoritmo se reduce a elegir una secuencia continua de celdas que definen una trayectoria.

Otro algoritmo de planificación es el mapa probabilístico o **PRM**. Uno de los puntos a favor de este algoritmo es la eficacia que presenta ante escenarios con muchos grados de libertad y aunque puede no ser el caso de este proyecto sí que es necesario, al menos, mencionarlo ya que es muy utilizado. Este algoritmo está dividido en dos grandes enfoques, los **PRM de consulta única** y los **PRM de múltiple consulta**.

Los *PRM de múltiple consulta* están basados en el uso de múltiples grafos. Este tipo de métodos generan una serie de puntos aleatorios dentro de C_{free} , posteriormente dichos puntos se unen entre pares de puntos cercanos, si dicho segmento presenta alguna colisión entonces se desecha, en caso contrario se añade a un grafo que representa todos los puntos que han logrado conectividad. El algoritmo se finaliza una vez que dicho recorrido une el punto de origen y el punto de destino (véase Figura 4.36).

Una de las ventajas del algoritmo es que reduce el número de cálculos a realizar ya que no es necesario calcular los límites de los obstáculos- C y de C_{free} dado que sólo es necesario una función que calcule si existe o no colisión. Otra de las ventajas de dicho algoritmo es la *completitud probabilística* que define que a medida que el número de puntos aumenta la probabilidad de que el algoritmo no encuentre una trayectoria, si existe, tiende a cero [LEKO96].

La otra vertiente, *PRM de consulta única*, define un único grafo y limita la búsqueda de trayectorias a aquellas configuraciones que son accesibles desde los nodos de dicho grafo. Este proceso se realiza construyendo un árbol de configuraciones desde el punto inicial o el

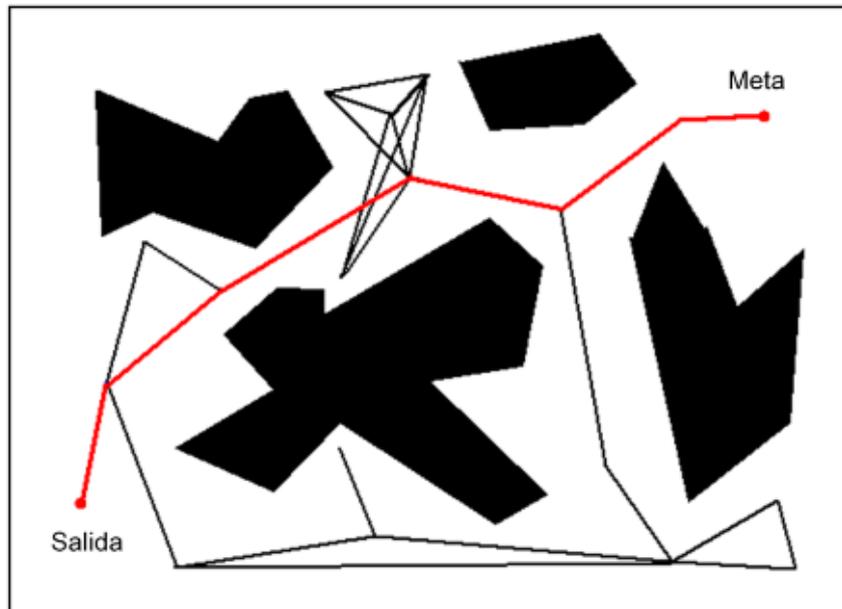


Figura 4.36: Ejemplo de grafo generado mediante PRM.

final hasta que dicho árbol establece una conexión con el otro punto. Esta construcción se realiza mediante trayectorias aleatorias añadiendo nuevos nodos y arcos. Esta solución, de un solo grafo, es denominada **búsqueda unidireccional**. Existe una variante de este algoritmo que define dos grafos, uno por cada punto (inicial y final) y que crecen buscándose. Dicho algoritmo está definido como **búsqueda bidireccional** y finaliza una vez se hayan encontrado dichos arboles.

Una pequeña comparativa entre los PRM y los *campos de potencial* hace evidente sus ventajas e inconvenientes. Como punto a favor cabe destacar que con el algoritmo PRM no es necesario solucionar el problema de los mínimos locales, aunque como desventaja es importante remarcar que este algoritmo no es muy eficiente frente a una planificación local, ya que requiere que se ejecute el algoritmo con bastante frecuencia y su carga computacional no es baja.

Cabe destacar que dicho algoritmo es capaz de obtener distintas trayectorias por lo que lo hace bastante competente para una planificación global.

El último de los algoritmos de planificación a tratar es el **Rapidly Exploring Random Trees (RRT)**. Inicialmente fue creado para dar soporte a otros algoritmos, pero sus numerosas ventajas y capacidades hicieron que fuese utilizado independientemente.

Este algoritmo está basado en la construcción de un árbol de configuraciones que, al igual que el PRM, crece explorando a partir del punto de origen. Antes de continuar es necesario definir una serie de conceptos útiles para entender la explicación de dicho algoritmo [Gar12]:

- ρ : define una métrica (ponderación de proximidad. Ej: distancia euclídea) dentro de C .
- q_{ini} : define la configuración inicial.
- q_{fin} : define la configuración final.
- q_{rand} : define una configuración aleatoria dentro de C .
- q_{near} : define la configuración más próxima, de entre las existentes en el grafo, a q_{rand} .
- q_{new} : define la configuración que se añadirá al árbol.
- ε : define la longitud del segmento de crecimiento.

Para generar el árbol de exploración se define el algoritmo 4.3.2, definido por Steven M. LaValle²⁸ [LaV98]. Este algoritmo busca extender el árbol según un punto q_{rand} elegido de forma aleatoria.

Algorithm 1 Algoritmo basico RRT

```

1: procedure RTT( $q_{ini}, Kmax$ )
2:   Arbol[0]  $\leftarrow$   $q_{ini}$ 
3:    $k \leftarrow 1$ 
4:   while  $k < Kmax$  do
5:      $q_{rand} = \text{CONFIGURACION\_ALEATORIA}()$ 
6:      $\text{EXTIENDE}(\text{Arbol}, q_{rand})$ 
7:      $k \leftarrow k + 1$ 
8:   end while
9:   return Arbol
10: end procedure

```

El algoritmo inicializa la tabla asociada al árbol con la configuración inicial q_{ini} . A continuación obtiene una configuración aleatoria, donde se obtiene un punto al azar dentro del espacio de configuraciones C , y con dicha configuración extiende el árbol en esa dirección. Este proceso se repite mientras k sea menor que $Kmax$. Una vez finalizado devuelve el árbol generado. $Kmax$ es la variable utilizada para finalizar el algoritmo en caso de que no se alcance la configuración de destino. Cabe destacar que esta variable depende de las características del problema.

Como se ha visto, este algoritmo utiliza a su vez la función *Extiende* (véase Algoritmo 4.3.2) encargada de ampliar el árbol en el sentido que marca q_{rand} . Este algoritmo empieza calculando el q_{near} mediante la función *VecinoMásProximo*(q_{rand}, Arbol) que aplica la métrica ρ a los vértices del árbol y obtiene así el punto más cercano a q_{rand} . A continuación, mediante la función *NuevaConfiguracion*($q_{rand}, q_{near}, q_{new}$) que a partir de q_{near} y a una distancia ε con dirección q_{rand} calcula q_{new} (véase Figura 4.37). Para este nuevo punto es necesario calcular si existe alguna colisión, en caso de existir se devuelve un aviso de colisión

²⁸<http://mssl.cs.illinois.edu/~lavalle/>

y dicho punto no se añade al árbol, en caso de no existir colisión alguna el punto se agrega al árbol y se marca como *alcanzado*, caso en el cual q_{rand} está dentro del círculo con centro q_{near} y radio ϵ , o en caso contrario como *avanzado*.

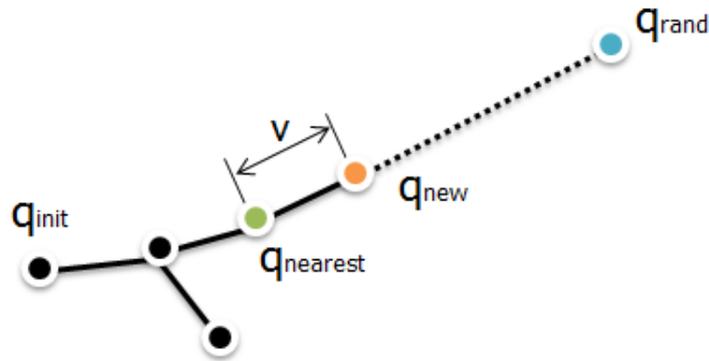


Figura 4.37: Función Extiende del algoritmo RRT. <http://joonlecture.blogspot.com.es/2011/02/improving-optimality-of-rrt-rrt.html>

Algorithm 2 Algoritmo Extiende

```

1: procedure EXTIENDE( $Arbol, q_{rand}$ )
2:    $q_{near} = \text{VECINOMASPROXIMO}(q_{rand}, Arbol)$ 
3:   if NUEVACONFIGURACION( $q_{rand}, q_{near}, q_{new}$ ) then
4:     AÑADEVERTICE( $Arbol, q_{new}$ )
5:     if  $q_{new} = q_{rand}$  then
6:       return Alcanzado
7:     else
8:       return Avanzado
9:     end if
10:  else
11:    return Rechazado
12:  end if
13: end procedure

```

Cabe destacar que el comportamiento de este algoritmo en comparación con los anteriores obtiene una homogeneidad superior del espacio explorado. El algoritmo avanza hacia zonas inexploradas, dado que es más probable de que exista q_{rand} , haciendo que en sus primeras iteraciones este se extienda en unas pocas ramas y posteriormente, conforme va cubriendo el espacio, este pase a ramificarse.

Para que se añada un punto al árbol y se defina una rama es necesario que q_{rand} sea el punto más próximo a ningún otro. Es decir, que q_{rand} pertenezca a la región Voronoi asociada a dicho punto ya que si se genera q_{rand} de forma que la probabilidad sea igual en todo el espacio de configuraciones y dicho espacio se divide en regiones de Voronoi, la probabilidad de que q_{rand} pertenezca a una de estas regiones estará definido por el cociente entre el tamaño de la región y el espacio de configuraciones (véase Figura 4.38) [LJS08].

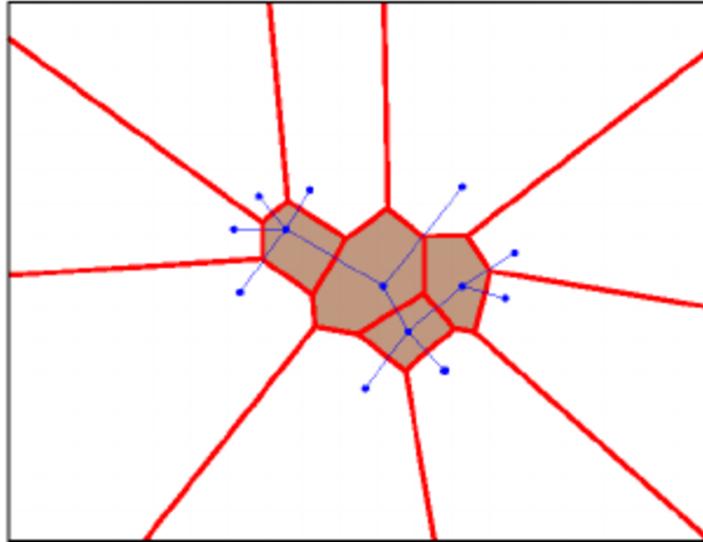


Figura 4.38: Ejemplo de árbol RRT con Voronoi.

Según *LaValle* [LaV98], los algoritmos basados en RRT poseen una serie de características:

- Se expanden hacia espacios inexplorados.
- Sus vértices se distribuyen siguiendo una función de probabilidad aproximadamente.
- Son simples de analizar y de implementar en cualquier escenario.
- Siempre permanecen conexos, aunque tengan pocos vértices.
- Pueden ser utilizados en otros planificadores.
- No se basan en un par origen-destino por lo que pueden ser utilizados en distintos ámbitos.
- No necesitan definir C_{free} sólo una función de colisión.
- Es probabilísticamente completo.

En las primeras propuestas de RRT como planificador autónomo surgieron algoritmos que se basaban en este RRT básico pero que añadían pequeñas modificaciones en la función que generaba q_{rand} , así surgieron **RRT-GoalZoom** y **RRT-GoalBias** [LJ00] englobados en los algoritmos RRT de consulta única. Otras de las posibles variaciones son²⁹:

- **Parti-game Directed RRTs (PDRRTs)**: [RK04] es un método que combina RRT con el método *Parti-game* [WMGA00] para refinar la búsqueda en casos concretos para así poder planificar más trayectorias de manera más eficiente.
- **Closed-loop Rapidly-exploring Random (CL-RRT)**: [YKKFH09] es una extensión de RRT que transforma una entrada en un sistema estable consistente en un vehículo y

²⁹<https://en.wikipedia.org/wiki/Rapidly-exploring-random-tree>

un controlador.

- Basadas en **RRT*** y **Rapidly-exploring Random Graph (RRG)**: [KF10] [KF11] estas técnicas están basadas en encontrar el coste óptimo de la trayectoria dado que las técnicas RRT no obtienen la ruta más óptima (véase Figura 4.39).

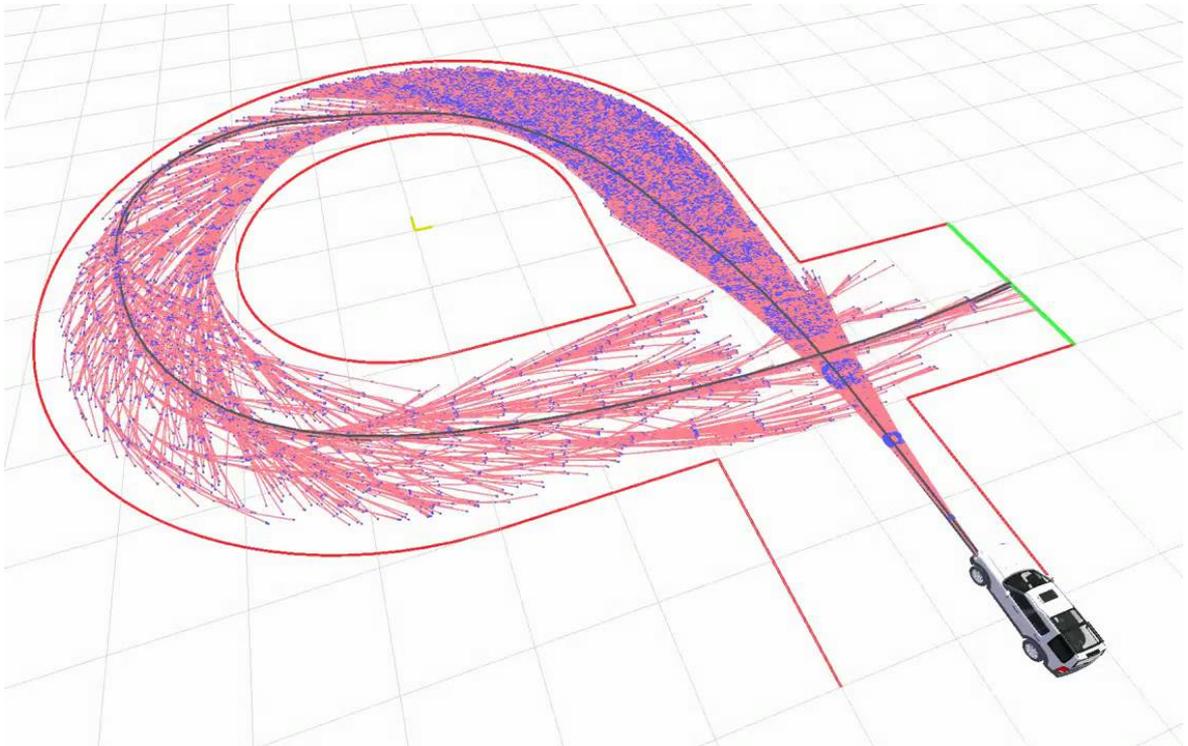


Figura 4.39: Ejemplo de árbol RRT*. <https://www.youtube.com/watch?v=p3nZHn0Whrg>

- **RRT*-Smart**: [FIAH12] es un método para acelerar las tasas de convergencia de RRT* utilizando la optimización de ruta y el muestreo inteligente.
- **A*-RRT** y **A*-RRT***: [MBS13] es un método de planificación de movimiento en dos fases, que, en una primera fase, utiliza un algoritmo de búsqueda de grafos para buscar una trayectoria inicial factible en un espacio de baja dimensión y en una segunda fase se realiza una búsqueda RRT* en un espacio continuo.
- **RRT*FN**: [AV13] es un RRT* con un número fijo de nodos y que elimina aleatoriamente un nodo hoja en cada iteración.
- **RRT*-AR**: [SCS13] que planifica rutas alternativas mediante muestreo.
- **Informed RRT***: [JGB14] mejora la velocidad de convergencia de RRT* mediante la introducción de una heurística, similar a la forma en que A* mejora con el algoritmo de Dijkstra.
- **RT-RRT***: [KNH15] es una variante de RRT* y *Informed RRT*, que utiliza una estrategia que permite a la raíz del árbol moverse con el agente sin descartar rutas

previamente muestreadas, con el fin de obtener una planificación de trayectoria en tiempo real.

- **Theta*-RRT**: [LPA16] es un método de planificación de movimiento en dos fases, similar a A^*-RRT^* , que utiliza una combinación jerárquica de una búsqueda desde cualquier ángulo con una planificación RRT para así generar una trayectoria rápida en entornos con restricciones complejas no oronómicas.

La mayoría de los algoritmos de RRT generan unos árboles poco óptimos para la mayoría de los vehículos por lo que requieren de un **postprocesado**[DGO10] para suavizar las trayectorias ya que, de no ser así, se consumiría mucho tiempo realizando maniobras inútiles para el vehículo dado que, debido a la naturaleza del algoritmo, es poco común encontrar una sucesión de segmentos alineados a lo largo de la trayectoria. Cuanto mayor es la precisión del algoritmo (menor tamaño de segmentos ϵ) más difícil será realizar maniobras por parte del vehículo.

Estas técnicas de *postprocesado*, al igual que ocurre en otros algoritmos, son de bajo coste computacional, y que, a partir de la salida del planificador obtienen una trayectoria optimizada para el vehículo. Cabe destacar que dicho algoritmo deberá estar configurado y diseñado para el caso en concreto siguiendo el modelo dinámico del vehículo.

El algoritmo de postprocesado busca unir nodos de la trayectoria siempre y cuando puedan ser unidos mediante una trayectoria rectilínea que atravesase un obstáculo en su recorrido. Este proceso se realiza inicialmente intentando unir, mediante una cuerda, el punto de origen con el punto de destino, en caso de colisión se reduce el número de vértices a sustituir y se repite el cálculo.

En la Figura 4.39-a puede observarse un árbol generado por un algoritmo RRT que aun no ha sido postprocesado y las distintas configuraciones que tendría el vehículo en los distintos nodos del árbol. La Figura 4.39-b remarca las cuerdas calculadas por el algoritmo, obteniendo como válida, libre de colisiones, la cuerda señalada en la figura. En la Figura 4.39-c indica el cambio de configuraciones al sustituir los vértices del árbol por la cuerda válida. Por último la Figura 4.39-d representa la trayectoria final del vehículo.

4.3.3 Planificación de trayectoria locales y globales mediante el uso de Splines

Existe otro algoritmo o método para la planificación local y global de trayectorias de vehículos autónomos. Este algoritmo está basado en la creación de trayectorias locales y globales mediante el uso de *Splines*³⁰ [RLF16]. El principal objetivo de este algoritmo, el cual debido a su importancia para el proyecto se detallara a continuación, es generar un camino suave y libre de obstáculos que empiece desde una posición inicial y se extienda a lo largo de toda la ruta global hasta la posición final. La Figura 4.41 muestra un ejemplo de un

³⁰<https://es.wikipedia.org/wiki/Spline>

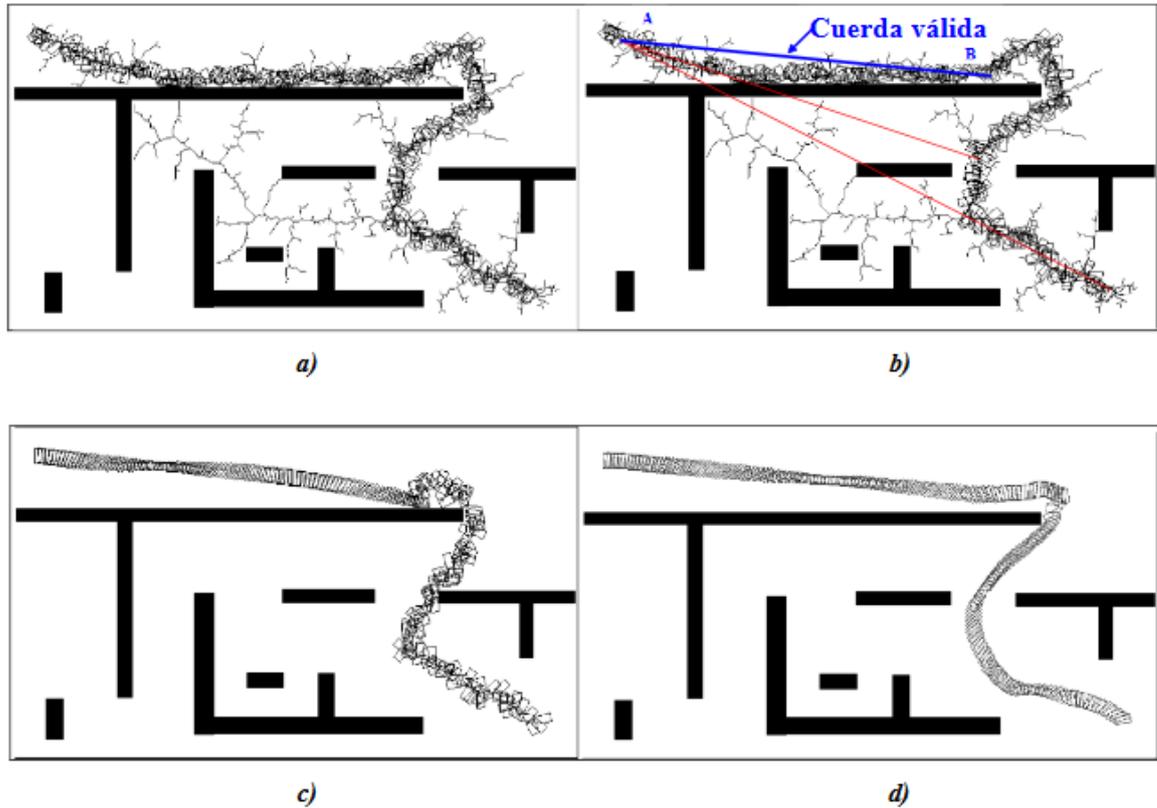


Figura 4.40: Distintas etapas del postprocesado de un árbol RRT. A)Estado inicial. B)Cálculo de cuerdas. C)Sustitución de vértices. D)Árbol final.

ruta global y una serie de rutas candidatas.

Esta técnica de planificación de trayectorias está dividida en cuatro etapas: *construcción*, *localización*, *generación* y *selección*.

4.3.3.1. Construcción

La fase de **construcción**, que pertenece a la planificación global, es responsable de construir la curva de la ruta global. La curva generada para la ruta global está parametrizada según la longitud del arco s y está formada por *splines* cúbicos a partir de los *waypoints* del camino. El proceso de parametrización está dividido en dos etapas: *la parametrización de la curva de la ruta global* y *la reparametrización de la longitud del arco s* .

La parametrización de la curva de la ruta total está relacionada con la longitud de la curva s y dado que esta no es conocida al inicio del proceso y no puede ser interpolada a partir de los *waypoints* del camino al comienzo la estrategia adoptada es fijar la longitud del arco a la distancia acumulativa d entre los puntos (*waypoints*) y posteriormente reparametrizar para obtener s . En la Figura 4.42, en línea discontinua, se observa la distancia acumulada d donde $d_x = \sum_{i=1}^n d_i$.

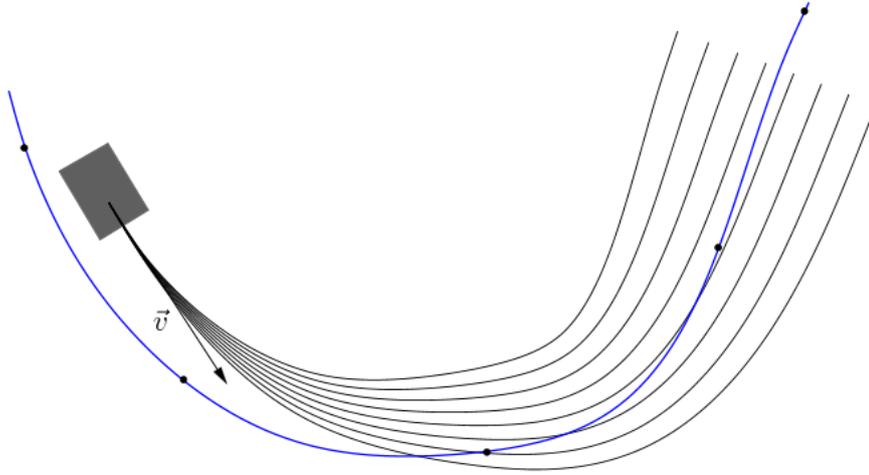


Figura 4.41: Ruta global junto a una serie de rutas candidatas generadas por el algoritmo.

El conjunto de ecuaciones 4.20 define la curva de la ruta global parametrizada respecto a la distancia acumulada entre los puntos d .

$$\begin{aligned} x_i(d) &= m_{x,i}(d - d_i)^3 + n_{x,i}(d - d_i)^2 + o_{x,i}(d - d_i) + p_{x,i} \\ y_i(d) &= m_{y,i}(d - d_i)^3 + n_{y,i}(d - d_i)^2 + o_{y,i}(d - d_i) + p_{y,i} \end{aligned} \quad (4.20)$$

Donde $x_i(d)$ y $y_i(d)$ son las coordenadas cartesianas de la curva de la ruta global parametrizada respecto a la longitud acumulada d , y m , n , o y p son los coeficientes de la *spline* cubica obtenidos de interpolar la *spline*.

Una vez generada la primera curva se pasa a la fase de reparametrización de la curva en base a la distancia s . Esta parametrización se realiza gracias a la formula 4.21 donde \dot{s} es la derivada de la distancia del arco según el parámetro t , y $\frac{dx}{dt}$ y $\frac{dy}{dt}$ son la descomposición geométrica de \dot{s} en la dirección x e y .

$$s(t) = \int_0^d \dot{s}(t) dt = \int_0^d \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt \quad (4.21)$$

Utilizando esta función es posible obtener la distancia d asociada a la longitud del arco s . La Tabla 4.1 muestra un ejemplo del proceso de obtención de la longitud del arco s .

Una vez se ha obtenido la longitud del arco s se realiza otra interpolación de *spline* cubico para obtener la curva de la ruta global parametrizada esta vez según la longitud del arco s .

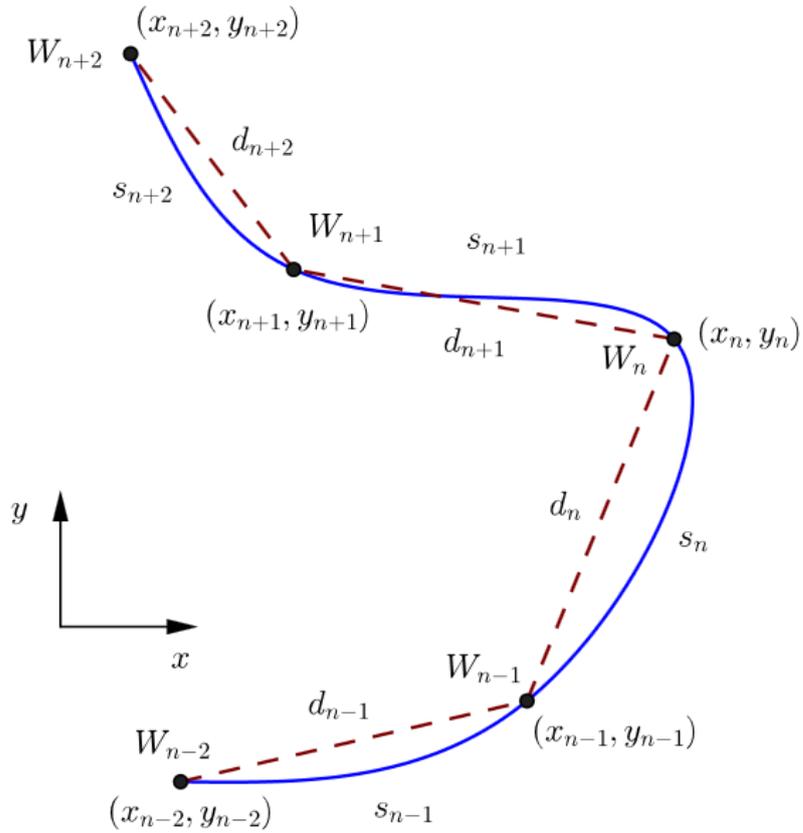


Figura 4.42: Distancia acumulativa d junto a la distancia del arco s . Ruta Global s en línea continua. Distancia acumulativa en línea discontinua.

Waypoints	Parametro d	Longitud del arco s
W_1	0	0
W_2	d_1	s_1
W_3	d_2	s_2
W_4	d_3	s_3
W_5	d_4	s_4
...
W_n	d_{n-1}	s_{n-1}

Cuadro 4.1: Ejemplo de la obtención de la longitud del arco s

Como se puede ver en la Tabla 4.2 se ha realizado una interpolación de la curva *spline* junto con el conjunto de datos x para generar $x(s)$ y otra con el conjunto de datos y para generar $y(s)$.

En el ejemplo presentado para la parametrización de la curva de la ruta global con respecto a la longitud del arco s , se han utilizado los waypoints originales y sus longitudes de arco s . Sin embargo, desde las ecuaciones de la curva de la ruta global parametrizada con

Waypoints	Conjunto x	Conjunto y
(x_1, y_1)	$(0, x_1)$	$(0, y_1)$
(x_2, y_2)	(s_1, x_2)	(s_1, y_2)
(x_3, y_3)	(s_2, x_3)	(s_2, y_3)
(x_4, y_4)	(s_3, x_4)	(s_3, y_4)
(x_5, y_5)	(s_4, x_5)	(s_4, y_5)
...
(x_n, y_n)	(s_{n-1}, x_n)	(s_{n-1}, y_n)

Cuadro 4.2: Interpolación del *spline* con el conjunto de datos x e y .

respecto a d , $(x(d), y(d))$, es posible obtener la longitud de arco de cualquier distancia acumulada d y calcular la longitud de arco parametrizado para cualquier conjunto de distancias acumuladas.

La ecuación 4.22 define la ruta global parametrizada según la longitud de arco s .

$$\begin{aligned}
 x_i(s) &= m_{x,i}(s - s_i)^3 + n_{x,i}(s - s_i)^2 + o_{x,i}(s - s_i) + p_{x,i} \\
 y_i(s) &= m_{y,i}(s - s_i)^3 + n_{y,i}(s - s_i)^2 + o_{y,i}(s - s_i) + p_{y,i}
 \end{aligned} \tag{4.22}$$

Donde $x_i(s)$ y $y_i(s)$ son las coordenadas cartesianas de la curva de la ruta global parametrizada según la distancia del arco s , y m , n , o , y p son los coeficientes del *spline* cubico obtenido de interpolar la *spline*.

4.3.3.2. Localización

Esta etapa está dedicada a localizar el vehículo respecto a la ruta global. Esta etapa calcula la distancia recorrida s_c del vehículo y su desplazamiento lateral q_c . El parámetro s_c y q_c están ilustrados en la Figura 4.43.

Para calcular los valores del parámetro s_c y q_c se realizara la siguiente minimización:

$$\sqrt{(x_c - x(d))^2 + (y_c - y(d))^2}$$

Donde el parámetro s está comprendido entre $0 \leq s \leq s_f$ y x_c e y_c son las posiciones del vehículo en las coordenadas cartesianas y los parametros $x(s)$ y $y(s)$ son un punto de la curva de la ruta global en el eje cartesiano. La minimización busca determinar la longitud del arco s desde el punto más cercado a la curva de la ruta global del vehículo y con dicho punto obtener los parametros s_c y q_c [HWA02].

4.3.3.3. Generación

Una vez construida la curva de la ruta global parametrizada con respecto a la longitud del arco s y localizado el vehículo, se pasa a realizar la **generación** de los trayectos candidatos.

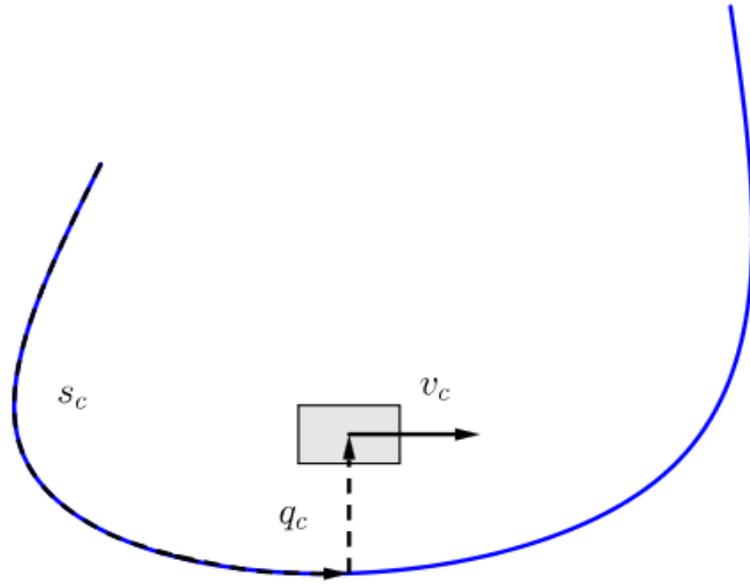


Figura 4.43: Localización del vehículo en relación a la ruta global.

Los trayectos candidatos son generados en tiempo real y están englobados en la planificación local. Los trayectos son inicialmente generados en un sistema de coordenadas curvilíneas definido por los parámetros s_c y q_c y posteriormente son mapeados al sistema de coordenadas cartesianas de la ruta global. En la Figura 4.44 se muestra el sistema de coordenadas curvilíneas s_0q y el conjunto de trayectos candidatos representados en las coordenadas cartesianas.

El conjunto de ecuaciones 4.23 define las ecuaciones utilizadas para generar el conjunto de trayectorias candidatas en el sistema de coordenadas curvilíneo.

$$\begin{aligned}
 q(s) &= \begin{cases} a\Delta s^3 + b\Delta s^2 + c\Delta s + d & s_i \leq s < s_f \\ q_f & s_f \leq s \end{cases} \\
 \frac{dq}{ds}(s) &= \begin{cases} 3a\Delta s^2 + 2b\Delta s + c & s_i \leq s < s_f \\ 0 & s_f \leq s \end{cases} \\
 \frac{d^2q}{ds^2}(s) &= \begin{cases} 6a\Delta s + 2b & s_i \leq s < s_f \\ 0 & s_f \leq s \end{cases}
 \end{aligned}
 \tag{4.23}$$

Donde la función $q(s)$ modela el desplazamiento lateral q de las trayectorias candidatas se-

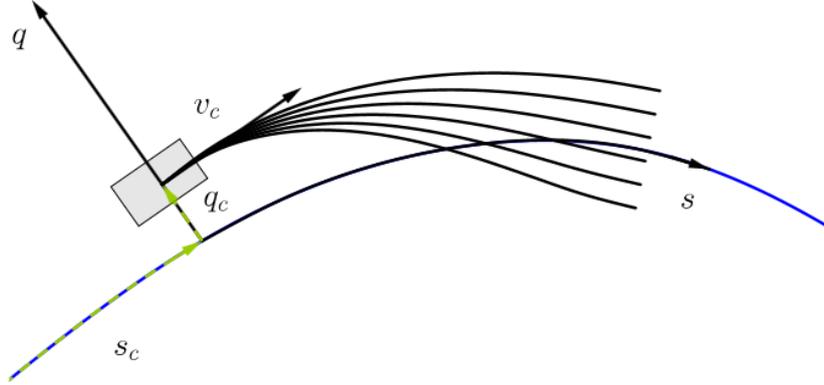


Figura 4.44: Sistema de coordenadas curvilíneas s_0q y el conjunto de trayectos candidatos representados en las coordenadas cartesianas.

gún la ecuación de tercer grado que está en función de la longitud del arco s . Los parámetros a , b , c y d son los coeficientes de la ecuación de tercer grado y el incremento es $\Delta s = s - s_i$. Esta ecuación está diseñada para proveer trayectorias suaves.

Para transformar el conjunto de trayectorias candidatas de un sistema de coordenadas curvilíneo a un sistema cartesiano se debe seguir un enfoque dividido en dos etapas. La primera, se basa en determinar la curvatura k de cada trayectoria candidata en el sistema de coordenadas cartesianas. A continuación, con la información de la curvatura y el conjunto de ecuaciones del modelo del vehículo se calcula la posición de las trayectorias candidatas en el sistema cartesiano. La ecuación 4.24 es utilizada para calcular la curvatura k de las trayectorias candidatas en el sistema cartesiano de la curva de la ruta global [BC04] [MWT10], donde k_b es la curvatura de la curva de la ruta global que se obtiene mediante la ecuación 4.25, y q es el desplazamiento lateral de la trayectoria candidata. Ambos parámetros están en función de la distancia del arco s .

$$k = \frac{S}{Q} \left(k_b + \frac{(1 - qk_b) \frac{dq^2}{ds^2} + k_b \frac{dq^2}{ds}}{Q^2} \right);$$

$$S = \text{sgn}(1 - qk_b);$$

$$Q = \sqrt{\left(\frac{dq}{ds}\right)^2 + (1 - qk_b)^2};$$

$$k_b = \frac{x_b' y_b'' - x_b'' y_b'}{(x_b'^2 + y_b'^2)^{\frac{3}{2}}}$$

Gracias a la curvaturas k de las trayectorias candidatas y mediante el uso de las ecuaciones diferenciales del modelo del vehículo es posible determinar la posición de estas en el sistema de coordenadas cartesiano de la curva de la ruta global.

4.3.3.4. Selección

Del conjunto de trayectorias o caminos candidatos se debe seleccionar uno a seguir por el vehículo autónomo. Como trayectoria elegida se selecciona aquella que minimiza el coste de la combinación lineal (véase Ecuación 4.26) de tres funciones. Estas ecuaciones buscan representar la seguridad intrínseca y extrínseca de la trayectoria elegida por el vehículo.

$$J[i] = w_s C_s[i] + w_k C_k[i] + w_c C_c[i] \quad (4.26)$$

El coste de la función $J[i]$ calcula el coste final asignado a la trayectoria i -ésima. Este coste es el resultado de combinar linealmente el coste de las tres funciones $C_s[i]$, $C_k[k]$ y $C_c[i]$. Cada una de estas funciones busca cuantificar la seguridad de la i -ésima trayectoria según un criterio específico. Los parámetros w_s , w_k y w_c son los pesos dados a las funciones de coste en la combinación lineal. Estos valores pueden ser modificados y deben ser ajustados según las necesidades del proyecto.

La ecuación $C_s[i]$ es la responsable de cuantificar la seguridad de las trayectorias candidatas. Esta ecuación comprueba cada ruta y verifica cuáles están libres de posibles colisiones con obstáculos y cuáles no. La función $C_s[i]$ asigna, para aquellas rutas que están libres de colisión, un valor que está relacionado con la distancia de estas trayectorias a los obstáculos. Para calcular el valor de seguridad se las trayectorias candidatas se utiliza una convolución Gaussiana³¹ (véase Ecuación 4.27) [KCS12] donde $C_s[i]$ es el valor de seguridad de la i -ésima trayectoria, N es el número de trayectorias candidatas, $c[k]$ es la detección de colisiones de la k -ésima trayectoria, $g[i]$ es una distribución gaussiana con promedio cero y una desviación estándar del riesgo de colisión σ y Δq es el incremento del desplazamiento lateral de las trayectorias candidatas. El parámetro de detección de colisión $c[k]$ toma el valor de 1 para trayectorias con colisión y cero para trayectorias sin colisión. La desviación estándar σ es un parámetro de diseño y determina el alcance efectivo de la detección de colisiones para cada ruta.

$$C_s[i] = \sum_{k=0}^N c[k] g[i - k]; g[i] = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\Delta q \cdot i)^2}{2\sigma^2}\right) \quad (4.27)$$

La función $C_k[i]$ es la encargada de cuantificar la *suavidad* de las trayectorias candidatas. Dicha ecuación asigna un valor elevado a las trayectorias que cambian repentinamente de dirección y un valor inferior a las que realizan cambios con mayor *suavidad*. La ecuación 4.28)

³¹<https://es.wikipedia.org/wiki/Convolución>

es la encargada de calcular el valor de *suavidad* de cada trayectoria, donde s_p es la longitud del arco a lo largo del camino, s es la longitud del arco en un instante y k_i es la curvatura de la trayectoria i -ésima.

$$C_k[i] = \int k_i^2(s) ds_p = \int k_i^2(s) Q(s) ds \quad (4.28)$$

Por último, la ecuación $C_c[i]$ es la responsable de cuantificar la consistencia de las trayectorias candidatas. Esta evalúa como de diferentes son las rutas candidatas generadas actualmente respecto a las rutas generadas/seleccionadas anteriores y asigna un valor alto a las rutas que difieren significativamente con la anterior. La ecuación 4.29) calcula el valor de consistencia de las trayectorias candidatas, donde l_i es la distancia euclidiana entre un punto en la i -ésima trayectoria candidata y un punto de la trayectoria seleccionada en el ciclo anterior.

$$C_c[i] = \frac{1}{s_2 - s_1} \int_{s_1}^{s_2} l_i ds \quad (4.29)$$

4.4 Toma de decisiones

4.4.1 Introducción

Un vehículo autónomo debe ser capaz de tomar decisiones que le permitan realizar la trayectoria determinada tal y como un humano las tomaría. El tema de toma de decisiones es el que más preocupa a todos, desarrolladores, clientes y peatones. Con los recientes avances en la tecnología de los vehículos autónomos y la publicidad que se les da a estos vehículos es normal que surjan preguntas. Una de las más polémicas cuestiona la decisión de a quién salvar en un accidente grave, siendo esta es una de las decisiones más complejas a las que se enfrenta un vehículo autónomo³²

En este documento no se entrada a debatir que situación es mejor, dado que este tema ya ha sido debatido en numerosas ocasiones, incluso esta situación fue planteada en el *dilema del tranvía* (véase Figura 4.45)³³, desarrollado por *Philippa Foot*³⁴.

En un vehículo autónomo, como se ha mencionado, esta decisión sucede en el SW que lo controla, en el algoritmo encargado de la toma de decisiones, pero ¿Que reglas debería de tomar?. Un estudio realizado por el Instituto de Ciencias Cognitivas de la Universidad de Osnabrück (Alemania) sugiere que dichas reglas deben ser lo más parecidas a las que sigue un humano.

Pero como se ha mencionado, no se entrara en debate dado que el planteamiento de este

³²<http://omicronno.espanol.com/2017/07/coche-autonomo-salvar-vidas/>

³³https://es.wikipedia.org/wiki/Dilema_del_tranvía

³⁴<https://es.wikipedia.org/wiki/Philippa.Foot>

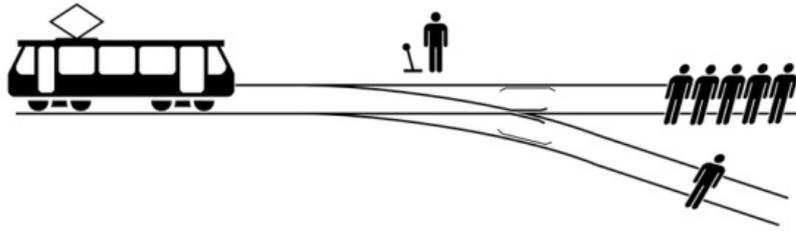


Figura 4.45: Dilema del tranvía.

documento está centrado a nivel tecnológico y no a nivel filosófico, y como tal, se busca que el vehículo autónomo, mediante su HW y su SW, no tenga esta serie de accidentes que son más comunes en el ser humano.

Por eso, la solución a este problema de toma de decisiones y control de vehículos de alto nivel depende de una variedad de áreas y temas de investigación relacionados (véase Figura 4.46).

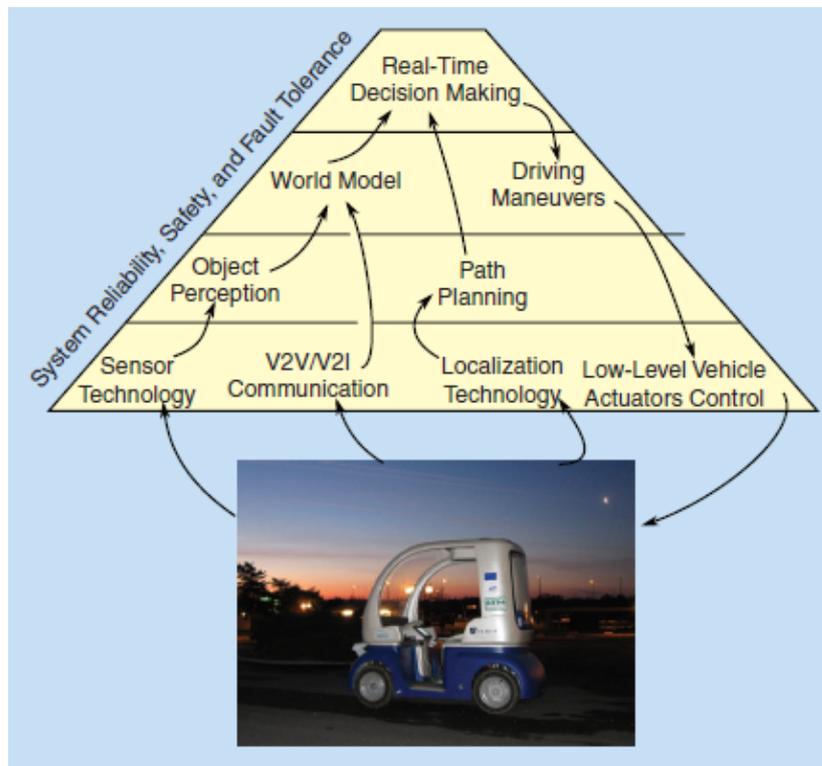


Figura 4.46: Visión general de los temas de investigación de los vehículos autónomos y el alcance de la toma de decisiones en tiempo real.

4.4.2 Toma de decisiones de vehículos autónomos en entornos urbanos

Existen distintos enfoques para el desarrollo de algoritmos de toma de decisiones. Fue en 2007, en el **DARPA Urban Challenge**³⁵, donde se realizó una carrera automovilística autónoma sobre un entorno urbano simulado. En dicha competición surgió el problema de la toma de decisiones en entornos urbanos y fue solventado utilizando distintas soluciones que van desde una máquina de estados finitos o Finite State Machine (FSM) [Mea08b] y árboles de decisión [Mea08a], hasta enfoques heurísticos [Uea08]. Sin embargo, estos enfoques se adaptaron a situaciones muy específicas y simplificadas y, según sus autores, «*no eran sólidos para un mundo variado*» [Uea08].

Actualmente, el enfoque más común para la toma de decisiones es modelar manualmente las reacciones a las situaciones [SBD14]. La mayoría de las máquinas de estado se utilizan para evaluar situaciones y decidir en un marco único. Un ejemplo es la máquina de estado jerárquica utilizada por *Team AnnieWAY* [Gea08] en el Urban Challenge. Las alternativas a las máquinas de estado son, por ejemplo, redes de comportamiento [JSD07], arquitecturas de comportamiento basadas en votación distribuidas [Ros97] o híbridos de modelos de comportamiento y toma de decisiones basados en reglas [Rea08]. Con este enfoque, las tareas complejas individuales, como seguir a un vehículo líder o unirse al tráfico, generalmente necesitan soluciones a medida. *J. Ziegler* utilizó una máquina de estado concurrente jerárquica para generar comportamientos básicos [Zea14]. El ganador del Urban Challenge, el equipo *Tartan Racing*³⁶, con su vehículo *BOSS*, realizó varias comprobaciones definidas manualmente que usaban métricas, por ejemplo, verificando el hueco y si el automóvil podía o no acelerar lo suficientemente rápido, para asegurar que la maniobra de fusión fuese segura [Uea08]. *Team Junior* usó una prueba de umbral, basada en la velocidad y la proximidad, para encontrar las zonas críticas donde el vehículo tenía que esperar [?].

Por otro lado, varios investigadores han abordado el problema de la toma de decisiones para la conducción autónoma a través de la optimización de la trayectoria [ACO15]. Tran y Diehl propusieron un método de optimización convexo que aplican a un vehículo automotriz simulado [TD09]. Sin embargo, su enfoque no considera objetos dinámicos. Gu y Dolan utilizaron la programación dinámica para generar trayectorias que sí consideran objetos dinámicos en el entorno [GD12]. Aunque sus resultados de simulación son prometedores, no demuestran su método en un vehículo real. El método de optimización trayectoria propuesta en [WXZJZ12] optimiza un conjunto de costes que buscan maximizar la eficiencia del camino y comodidad. Este método está demostrado mediante una simple maniobra de adelantamiento, pero los resultados reales de su vehículo están limitados a obstáculos estáticos. El problema clave con la optimización de trayectoria es predecir que harán los vehículos del entorno.

³⁵<http://archive.darpa.mil/grandchallenge/>

³⁶<http://www.tartanracing.org/>

Otro enfoque está basado en el modelo Partially Observable Markov Decision Process (POMDP)³⁷ proporciona una formalización matemáticamente rigurosa del problema de toma de decisiones en escenarios dinámicos e inciertos, como es la conducción autónoma [EGO17]. Desafortunadamente, encontrar una solución óptima para la mayoría de los POMDP es intratable [PT87] [OMC03]. Existen distintas soluciones generales de POMDP que buscan aproximarse a la solución [Thr00] [HKL08] [SV10] [HBL14]. Aunque estos métodos generalmente requieren de varias horas de cálculo para problemas incluso con pequeños estados, observación y espacios de acción, en comparación con escenarios del mundo real [SCH10]. Cabe destacar que ha habido algún progreso reciente que explota la paralelización de las unidades de procesamiento gráfico o Graphics Processing Unit (GPU)s [LK16].

A lo largo de los años han aparecido diversos enfoques del POMDP: J. Wei, entre otros, propuso el uso de un *proceso de decisión de Markov*, o Markov Decision Process (MDP)³⁸, basado en puntos para la conducción y fusión de un solo carril. Se presentó un enfoque basado en Point-based Markov Decision Process (QMDP) para comportamientos de carril único. Defendieron que la incertidumbre en el comportamiento del vehículo líder y las limitaciones de percepción mejoran la robustez. Sin embargo, utilizaron un espacio de estado que se adapta a la conducción de un solo carril [JWL11]. S. Ulbrich y M. Maurer aplicaron una formulación POMDP para manejar los cambios de carril de la carretera mostrando como los POMDP con un espacio de estado discreto pueden usarse para derivar decisiones de cambio de carril. En este método se definió, manualmente, un pequeño conjunto de solo 8 estados para describir situaciones de autopistas. Aunque este enfoque permitió poder procesar los estados, es cierto que limitó la aplicación de este método [UM13]. S. Brechtel, entre otros, formularon un MDP para conducir en carretera utilizando comportamientos que reaccionan a otros objetos [SBD11]. El enfoque POMDP de T. Bandyopadhyay, entre otros, considera la observación parcial de las intenciones de los usuarios de la carretera [Dea13], mientras que Brechtel, entre otros, resuelve un POMDP que tiene en cuenta objetos potencialmente ocultos y la incertidumbre de la observación, considerando las interacciones de los usuarios de la carretera [SBD14]. Estos son, entre muchos otros, algunos ejemplos de las variaciones de POMDP.

4.4.3 Estructura de la toma de decisiones

A continuación se define unas bases y una estructura para la definición de un subsistema de toma de decisiones [FV11].

Un sistema de toma de decisiones y control de vehículos autónomos consta de los siguientes subsistemas funcionales (véase Figura 4.47):

- Subsistema de percepción

³⁷https://en.wikipedia.org/wiki/Partially_observable_Markov_decision_process

³⁸https://en.wikipedia.org/wiki/Markov_decision_process

- Toma de decisiones en tiempo real y control de maniobras de conducción.
- Maniobras de conducción.
- Interfaz del vehículo.

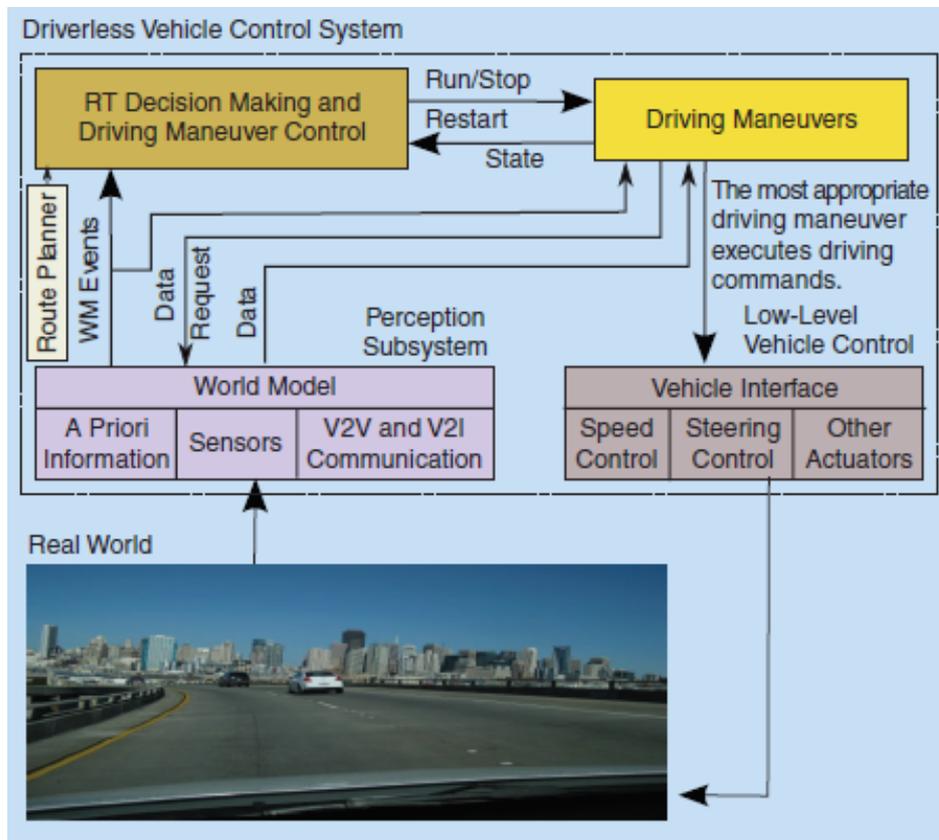


Figura 4.47: Vista simplificada de la arquitectura de software de control y toma de decisiones del vehículo autónomo y su flujo de datos [FV11].

La finalidad del subsistema de percepción es recopilar información del entorno que rodea al vehículo mediante los sensores de a bordo [MMD10] o de sensores externos [LJ09] [Zea09], para administrarla, procesarla y proporcionarla de manera adecuada a los procesos de toma de decisiones en tiempo real y control de maniobras de conducción. Los componentes del subsistema de percepción son (véase Figura 4.48) [FV11]:

- **Una información *a priori***: consiste en toda información que es proporcionada antes de que el vehículo comience su recorrido. Suele definir mapas y rutas, como por ejemplo los Route Network Definition File (RNDF)³⁹ que otorga **DARPA** en sus competiciones.
- **Componentes de los sensores**: componentes SW y HW que proporcionan información obtenida de los sensores integrados.
- **Comunicación**: componentes SW y HW que proporcionan información a partir de las comunicaciones con otros vehículos o infraestructuras.

³⁹<https://en.wikipedia.org/wiki/RNDF>

- Modelado del entorno o del mundo: componente SW que aúna toda la información y genera una interpretación del entorno al vehículo y a los demás subsistemas y que proporciona acceso a toda su información a otros componentes de software a través de una Application Programming Interface (API)⁴⁰.

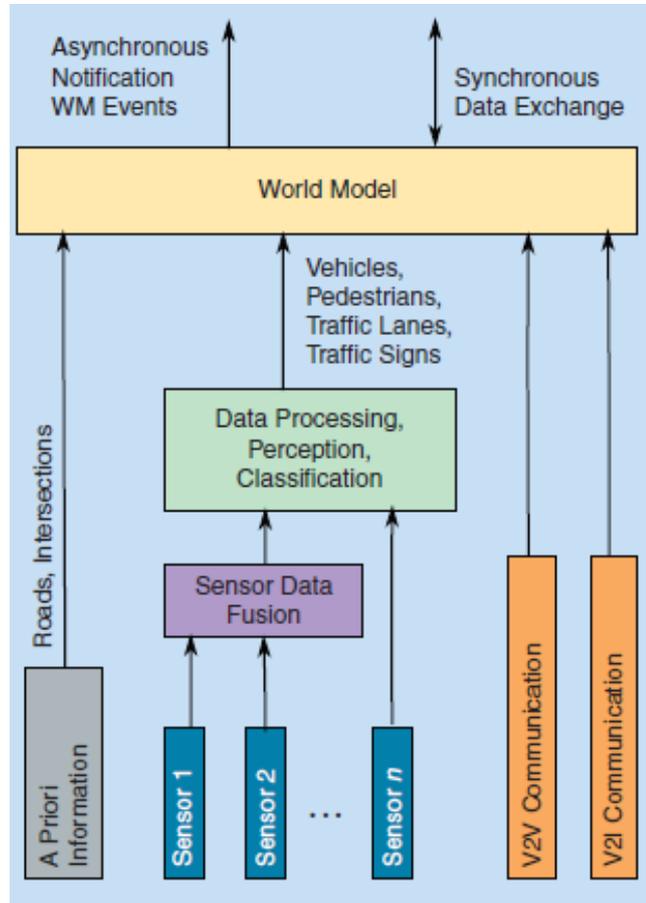


Figura 4.48: Entradas y salidas del modelado del entorno [FV11].

Las decisiones del *subsistema de toma de decisiones en tiempo real y control de maniobras* son tomadas en base a la información proporcionada por el *subsistema de percepción*. Este subsistema decide, mediante un SW, cuales son las maniobras más adecuadas para cualquier situación de tráfico determinado. Por consiguiente, el *subsistema de maniobras de conducción* contiene un conjunto de algoritmos de control, cada uno enfocado a realizar una maniobra en una situación de tráfico específica. Las maniobras de conducción dirigen su salida hacia al *subsistema de interfaz del vehículo*. Por último, el *subsistema de control de vehículos de bajo nivel* contiene componentes de HW, SW, que controlan la velocidad del vehículo, el ángulo de dirección, y otros actuadores.

De todos estos pasos es posible extraer información útil para el proceso de toma de decisiones.

⁴⁰<https://es.wikipedia.org/wiki/Interfaz.de.programación.de.aplicaciones>

Uno de los objetivos del modelado del entorno es proporcionar información para el subsistema de toma de decisiones en tiempo real. Este proceso fusiona los diversos tipos de información y proporciona en un momento determinado una representación precisa y actualizada del entorno del vehículo. A continuación se resume cual es la funcionalidad del modelado del entorno (véase § 4.1.3 para más información):

- Almacena información *a priori*, como carreteras, intersecciones y señales de tráfico.
- Almacena información proporcionada por sensores, como obstáculos, carriles de tráfico y señales de tráfico percibidas.
- Almacena la información obtenida a través de la comunicación con otros vehículos o un centro de gestión del tráfico.
- Combina y actualiza la información *a priori* con la información obtenida mediante sus sensores.
- Notifica a otros subsistemas los eventos relevantes del entorno.
- Proporciona a otros subsistemas un acceso completo a toda la información almacenada.

Existen distintos enfoques para el modelado del entorno, pero uno de los más útiles para el proceso de toma de decisiones es el basado en el principio de Discrete Event Dynamic System (DEDS) [CL09].

Este principio define un conjunto de eventos discretos que pueden indicar la disponibilidad de cierta información, el cumplimiento de ciertas condiciones, o la ocurrencia de un evento determinado que ocurre en el mundo real. Los estados de un evento se modelan como una variable booleana.

Este enfoque utiliza los eventos para notificar a otros componentes de SW sobre el estado de ciertas condiciones predefinidas. A su vez, cada maniobra de conducción requiere que, para poder realizarse de forma segura, se tengan que cumplir ciertas condiciones o que haya cierta información disponible. La definición de eventos durante el modelado del entorno permite definir, de forma flexible, qué información es relevante para la toma de decisiones y, así, proporcionar un mecanismo fácil para el intercambio rápido de información.

El estado de todos los eventos definidos en el modelo del entorno es definido mediante la ***k*-tupla** 4.30, donde cada elemento $w_l (l = 1, 2, \dots, k)$ representa un evento.

$$WM_{events} = (w_1, w_2, \dots, w_k) : w_l \in \{true, false\} \quad (4.30)$$

La *k*-tupla WM_{events} es actualizada cada iteración del algoritmo y los componentes SW son notificados.

Otro sistema importante para la toma de decisiones es el *planificador de trayectorias* ya que este sistema tiene un impacto significativo en la toma de decisiones.

El planificador de trayectorias puede otorgar al sistema de toma de decisiones información sobre la trayectoria (giros, cambios de rasante, cambios de carril, etc...) por lo que conocer esta información por adelantado puede ser beneficioso para el sistema de toma de decisiones. Por ejemplo, adelantar a un vehículo más lento justo antes de un giro o un cambio de carril planeado puede no ser adecuado. Para que ambos sistemas funcionen correctamente es necesario que el planificador de trayectorias proporcione dicha información al sistema de toma de decisiones mediante un conjunto ordenado de acciones a realizar o puntos de interés junto con su posición.

Relacionado también con la toma de decisiones está el sistema de **maniobras de conducción**. Estas son algoritmos de control capaces de maniobrar el vehículo autónomo durante un periodo de tiempo. Todas las maniobras de conducción están estructuradas de una manera común. Sus comportamientos son modelados como autómatas finitos deterministas [JHU06][CL09] (véase Figura 4.48):

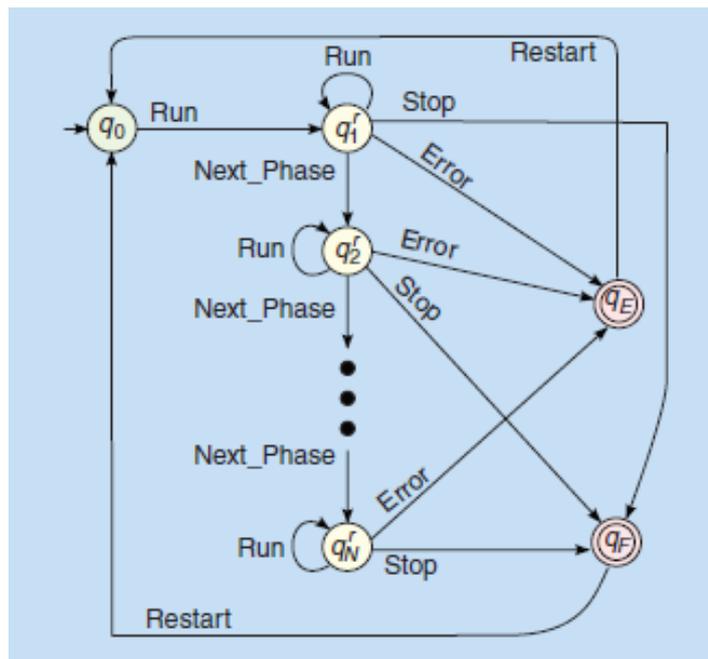


Figura 4.49: Ejemplo de autómata finito de una maniobra de conducción con múltiples estados de marcha. El número de estados de ejecución q_i^r $i = 1, 2, \dots, N$ es igual al número de fases de maniobra de conducción [FV11].

Estas máquinas de estado tienen las siguientes características:

- Un **estado de inicio** q_0 donde el autómata espera la señal de inicio.
- Un **conjunto de estados** de ejecución $Q^{run} = \{q_1^r, q_2^r, \dots, q_n^r\} \subset Q$, donde cada estado representa una fase de una maniobra de conducción.

- **Dos estados finales** $\{q_F, q_E\} = F$. El estado final q_F (finalizado) representa que la maniobra ha sido realizada con éxito, mientras que el estado q_E (error) indica que la maniobra ha sido cancelada debido a un error.
- Un **conjunto de eventos de entrada** Σ , que consisten en, como mínimo: *Ejecutar*, *Parar*, *Reiniciar*, *Error* y la función de transición de estado $\delta : Q \times \Sigma \rightarrow Q$.

Además de estas características, cada maniobra de conducción ofrece un *conjunto de alternativas de ejecución*, que se especifican mediante parámetros discretos. Los parámetros son valores de referencia para los algoritmos de control implementados en los estados de ejecución Q^{run} .

Con las bases ya definidas, se puede elaborar el sistema de toma de decisiones en base a los prerequisites elaborados. La tarea de toma de decisiones se puede especificar de la siguiente manera:

- Un conjunto $M_{all} = \{m_1, m_2, \dots, m_n\}, m \in \mathbb{N}^1$ de todas las maniobras de conducción disponibles que puede realizar el vehículo.
- Una *k-tupla* $(w_1, w_2, \dots, w_k) \in W_{events}$ de eventos del modelo del entorno, $w_l \in \{0, 1\}, l = 1, 2, \dots, k$
- Una indicación de dirección del planificador de ruta $d_i \in D_{route}$.

En este enfoque, la tarea del sistema de toma de decisiones es elegir, basándose en el contexto, la maniobra más adecuada $m_{most_appr} \in M_{all}$ que origina un comportamiento conforme a lo especificado.

La toma de decisiones se descompone en dos etapas que se realizan de forma consecutiva y que deciden sobre que maniobra realizar.

La primera etapa está centrada en la **toma de decisiones sobre maniobras de conducción factibles y críticas para la seguridad** según los eventos producidos. Se define como maniobra *factible* si es posible realizarla sin poner el vehículo, o al entorno, en una situación de riesgo siguiendo las normas de tráfico. Cabe destacar que en cualquier situación de tráfico es posible que se den numerosas maniobras factibles.

Los siguientes aspectos son relevantes para esta etapa:

- Información sobre el entorno del vehículo. Es proporcionada por los eventos del modelo del entorno.
- Conocimiento sobre las reglas de tráfico y cumplimiento de ellas.
- Información sobre la dirección de viaje planificada. Proporcionado por el planificador de ruta.

En la Figura 4.50 se muestra los pasos del procesamiento que se realiza para la selección de una maniobra de conducción factible. De entre todas las maniobras de conducción sólo

las que cumplen sus requisitos son seleccionadas, es decir, son maniobras operativas, por eso dichas maniobras requieren información del entorno que es otorgada en forma de eventos por parte del sistema de modelado del entorno. Para tener en cuenta las reglas de tráfico es necesario añadir más restricciones a las maniobras de conducción ya que el conocimiento sobre las reglas de tráfico está incluido en el primer paso de esta etapa (**DMU1A** en Figura 4.50).

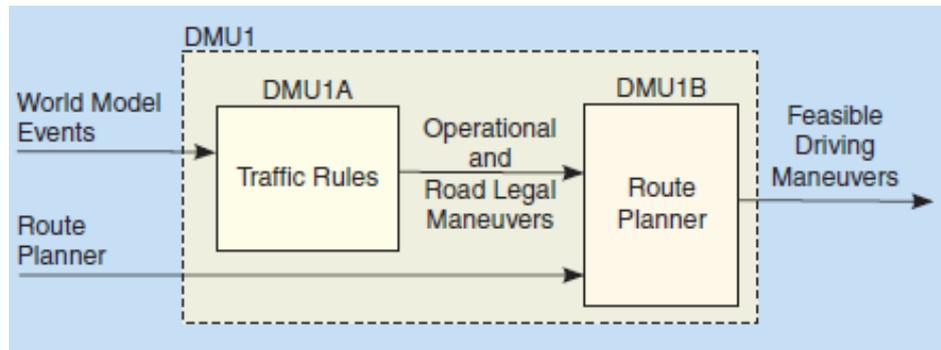


Figura 4.50: La unidad de toma de decisiones para la selección de maniobras de conducción factibles¹ [FV11].

La ruta de viaje planificada desempeña un papel más en la reducción del número de maniobras de conducción candidatas (**DMU1B** en Figura 4.50). Las maniobras de conducción que conducen al vehículo en la dirección incorrecta, o lo maniobran inadecuadamente con respecto a la ruta planificada, se omiten del conjunto de maniobras de conducción factibles. Teniendo en cuenta la gran cantidad de factores que se deben considerar en esta etapa de decisión se requiere de un modelo que permita el diseño y el análisis de un comportamiento operativo altamente complejo. Esta complejidad puede ser resuelta mediante redes de Petri⁴¹ [Pet81].

La unidad de toma de decisiones que se muestra en la Figura 5 puede ser modelada como una red de Petri (véase Figura 4.51) que consta de dos subredes. Cada subred modela las unidades de toma de decisiones *DMU1A* y *DMU1B*, respectivamente.

Después de realizar la ejecución de la red de Petri **DMU1** se obtienen las maniobras de conducción factibles. Después de cada ciclo de toma de decisiones, todos los lugares marcados de la red de Petri se borran, y comienza un nuevo ciclo de toma de decisiones.

En la segunda fase, o etapa, se busca tomar una maniobra, entre todas las factibles, lo más apropiada posible. Esta etapa es la encargada de seleccionar la maniobra que será ejecutada por el vehículo. Cada una de las maniobras de conducción factibles ofrece múltiples alternativas de ejecución, que pueden ser seleccionadas a través de ciertos parámetros de maniobra discretos. Dado que en esta etapa sólo se consideran las maniobras de conducción que se han

⁴¹https://es.wikipedia.org/wiki/Red_de_Petri

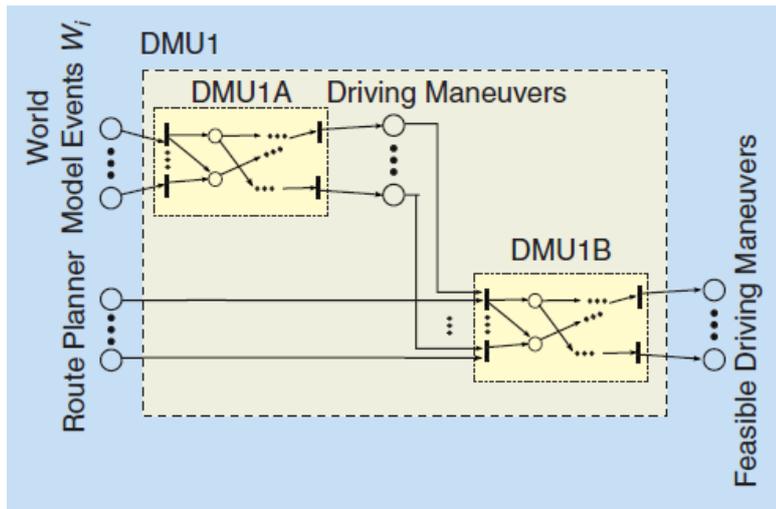


Figura 4.51: Red de Petri para la selección de maniobras de conducción factibles [FV11].

seleccionado como factibles, esta etapa no está considerada como crítica para la seguridad del vehículo o el entorno.

Para seleccionar la maniobra de manejo más adecuada y para ella la alternativa de ejecución más adecuada se aplica Multiple-Criteria Decision-Making (MCDM). Para aplicar este método se define una jerarquía de objetivos a partir de un objetivo general. Dichos objetivos pueden ser medidos a partir de sus atributos. Los objetivos menores pueden ir desde mantenerse dentro de los límites de un carril hasta minimizar las paradas. Por otro lado los atributos son asignados a dichos objetivos siendo una propiedad específica de dicho objetivo. Es posible que un objetivo tenga distintos atributos, incluso que es posible definir la importancia de cada atributo.

Las alternativas corresponden a la ejecución de una maniobra de conducción. Es posible que una maniobra de conducción tenga múltiples alternativas según los valores de sus parámetros discretos. Los parámetros de las maniobras de conducción corresponden, en términos de MCDM, a variables de decisión, donde cada alternativa está representada por un vector de variable de decisión. Se define una función de utilidad que especifica el nivel de valor de un objetivo para una alternativa en relación a sus atributos. Por cada atributo se define una función de utilidad y por consiguiente se define una función de utilidad, unión de los atributos, para cada alternativa.

El último paso es calcular la mejor solución entre las alternativas factibles a partir de los valores obtenidos a partir de su función de utilidad. Se puede aplicar diversas formas del método MCDM para resolver este problema, como por ejemplo: método de dominancia, método de satisfacción, métodos de eliminación secuencial o métodos de puntuación [YH95]

Esta descomposición en dos fases, con objetivos bien diferenciados, conduce a una la realización de unas subtarear con menor complejidad lo que permite un mejor control de

ellas. Cabe destacar que esta división permite verificar y probar cada etapa por separado.

Este método de desarrollo de un sistema de toma de decisiones, como puede observarse, es bastante intuitivo y sencillo de aplicar, y describe la toma de decisión tal y como es realizada por parte de un humano, define una serie de maniobras, unas alternativas a partir de su entorno en tiempo real y decide cual es la más óptima por eso se ha elegido y utilizado, entre todos los mencionados, como ejemplo de definición de un sistema de toma de decisiones.

4.5 Modelo cinemático y control del movimiento

4.5.1 Introducción

Para finalizar este estado del arte es necesario tener en cuenta el control del movimiento del vehículo autónomo. Existen diversas investigaciones que discuten y comparan distintos modelos cinemáticos a partir de datos empíricos [SSDW99] [JRSWC00]. Por ejemplo, Y. Yavin en «*Modelling the motion of an underground mining vehicle*» [Yav05] define un modelo dinámico simple para el control de movimiento de un camión Load Haul Dump (LHD) con carga mediante el uso de ecuaciones Lagrange con restricciones cinemáticas que tienen en cuenta las fuerzas implicadas. En el caso que nos atañe, el de un vehículo articulado con remolque, existen distintas restricciones inherentes a la estructura. Por un lado existen restricciones no-holonómicas en el movimiento y restricciones holonómicas en las articulaciones de dichos vehículos [JMGC12].

Antes de continuar es necesario explicar que son los sistemas holonómicos ⁴² y los no-holonómicos. Dicha clasificación va ligada a la movilidad del vehículo o robot. Se dice que un vehículo es, simplificando, holonómico, si, es capaz de modificar su dirección instantáneamente y sin necesidad de rotar previamente. Por consiguiente, un vehículo similar a un coche no sería holonómico dado que para desplazarse lateralmente debe realizar maniobras previas.

Entrando en detalle es necesario, a su vez, definir que es el número de grados de libertad⁴³ de un robot o vehículo. Se dice que toda articulación presente en un vehículo o robot le proporciona, habitualmente, un grado de libertad. Volviendo al ejemplo del coche, y dado que su desplazamiento es sobre un plano, estaríamos hablando de tres grados de libertad: dos para la posición sobre el plano y uno para la orientación del vehículo.

Muy relacionado con los grados de libertad está el control de ellos. Para obtener dicho control se requiere de un actuador. Cuando se dispone de un actuador por grado de libertad se puede decir que todos los grados son controlables, aunque esto no pasa siempre.

Por consiguiente, diremos que un robot o vehículo es holonómico si todos sus grados de libertad son controlables y no-holonómico si no todos sus grados de libertad son controlables.

⁴²<http://blog.electricbricks.com/2010/07/sistemas-holonomicos/>

⁴³[https://es.wikipedia.org/wiki/Grado_de_libertad_\(ingeniería\)](https://es.wikipedia.org/wiki/Grado_de_libertad_(ingeniería))

En caso de que existan más grados controlados que grados de libertad se habla de un robot o vehículo redundante.

Cabe destacar que cuanto más diferencia exista entre grados de libertad totales y controlables más difícil será controlar el vehículo o robot.

Se habla de restricciones holonómicas⁴⁴ cuando son restricciones que no intervienen las velocidades. Por ejemplo, una rueda moviéndose en una sola dirección representa una restricción holonómica (véase Figura 4.52). Y se define como restricción no-holonómica a aquella que depende de la velocidad con la restricción de que debe ser integrable y no puede ser derivada de una restricción holónoma. Si en el ejemplo anterior se cambia de plano y se tiene en cuenta un plano 2D, este, se transforma en una restricción del tipo no-holonómica dado que tiene una dirección en la que no puede moverse de forma instantánea (véase Figura 4.53).

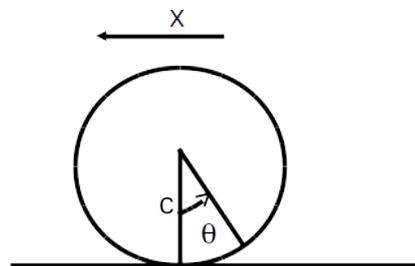


Figura 4.52: Rueda moviéndose en una sola dirección. Restricción holonómica. <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

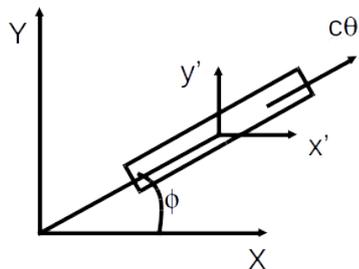


Figura 4.53: Rueda moviéndose en plano 2D. Restricción no-holonómica. <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

4.5.2 Modelo cinemático

Entendemos como **cinemática** a la rama de la física que estudia el movimiento sin considerar las fuerzas o pares que lo causan, en otras palabras, estudia las leyes de movimiento sin tener en cuenta aspectos tales como masas e inercias⁴⁵. Dicha ciencia se limita al estudio de la trayectoria de un vehículo o robot a lo largo del tiempo. Estos cálculos son realiza-

⁴⁴<http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

⁴⁵<http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

dos teniendo en cuenta la posición del vehículo o robot, la velocidad y, en algunos casos, la aceleración. Es una descripción analítica del movimiento espacial en base al tiempo.

Gracias a la *cinemática* es posible estudiar el movimiento respecto a un sistema de referencias. Dicho sistema de referencias determina la relación existente entre la posición y orientación de las partes móviles del vehículo o robot y sus coordenadas articulares (véase Figura 4.54).



Figura 4.54: Relación entre la cinemática directa e indirecta. <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

Entendemos la **cinemática directa** como los modelos matemáticos que permiten describir, y así calcular, la posición de un vehículo o robot a partir de sus componentes fijos y articulaciones.

Esta cinemática, hablando en concreto de vehículos móviles, está centrada en el cálculo de la posición y orientación de la base de un vehículo a partir de la velocidad y orientación de las ruedas. Aunque cabe destacar que la mayoría de los robots, vehículos móviles o no, disponen de un modelo cinemático directo que permite calcular la posición y orientación de sus elementos articulados. En la Figura 4.55 puede observarse la descripción de la posición de un brazo articulado que genera el modelo cinemático directo de la Ecuación 4.31).

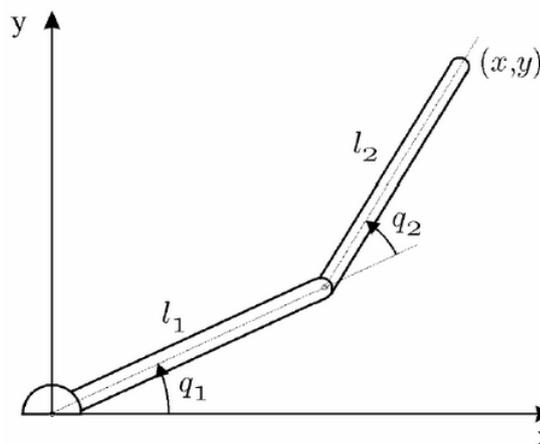


Figura 4.55: Esquema de un robot articulado. <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

$$\begin{aligned}
 x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\
 y &= l_1 \operatorname{sen} q_1 + l_2 \operatorname{sen}(q_1 + q_2)
 \end{aligned}
 \tag{4.31}$$

La **cinemática inversa**, por lo contrario, está relacionada con la obtención de los modelos matemáticos que a partir de una posición específica del actuador final son capaces de obtener los ángulos y el desplazamiento de los actuadores. En robots articulados es posible que se dé el caso de que no sea posible encontrar una configuración factible o situaciones donde se encuentren múltiples soluciones (codo arriba, codo abajo) (véase Figura 4.56). Para solucionar la gran mayoría de estos problemas se utilizan métodos aproximados para el cálculo de las variables.

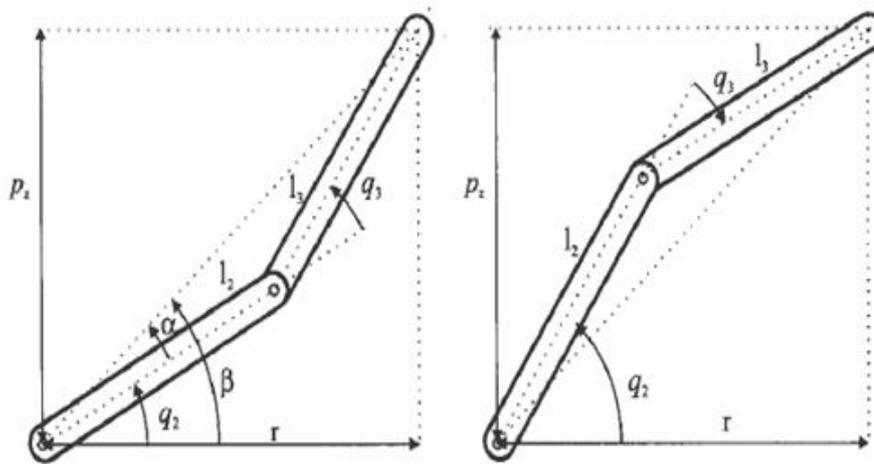


Figura 4.56: Problema del codo arriba, codo abajo en un robot de brazos articulado. <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>

Dado que este proyecto está enfocado a vehículos articulados a continuación se define un modelo dinámico directo para el cálculo de la posición y orientación del vehículo.

Modelar la dinámica de un vehículo articulado en un espacio bidimensional, utilizando un sistema de coordenadas cartesianas estándar (x, y) necesita, como mínimo, cuatro estados [Elh15]. Dos estados para obtener la orientación del vehículo y dos para obtener la posición. Los estados de orientación serán dados por los ángulos de orientación θ_1 y θ_2 del vehículo tractor y del remolque, relativos al eje x , respectivamente. Los estados de posición pueden ser las coordenadas de cualquier punto del vehículo articulado. Para los siguientes ejemplos y para simplificar las operaciones se ha elegido el eje de la rueda del remolque, (x_C, y_C) . En la Figura 4.57 puede verse el modelo del vehículo articulado definido junto con estos estados.

Donde el ángulo de enganche, es decir, el ángulo entre el vehículo tractor y el remolque,

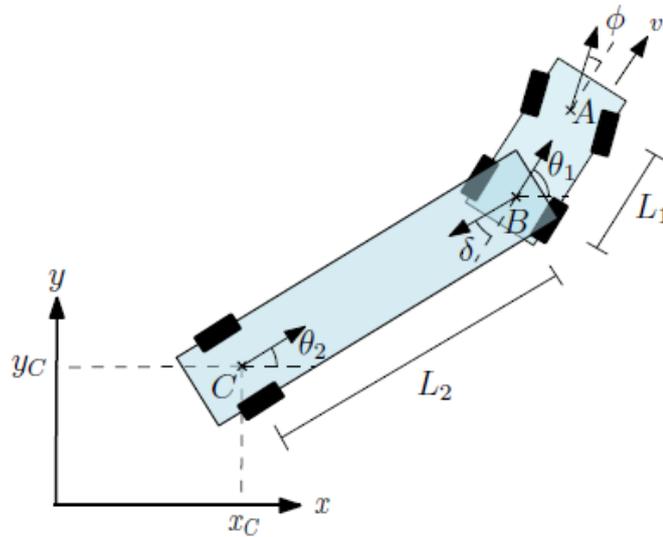


Figura 4.57: Modelo cinemática de un vehículo articulado compuesto de vehículo tractor y remolque[Elh15].

es $\delta = \theta_1 - \theta_2$. La Ecuación 4.32) define el modelo cinemático donde v es la velocidad del vehículo tractor y ϕ su ángulo de dirección.

$$\begin{aligned}
 \dot{x}_C &= v \cos(\theta_1 - \theta_2) \cos(\theta_2) \\
 \dot{y}_C &= v \sin(\theta_1 - \theta_2) \cos(\theta_2) \\
 \dot{\theta}_1 &= \frac{v \tan(\phi)}{L_1} \\
 \dot{\theta}_2 &= \frac{v \sin(\theta_1 - \theta_2)}{L_2}
 \end{aligned} \tag{4.32}$$

De la Figura 4.57 se puede definir la Ecuación 4.33) que establece una relación entre la velocidad angular del ángulo de enganche y el de dirección.

$$\dot{\delta} = \dot{\theta}_1 - \dot{\theta}_2 = \frac{v \tan(\phi)}{L_1} - \frac{v \sin(\delta)}{L_2} \tag{4.33}$$

Por lo que el sistema tendría dos puntos de equilibrio respecto a δ , uno estable y otro inestable. El punto de equilibrio estable es aquel donde el remolque gira sobre el punto de enganche, $\delta = \pi$. Cabe destacar que dicho punto no es posible ya que el remolque colisionaría con el vehículo tractor a medio giro. El punto de equilibrio inestable, por otro lado, es dado cuando el vehículo articulado está circulando alrededor de un punto fijo. Este estado es definido como *estado circulante* (nótese que el caso donde el movimiento es $\phi = \delta = 0$ es

un caso especial donde el punto está en el infinito). El ángulo de dirección ϕ_{circ} en el estado circulante donde $\dot{\delta} = 0$ es obtenido mediante la Ecuación 4.34).

$$\phi_{circ} = \tan^{-1} \left(\frac{L_1 \sin(\delta)}{L_2} \right) \quad (4.34)$$

Incrementando el ángulo de dirección en el estado ($\phi > \phi_{circ}$) se obtiene como resultado un descenso en el ángulo de enganche ($\dot{\delta} < 0$), y una disminución genera el caso contrario, como se puede deducir de la Ecuación 4.34) (v es negativa en la marcha atrás). Por lo tanto, se establece una relación directa para el control ya que el ángulo de dirección junto con la velocidad constituyen las dos variables de control.

Como puede observarse, el número de variables de control es menor que el número de variables de estado, lo que da como resultado que este sistema sea no-holonómico [SM09].

Otro tema importante a tratar es el efecto **jackknife**. Antes de explicar dicho efecto es necesario detallar ciertos aspecto del modelo. Es posible calcular el punto de circulación P alrededor del cual circula el vehículo articulado mediante la intersección de las perpendiculares de todos los pares de ruedas [NA] como puede verse en la Figura 4.58.

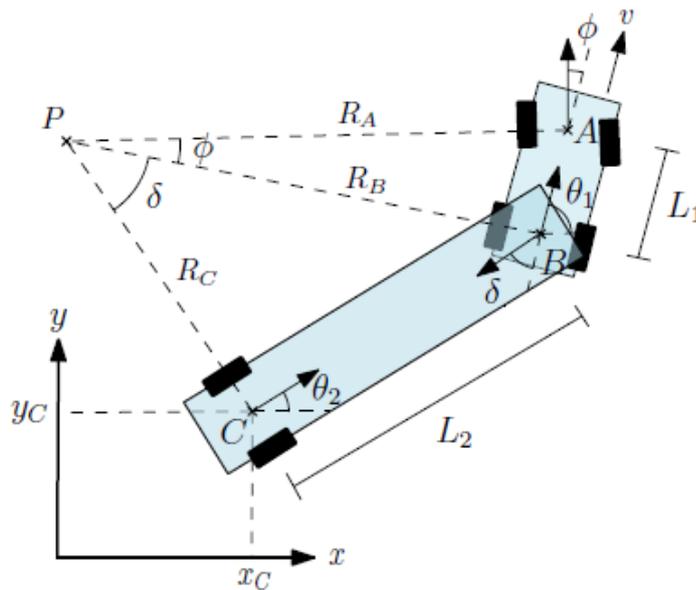


Figura 4.58: Vehículo articulado en estado de circulación con centro de circulación P [Elh15]

El hecho de que el ángulo de dirección tenga un punto máximo plantea la siguiente pregunta: ¿Qué ocurre si se alcanza el estado de circulación, marcha atrás (v negativa), con el ángulo de dirección máximo y se desea enderezar el vehículo?. Este supuesto no sería posible sin una maniobra hacia adelante ya que una disminución en el ángulo de enganche requeriría un incremento en el ángulo de dirección. Esta situación podría llevar a un estado de atasco por parte del vehículo o, en caso de que el ángulo de enganche aumente ligeramente (por

algún deslizamiento o error), podría llevar a un estado donde la trayectoria RA de la Figura 4.58 se cruce con la trayectoria RC antes de cruzarse con RB , (vease Figura 4.59). Si este supuesto caso se da con el ángulo máximo de dirección se produciría el efecto *jackknife* ya que el sistema comenzaría a converger al punto de equilibrio estable donde ambos cuerpos ocupan el mismo lugar llevando a una colisión (véase Figura 4.60). Cabe destacar, que aunque se ha planteado una situación donde el vehículo está realizando una maniobra marcha atrás, es posible que este efecto se de también con una maniobra hacia delante donde se supere dicho ángulo, ya sea al realizar un giro muy pronunciado o por un deslizamiento del vehículo debido a la carga del remolque.

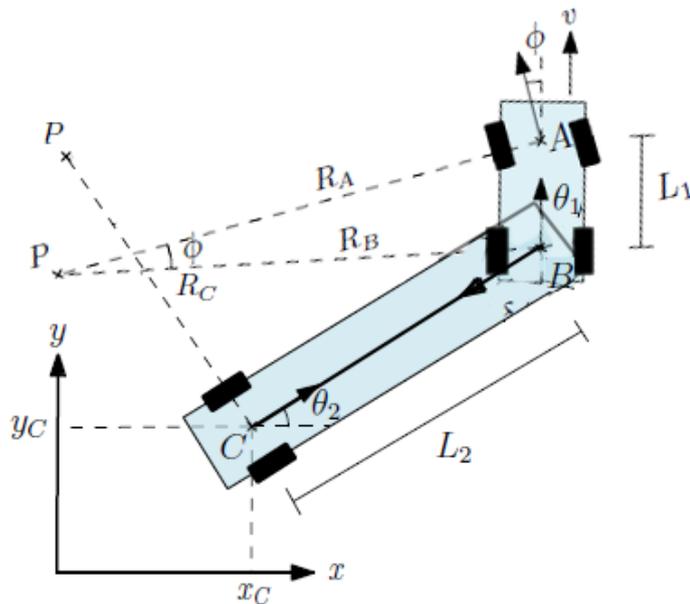


Figura 4.59: Vehículo articulado en estado de circulación donde la trayectoria RA interseca antes con la RC que con RB .

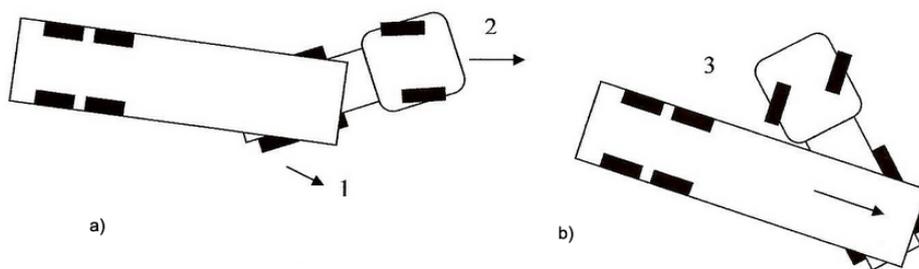


Figura 4.60: Etapas del efecto Jackknife. a) Vehículo en movimiento circular intentando enderezarse. b) El remolque continua su trayectoria haciendo que el ángulo de enganche tienda a un punto de equilibrio https://commons.wikimedia.org/wiki/File:Jackknifing_and_Trailer_swing.jpg

Para evitar esta situación y siempre ser capaz de aumentar o disminuir el ángulo de enganche, el estado de circulación nunca debe alcanzar el ángulo máximo de dirección. Para calcular este ángulo de enganche, denominado **ángulo de enganche crítico** δ_{crit} , y asumiendo que el ángulo de dirección de la Figura 4.58 es el ángulo máximo ϕ_{max} se siguen unas ecuaciones básicas de trigonometría (véase Ecuación 4.34).

$$\delta_{crit} = \sin^{-1}\left(\frac{L_2}{R_B}\right) = \sin^{-1}\left(\frac{L_2 \tan(\phi_{max})}{L_1}\right) \quad (4.35)$$

Por consiguiente, el δ_{crit} depende de la relación de longitud entre el vehículo tractor y el remolque y el ángulo de dirección máximo ϕ_{max} . Una vez conocido el δ_{crit} es posible añadir una restricción al sistema para no sobrepasar dichos límites.

Cabe destacar que existen situaciones donde el ángulo de enganche crítico δ_{crit} no existe por lo que no es posible que se de el efecto *jackknife*. Esta situación se da siempre y cuando la Ecuación 4.36 sea satisfecha.

$$\frac{L_2 \tan(\phi_{max})}{L_1} > 1 \quad (4.36)$$

En este caso, el ángulo de enganche máximo δ_{crit} debe establecerse en 90° . La razón de esto es que el vehículo sería capaz de lograr un giro más brusco con un ángulo de enganche máximo mayor, incluso hasta llegar a un ángulo perpendicular, momento en el que sería capaz de girar sobre el punto, por lo tanto, no existe una necesidad de superar dicho ángulo, aunque sea posible. Cabe destacar que, en los casos reales, no es muy común encontrar un vehículo articulado sin ángulo de enganche crítico por lo que es necesario tener en cuenta el efecto de *jackknife*.

4.5.3 Control de movimiento

Todo sistema de conducción autónoma necesita de un controlador de movimiento robusto que sea capaz de seguir una trayectoria determinada. El movimiento hacia delante de un vehículo articulado forma un sistema estable, lo que significa que es posible ignorar la presencia del remolque, en ciertos casos, y desarrollar un controlador enfocado sólo al vehículo tractor. En el caso de que se quieran realizar maniobras marcha atrás es necesario tener en cuenta el remolque, ya que, como se ha visto anteriormente, el sistema es inestable.

También es necesario entender que en las maniobras marcha atrás el ángulo de dirección del vehículo influye sobre el ángulo de enganche y dicho ángulo es el que influye directamente en como gira el vehículo. A diferencia del movimiento hacia adelante, la influencia del ángulo de dirección sobre la forma en que gira el vehículo es indirecta, y el ángulo de enganche es tomado, en este caso, como la variable de control real para dirigir el vehículo.

A continuación se detallaran una serie de puntos a tener en cuenta, tanto en la conducción con un movimiento hacia delante como marcha atrás. El controlador de movimiento se basa en un bucle de control de dos capas: el bucle de control externo estabiliza el sistema en la ruta (**controlador de estabilización de trayectoria**) y el bucle de control interno estabiliza el ángulo de enganche (**controlador de ángulo de enganche**). El *controlador de estabilización de trayectoria* es el encargado de controlar que el vehículo sigue la trayectoria lo mejor posible. Como se ha visto en la Sección *Planificación de trayectorias* (véase § 4.3), los planificadores de trayectorias generan un conjunto de puntos cercanos. Una buena manera de estimar si estos puntos están lo suficientemente cerca es haciendo la ruta circular más pequeña posible entre ellos, dicha ruta depende del radio de giro mínimo del vehículo. Una vez realizado dicho proceso se conectan los puntos con líneas y si la ruta sigue pareciendo un círculo y no un polígono, entonces, la trayectoria, debería ser lo suficientemente buena para que el controlador de movimiento la siga.

Cabe destacar, que aumentar la densidad de los puntos sería redundante, ya que el radio de giro mínimo del vehículo es fijo y agregar puntos intermedios sólo añadiría ruido y tiempo de cálculo. Por lo tanto se debe calcular la densidad correcta para evitar esta situación.

Otro punto que favorece los cálculos, y por consiguiente a la estabilidad de la trayectoria, es añadir información adicional a los puntos, como la orientación del vehículo y la curvatura del segmento de trayectoria en el que se encuentra, aparte de la posición del vehículo. Esta información sería calculada al principio, tras la planificación de la trayectoria por lo que cada punto contendría la configuración del vehículo en dicho punto, es decir, contendría una captura del vehículo y su remolque.

La estabilización de la trayectoria del vehículo se basa en la unión de tres tipos de errores: error lateral $e_{\bar{y}}$, error de orientación e_{θ} y error de curvatura e_c . Estos tres errores unidos pueden usarse para decidir el ángulo de enganche que debe de tener el vehículo. La Ecuación 4.37 define una función lineal donde $K_{\bar{y}}$, K_{θ} y K_c son parámetros de ajuste. Cabe destacar que son dependientes de la velocidad, aunque, en maniobras de marcha atrás, y dado que la velocidad es mínima, son independientes [RC03].

$$\delta_{ref} = K_{\bar{y}}e_{\bar{y}} + K_{\theta}e_{\theta} + K_c e_c \quad (4.37)$$

Aunque este método está enfocado a maniobras marcha atrás, es posible realizarlo con un movimiento hacia delante. Este método define un punto P_s , o punto de referencia, a una distancia d_s por delante del eje del vehículo tractor o por detrás del eje trasero del remolque P_t . Posteriormente se localiza el punto más cercano de la ruta a dicho punto P_1 y se calcula una trayectoria circular que una el punto P_1 y con el eje trasero o delantero. Esta operación se realiza utilizando un punto de referencia en vez del punto del eje dado que si se buscara el punto más cercano al eje, P_0 , se obtendría una maniobra que supondría una conducción más

agresiva. En la Figura 4.61 se define este esquema en una maniobra marcha atrás.

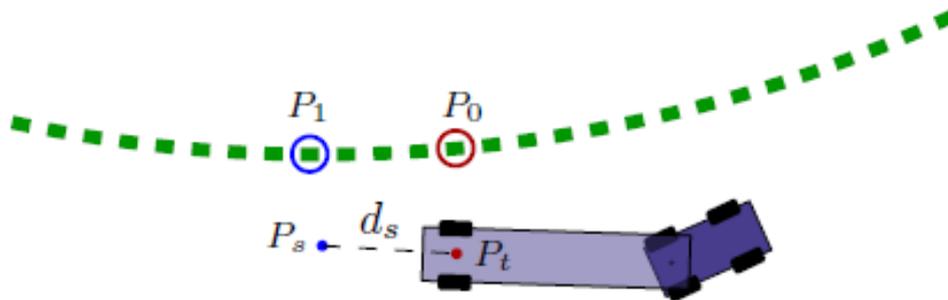


Figura 4.61: Maniobra de acercamiento marcha atrás. Siendo P_t el punto del eje trasero, se define un punto P_s , o punto de referencia y se obtiene el punto más cercano de la trayectoria P_1 o punto de destino. Como puede observarse, si se define P_0 como punto de destino la maniobra a realizar sería menos suave[Elh15].

La distancia d_e marca la suavidad de la trayectoria de corrección por lo que la elección de dicha distancia es muy delicada para una buena conducción ya que el punto importante de este método es la obtención de una trayectoria suave. Si la distancia elegida es muy elevada podría llevar a una trayectoria con colisión ya que no se tendrían en cuenta variaciones de la trayectoria en los segmentos del camino que se encuentran a menos de una distancia d_s además de que se podrían obtener cambios bruscos en la dirección ya que un pequeño cambio en la orientación del remolque podría provocar un cambio sustancial en la ubicación del punto de referencia P_s . Una posible solución sería regular la distancia d_s de forma dinámica.

Otro caso particular de este planteamiento se produce cuando la trayectoria, o ruta, del vehículo se interseca a sí misma como puede verse en la Figura 4.62. Este caso particular podría causar que el punto de referencia se moviese del segmento actual al segmento del cruce obteniendo así un punto de destino erróneo. Una posible solución es la utilización de segmentos de ruta y limitar la búsqueda del punto de destino a dicho segmento. La longitud de este segmento debería ser, por supuesto, más corta que el segmento más corto posible necesario para hacer una intersección. En otras palabras, tiene que ser más corto que la circunferencia del círculo más pequeño que pueda formarse, que es un círculo con un radio correspondiente al radio mínimo de giro r_{min} del vehículo articulado. Además, dado que el punto de referencia sólo debe moverse hacia adelante, el segmento de trayectoria utilizado para determinar el siguiente punto de referencia debe comenzar desde el actual para así evitar obtener puntos ya transitados. Esto significa que la distancia del segmento será constante hasta que la distancia al punto final sea menor a la del segmento, momento en el cual la distancia del segmento será igual a la distancia hasta el punto final.

Volviendo al tema de los tres errores que surgen de seguir una trayectoria se puede definir

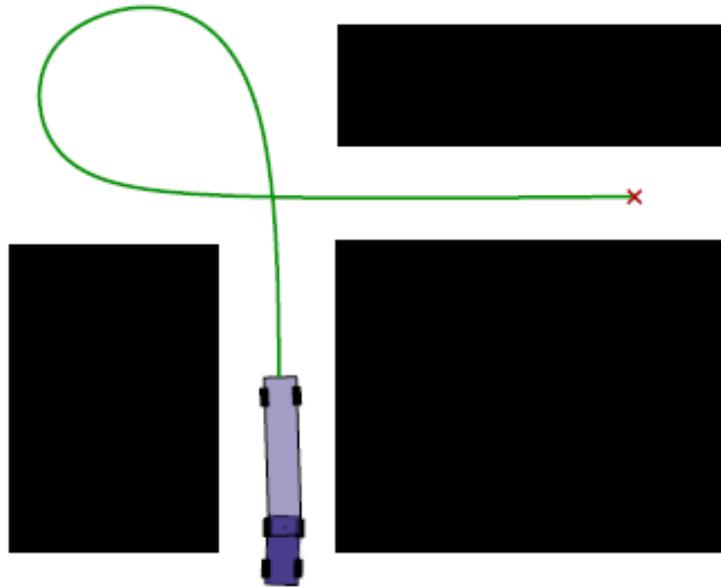


Figura 4.62: Ejemplo de trayectoria que se interseca a sí misma[Elh15].

que el *error lateral* $e_{\tilde{y}}$ se utiliza para que el vehículo articulado converja al punto de destino en la ruta. Para calcular dicho error es necesario considerar el propio sistema de coordenadas del vehículo: el sistema de coordenadas cartesianas con origen en el punto medio del eje. En la Figura 4.63 se ilustra el sistema de coordenadas y la posición de error respecto a un punto (x_r, y_r, θ_r) de la trayectoria.

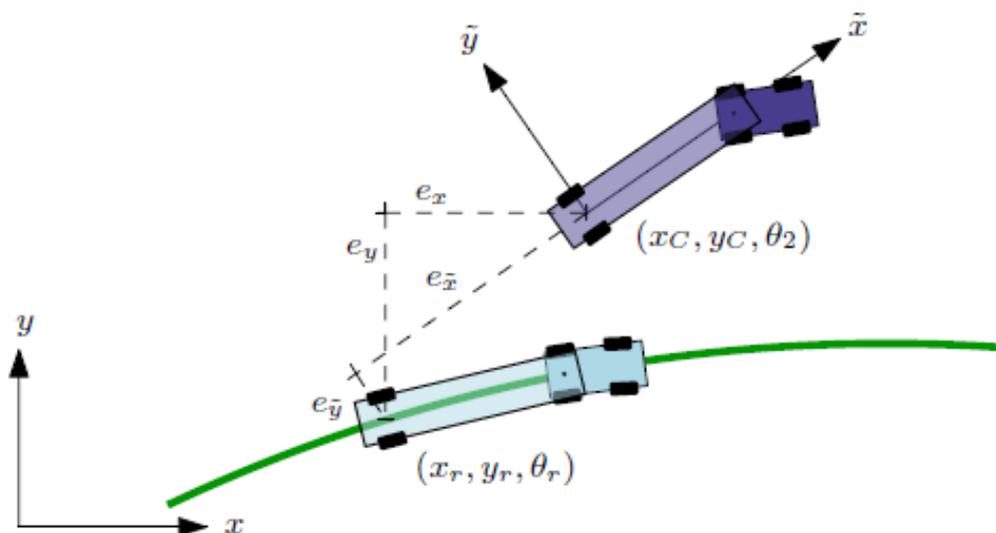


Figura 4.63: Esquema de los errores longitudinales y laterales, $e_{\tilde{x}}$ y $e_{\tilde{y}}$, junto con los errores de posición, e_x y e_y , a un punto de referencia (x_r, y_r, θ_r) en la ruta.[Elh15].

Es posible calcular el *error longitudinal*, $e_{\bar{x}}$, y lateral, $e_{\bar{y}}$, ya que gracias al esquema de la Figura 4.63 es posible determinar la Ecuación 4.38 donde e_x y e_y vienen definidos por la Ecuación 4.39.

$$\begin{aligned} e_{\bar{x}} &= e_x \cos(\theta_2) + e_y \sin(\theta_2) \\ e_{\bar{y}} &= e_y \cos(\theta_2) - e_x \sin(\theta_2) \end{aligned} \quad (4.38)$$

$$\begin{aligned} e_x &= x_r - x_C \\ e_y &= y_r - y_C \end{aligned} \quad (4.39)$$

Dicho error determina en que posición se encuentra el vehículo respecto a la trayectoria. En la Figura 4.63 puede observarse que el error lateral $e_{\bar{y}}$ es negativo ya que el punto de destino está a la derecha. En caso contrario el error lateral sería positivo.

El *error de orientación* e_θ es utilizado para corregir la orientación del remolque en el punto de referencia. Para realizar dicha corrección se tiene que tener en cuenta que la corrección de dirección debe ser la más pequeña para así tomar el camino más corto hacia la ruta, es decir, la magnitud de la corrección no puede superar el valor de π . La Ecuación 4.41 define el error de orientación en más detalle.

$$e_\theta = \begin{cases} \theta_r - \theta_2 & \text{si } |\theta_r - \theta_2| \leq \pi \\ \theta_r - \theta_2 - 2\pi \cdot \text{sign}(\theta_r - \theta_2) & \text{si } |\theta_r - \theta_2| > \pi \end{cases} \quad (4.40)$$

Cabe destacar que el problema surge del hecho de que la orientación está limitada dentro de una región, lo que da como resultado una discontinuidad en los puntos de inicio y fin de la región. El problema surge cuando los puntos de discontinuidad se encuentran, en el tramo más corto, entre la orientación actual y el punto de referencia. Este procedimiento es necesario siempre y cuando la orientación este limitada a una región.

Por último, el *error de curvatura* e_c se utiliza para corregir la curvatura del vehículo y hacer que esta sea la misma que la curvatura del segmento de trayectoria en el que se encuentra. La curvatura del segmento está determinada por el círculo formado por el punto de referencia y dos puntos vecinos. La curvatura de referencia es simplemente la inversa del radio r_p y su dirección depende del signo del radio. La dirección de este círculo siempre será el mismo que la dirección de la curva en la que se encuentra el punto de referencia, es decir, en el sentido de las agujas del reloj cuando la curva gira hacia la derecha y hacia la izquierda cuando gira a la izquierda. Definiremos el radio como positivo para los giros a la derecha y

negativo para los giros a la izquierda. Por lo tanto, el rango de la curvatura de referencia $1/r_p$ es $[-1/r_{min}, 1/r_{min}]$. Esta ecuación muestra que una línea recta tiene un radio infinito y, por lo tanto, su curvatura es cero.

Cada ángulo de enganche del vehículo articulado tiene, por consiguiente, un estado circular correspondiente que puede formarse con un cierto ángulo de dirección. La curvatura del vehículo viene definida por la inversa del radio r_t del círculo que está formado por este estado circular. El radio será positivo para círculos que sigan el sentido de las agujas del reloj y negativo en sentido anti-horario.

La Figura 4.64 ilustra los radios utilizados para determinar las curvaturas del punto de referencia y del vehículo.

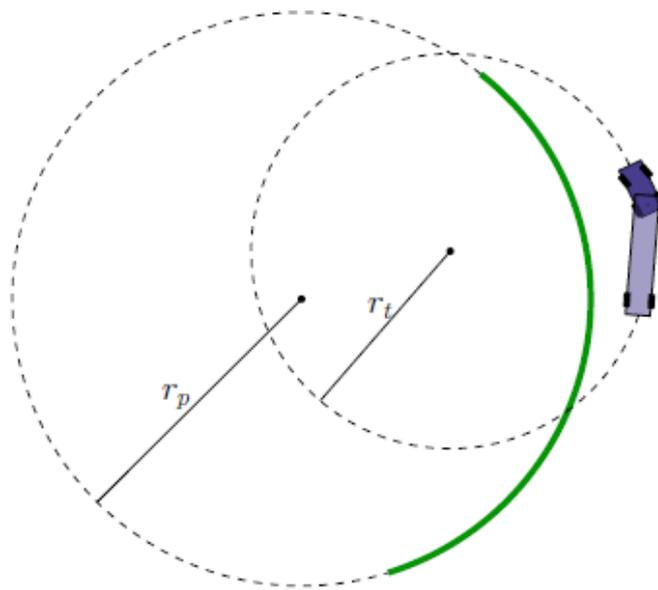


Figura 4.64: Esquema del cálculo del error de curvatura. El radio de giro r_t del vehículo y el radio r_p de la ruta se utilizan para determinar el error de curvatura[Elh15].

La Ecuación 4.41 determina, una vez que se conocen tanto la curvatura de referencia como la curvatura del remolque del tractor, el error de curvatura.

$$e_c = 1/r_p - 1/r_t \quad (4.41)$$

Una vez se han calculado los errores es necesario ajustar los signos de los valores $K_{\bar{y}}$, K_{θ} y K_c . Este procedimiento consiste en definir todos los errores de seguimiento del escenario y aislar cada termino en el controlador con el propósito de determinar si el termino debe generar una corrección negativa o positiva en el Angulo de enganche. Si el signo del error aislado y el cambio del Angulo de enganche son el mismo entonces el parámetro de ajuste tendrá un signo positivo, en caso contrario será negativo.

El método utilizado para ajustar los parámetros inicialmente está basado en posicionar el vehículo articulado paralelo a una ruta recta con un pequeño error lateral. Inicialmente se configuran los parámetros $K_{\bar{y}}$ y K_{θ} para dar un Angulo máximo e enganche de referencia una vez se han alcanzado ciertos errores. El parámetro K_c es, inicialmente, establecido a cero. El propio controlador deberá ser capaz de estabilizarse en el camino recto con sólo el parámetro lateral y de orientación. El parámetro de curvatura sólo entra en juego una vez se ha alcanzado una curva. Una vez eliminados los errores laterales y de orientación el vehículo comenzara a enderezarse y cada vez que se separe de la ruta volverá a corregirse llevando a cabo pequeños cabeceos. Este comportamiento de cabeceo, o tambaleo, es lo que se trata con el error de curvatura. Para ajustar este parámetro se debe realizar una prueba donde el vehículo siga una ruta circular mientras se aumenta gradualmente este parámetro hasta que el cabeceo haya desaparecido.

El controlador de ángulo de enganche entra en juego una vez se ha obtenido el ángulo de enganche de referencia, δ_{ref} del controlador de estabilización de trayectoria, para calcular el ángulo de dirección requerido ϕ . Existen distintos tipos de controladores de ángulo de enganche. A continuación se detallaran brevemente un método lineal y otro no lineal.

El control del ángulo de enganche puede ser realizado mediante la Ecuación 4.42 definida a partir de un controlador Proporcional Integra (PI)⁴⁶, donde K_p y K_I son parámetros de ajuste. Sin embargo, debido a las no linealidades, siempre se existirá un error de estado estacionario para puntos de operación que no sean igual a cero en la parte proporcional del controlador PI, es decir, el controlador P . En la Figura 4.65 se puede ver la respuesta escalonada del controlador P . Por otro lado, la parte I, parte integral, del controladorPI maneja este error, aunque este proceso ralentizara al controlador.

$$\phi = K_p(\delta_{ref} - \delta) + K_I \int_0^t (\delta_{ref} - \delta) dt \quad (4.42)$$

Es posible realizar cambios mediante la pre-modificación del ángulo de enganche de referencia [PU08]. La Ecuación 4.43 define la nueva expresión de modificación correspondiente a nuestro caso.

$$\bar{\delta}_{ref} = \left(1 - \frac{L_1}{K_p L_2}\right) \delta_{ref} \quad (4.43)$$

Esta modificación solos e usa en el termino proporcional del controlador PI. Por lo tanto la Ecuación 4.42 se transforma en la Ecuación 4.44.

$$\phi = K_p(\bar{\delta}_{ref} - \delta) + K_I \int_0^t (\delta_{ref} - \delta) dt \quad (4.44)$$

⁴⁶https://en.wikipedia.org/wiki/PID_controller#PI_controller

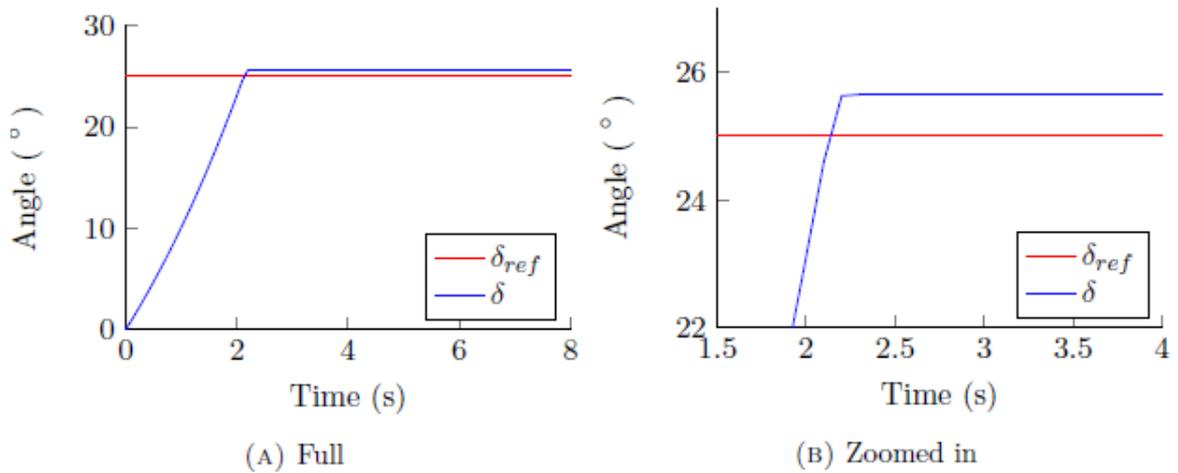


Figura 4.65: Gráfico del controlador P con error de estado estable[Elh15].

En la Figura 4.66 se puede observar la nueva gráfica generada mediante la modificación de la ecuación. Casi se elimina el error por completo.

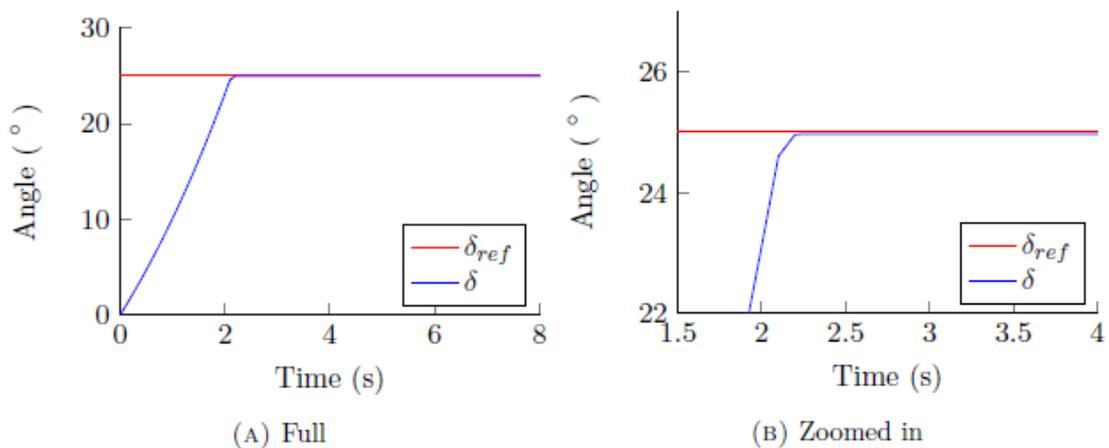


Figura 4.66: Gráfico del controlador P con error de estado estable casi eliminado mediante la modificación del ángulo de referencia[Elh15].

Por último, para eliminar por completo el error es posible incorporar la integral en este punto a costa de un pequeño exceso. Este exceso proviene de la fase de integración y puede corregirse mediante el uso de integrales *anti-windup*⁴⁷. El rendimiento del controlador PI con la parte proporcional modificada y la integral *anti-windup* se puede ver en la Figura 4.67.

Por otro lado, la teoría de **Lyapunov**⁴⁸ es una de las herramientas más utilizadas para el

⁴⁷https://es.mathworks.com/help/simulink/examples/anti-windup-control-using-a-pid-controller.html?stid=gn_loc_drop

⁴⁸https://en.wikipedia.org/wiki/Lyapunov_stability

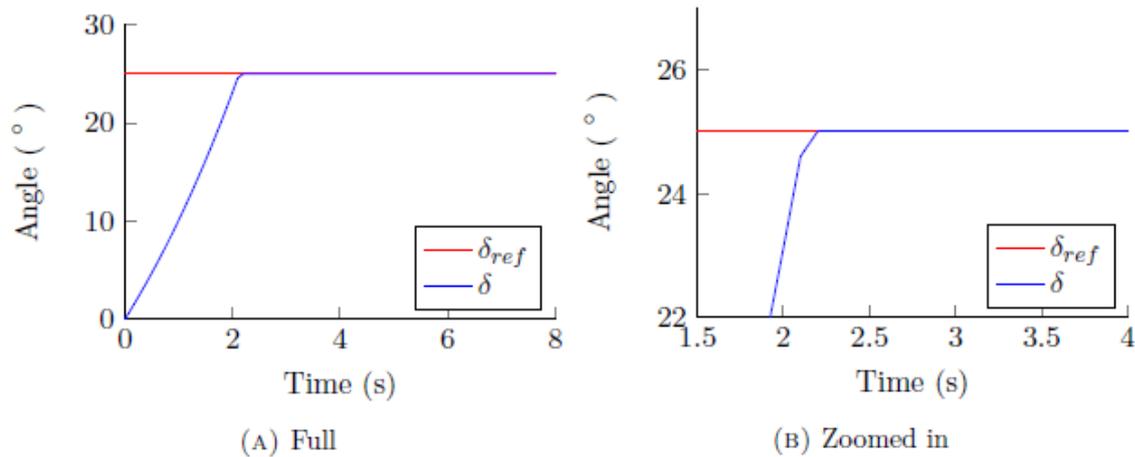


Figura 4.67: Gráfico del controlador P sin error mediante la modificación del ángulo de referencia y el uso de integrales *anti-windup*[Elh15].

análisis de sistemas dinámicos no lineales. El método directo de *Lyapunov* es utilizado en el análisis de la estabilidad y el control del diseño de estos sistemas. En resumen, define que si la energía total de un sistema disminuye, el sistema eventualmente alcanzara un punto de equilibrio, lo que lo convertirá en estable. Toda función de energía que cumpla el teorema del método directo de *Lyapunov* se denomina *función de Lyapunov*.

Dado que el objetivo de un controlador, a grandes rasgos, es llevar a cero un error, es posible utilizar la *función de Lyapunov* para ello. La Ecuación 4.45 establece la *función de Lyapunov* cuya derivada resultante es la Ecuación 4.46.

$$V(x) = \frac{x^2}{2} = \frac{1}{2}(\delta_{ref} - \delta)^2 \quad (4.45)$$

$$\dot{V}(x) = \dot{x}x \approx \dot{\delta}(\delta - \delta_{ref}) \quad (4.46)$$

La aproximación $\dot{\delta}_{ref} \approx 0$ se basa en la suposición de que la tasa de variación en el ángulo de referencia de enganche δ_{ref} es baja en comparación con la del ángulo de enganche real, δ . La Ecuación 4.47 define $\dot{\delta}$ para que se cumpla la condición de que $\dot{V}(x)$ sea negativo, donde K es un parámetro de ajuste positivo.

$$\dot{\delta} = -K(\delta - \delta_{ref}) \quad (4.47)$$

De la Ecuación 4.33 es posible extraer el ángulo de dirección ϕ como la Ecuación 4.48.

$$\phi = \tan^{-1} \left(\frac{L_1}{v} \left(\frac{v}{L_2} \sin(\delta) + K(\delta_{ref} - \delta) \right) \right) \quad (4.48)$$

Así se establece el controlador *Lyapunov* con un solo parámetro de ajuste. Observe también cómo el sistema converge al estado de circulación a medida que disminuye el error de ángulo de enganche. El ángulo de dirección se vuelve ϕ_{circ} cuando el error de ángulo de enganche se pone a cero. La Figura 4.68 muestra la gráfica del uso del controlador *lyapunov*.

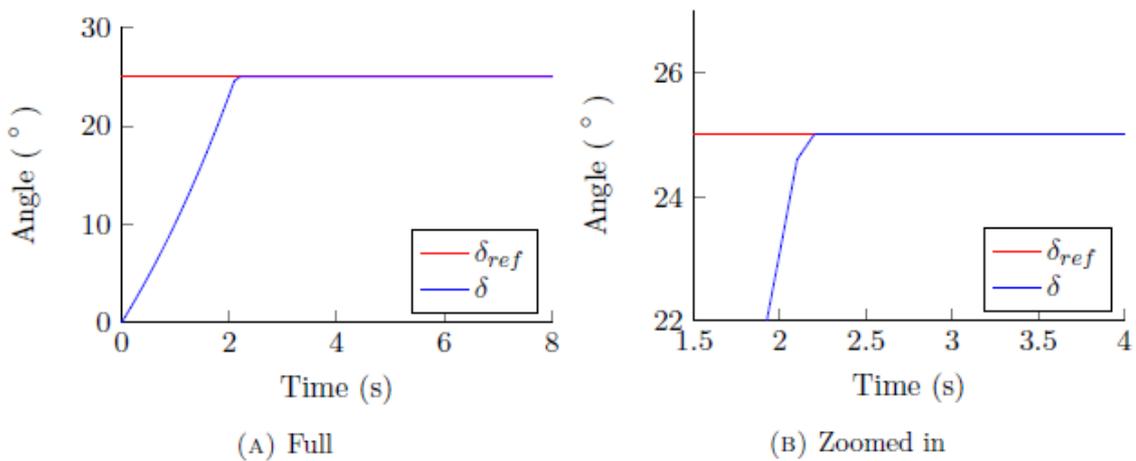


Figura 4.68: Gráfico del controlador de Lyapunov[Elh15].

En resumen, estas son las partes más importantes de la estabilización del controlador de movimiento de un vehículo autónomo, el seguimiento de la trayectoria y la corrección de errores. Existen otros puntos interesantes que no serán detallados en este documento, como el control de la velocidad del ángulo de enganche y el control de frenado.

Capítulo 5

Soluciones comerciales

EN la actualidad existe un amplio mercado relacionado con el automatismo de vehículos que abarca todos los procesos de la fabricación, diseño y desarrollo de un vehículo autónomo (véase Figura 5.1). En el siguiente apartado se detallara el estado actual del mercado detallando así los vehículos autónomos que existen actualmente haciendo especial mención a los camiones. También se mencionaran los sensores existentes y las compañías de HW que los desarrollan. Por último, y para cubrir todos los aspectos del mercado, se detallaran los entornos de desarrollo utilizados tanto para desarrollar nuevos algoritmos como para controlar los vehículos.

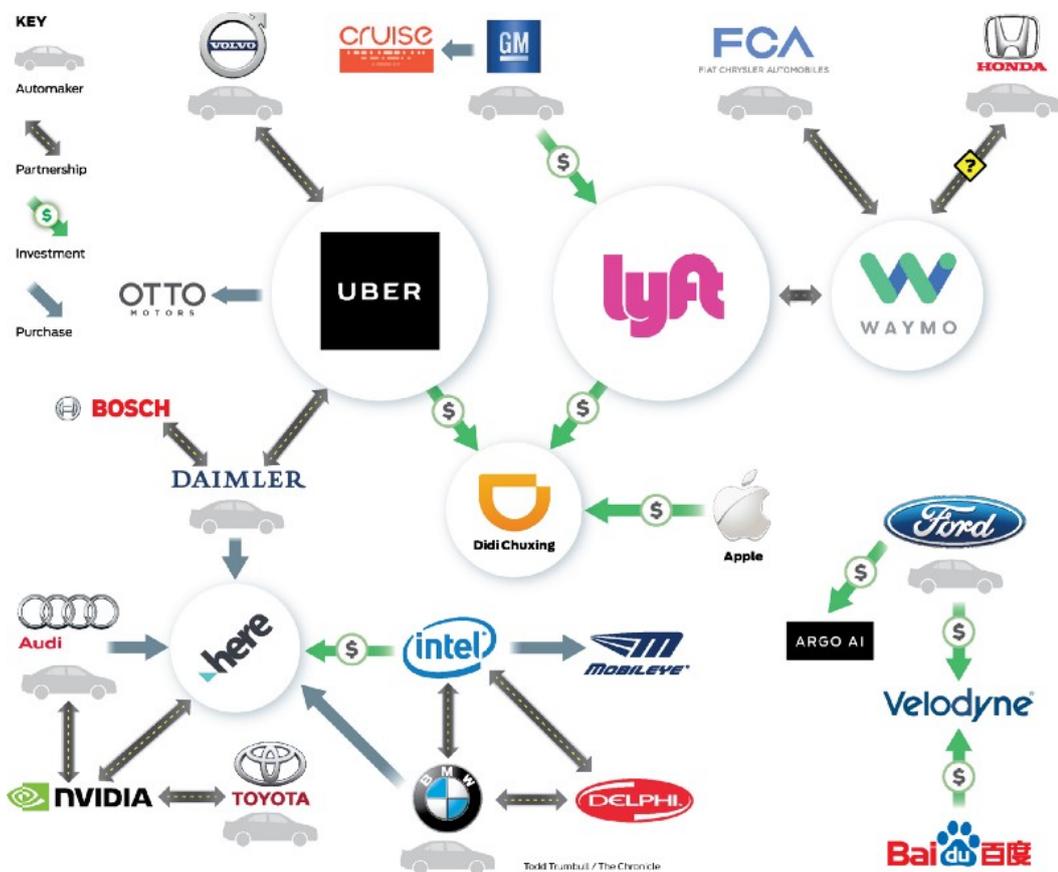


Figura 5.1: Mapa del mercado actual de vehículos autónomos.

5.1 Introducción

En la actualidad, cuando a alguien inexperto en la materia se le habla de vehículos autónomos enseguida lo relaciona con **Tesla** o **Google**, pero no es de sorprender que existan casi medio centenar de grandes empresas que están invirtiendo grandes cantidades en realizar I+D en este área.

El número de acuerdos empresariales relacionados con los vehículos autónomos ha superado a los acuerdos de otras tecnologías. Este área de investigación ha movido \$3.994 millones en el año 2017 frente a los \$964 millones que han sido utilizados para investigar otras tecnologías automovilísticas y se ha recuperado un 76 % de lo invertido (véase Figura 5.2).

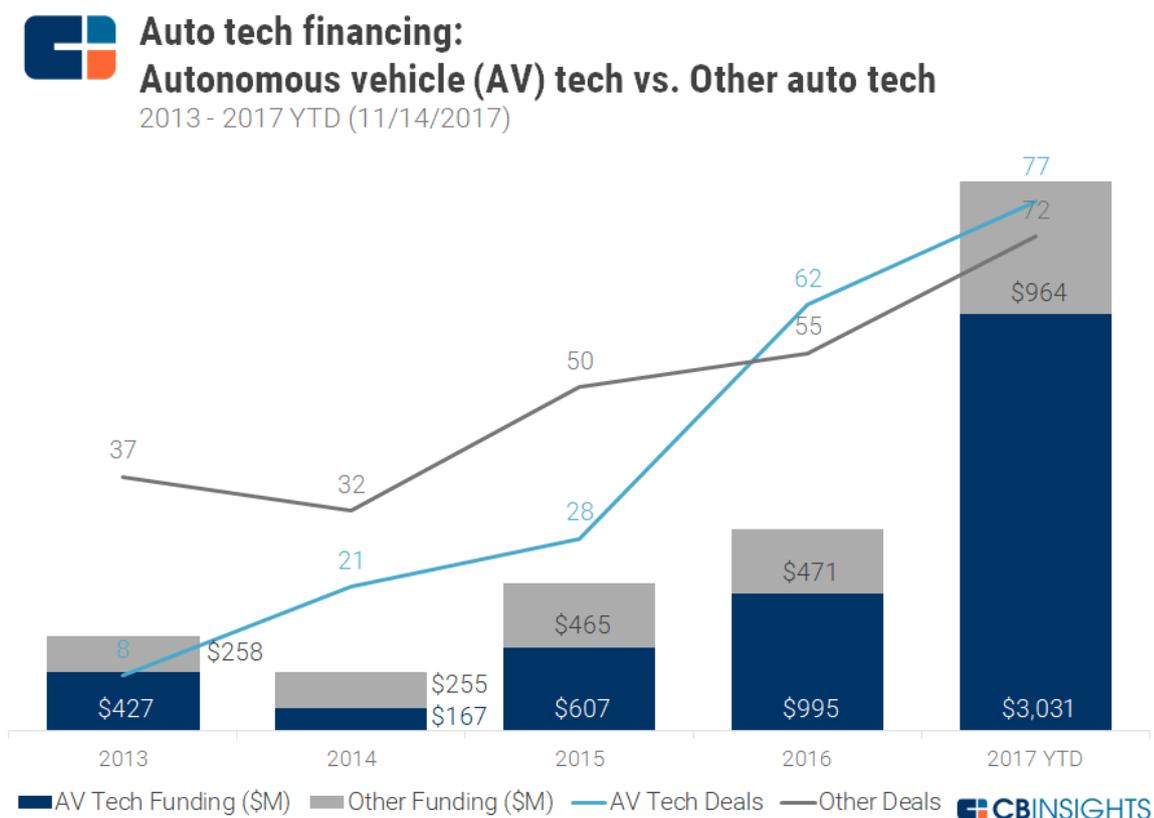


Figura 5.2: Financiación del mercado automovilístico en 2017.

La empresa China **NIO**, antes conocida como **NextEV**, ha invertido hasta la fecha unos \$2.200 millones en la creación de vehículos eléctricos y autónomos aliándose con grandes empresas como **Mobileye** (comprada recientemente por **Intel**¹), **NVIDIA**² y **NXP**³ habiendo recuperado más de \$1.600 millones en 2017.

¹<http://www.eleconomista.es/tecnologia/noticias/8217243/03/17/Intel-acuerda-la-compra-de-la-israeli>

²<https://twitter.com/nvidia/status/840267131354677248?lang=es>

³<https://smartcitynews.global/nio-to-join-forces-with-mobileye-nvidia-and-nxp/>

Otras empresas como **Nauto**⁴, empresa que ofrece un sistema que mejora la conducción y la hace más segura mediante la recopilación de datos de la conducción, ha recaudado \$159 millones⁵, **LeddarTech**, empresa que fabrica sensores LiDAR de estado sólido, ha invertido \$101 millones en investigación⁶ y Peloton Technology, empresa centrada en la conexión entre vehículos, *platooning*⁷ y la automatización, ha recaudado \$60 millones⁸.

A continuación se detallan las últimas alianzas, compra-ventas, creaciones y fusiones de empresas relacionadas con la tecnología:

- **Apple**: con su proyecto Titan⁹ quiere entrar en el mercado de los vehículos autónomos.
- **Audi**: crea una filial de conducción autónoma. La empresa pretende dotar a su modelo A8 de una automatización de nivel SAE 3. *Audi* es parte del consorcio alemán, que incluye a **Daimler** y **BMW**, que compro la tecnología **HERE** de **Nokia** por \$3.100 millones y recientemente ha demostrado interés en diseñar una especificación abierta para la recopilación y transmisión de datos obtenidos a partir de los sensores de los vehículos¹⁰.
- **Autoliv** y **Volvo**: ambas empresas han unido esfuerzos para comercializar sus primeros sistemas de asistencia al conductor en 2019.
- **Baudi**: esta empresa se asoció con **BWM** para lanzar un prototipo semiautomático en 2014, pero se separaron tras desacuerdos entre las compañías. En 2016 se marco el objetivo de cinco años para la creación en masa de vehículos autónomos.
- **BWM**, **Intel** y **Mobileye**: las tres empresas han creado una alianza para la creación de una plataforma abierta basada en estándares para vehículos autónomos. *Intel* adquirió *Mobileye* por \$15.300 millones y en 2016 creó su Autonomous Driving Group (ADG) y destinó \$250 millones para inversiones. A su vez, *Mobileye* tiene una plataforma de mapeo llamada Road Experience Management (REM)¹¹ que incluye a empresas como **Nissan**, **VW** y *BWM*.
- **BOSCH** y **Mercedes**: se unieron a principios de 2017 para desarrollar vehículos de nivel SAE 4 y 5.
- **Continental AG.**: proveedor alemán de automóviles. En 2017 abrió un laboratorio de I+D y dijo que invertiría \$300 millones para expandir las tecnologías de vehículos híbridos y eléctricos.

⁴<https://www.nauto.com/>

⁵<https://www.tinuku.com/2017/07/Nauto.html>

⁶<https://www.prnewswire.com/news-releases/leddartech-leads-the-charge-in-solid-state-lidar->

⁷<https://es.wikipedia.org/wiki/Platooning>

⁸<http://www.finmes.com/2017/04/peloton-technology-closes-60m-series-b-funding-round.html>

⁹<http://quetediga.com/proyecto-titan-el-coche-electrico-de-apple-podria-ser-realidad-en-2019/>

¹⁰<https://www.slashgear.com/here-makes-bold-play-to-define-common-language-for-self-driving-cars>

¹¹<https://www.mobileye.com/our-technology/rem/>

- **Daf, Daimler, Iveco, Man, Scania y Volvo:** en 2016 se realizó la primera carrera transfronteriza de convoyes de *pelotones*¹².
- **Delphi:** se ha aliado con **Quarnergy** para desarrollar un LiDAR de estado sólido. A finales de 2016 se asoció con *Mobileye*.
- **DiDi Chuxing:** absorbió a la unidad de **Uber** en China y abrió su propio laboratorio de IA enfocado a sistemas de conducción inteligentes.
- **Ford:** creó en 2016 una filial llamada **Ford Smart Mobility LLC**. Está centrando sus esfuerzos en pruebas en entornos menos amigables. Tiene previsto desplegar vehículos autónomos para 2021.
- **GM y LYFT:** la empresa *GM* invirtió \$500 millones en *Lyft*.
- **Honda:** ha introducido el sistema semiautomático de ADAS en su modelo Civic básico y ha entablado conversaciones con **Waymo** de **Alphabet** para implementar un sistema de auto-conducción.
- **Huawei:** ha creado su propio equipo de I+D para la investigación de vehículos autónomos.
- **Hyundai:** quiere desarrollar un sistema más asequible.
- **Jaguar y Land Rover:** desean implementar la conducción asistida pero no eliminar al conductor de la conducción. Disponen de una flota de 100 vehículos para la investigación.
- **Magna:** esta empresa se asoció con **Innoviz** para desarrollar LiDAR de estado sólido.
- **Microsoft:** se ha aliado con *Volvo* para aprovechar la tecnología *HoloLens* en los vehículos autónomos.
- **Nissan / Renault :** quiere poseer 10 vehículos con una funcionalidad autónoma significativa para 2020.
- **NVIDIA y Paccar:** la empresa *NVIDIA* se ha asociado con el fabricante de camiones *Paccar* para desarrollar camiones autónomos.
- **Samsung:** informó al Ministerio de Transportes de Corea del Sur de su intención de probar vehículos autónomos.
- **SoftBank:** se centra en el avance de la tecnología de conducción autónoma, especialmente relacionada con el público y servicios comunitarios tales como autobuses, así como el transporte por carretera.
- **PSA Groupe:** informó que dos automóviles Citroën habían recorrido 300km sin supervisión llegando a un nivel SAE 3. En Mayo de 2017 anunció una alianza con **nuTonomy**, spin-off del MIT, para instalar sistemas en vehículos **Peugeot 3008**.

¹²<https://connectedautomateddriving.eu/mediaroom/european-truck-platooning-challenge/>

- **TATA Elixsi**: definió un sistema de aparcamiento automático.
- **Tesla**: las últimas noticias de la empresa afirman que su nuevo HW permite una *autoconducción* total del vehículo. A finales de 2017 la empresa anunció su nuevo camión eléctrico.
- **Toyota**: en 2015 anunció un presupuesto de \$1.000 millones para investigación y creó su Toyota Research Institute (TRI). En 2016 la empresa invirtió \$22 millones en la Universidad de Michigan para impulsar la investigación robótica y la autoconducción.
- **Uber**: la compañía adquirió **Otto**, una empresa puntera de camiones autónomos.
- **Valeo**: la empresa aumentó sus ventas un 22 % en el primer trimestre de 2017 gracias al impulso de la compañía en las tecnologías ADAS.
- **Volkswagen**: afirmó que su objetivo era llevar dichas tecnologías al mercado más rápido que sus competidores y afirma que los vehículos autónomos serán comunes en 2025. En 2016 dio a conocer su nueva marca **Moia**¹³.
- **Waymo**: la empresa presentó en 2017 su nuevo HW de autocontrol diseñado a medida y planea vender un sistema integrado de HW y SW.
- **Yutong**: ha probado con éxito los autobuses sin conductor en una carretera interurbana en la provincia central china de Henan.

Como se puede observar el mercado de los vehículos autónomos está en auge y se espera que crezca con los años. Un punto en el que suelen coincidir todas las grandes empresas es que alrededor del año 2020-2025 estos productos estarán a la orden del día.

5.2 Camiones autónomos

Centrándose en el mercado de los vehículos de transporte se puede encontrar una amplia variedad de proyectos y productos. El mercado relacionado está dividido: **platooning** y **minería**, aunque también existen otros proyectos.

Existen distintas empresas centradas en el *platooning* de vehículos pesados o articulados. El *platooning*¹⁴ se basa en agrupar vehículos formando pelotones para así aumentar la efectividad. Esta agrupación se basa en comunicar dichos vehículos para reducir su distancia y controlar sus velocidades de forma conjunta. Los mayores beneficios del *platooning* son:

- Uso eficiente del combustible debido a la reducción de la resistencia del aire.
- Reducción de la congestión del tráfico.
- En viajes largos el control por parte del operario se ve reducido drásticamente.
- Menos colisiones de tráfico debidas a cambios bruscos en la velocidad.

¹³<https://www.volkswagenag.com/en/brands-and-models/moia.html>

¹⁴[https://en.wikipedia.org/wiki/Platoon_\(automobile\)](https://en.wikipedia.org/wiki/Platoon_(automobile))

Tanto es el auge de este mercado que en Abril de 2016 se realizó la **European Truck Platooning Challenge**, esta competición, organizada por Holanda, unió a grandes empresas de Europa con un mismo propósito: mostrar un modelo de transporte más seguro, limpio y eficiente. En este reto participaron cinco fabricantes europeos y se presentaron distintos sistemas para la conexión entre vehículos (véase Figura 5.3).



Figura 5.3: Foto del European Truck Platooning Challenge 2016. Se puede observar como los camiones traseros muestran el texto «*This truck is connected to the next truck*».

Aunque no sólo en Europa se investiga en este terreno, en Estados Unidos, la empresa **Peloton Technology**¹⁵ está centrada en la automatización y conexión de vehículos. Actualmente está desarrollando un sistema de *platooning* para vehículos de transporte que permite conectar pares de vehículos para que estos operen a corta distancia mejorando así la seguridad y la eficiencia del combustible (véase Figura 5.4).

Esta empresa ha desarrollado una tecnología que utiliza un DSRC de 5.9GHz para establecer comunicaciones de Vehicle-to-Vehicle (V2V) entre pares de camiones¹⁶ (véase Figura ?? y Figura 5.6). Este sistema combinado con la detección de colisiones basada en radar hace posible mantener una distancia mínima de seguridad. El resultado es que los camiones son capaces de circular en una formación de pelotón, mejorando la aerodinámica para reducir el

¹⁵<https://peloton-tech.com/>

¹⁶<https://peloton-tech.com/how-it-works/>

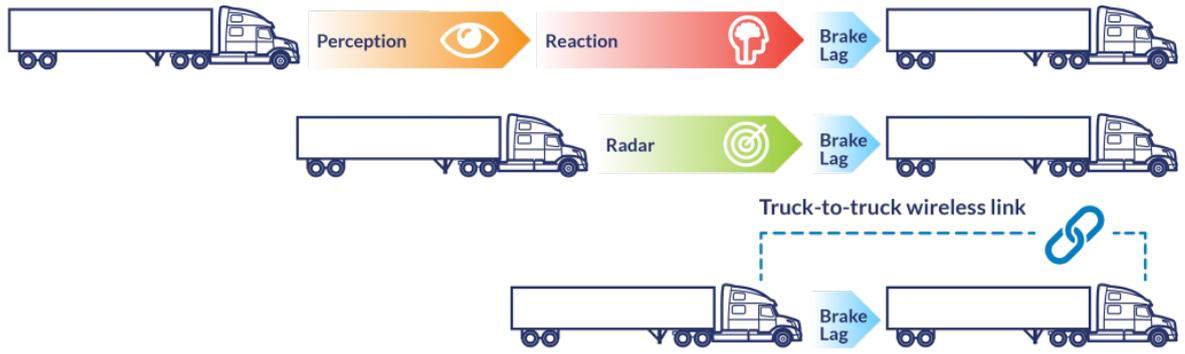


Figura 5.4: Mejora de conducción tras instalar el sistema V2V. <https://peloton-tech.com/how-it-works/#1452644330761-1e5676cf-7227>

consumo de combustible. El propósito de mantener un espacio constante entre los vehículos es denominado: Cooperative Adaptive Cruise Control (CACC)¹⁷.

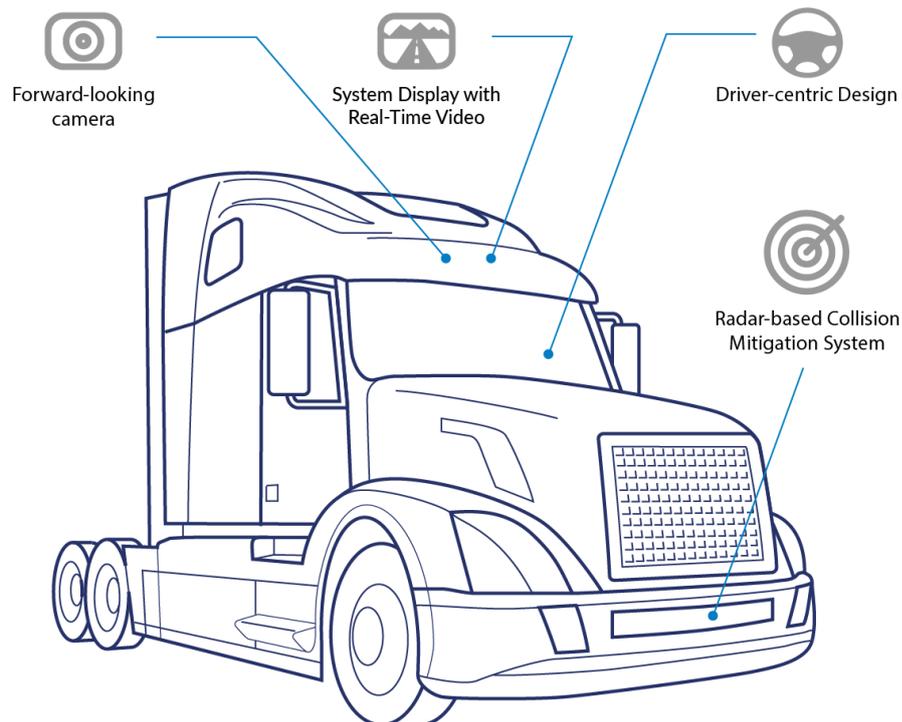


Figura 5.5: Sensores del sistema *Peloton*. <https://peloton-tech.com/how-it-works/#1452644330761-1e5676cf-7227>

¹⁷https://en.wikipedia.org/wiki/Cooperative_Adaptive_Cruise_Control

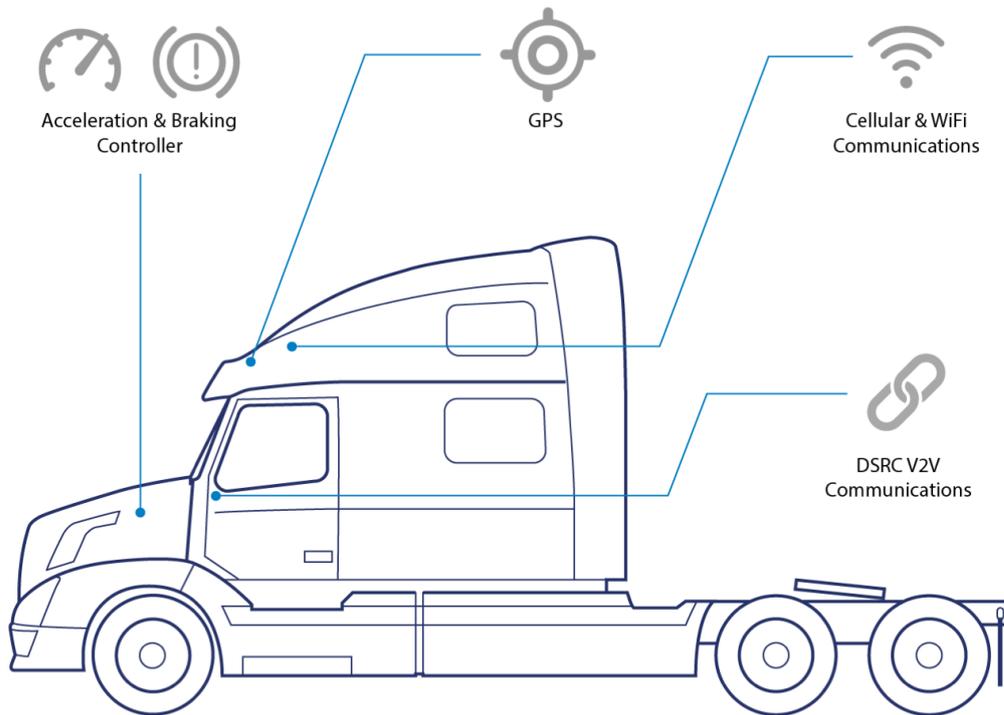


Figura 5.6: Componentes del sistema *Peloton*. <https://peloton-tech.com/how-it-works/#1452644330761-1e5676cf-7227>

El sistema de *Peloton* incluye la conexión de cada camión individual a un sistema de monitorización y administración basado en la nube, al que se refieren como el Network Operations Center (NOC). Este centro está diseñado para monitorear la seguridad de cada camión y su perímetro al aprobar la vinculación de pares de camiones en un orden específico siempre y cuando estén en caminos adecuados y bajo condiciones adecuadas de clima, vehículos y tráfico.

El sistema opera en SAE 1, donde los conductores de ambos vehículos continúan maniobrando mientras que el control de velocidad queda asignado al sistema y a su vez al camión líder del pelotón.

Cambiando de ámbito, la empresa **Komatsu Limited**, centrada en todo tipo de maquinaria para minería, ha desarrollado el Autonomous Haulage Vehicle (AHV) (véase Figura 5.7), un volquete autónomo sin cabina que incorpora el Autonomous Haulage System (AHS)¹⁸. Este sistema está incorporado con una serie de controladores, un sistema GPS, detección de obstáculos y un sistema de red inalámbrico. Estos volquetes están operados y controlados a

¹⁸<http://www.komatsuamerica.com/innovation/autonomous-navigation>

través de una estación remota. La información sobre el rumbo del objetivo y la velocidad se envía de forma inalámbrica desde la computadora de supervisión a los volquetes autónomos, mientras que el GPS se usa para determinar su posición. Al cargar, los volquetes se guían automáticamente al punto de carga después de calcular la posición de la excavadora o cargador localizados mediante GPS. La computadora de supervisión también envía información sobre un curso específico al lugar de vertido. Dado que están provistos de un sistema de detección de obstáculos son capaces de detener la marcha en caso de detectar un obstáculo en su trayectoria por lo que el sistema es extremadamente seguro y confiable.



Figura 5.7: Volquete sin cabina de Komatsu, denominado Autonomous Haulage Vehicle (AHV).

Otra empresa en incorporar el AHS a sus vehículos mineros es **Hitachi**¹⁹ (véase Figura 5.8). La empresa ha desarrollado su AHS en asociación con **Wenco**. Este sistema puede ser instalado en maquinaria ya existente mediante un módulo de dirección eléctrico insertado en el volante y una válvula. El mismo sistema también incluye sensores y controles vitales de comunicación que pueden ser adaptados a los volquetes ya existentes. Una máquina equipada con este sistema puede llevar a cabo tareas basadas en los datos del mapa, las rutas, los límites de velocidad y las tareas del trabajo.

Volviendo al mercado del transporte de mercancías, pero ya más enfocado al desarrollo de un sistema que haga autónomos a los camiones, tenemos a la empresa **Uber**, que actualmente está desarrollando vehículos, tanto camiones como coches, autónomos. Para impulsar su investigación adquirió la empresa *Otto*(Ottomotto LLC)²⁰ en Agosto del 2016 por \$680

¹⁹<https://www.hitachicm.eu/press-center/hitachi-to-develop-autonomous-haulage-system/>

²⁰<https://www.ottomotors.com>

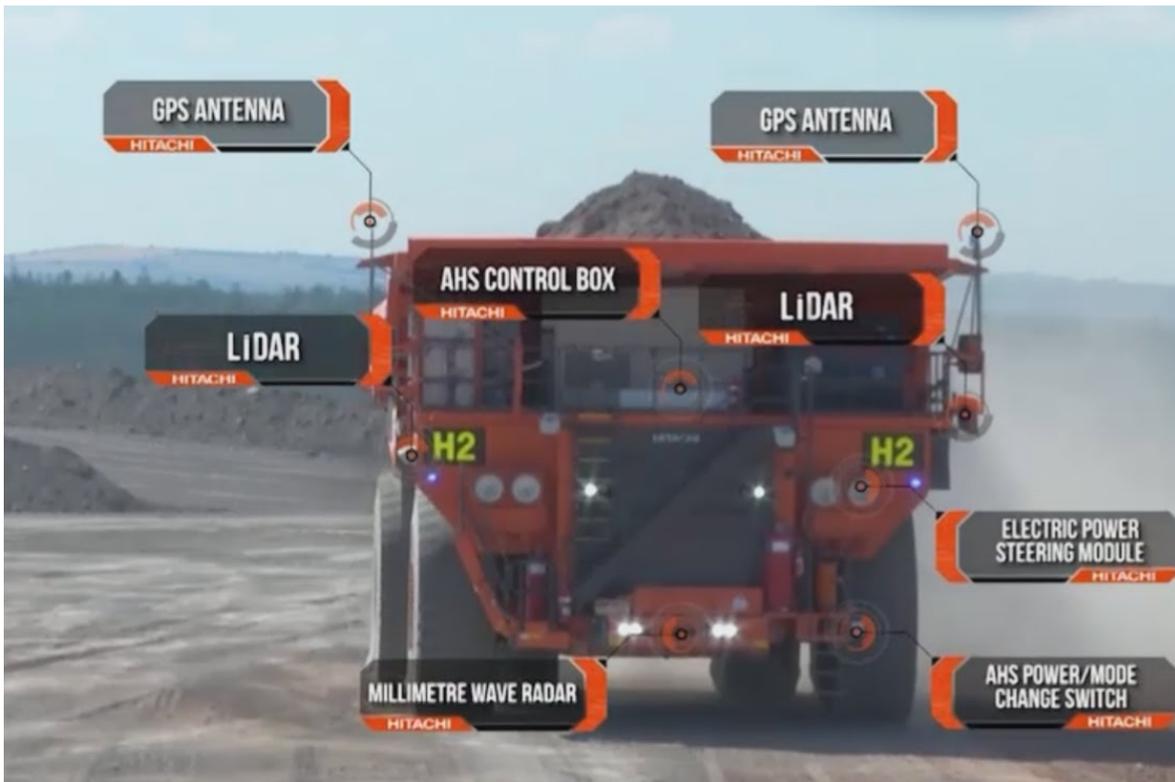


Figura 5.8: Volquete de Hitachi con AHS.

millones aunque dicha empresa se mantiene independiente de *Uber*.

La empresa *Otto* fabrica kits HW que pueden ser instalados en camiones. Actualmente tiene instalado el kit en cinco semirremolques Volvo 780. Estos kits utilizan una tecnología similar a la desarrollada por *Google* y *Nissan* equipando los camiones con radar, cámaras y lidar (véase Figura 5.9).

En Mayo de 2017 **Uber** retiró oficialmente de sus vehículos la marca *Otto* tras una disputa por la marca registrada *Otto*. Esto llevó a que *Uber* sacara un nuevo modelo del camión autónomo²¹. Dicho modelo dispone de un dispositivo LIDAR de 64 canales rotatorio (muy similar al modelo HDL-64E²² de **Velodyne**) además de nuevos sensores HW y nuevos componentes de SW (véase Figura 5.10). Estas mejoras han llevado a este vehículo a adquirir un nivel SAE 2.

Otra empresa encargada de adaptar vehículos, aunque no está centrada únicamente en el transporte de mercancías, es **Volvo**. Actualmente esta empresa ha desarrollado un camión de la basura autónomo. Este vehículo dispone de unos sensores que monitorean continuamente la proximidad del vehículo y que mandan detenerse al camión inmediatamente si aparece un

²¹<https://techcrunch.com/2017/06/29/uber-atg-upgrades-its-autonomous-truck-test-fleet-with-new-tech/>

²²<http://velodynelidar.com/hdl-64e.html>

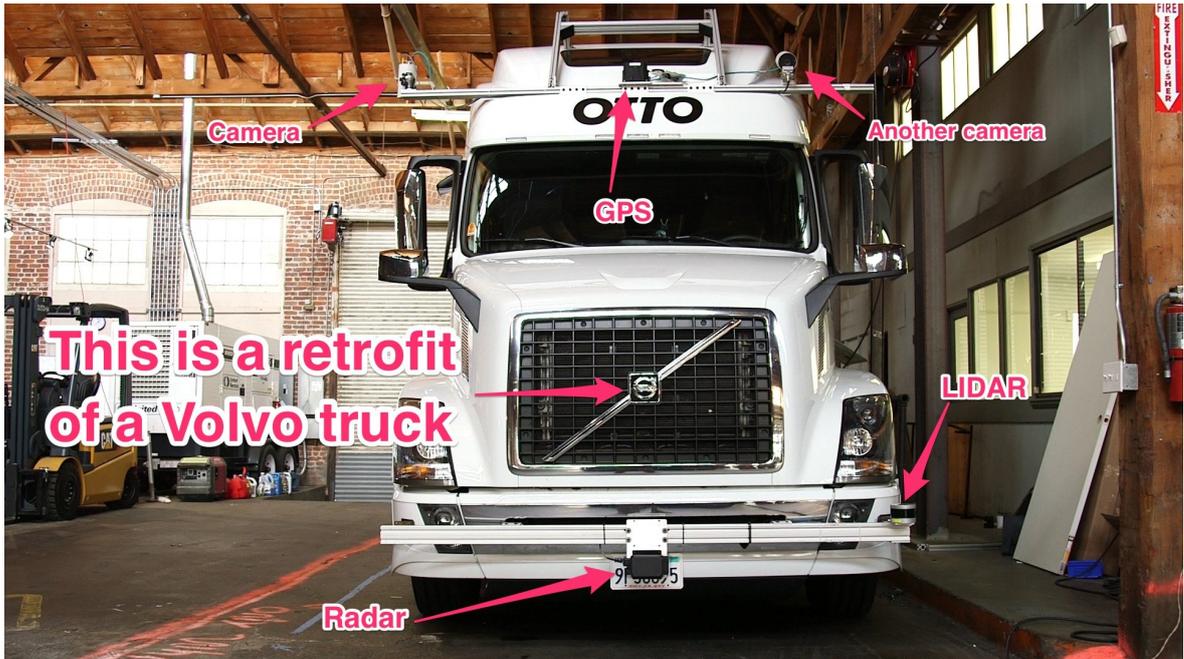


Figura 5.9: Sensores del modelo inicial de Otto.<http://www.businessinsider.com/uber-otto-self-driving-truck-technology-miracle-2016-8>



Figura 5.10: Nuevo conjunto de sensores de los camiones de *Uber*.<https://www.constructionequipment.com/video-new-look-and-sensors-uber-self-driving-semi>

obstáculo repentinamente en su camino. La ruta está pre-programada y el camión se conduce de un contenedor a otro. El conductor, que camina delante del vehículo, puede concentrarse en la recolección de basura y no tiene que subir y bajar de la cabina cada vez que el camión se traslada a un nuevo contenedor (véase Figura 5.11).



Figura 5.11: Camión de la basura de Volvo. <http://www.volvogroup.com/en-en/news/2017/may/news-2561936.html>

Otro desarrollo de *Volvo* está centrado en la recolección de caña de azúcar en Brasil²³. En este proyecto se ha desarrollado un vehículo capaz de recolectar la caña de azúcar de forma autónoma que permite a los agricultores aprovechar mejor los cultivos.

Como último proyecto, la empresa *Volvo* tiene en marcha uno para la creación de un vehículo autónomo en entornos mineros y que según la empresa *revolucionara la industria minera*. Este proyecto utiliza un *Volvo* FMX que fue probado en las profundidades de la mina Kristineberg. Durante el test realizado el camión recorrió 7Km llegando a una profundidad de 1.320 metros con lo que ello conlleva (véase Figura 5.12). El camión está equipado con tecnología GPS y sensores LIDAR, que escanean continuamente sus alrededores. El sistema crea un mapa de la geometría de la mina y la utiliza para crear una ruta a través de los túneles para que la siga el camión, así como para informar la dirección, los cambios de marcha y la velocidad. Gracias a los datos obtenidos es capaz de navegar entre obstáculos fijos y en movimiento, así como operar de manera totalmente autónoma tanto en la superficie como bajo tierra. De los sensores instalados en el camión, la empresa informa de que al menos dos, y a menudo tres, pueden controlar cualquier punto de su entorno en un momento dado. Esto es debido a que usa cuatro sensores LIDAR de 360 grados posicionados en las

²³<https://www.multivu.com/players/es-es/8117251-self-steering-volvo-truck-brazil-harvest/>

esquinas permitiendo así obtener una redundancia de los datos (véase Figura 5.13). del En caso de que se detecte un obstáculo, la carretilla se detendrá y se pondrá en contacto con su centro de control. Al mismo tiempo, un sistema de transporte a bordo reúne datos para optimizar y coordinar la ruta y el consumo de combustible ya que cada visita permite que el camión refine aún más su modelo de la mina y posteriormente optimice su ruta y consumo de combustible.



Figura 5.12: Volvo FMX totalmente autónomo realizando un recorrido de prueba en las profundidades de la mina Kristineberg.

La última empresa a mencionar es **ASI Robotics**, centrada en el desarrollo de kits que permite a vehículos normales transformarse en autónomos. Esta empresa dispone de numerosos kits enfocados para la minería, agricultura y ámbito militar, que pueden ser instalados en diversos vehículos.

Estos kits están formados por un conjunto de sensores, una unidad de control y los actuadores. Los sensores recopilan información ambiental y la retransmiten a la VCU integrada para dar a conocer el sistema de automatización, cabe destacar que este sistema de sensores es compatible con el sistema de cámara *Mobileye*, dichos sensores permiten al software **Mobius**²⁴ obtener información del vehículo. La unidad de control VCU está desarrollada íntegramente por *ASI Robotics* y es la encargada de comunicar los sensores del vehículo y el control del vehículo. Los actuadores universales se colocan en el vehículo, en caso de que

²⁴<https://www.asirobots.com/platforms/mobius/>

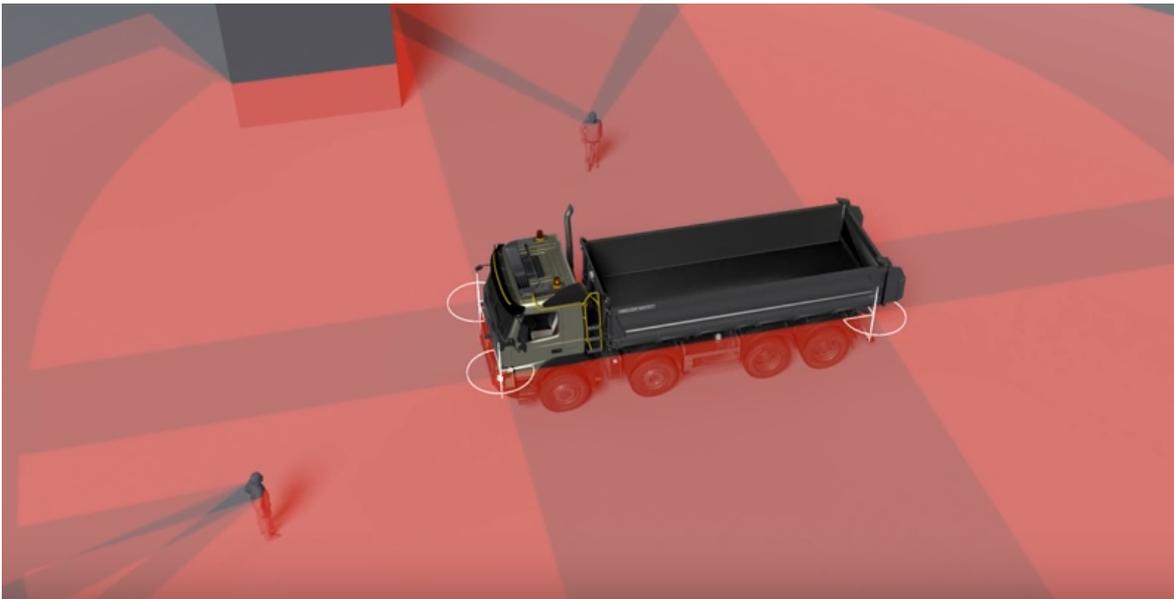


Figura 5.13: Volvo FMX. Solapamiento de los sensores LiDAR

sea necesario ya que también puede conectarse la VCU directamente al vehículo a través del módulo de interfaz del vehículo (véase Figura 5.14).

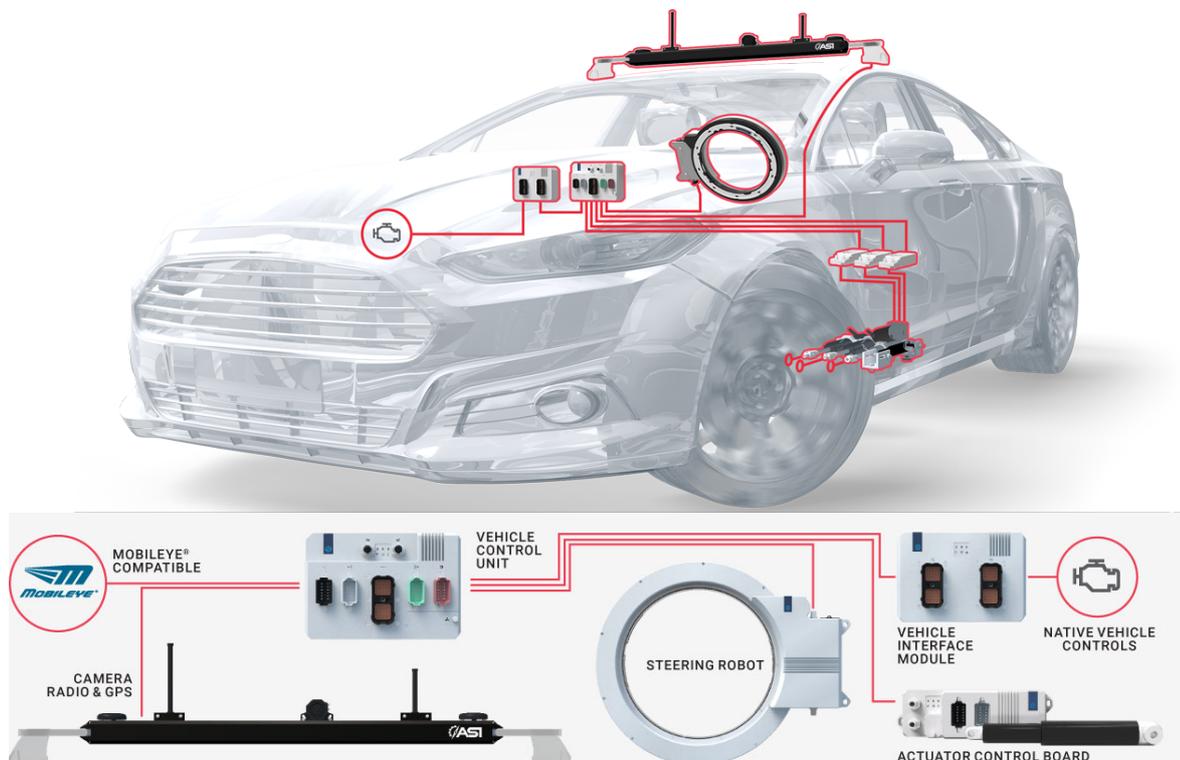


Figura 5.14: Kit de automatización realizado por la empresa *ASI Robotics*. <https://www.asirobots.com/automotive/>

5.3 Hardware de sensorización

Como se ha visto en la Sección 4.1 existen multitud de sensores, uno para cada magnitud medible. La mayoría de ellos están en continuo avance tecnológico y sobre todo aquellos que están diseñados especialmente para vehículos.

En la actualidad existen diversas empresas dedicadas al desarrollo de sensores orientados a vehículos autónomos. Cabe destacar que, por lo general, estas empresas están centradas en un tipo de sensores: LiDAR, visión, radar, etc. . . , aunque eso no quita que traten sensores de todo tipo.

Una de estas empresas que trata todo tipo de sensores es **Sick AG**²⁵, aunque está centrada en escáneres laser, por lo que sus sensores LiDAR son bastante conocidos. Se utilizaron cinco **LIDAR!** desarrollados por *Sick AG* para la detección de corto alcance en Stanley²⁶, el automóvil autónomo que ganó en 2005 el DARPA Grand Challenge (véase Figura 5.15). Una mezcla de sensores LiDAR de *SICK AG* e **IBEO**²⁷ fueron usados para evitar obstáculos, mapear y rastrear peatones en WAMbot²⁸, el vehículo autónomo que quedó 4º en la competencia MAGIC 2010.

Otra empresa centrada en sensores LiDAR es **Velodyne LiDAR**²⁹. Esta empresa se centra en las aplicaciones de la tecnología LiDAR para su uso en vehículos autónomos, sistemas de seguridad de vehículos, mapeo móvil 3D, mapeo aéreo en 3D y seguridad. Los sensores de Velodyne tienen un alcance de alrededor de 120 metros, un alcance mayor que el de las cámaras (véase Figura 5.16).

A finales de 2016 emprendió el desarrollo de sensores LiDAR de estado sólido que reducirán el precio de los sensores.

En abril de 2017, Velodyne anunció Velarray³⁰, un sensor que usa un conjunto fijo de láseres y receptores, en lugar de la matriz giratoria utilizada hasta ahora. El sensor Velarray es más pequeño que sus predecesores, aunque este mapea un arco de 120 grados, en vez de los 360 grados normales, haciéndolo perfecto para un uso conjunto con otros sensores (véase Figura 5.17).

Los nuevos sensores LiDAR de estado sólido son el futuro tecnológicamente hablando. Poseen un tamaño reducido y su coste es muy reducido en comparación con los sensores actuales. La empresa **LeddarTech**³¹ es una de las pioneras en este campo. La empresa ha desarrollado y patentado **Leddar**³². Es esencialmente un conjunto de software, algoritmos y

²⁵<https://en.wikipedia.org/wiki/Sick.AG>

²⁶[https://en.wikipedia.org/wiki/Stanley_\(vehicle\)](https://en.wikipedia.org/wiki/Stanley_(vehicle))

²⁷<https://www.ibeo-as.com/>

²⁸<http://www.wambot.org/>

²⁹<http://www.velodynelidar.com/>

³⁰<http://www.businesswire.com/news/home/20170419005516/en/Velodyne-LiDAR-Announces>

³¹<https://leddartech.com/>

³²<https://en.wikipedia.org/wiki/Leddar>



Figura 5.15: Sensores LIDAR de la empresa Sick AG montados sobre el vehículo Stanley.

conocimientos que se utilizan para diseñar u optimizar varios tipos de sensores LIDAR de estado sólido.

Otro sector está enfocado en el desarrollo de cámaras que puedan ser utilizadas en sistemas autónomos. La empresa **Mobileye** es el líder mundial en el desarrollo de tecnología de visión para ADAS y vehículos autónomos. La tecnología desarrollada se basa en el uso de sistemas de visión óptica con algoritmos de detección de movimiento que se ejecutan en un acelerador de hardware personalizado: el chip EyeQ³³.

La empresa también desarrolla sistemas de advertencia de abandono del carril que son sistemas electrónicos que van dentro del vehículo y que monitorean la posición de un vehículo dentro de una línea de carretera y advierten al conductor si el vehículo se desvía o está a punto de desviarse fuera del carril. La versión de *Mobileye* fue lanzada en múltiples plataformas de producción durante 2007 y 2008 con *GM*, *BMW* y *Volvo*.

Desde 2008, los automóviles de la serie 7 de *BMW* están equipados con los sistemas de reconocimiento de señales de tráfico de *Mobileye*, este sistema esta desarrollado en cooperación con *Continental AG*.

³³<https://www.mobileye.com/our-technology/evolution-eyeq-chip/>



Figura 5.16: Distintos sensores LiDAR de la empresa *Velodyne*.



Figura 5.17: Sensor *Velarray* de la empresa *Velodyne*.

La empresa también utiliza fusión de datos para unir sus sensores de visión con sensores de radar y así en Mayo de 2009 como parte del sistema de fusión de visión radar *Volvo* de próxima generación que también proporciona advertencia de salida de carril y detección de

vehículo.

Actualmente ofrecen sistemas de advertencia de salida de carril, advertencia de colisión delantera, monitorización y advertencia de avance, advertencia de colisión urbana de baja velocidad, control inteligente de faro, indicación de límite de velocidad (tsr) y advertencia de colisión peatonal (incluidas bicicletas) (véase Figura 5.18).



Figura 5.18: *Mobileye Serie 5*. Sistema de prevención de accidentes.<http://www.mobileye.com/es-es/productos-de-mobileye/series-5-de-mobileye/>

5.3.0.1. Comparativa de sensores

Con la existencia de tantos sensores es normal que existan distintos enfoques de cómo debe percibir el vehículo el entorno, por eso es necesario hacer una comparación entre tecnologías y enfoques utilizados para así discernir que enfoque es mejor. Para ello, a continuación se comparará el uso de LIDAR o no LIDAR en los vehículos autónomos.

La empresa **Google** es defensora del uso de los sensores LIDAR ya que según su criterio es un sensor necesario para la conducción autónoma. Por otro lado, la empresa **Tesla** defiende que no es tan necesario y que su elevado coste lo hace un sensor inutilizable para los vehículos del futuro³⁴, aunque el *modelo S de Tesla* incorpora un sensor LIDAR en su

³⁴<https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

techo³⁵.

En la Figura 5.19 se puede ver una comparativa, realizada en 2016, de los distintos sensores enfrentados a distintos entornos. El valor de rango o *range* está medido en metros y el valor de agudeza o *acuity* está basado en la combinación de resolución, detección de contraste y color de los sensores.

Como se puede observar las *cámaras de video a color*, o *passive visual* como las denomina el estudio, son las que mejor agudeza y rango tienen en condiciones optimas pero bajan su puntuación una vez las condiciones empeoran.

El rango en situaciones de oscuridad está basado en los faros actuales y en el tipo de iluminación que describen aunque esta puede variar. En dicho estudio se define un entorno donde el coche no está en un camino bien iluminado y que las condiciones lumínicas pueden variar generando así zonas en penumbra o en total oscuridad. Un punto a tener en cuenta son los resultados que ha obtenido el radar, que, aunque no es el mejor sensor, sus resultados son similares en todos los entornos manteniendo un rango óptimo para detectar vehículos y objetos.

El sensor LIDAR obtiene unos buenos resultados hasta que las condiciones atmosféricas empeoran reduciendo su agudeza drásticamente. Estos datos llevan a preguntarse hasta qué punto se debe confiar en la calidad de la información obtenida por los sensores. ¿La agudeza del radar es suficiente para manejar un vehículo de forma autónoma a una velocidad estándar o es necesario una agudeza superior?

Otra pregunta que surge con estos datos es ¿Se debe diseñar el sistema basándose en la información mas confiable disponible en todas las condiciones o basarse en un sistema multi-sensor y hacer las suposiciones en base a un conjunto de datos que se degradan según las condiciones del entorno?

Según la empresa *Tesla* el uso de sensores LIDAR no es necesario para que un sistema autónomo sea eficaz.

El modelo de Tesla con el sistema Autopilot tiene ocho cámaras de vídeo repartidas por el perímetro del coche que permiten ver casi todo el entorno que le rodea:³⁶

- **Tres cámaras frontales** que miran hacia adelante. Una de mayor alcance, 250m, con un ángulo de visión estrecho. Otra de alcance intermedio, 150m, con un ángulo de visión un poco más abierto. Y la ultima, con un objetivo gran angular, sólo alcanza a ver hasta 60 m.
- **Dos cámaras laterales** situadas delante. Situadas en el lado derecho e izquierdo visualizan lo que no se llega a ver debido a los angulos muertos. Tienen una distancia de 80m.

³⁵<https://evobsession.com/self-driving-tesla-model-s-spotted-near-company-hq>

³⁶<https://www.motorpasion.com/tecnologia/lidar-si-o-lidar-no-esa-es-la-cuestion-con-los-coches-autonomos>

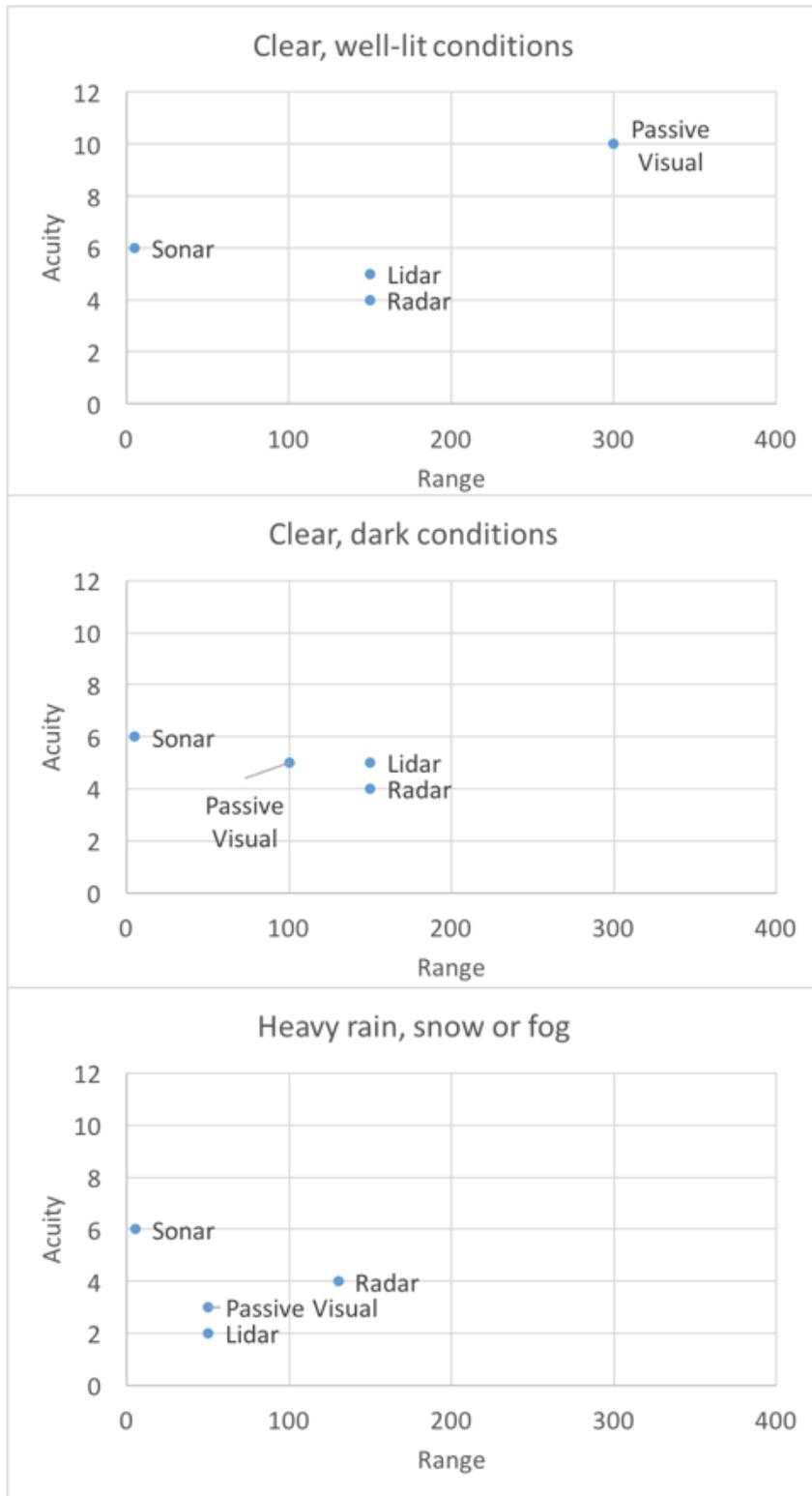


Figura 5.19: Comparativa de distintos sensores en distintos entornos. Range medido en metros. Acuity basado en la combinación de resolución, detección de contraste y color de los sensores. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

- **Dos cámaras laterales** situadas en la parte trasera. Simulan la vision de los retrovisores enfocando hacia el lado izquierdo y derecho trasero.
- **Una cámara trasera** centrada. Simula la vision del espejo retrovisor interior.

Estas cámaras permiten una visión 360 grados del entorno permitiendo al vehículo reconocer los obstáculos que le rodean (véase Figura 5.20).



Figura 5.20: Visión conjunta del autopilot del vehículo *Tesla*. <https://www.motorpasion.com/tecnologia/lidar-si-o-lidar-no-esa-es-la-cuestion-con-los-coches-autonomos>

Para determinar si el uso de múltiples sensores puede equipararse al uso de LiDAR en la Figura 5.21 se han superpuesto las cualidades de los distintos sensores y se ha mostrado que pueden proporcionar mediciones correctas en la mayoría de las condiciones.

Como punto a favor del uso de múltiples sensores está que el gasto de dichos sensores está muy por debajo del gasto de un sensor LiDAR actual, que puede rondar los \$30.000 - \$70.000 lo que lo hace inaccesible para vehículos normales, aunque el gasto de un sensor LiDAR de estado sólido es similar al gasto del conjunto de sensores de un sistema que lo excluya, por lo que no lo hace tan inaccesible.

Aunque uno de los puntos negativos del uso de múltiples sensores es que el radar no dispondría de buena resolución de imagen en la oscuridad en comparación con un sistema con sensor LiDAR ya que este no tiene una buena resolución nocturna.

En resumen, y viendo las gráficas, no existe una propuesta ganadora ya que todos los sensores poseen unos puntos a favor y unos puntos en contra y la evolución de dichos sensores

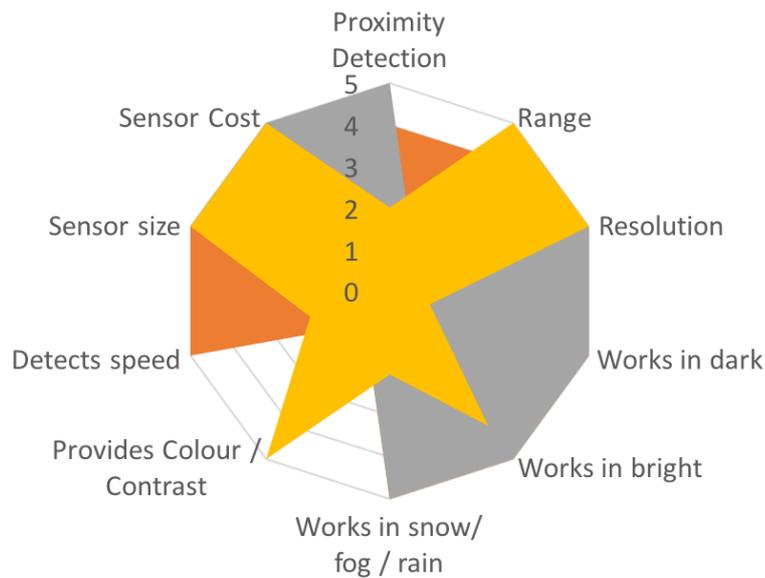


Figura 5.21: Superposición de los atributos del radar, ultrasonidos y cámara de video. <https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>

y el abaratamiento de su coste hace que el uso de múltiples sensores, incluido el LIDAR sea la elección más correcta para realizar un proyecto.

5.4 Entornos de desarrollo

La parte más importante de un vehículo autónomo es su inteligencia, es el cerebro del vehículo, el procesar la información, saber ir de un punto de origen a un punto de destino, el reaccionar ante obstáculos y el tomar decisiones. En la actualidad existen soluciones de bajo nivel, como **ROS**³⁷, alto nivel, **Mobius** de *ASI Robotics*, y un nivel intermedio como el **NVIDIA Driveworks**³⁸. Otro tipo de SO utilizado indirectamente para el desarrollo es aquel utilizado para la simulación y desarrollo de nuevos métodos o algoritmos matemáticos relacionados con las etapas por las que pasa un vehículo autónomo: fusión de datos, modelado del entorno, planificación, control, etc. ROS es un framework orientado al desarrollo de SW para robots que provee la funcionalidad de un SO en un clúster heterogéneo. Fue creado por el Laboratorio de Inteligencia Artificial de Stanford en 2007. Es tratado como un SO dado que provee de los servicios típicos que otorga un SO. Se basa en una arquitectura de grafos, donde cada nodo procesa la información y puede recibir, mandar y multiplexar mensajes de sensores, control, estados, planificación, y actuadores, entre otros. Actualmente es compatible con sistemas *UNIX* aunque se está adaptando a otros sistemas operativos. Aparte de la

³⁷https://es.wikipedia.org/wiki/Sistema_Operativo_Robótico

³⁸<https://developer.nvidia.com/driveworks>

librería que otorga el SO también existe un gestor de paquetes, *ros-pkg*, que otorga un conjunto de paquetes ya desarrollados y aportados por usuarios que implementan funcionalidad útil como: localización, percepción, mapeo, simulación, etc. Aunque ROS es software libre con licencia *Berkeley Software Distribution (BSD)*, los paquetes están bajo distintas licencias. Dada su experiencia con la robótica, ROS es una plataforma madura y flexible para la programación de vehículos. Como puntos a favor de su uso tenemos³⁹:

- **Paquetes para vehículos autónomos:** en la actualidad existen multitud de paquetes enfocados a vehículos autónomos. Además, dada la similitud de los vehículos con los robots con ruedas, es posible utilizar paquetes orientados a estos últimos.
- **Visualización de datos:** ROS otorga un conjunto de herramientas gráficas que permiten visualizar los datos capturados por los sensores y gracias a ellos representar el vehículo de una manera integral (véase Figura 5.22).
- **Es simple empezar:** ROS permite empezar de forma rápida y sencilla a partir de un par de sensores y un robot con ruedas y dada su escalabilidad permite ir aumentando la complejidad del proyecto.

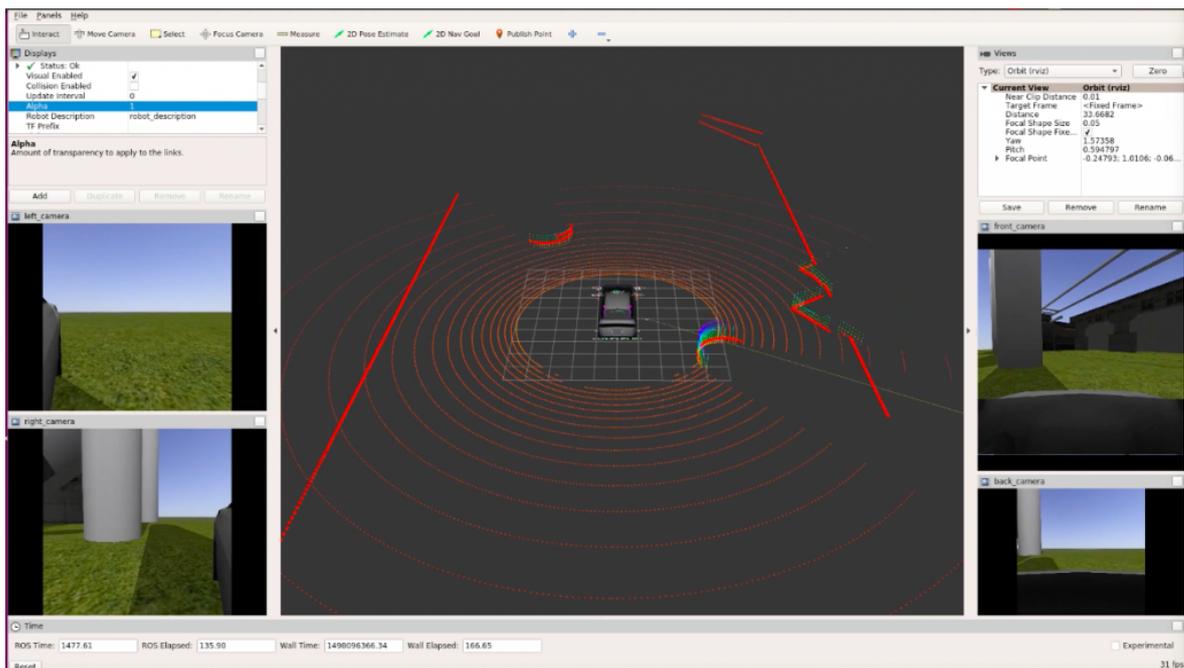


Figura 5.22: Interfaz gráfica de ROS.

Como puntos negativos del uso de ROS tenemos:

- **RosCore:** es el núcleo de ROS, todos los sistemas dependen de este núcleo y si este falla entonces el vehículo falla por completo.

³⁹<http://www.theconstructsim.com/start-self-driving-cars-using-ros/>

- **No tiene seguridad:** es posible ingresar a la red de ROS y enviar paquetes fraudulentos y tomar el control del vehículo.

Aunque, estos puntos negativos de la librería ROS se han arreglado en su nueva versión ROS 2⁴⁰. Actualmente existen proyectos de vehículos autónomos que utilizan ROS para dirigir sus vehículos, como *BMW*, *Bosch* y *nuTonomy*. También existen distintos kits en el mercado que vienen con ROS integrado, como los desarrollados por **AutonomouStuff**⁴¹.

A un nivel más alto de programación tenemos el SW **Mobius** de *ASI Robotics*. Este SW no sirve para el desarrollo de vehículos autónomos, pero sí que permite al usuario controlarlos y dotarlos de cierta inteligencia, dentro de unos límites. *Mobius* está desarrollado para controlar los vehículos que incorporen los kits desarrollados por *ASI Robotics* e internamente los dota de cierta inteligencia que les permite realizar trayectorias, evitar obstáculos y tomar decisiones y externamente, permite al usuario:

- **Controlar el vehículo:** el SW permite establecer tareas específicas y controlar las funciones críticas del vehículo, incluidas la dirección, la transmisión, la aceleración, el frenado y el encendido desde una ubicación remota.
- **Controlar múltiples vehículos:** es posible controlar varios vehículos autónomos que interactúan en el misma área o en lugares diferentes.
- **Creación de mapas:** el usuario puede crear mapas personalizados, definir carriles y carreteras y establecer perímetros de seguridad aparte de añadir imágenes de satélite de la zona.
- **Creación de rutas:** es posible determinar trayectorias para que el vehículo autónomo las siga de forma segura evitando así obstáculos y colisiones con otros vehículos (véase Figura 5.23).

El SW tiene una interfaz simple y limpia que permite controlar los vehículos al usuario de manera fácil y sencilla.

Bajando un nivel podemos encontrar la solución de **NVIDIA** con su Software Development Kit (SDK) **DriveWorks** y su plataforma **NVIDIA Drive**⁴². La plataforma *NVIDIA DRIVE* es una plataforma informática autónoma que permite a los fabricantes de automóviles y proveedores de nivel SAE 1 acelerar la producción de vehículos automáticos y autónomos.

La empresa *NVIDIA* dispone de distintos controladores desarrollados para ser el cerebro de los vehículos autónomos. En 2015 empezó con el modelo *Drive PX*, de primera generación, centrado en transformar los coches en semi-automáticos, que fue utilizado por *Toyota*⁴³.

⁴⁰<https://github.com/ros2/ros2/wiki>

⁴¹<https://autonomoustuff.com/product/adas-kit/>

⁴²<https://developer.nvidia.com/drive>

⁴³<https://electrek.co/2017/05/10/toyota-nvidia-autonomous-cars/>

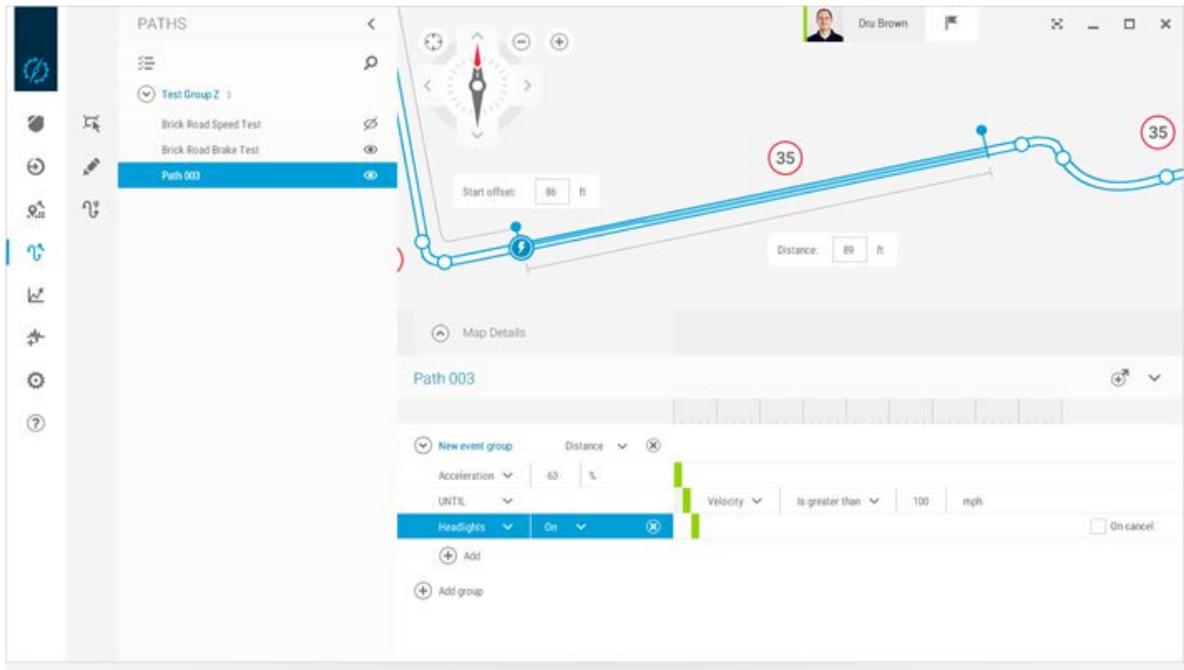


Figura 5.23: Interfaz gráfica de MOBIUS!. Planificador de trayectorias.

Posteriormente, en 2016 , saco el modelo *Drive PX 2* que está siendo distribuido listo para montar y funcionar en sus modelos *Parker AutoCruise*, diseñado para manejar funciones que incluyen la conducción automática en autopistas, así como el mapeo del entorno en HD y *Parker AutoChauffeur* que permite realizar trayectorias punto a punto. En 2017 se anuncio el *Xavier AI Car Supercomputer*, 50 % más potente que el *Drive XP 2 Autochauffeur*. A finales de año se anuncio el *Drive PX Pegasus* que tiene como base dos dispositivos *Xavier* es capaz de realizar cálculos de aprendizaje profundo y la capacidad de ejecutar numerosas redes neuronales profundas al mismo tiempo, esta computadora de IA de alto rendimiento proporciona todo lo necesario para una conducción autónoma y segura de nivel SAE 5⁴⁴. Por otro lado, la empresa, otorga a los desarrolladores un SDK, el *DriveWorks*, especialmente diseñado para el desarrollo de vehículos autónomos. Este SDK contiene aplicaciones de referencia, herramientas y biblioteca. También incluye un marco de trabajo en tiempo de ejecución que va desde la detección, la localización, la planificación hasta la visualización.

Este SDK está disponible como parte del SW *NVIDIA Drive* que es suministrado a los fabricantes de automóviles, proveedores de nivel SAE 1, instituciones de investigación, etc. . . es decir, todo desarrollador de sistemas útiles para los vehículos autónomos.

El SDK provee de la funcionalidad necesaria para tratar problemas de detección, localización, planificación y visualización. Es compatible con SW CUDA, NPP, cuDNN, OpenGL y V4L/V4Q entre otros y es capaz de gestionar distintos sensores como cámaras, LiDAR,

⁴⁴<http://omicrono.lespanol.com/2017/10/ordenador-coches-nvidia-drive-px-pegasus/>

radar, GPS, ultrasonidos, edometría y mapas. Las principales características de los módulos del SDK son:

- **Detección:** incluye procesamiento de sensores, correlación de sensores, fusión de sensores, segmentación y detección y clasificación de aprendizaje profundo (véase Figura 5.24).



Figura 5.24: Detección de obstáculos del SDK *NVIDIA DriveWorks*.

- **Localización:** incluye la localización de mapas, la interfaz HD-Map y egomotion⁴⁵ (véase Figura 5.25).

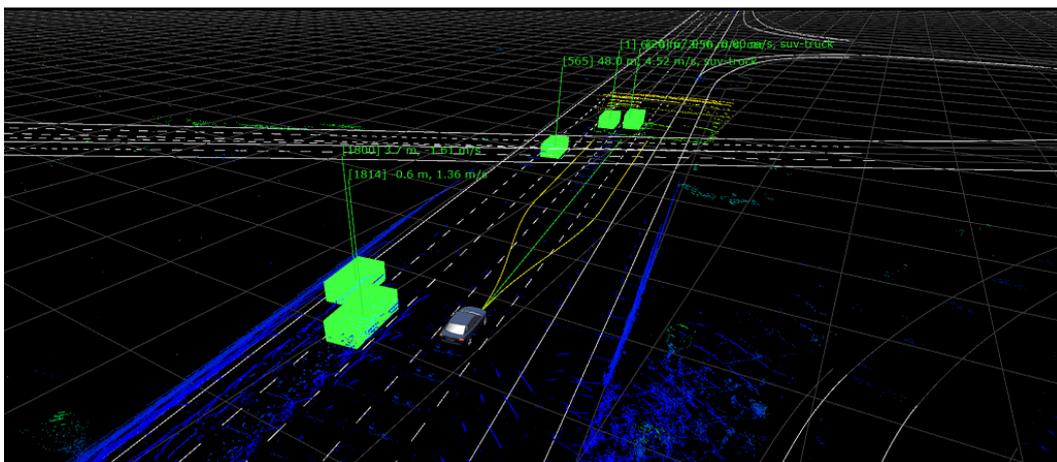


Figura 5.25: Localización y planificación de trayectorias del SDK *NVIDIA DriveWorks*.

- **Planificación:** incluye control de vehículos, comprensión de escenas y solucionadores de problemas de planificación de caminos.

⁴⁵ Egomotion is defined as the 3D motion of a camera within an environment.

- **Visualización:** incluye la transmisión a un clúster de visualización, procesamiento ADAS y depuración.

Además entre las herramientas que incluye tiene un sistema de calibración de sensores, calibración de la plataforma, configuración de valores intrínsecos y extrínsecos de la cámara y configuración de los valores extrínsecos del sensor LiDAR.

NVIDIA DriveWorks está diseñado para ser modular, educativo, optimizado y de código abierto. Está basado en OpenGL y en las tecnologías *NVIDIA* existentes como DRIVE PX 2, Vibrante 4 Linux (V4L), CUDA, cuDNN, NPP y GPU Inference Engine (GIE).

Por último, existe otro tipo de SW utilizado para el desarrollo de vehículos autónomos que no está creado con este propósito. Es aquel SW de simulación o de ámbito matemático que es utilizado para desarrollar algoritmos o métodos útiles para los pasos o procesos que sigue un vehículo autónomo.

Uno de los más utilizados es **MatLab** dado que es el SW matemático por excelencia es comúnmente utilizado para simular algoritmos de planificación, fusión de datos o toma de decisiones.

Entre los simuladores disponemos de **Gazebo**⁴⁶, **V-Rep**⁴⁷, **ROS MoveIt**⁴⁸ (véase Figura 5.26) y **Webots**⁴⁹.

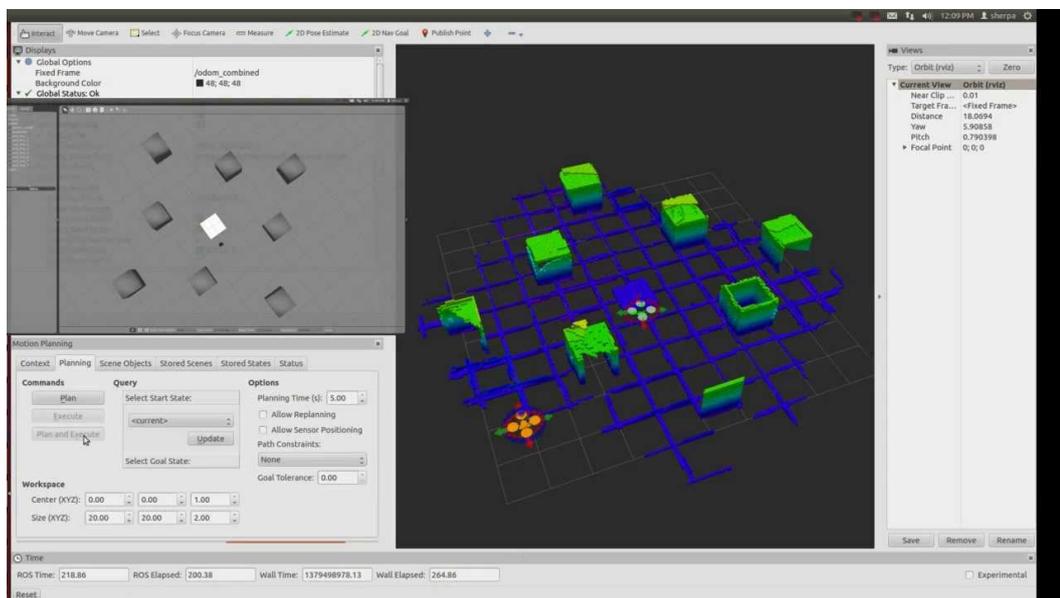


Figura 5.26: Interfaz gráfica de Ros MoveIt.<https://www.youtube.com/watch?v=VlBQLbmc03g>

⁴⁶<http://gazebosim.org/>

⁴⁷<http://www.coppeliarobotics.com/>

⁴⁸<http://moveit.ros.org/>

⁴⁹<https://www.cyberbotics.com/>

Propuesta de camión autónomo

EN este capítulo se detallara el estándar **J3016A** propuesto por SAE que define una taxonomía para sistemas de automatización de la conducción de vehículos. Esta taxonomía será utilizada para realizar la definición de un esquema de trabajo que establezca los pasos recomendados desde que se comienza a diseñar una solución hasta que el camión autónomo se implanta realmente. También en este capítulo se detallara el entorno en el que está englobado este proyecto, los requisitos funcionales del prototipo y el coste de realizarlo.

6.1 Introducción

Como base para la elaboración del esquema que se desea proponer se ha utilizado el estándar **J3016A** *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* [Int17], propuesto por SAE.

SAE proporciona un **taxonomía** para sistemas de automatización de la conducción de vehículos, o **driving automation system**, que llevan a cabo una parte o la totalidad de la tarea de conducción dinámica, o en inglés Dynamic Driving Task (DDT) de manera sostenida y continuada y que pueden variar en nivel desde vehículos sin automatización de la conducción, nivel SAE 0, hasta un nivel de automatización total de la conducción, nivel SAE 5.

Estas definiciones de nivel, junto con los términos y definiciones de soporte adicionales proporcionados en este documento, se pueden usar para describir la gama completa de características de conducción autónoma equipadas en los vehículos de una manera coherente y funcionalmente consistente. Por ejemplo, el término «*En ruta*» se refiere a vías públicas que prestan servicios colectivos a usuarios de vehículos de todas las clases y niveles de automatización de conducción, así como a motociclistas, ciclistas a pedal, y peatones.

Los niveles tienen en cuenta las características de automatización que están involucradas en cualquier momento dado. Esto significa que aunque el vehículo esté equipado con un *driving automation system* que ofrezca múltiples características de automatización que funcionen a diferentes niveles, el nivel de automatización está determinado por las características involucradas.

El estándar de SAE define **tres actores principales** en la conducción: el conductor, el *driving automation system* y otros sistemas y componentes del vehículo. A su vez, los niveles de automatización se definen por referencia al papel específico desempeñado por cada uno de los tres actores principales en la ejecución del DDT. El término «rol» se refiere al rol esperado de un actor primario dado, basado en el diseño del *driving automation system* en cuestión y no necesariamente en el desempeño real de un actor primario dado. Por ejemplo, un conductor que no supervisa la carretera durante el acoplamiento de un sistema de control de cruce adaptable, o Adaptive Cruise Control (ACC), sigue teniendo el papel de conductor, incluso cuando lo descuida.

Cabe destacar que los **sistemas de seguridad activos**, como el control de estabilidad electrónico y el frenado de emergencia automático, y ciertos tipos de sistemas de asistencia al conductor, como la asistencia para el mantenimiento de carriles, están excluidos del alcance de esta taxonomía de automatización de conducción porque no realizan parte o todo el DDT de forma sostenida y, más bien, simplemente proporcionar intervención momentánea durante situaciones potencialmente peligrosas. Debido a la naturaleza momentánea de las acciones de los *sistemas de seguridad activos*, su intervención no cambia ni elimina el rol del conductor en la realización de parte o la totalidad del DDT, y por lo tanto no se considera que dirija la automatización. A continuación se detallan una serie de términos que son definidos en el estándar SAE.

SAE define el término **Automated Driving System (ADS)**, que es el HW y SW que son colectivamente capaces de realizar todo el DDT de forma sostenida, independientemente de si está limitado a un dominio de diseño operativo, o Operational Design Domain (ODD)¹, específico. Este término se usa específicamente para describir un sistema de automatización de conducción de nivel 3, 4 o 5. Cabe destacar que difiere del término genérico *driving automation system* que se refiere a cualquier sistema o característica de nivel 1-5 que realiza una parte o la totalidad del DDT de manera sostenida. Dada la similitud entre el término genérico, *driving automation system* y el término específico de nivel 3-5, *automated driving system*, este último término debe escribirse con mayúscula cuando se deletrea y se reduce a su sigla, ADS, tanto como posible, mientras que el término anterior no se debería.

Se define **ADS-Dedicated Vehicle (ADS-DV)** como aquel vehículo diseñado para ser operado exclusivamente por un ADS de nivel 4 o 5 para todos los viajes. Cabe destacar que un ADS-DV es un vehículo verdaderamente «*sin conductor*». Sin embargo, SAE, no utiliza dicho término, «*sin conductor*», ya que ha sido, y sigue siendo, ampliamente utilizado indebidamente para referirse a cualquier vehículo equipado con un sistema de automatización de conducción, incluso si ese sistema no es capaz de realizar siempre el DDT completo y por lo tanto, implica un controlador (humano) para parte de un viaje determinado. Esta es la única categoría de vehículos operados por ADS que no requiere un controlador convencional

¹<https://vsi-labs.com/knowledge/2017/2/13/understanding-operational-design-domains>

ni remoto durante la operación de rutina.

SAE define **Dynamic Driving Task (DDT)** como todas las funciones operativas y tácticas en tiempo real requeridas para operar un vehículo en una ruta, excluyendo las funciones estratégicas tales como la programación de viajes y la selección de destinos y puntos de referencia, e incluyendo, entre otros:

- Control de movimiento lateral del vehículo a través de la dirección (operacional).
- Control de movimiento longitudinal del vehículo a través de aceleración y deceleración (operacional).
- Monitoreo del entorno de conducción a través de la detección de objetos y eventos, reconocimiento, clasificación y preparación de respuestas (operativa y táctica).
- Ejecución de respuesta de objetos y eventos (operacional y táctica).
- Planificación de la maniobra (táctica).
- Mejorar la visibilidad a través de iluminación, señalización y gesticulación, etc. (táctica).

Las sub-tareas de monitoreo del entorno y ejecución de respuestas es denominada como Object and Event Detection and Response (OEDR). En la Figura 6.1 muestra una vista esquemática de la tarea de conducción.

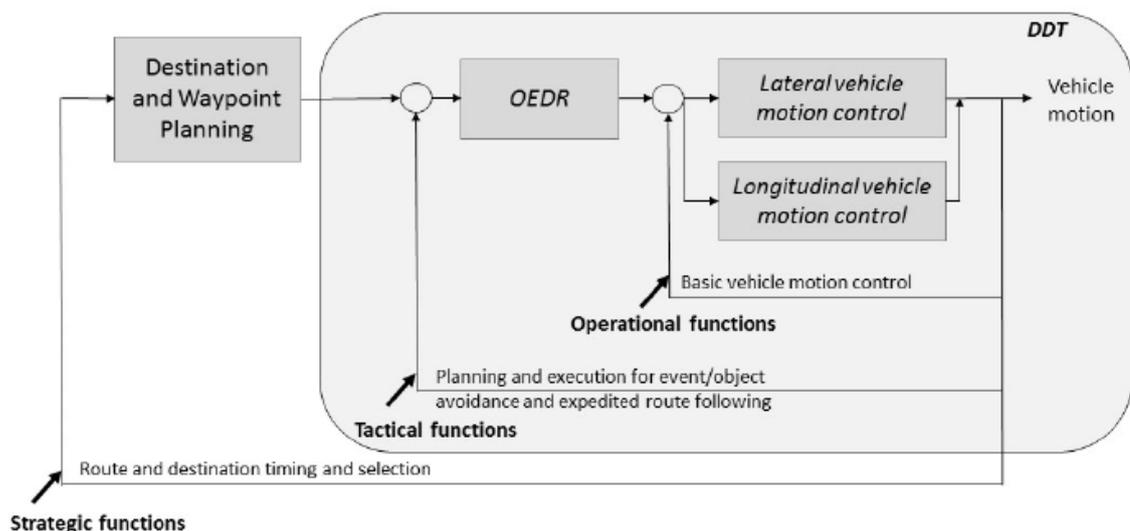


Figura 6.1: Vista esquemática de la tarea de conducción que muestra la sección de DDT [Int17].

Como se puede observar el nivel 1 abarca la automatización del bucle interior, es decir, control lateral o longitudinal del vehículo y el OEDR limitado a dicho eje. El nivel 2 abarca

el bucle de interior, es decir control lateral y longitudinal y el OEDR limitado al control de movimiento. Los niveles de 3 al 5 abarcan la automatización de ambos circuitos internos, es decir control lateral y longitudinal del movimiento del vehículo y OEDR completo.

Muy relacionado con el término anterior está el **DDT Fallback** o DDT de emergencia. Es definida como la respuesta del usuario o del ADS ante una emergencia ya sea para realizar el DDT o para lograr una **condición de riesgo mínima** tras un fallo del sistema relevante al DDT o la salida de la zona ODD.

Cabe destacar que el DDT y el DDT *Fallback* son funciones distintas y la capacidad de realizar uno no implica necesariamente la capacidad de realizar el otro. Los vehículos de nivel 3 pueden no realizar todo el DDT dentro de su ODD, ya que pueden no ser capaces de realizar la tarea DDT de emergencia en todas las situaciones requeridas por lo que el sistema emitirá una *solicitud de intervención* DDT al usuario. Los vehículos con un nivel 4 y 5 deben ser capaces de realizar la acción DDT de emergencia, así como lograr una *condición de riesgo mínima*. Aunque estos vehículos pueden llevar pasajeros, no tienen porque incluir un sistema físico para permitirles realizar maniobras DDT de emergencia.

Otros dos términos definidos por SAE son el **control de movimiento lateral** y **control de movimiento longitudinal** del vehículo. El primero, *movimiento lateral*, define la sub-tarea DDT que comprende las actividades necesarias para la regulación sostenida, en tiempo real, del eje *y* (véase Figura 6.2) del componente de movimiento del vehículo. Este control incluye la detección del posicionamiento del vehículo en relación con los límites del carril y la aplicación de las entradas de frenado de dirección y diferencial para mantener el posicionamiento lateral apropiado. El segundo, *movimiento longitudinal*, define la sub-tarea DDT que comprende las actividades necesarias para la regulación sostenida, en tiempo real, del eje *x* del componente de movimiento del vehículo. Este control del vehículo incluye mantener la velocidad establecida así como detectar un vehículo precedente en la trayectoria del vehículo sujeto, manteniendo un espacio apropiado al vehículo precedente y aplicando las entradas de propulsión o frenado para hacer que el vehículo mantenga esa velocidad o espacio.

Una **solicitud de intervención** es una notificación por parte del ADS al conductor indicándole que debe realizar rápidamente un retorno de DDT.

La **Object and Event Detection and Response (OEDR)** es definida como el conjunto de sub-tareas DDT que incluyen supervisar el entorno de conducción, es decir, detectar, reconocer y clasificar objetos y eventos además de prepararse para responder según sea necesario, y ejecutar una respuesta adecuada a dichos objetos y eventos, es decir, completar el DDT o efectuar una DDT de emergencia.

El **Operational Design Domain (ODD)** es definido como las condiciones específicas bajo las cuales un sistema de automatización de conducción determinado, o una característica del mismo, está diseñado para funcionar. Un ODD puede incluir limitaciones de diversos tipos:

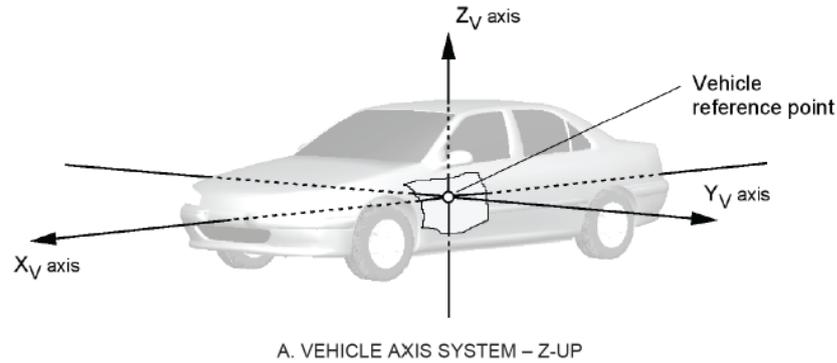


Figura 6.2: Diagrama que muestra los ejes de movimiento del vehículo [Int08].

geográficas, tipos de carreteras, condiciones ambientales, densidad de tráfico, velocidad, etc. Un ADS, en concreto, puede diseñarse para operar sólo bajo ciertas condiciones. Pueden existir distintos perfiles de un ODD bajo el cual se realizan distintas ADS.

Referente a los **roles** SAE define los siguientes términos:

- **Conductor:** un usuario que realiza en tiempo real una parte o la totalidad del DDT o DDT de emergencia de un vehículo en particular. Este término difiere del término convencional de conductor que controla manualmente el vehículo ya que puede ser que la conducción sea autónoma.
- **Usuario (humano):** un término general que hace referencia al rol humano en la conducción de la automatización. Los siguientes términos definen categorías de usuarios: conductor, pasajero, usuario preparado para la acción DDT de emergencia y despachador. Estas categorías humanas definen roles que no se superponen y pueden realizarse en secuencias variables durante un viaje determinado.
- **Pasajero:** es un usuario en un vehículo que no tienen ningún rol en la operación de dicho vehículo.

Hasta aquí los términos más importantes del estándar SAE que serán utilizados para definir el esquema que plantea este documento. Todos los términos, incluidos estos, pueden ser encontrados en el estándar **J3016A** [Int17]. Cabe destacar, que SAE define los términos: *Autonomous, Self-Driving, Driverless, Unmanned, Robotic* y *Automated* o *Autonomous Vehicle*, como obsoletos, aunque en este documento se han utilizado algunos con el fin de utilizar términos utilizados comúnmente hoy en día.

Gracias a estos términos se puede definir una taxonomía para la automatización de la conducción que consta de seis niveles discretos y mutuamente excluyentes.

Para la definición de esta taxonomía son fundamentales los roles respectivos del *usuario (humano)* y el sistema de automatización de manejo y la relación el uno con el otro. Debido a

que los cambios en la funcionalidad de un sistema de automatización de conducción pueden cambiar el rol del *usuario (humano)*, estos proporcionan una base para categorizar dichos sistemas. Por ejemplo:

- Si el sistema de conducción automatizada realiza las tareas de control de movimiento longitudinal y/o lateral del DDT entonces el conductor no realiza dicha acción, aunque este espere a que se complete la tarea DDT. Esta división de roles corresponde a los niveles 1 y 2.
- Si el sistema de conducción automatizada realiza todo el DDT entonces el *usuario (humano)* no lo hace, aunque, si se espera que un *usuario (humano)* esté preparado para asumir la tarea DDT de emergencia cuando ocurre un fallo del sistema o cuando el sistema abandona su ODD entonces se habla de un nivel 3.
- Si el sistema de conducción automatizada es capaz de realizar todas las acciones DDT y DDT de emergencia, dentro de su ODD o en cualquier situación, entonces cualquier usuario presente en el vehículo toma el rol de *pasajero*. Esta división de roles corresponde a los niveles 4 y 5.

Cabe destacar que el vehículo también cumple una función en esta taxonomía, aunque el papel del vehículo no cambia la función del usuario en la realización del DDT. Por ello, los sistemas de conducción autónoma se clasifican en niveles según:

- **Nivel 1:** Si el sistema de automatización de la conducción realiza la sub-tarea DDT de control de movimiento longitudinal o lateral.
- **Nivel 2:** Si el sistema de automatización de la conducción realiza simultáneamente la sub-tareas DDT de control de movimiento longitudinal y lateral.
- **Nivel 3:** Si el sistema de automatización de la conducción también realiza la sub-tarea DDT OEDR.
- **Nivel 4:** Si el sistema de automatización de la conducción también realiza tareas DDT de emergencia.
- **Nivel 5:** Si el sistema de automatización de la conducción está limitado por un ODD.

Dado que son niveles mutuamente excluyentes se puede definir el siguiente diagrama de flujos (véase Figura 6.3). En dicho diagrama podemos observar como ciertas condiciones hacen que el nivel descienda. Así, si un vehículo tiene una limitación ODD como mucho será de nivel 4, además, si no puede realizar una acción DDT de emergencia, como mucho será de nivel 3. Si ese mismo vehículo no puede realizar operaciones OEDR entonces será como mucho de nivel 2, y ya si no puede realizar ambas maniobras de control simultáneamente entonces será de nivel 1 y de nivel 0 si no puede realizar ninguna.

Does the feature:

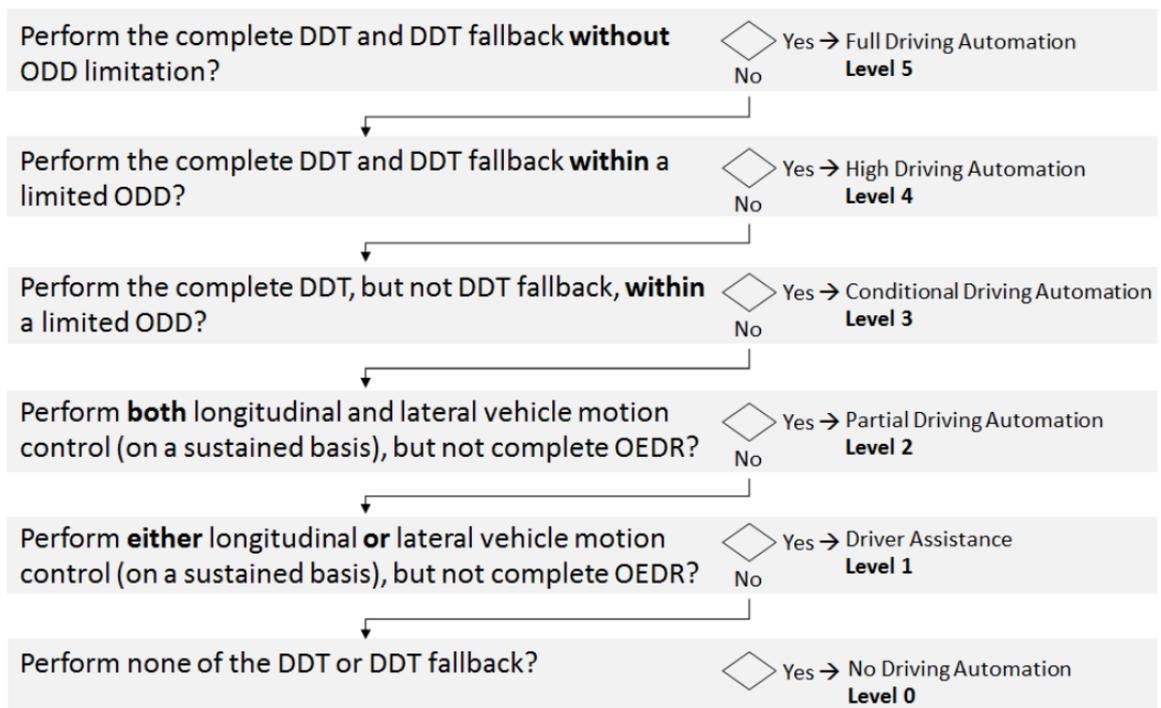


Figura 6.3: Diagrama de flujo lógico simplificado para asignar el nivel de automatización de conducción a una función [Int17].

Por esto, SAE, define los siguientes niveles:

- **Nivel 0:** la no automatización de la conducción. Define el control de todo el DDT por parte del *usuario (humano)* incluso cuando está mejorado por los sistemas de seguridad activa.
- **Nivel 1:** asistencia al conductor. La ejecución sostenida y limitada a un ODD por parte de un sistema de automatización de la conducción de las sub-tareas DDT de control de movimiento lateral o longitudinal (pero no ambas simultáneamente) con la expectativa de que el conductor realice el resto del DDT.
- **Nivel 2:** automatización de la conducción parcial. La ejecución sostenida y limitada a un ODD por parte de un sistema de automatización de la conducción de las sub-tareas DDT de control de movimiento lateral y longitudinal con la expectativa de que el conductor realice el resto del DDT.
- **Nivel 3:** automatización de conducción condicional. El rendimiento sostenido y específico a un ODD por un ADS de todas las acciones DDT con la expectativa de que el usuario de DDT de emergencia esté listo para recibir solicitudes de intervención emitidas por el ADS, así como también de fallas del sistema relevantes para el rendimiento de la tarea DDT en otros sistemas del vehículo.
- **Nivel 4:** alta automatización de la conducción. El rendimiento sostenido y específico a un ODD por parte de un ADS de todas las acciones DDT y DDT de emergencia, sin ninguna expectativa de que un usuario responda a una solicitud de intervención.
- **Nivel 5:** completa automatización de la conducción. El rendimiento sostenido e incondicional, no específico a un ODD de un ADS de todas las acciones DDT y DDT de emergencia sin ninguna expectativa de que un usuario responda a una solicitud de intervención.

6.1.1 Esquema propuesto

Siguiendo el estándar de SAE este documento propone un esquema que busca definir una solución escalable que alcance, teóricamente, un mínimo nivel 4. Para ello, y tras todos los datos recopilados y analizados, se definen los siguientes pasos a realizar para desarrollar un *vehículo autónomo*:

- **Paso 1: definición del entorno.** Como primer paso se busca definir el entorno ODD a partir de una serie de características inequívocas. Este entorno centrará el proyecto a realizar.
- **Paso 2: requisitos funcionales.** Como segundo paso a realizar se busca definir la serie de tareas que debe poder realizar el vehículo. Las denominadas DDT, DDT de emergencia y OEDR. Una vez definidas se deben enlazar con las características definidas

en el ODD. En este apartado también se definirán los algoritmos, métodos o sistemas utilizados para llevar a cabo dichas tareas.

- **Paso 3: iteración por niveles.** Como tercer paso se debe repetir el primer y segundo paso hasta elaborar una tabla con todas las características del entorno y las tareas DDT, DDT de emergencia y OEDR que se llevaran a cabo en cada nivel.
- **Paso 4: integración de sensores.** Como cuarto paso se busca definir el conjunto de sensores necesarios, y su localización, para realizar cada tarea DDT, DDT de emergencia y OEDR.
- **Paso 5: Análisis de costes.** Se analizarán los costes relativos a implementar la solución. Estos costes irán divididos por niveles y sus costes serán acumulativos según el nivel, es decir, implementar el nivel 2 implicará el coste del nivel 1 si el vehículo utilizado no lo tiene implementado.
 - **Paso 5.1: selección de componentes.** Como parte del quinto paso se seleccionarán los componentes que han sido definidos en el paso 4 entre los disponibles en el mercado.
- **Paso 6: implementación del nivel deseado.** Se implementará el nivel deseado en el vehículo y por consiguiente todos los niveles inferiores.

Cada iteración del paso 1 y 2 obtendrá como salida un esquema que defina el entorno correspondiente, las tareas involucradas en dicho nivel y el conjunto de sensores que servirá al nivel anterior como base. Una vez realizado el esquema será posible integrar nivel a nivel las necesidades del *vehículo autónomo* y se dispondrá de un conjunto de datos a utilizar e implementar en cada nivel.

A continuación se detallarán los pasos del esquema y se realizará el esquema del vehículo propuesto. Cabe destacar que el vehículo propuesto será un vehículo articulado que pueda maniobrar en un entorno minero y que alcance, como mínimo, un nivel SAE 4. Este vehículo será capaz de maniobrar con obstáculos dinámicos y estáticos, realizar tareas LHD, operar en condiciones de baja luminosidad como túneles mineros pero también a la luz del día, y por ello deberá implementar un sistema de localización no basado en GPS.

6.2 Definición del entorno

Como el estándar SAE define, los niveles 1 a 4 contemplan expresamente las limitaciones de ODD. Por el contrario, el nivel 5 niega expresamente tales limitaciones. En consecuencia, describir con precisión una característica (que no sea el nivel 5) requiere identificar tanto su nivel de automatización de conducción como su ODD. Como se proporciona en las definiciones anteriores, esta combinación de *nivel de automatización de conducción* y ODD se denomina *especificación de uso*, y por ello, una característica dada satisface una *especificación de uso* determinada.

Debido a la amplia gama de ODD posibles, puede existir una amplia gama de características posibles en cada nivel. Por ejemplo, el nivel 4 incluye estacionamiento, alta velocidad, baja velocidad, zona geográfica, etc. A diferencia de un ADS de nivel 5, un ADS de nivel 4 tiene un ODD limitado. Las restricciones geográficas o ambientales en un ADS-DV pueden reflejar las limitaciones de ODD de sus ADS.

La Figura 6.4 ilustra la ortogonalidad de ODD con relación a los niveles de automatización de conducción.

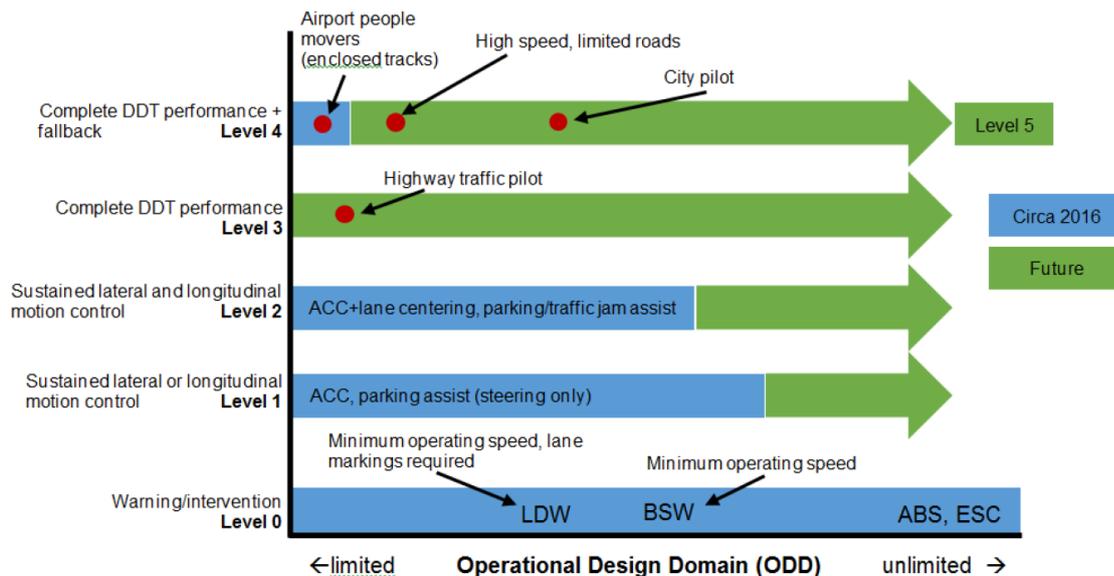


Figura 6.4: Ilustración de la importancia de ODD en relación con los niveles. [Int17].

Por ello es necesario definir bien, en un primer paso, el entorno que engloba el esquema que se desea proponer. Para ello se definirán una serie de características del entorno que posteriormente serán asignadas a tareas DDT. Estas son las características que se proponen como básicas, aunque opcionales, en el esquema propuesto:

- **Velocidad:** la velocidad a la que circulan los vehículos del entorno es muy importante. Por ejemplo en un entorno minero la velocidad es reducida debido a las grandes cargas que portan los vehículos.
- **Zona geográfica:** el entorno será limitado a una zona geográfica determinada. Por ejemplo una mina a cielo abierto, una mina bajo tierra o incluso una zona delimitada entre dos ciudades.
- **Condición ambiental:** dadas las tareas a realizar LHD se limitará el entorno a situaciones ambientales en las que un operario humano realizaría dicha tarea con soltura. Por lo tanto, en situaciones normales, el entorno no estará definido en situaciones ambientales fuertes o extremas de lluvia, granizo o nieve.

- **Horarias:** no existen, o no deberían existir, limitaciones horarias del entorno.
- **Tipos de carretera:** el tipo de carretera limitará las tareas a realizar. Esto quiere decir que una tarea LHD de gran tonelaje realizada en una cantera contemplará un tipo de carretera de tipo *camino* pero una tarea LHD de transporte de mercancías podrá ser realizada en una autovía.
- **Presencia de obstáculos:** se definirá un entorno donde será necesario evitar obstáculos móviles y estáticos.
- **Luminosidad:** se definirá un entorno donde será necesario realizar DDT con escasa iluminación.

Estas son las características básicas del entorno definido en el esquema propuesto en este documento. Estas características pueden ser ampliadas, u omitidas, según el proyecto.

Dado que se espera que el vehículo, definido por este esquema, pueda maniobrar en un entorno minero o un entorno de transporte de mercancías como mínimo se espera que este alcance un nivel 4.

6.2.1 Entorno propuesto

Para la elaboración del vehículo propuesto y siguiendo el paso 1 del esquema definido se obtienen la siguiente Tabla 6.1 con características del entorno separadas por nivel SAE.

Característica	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Velocidad	Baja velocidad	-	-	-
Zona geográfica	Túneles mineros.	-	Área externa de la mina.	-
Condición ambiental	-	-	Ausencia de fuertes lluvias, granizo o nevadas.	-
Horarias	Sin limitación de horario.	-	-	-
Tipos de carretera	Túneles mineros de doble carril definidos en un mapa.	-	Caminos de tierra señalizados.	-
Presencia de obstáculos	Vehículos autónomos.	-	Vehículos no autónomos y operarios.	Vehículos no autónomos y operarios localizados mediante GPS
Luminosidad	Condiciones casi nulas de iluminación.	-	Zonas iluminadas con luz natural.	-

Cuadro 6.1: Esquema características del entorno según nivel propuestas para el vehículo

Como se puede observar mediante esta definición de características, los niveles 1 y 2 están centrados en tareas realizadas en los túneles de la mina y los niveles 3 y 4 pueden ser

realizadas en el exterior de los túneles mineros.

6.3 Requisitos funcionales

En este paso del esquema es necesario identificar las maniobras DDT que es capaz de realizar el vehículo, como también las DDT de emergencia y las OEDR. Cabe destacar que hasta el nivel 3 no se requieren tareas DDT de emergencia y hasta el nivel 4 no se implementa en su totalidad el OEDR, por lo tanto dichas tareas no se añadirán en esos niveles del esquema.

Por ello se recomienda definir las siguientes DDT básicas y los algoritmos a utilizar:

- **DDT**
 - **Control de movimiento lateral del vehículo.** Se utilizara un algoritmo de planificación local para trazar el recorrido lateral a realizar.
 - **Control de movimiento longitudinal del vehículo** a través de aceleración y deceleración.
 - **Unión a carril** realizando una maniobra de inserción a una vía ocupada por más vehículos.
 - **Salida de carril** realizando una maniobra de abandono de una vía ocupada por más vehículos.
 - **Operación de descarga.**
 - **Planificación de la maniobra.**
- **DDT de emergencia:**
 - **Retorno al origen** realizando una maniobra que devuelva el vehículo al punto de inicio que tenía antes de realizar la tarea DDT.
 - **Incorporación fallida** maniobra de control de movimiento longitudinal que acelera o desacelera el vehículo para así evitar un choque en una incorporación fallida.
 - **Frenado de emergencia** maniobra que realiza un frenado de emergencia siempre y cuando el estado final suponga una *condición de riesgo mínima*.
- **OEDR:**
 - **Monitoreo del entorno** de conducción a través de la detección de objetos y eventos, reconocimiento, clasificación y preparación de respuestas. Toma de decisiones mediante redes de Petri.
 - **Ejecución de respuesta** de objetos y eventos.
 - **Monitoreo de componentes** que permite al vehículo comunicarse con el entorno y localizar a otros vehículos (autónomos o no), a los operarios de la mina y conocer eventos del entorno sin verlos.

Estas son las tareas básicas que deberá implementar un sistema ya que su conjunto permite maniobrar un vehículo en cualquier entorno. Cabe destacar que existen otras maniobras y por lo tanto estas pueden ser ampliadas.

Los algoritmos utilizados para la mayoría de las DDT son los relacionados con la planificación de trayectorias local, como *campos potenciales* y RRT y los relacionados con la creación de mapas como SLAM en especial OGM para niveles bajos y un modelado 3D en niveles altos. Para las tareas de planificación o DDT de emergencia y OEDR se utiliza un sistema de toma de decisiones mediante redes de Petri además del uso del estándar *HERE* definido por *Nokia* para comunicaciones *v2e* o vehículo a entorno. Además, todas las maniobras relacionadas con el control del vehículo conllevan sistemas de control de maniobras y un modelo cinemático acorde al vehículo.

6.3.1 Requisitos funcionales propuesto

: Para la realización del paso 2 se utilizarán las tareas básicas definidas en el esquema básico de requisitos funcionales. En la Tabla 6.2 a 6.5 se detallan como se irán implementando dichas tareas según el nivel SAE y los algoritmos a utilizar.

Nivel	Nombre	Descripción	Tipo	Entorno
1	Ctrl de mov. longitudinal del vehículo	Se espera que el vehículo pueda seguir los túneles de la mina en el sentido determinado o seguir a un vehículo que tenga delante.	DDT	Baja luminosidad. Red de túneles. Túnel. Presencia de otros vehículos autónomos.
1	Operación de descarga	Se espera que el vehículo pueda realizar una operación de descarga de materiales en un punto determinado señalado debidamente.	DDT	Baja luminosidad. Red de túneles. Túnel. Presencia de otros vehículos autónomos.
1	Planificación de la maniobra	Se espera que el vehículo pueda decidir que maniobra realizar según una planificación previamente definida.	DDT	Presencia de otros vehículos autónomos.

Cuadro 6.2: Esquema tareas DDT de nivel 1 propuestas para el vehículo

Nivel	Nombre	Descripción	Tipo	Entorno
2	Ctrl. de mov. lateral del vehículo	Se espera que el vehículo pueda cambiar de carril para realizar adelantamientos.	DDT	Baja luminosidad. Red de túneles. Túnel. Presencia de otros vehículos autónomos.
2	Salida de carril	Se espera que el vehículo pueda salir del carril.	DDT	Baja luminosidad. Red de túneles. Túnel. Presencia de otros vehículos autónomos.

Cuadro 6.3: Esquema tareas DDT de nivel 2 propuestas para el vehículo

Por último los principales algoritmos para realizar dichas tareas serán:

Nivel	Nombre	Descripción	Tipo	Entorno
3	Unión a carril	Se espera que el vehículo realice una maniobra de abandono de una vía ocupada por más vehículos.	DDT	Baja luminosidad. Red de túneles. Túnel. Presencia de otros vehículos.
3	Retorno al origen	Se espera que el vehículo realice una maniobra que devuelva el vehículo al punto de inicio que tenía antes de realizar la tarea DDT.	DDT Fallback	Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios.
3	Incorporación fallida	Se espera que el vehículo realice una maniobra de control de movimiento longitudinal que acelera o desacelera el vehículo para así evitar un choque en una incorporación fallida.	DDT Fallback	Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios.
3	Frenado de emergencia	Se espera que el vehículo realice una maniobra que realice un frenado de emergencia siempre y cuando el estado final suponga una <i>condición de riesgo mínima</i> .	DDT Fallback	Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios.
3	Monitoreo del entorno	Se espera que el vehículo realice una conducción a través de la detección de objetos y eventos, reconocimiento, clasificación y preparación de respuestas.	OEDR	Luminosidad variable. Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios.
3	Ejecución de respuesta	Se espera que el vehículo realice una ejecución de respuesta en base a unos objetos y eventos.	OEDR	Luminosidad variable. Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios.

Cuadro 6.4: Esquema tareas DDT, DDT Fallback y OEDR de nivel 3 propuestas para el vehículo

Nivel	Nombre	Descripción	Tipo	Entorno
4	Monitoreo de componentes	Se permite al vehículo comunicarse con el entorno y localizar a otros vehículos (autónomos o no), a los operarios de la mina y conocer eventos del entorno sin verlos.	OEDR	Luminosidad variable. Área total de la mina. Túneles y caminos. Presencia de otros vehículos y operarios localizados.

Cuadro 6.5: Esquema tareas OEDR de nivel 4 propuestas para el vehículo

- **OGM:** para detectar señales o balizas mediante imagen.
- **RRT:** para planificar trayectorias donde se vean involucrados obstáculos dinámicos.
- **Campos potenciales:** para planificar trayectorias a partir de OGM.
- **Redes de Petri:** para toma de decisiones.

6.4 Integración de sensores

A partir de las tareas DDT del paso anterior es posible definir los sensores a utilizar, según las características del entorno, los algoritmos utilizados y la acción a realizar. Para ello se define una tabla que contenga la tarea DDT a realizar y los sensores: LiDAR, radar, cámaras infrarroja y GPS y un sistema INS y se selecciona *Si* o *No* si son necesarios o no para realizar dicha tarea. Esta tabla se va actualizando según se van realizando iteraciones.

A continuación se detalla la colocación de los sensores según los documentos analizados.

Para los sensores LiDAR se plantean dos soluciones: cuatro sensores de 360 grados, o ocho sensores de 120 grados. La primera solución es la solución aplicada por el **Volvo FMX** que como ya se ha visto obtiene una buena cobertura redundante de hasta tres sensores simultáneos y un 80 % de cobertura mediante la colocación de cuatro sensores de 360 grados en las esquinas del vehículo. Para esta solución se utilizarían 4 sensores LiDAR de **Velodyne** modelo *Puck Lite*² con un coste de \$7.999³, seleccionados por su coste, rango y puntos por toma, lo que generaría un gasto de \$32.000.

La otra solución supone colocar ocho sensores de 120 grados en, cuatro en las esquinas del vehículo y dos, frontal y trasero, colocados según la Figura 6.5.

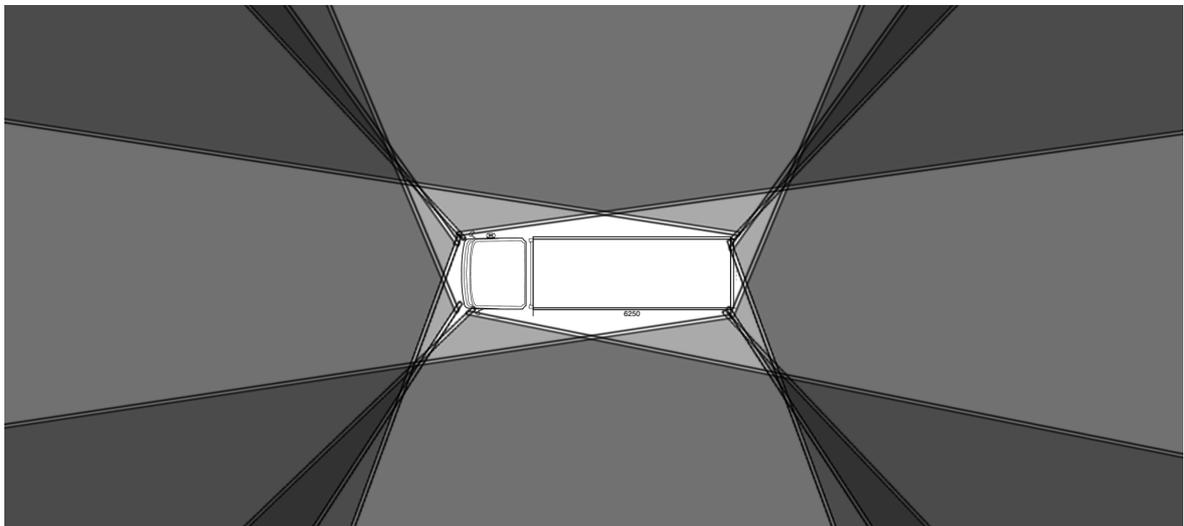


Figura 6.5: Colocación propuesta de ocho LiDAR de estado sólido de 120 grados de apertura.

Gracias a esta colocación se obtiene una buena cobertura de hasta tres sensores LiDAR similar a la de *Volvo* aunque con menos porcentaje de cobertura. El modelo S3 de **Quanergy**⁴

²<http://www.velodynelidar.com/vlp-16-lite.html>

³<https://www.spar3d.com/news/lidar/vol12no37-velodyne-announces-puck-lidar-sensor/>

⁴<http://quanergy.com/s3>

y el modelo *Velarray*⁵ de *Velodyne* son sensores de estado sólido y rondan un coste de \$250 unidad, \$1.500 total, lo que los hace una buena solución para un primer vehículo.

La solución elegida que se aconseja en este documento es la colocación de ocho sensores LIDAR de estado sólido de 120 grados.

En lo relativo a los sensores radares se aconseja que sean colocados tres delanteros, dos traseros y otros dos en las esquinas traseras ya que así cubren los puntos ciegos del vehículo. Como sensor se recomienda el modelo *Mid-Range Radar* (MRR) de **Bosch**⁶ con un alcance máximo entre 12m y 160m dependiendo del ángulo de apertura.

Por último, la colocación de las cámaras infrarrojas que se aconseja es la elegida por los vehículos *Tesla* con ocho cámaras repartidas por el vehículo (tres frontales, dos laterales delanteras, dos laterales traseras y una trasera) ya que se obtiene una visión 360 grados del entorno que se puede complementar con los datos LIDAR y radar⁷. En concreto se recomienda el uso del modelo *Viper* de **Adasky**⁸.

Como sistema GPS e INS se recomienda el uso de *Spatial*⁹. Otra opción es el *MTi-G-710 GNSS / INS* que es un *GNSS / INS* asistido por *GNSS* e *IMU* que ofrece una posición y orientación de alta calidad¹⁰.

Respecto al controlador que aporta el cerebro del vehículo, se recomienda el uso del *NVIDIA Drive PX* en concreto el último modelo *Pegasus* y su SDK, aunque los modelos *Parker AutoCruise* y *Parker AutoChauffeur*¹¹ pueden ser utilizados para los niveles SAE 1 y 2.

Cabe destacar que *NVIDIA* en alianza con **AutonomousStuff** ofrece una serie de kits con todo lo necesario para montar un *vehículo autónomo*¹² por lo que como se puede observar, los sensores utilizados para un vehículo pueden ser muy variados y todo depende del presupuesto y los puntos que se quieran cubrir.

6.4.1 Integración de sensores propuestos

Siguiendo el paso 4 del esquema y ya una vez definidos los sensores que se van a utilizar para realizar las tareas se pasa a definir la Tabla 6.6 que detalla la relación existente entre las tareas DDT y los sensores necesarios para llevarlas a cabo.

Como se ve en la tabla para la mayoría de las tareas es necesario el uso de todos los sensores, sobre todo para las tareas de monitorización del entorno ya que necesita utilizar todos los datos del entorno para obtener información.

⁵[http://www.businesswire.com/news/home/20170419005516/en/Velodyne-LiDAR-Announces-New-\"Velarray\"-LiDAR-Sensor](http://www.businesswire.com/news/home/20170419005516/en/Velodyne-LiDAR-Announces-New-\)

⁶<http://www.bosch-mobility-solutions.com>

⁷<https://www.motorpasion.com/tecnologia/lidar-si-o-lidar-no-esa-es-la-cuestion-con-los-coches-autonomos>

⁸<http://www.adasky.com/>

⁹<https://www.advancednavigation.com.au/product/spatial>

¹⁰<https://autonomoustuff.com/product/xsens-mti-g-710/>

¹¹<https://www.nvidia.com/en-us/self-driving-cars/drive-px/>

¹²<https://autonomoustuff.com/product/xsens-mti-g-710/>

Tarea	LiDAR	Radar	Cameras	GPS
Ctrl de mov. longitudinal del vehículo	No	Si	Si	No
Ctrl de mov. lateral del vehículo	No	Si	Si	No
Operación de descarga	No	Si	Si	No
Planificación de la maniobra	No	Si	Si	Si
Salida de carril	No	Si	Si	Si
Unión a carril	No	Si	Si	Si
Retorno de Origen	No	No	No	Si
Incorporación fallida	No	Si	No	No
Frenado de emergencia	No	Si	No	No
Monitoreo del entorno	Si	Si	Si	Si
Ejecución de respuestas	Si	Si	Si	Si
Monitoreo de componentes	No	No	No	Si

Cuadro 6.6: Esquema de sensores según nivel propuestos para el vehículo

6.5 Análisis de costes

Con los pasos del 1 al 4 del esquema ya realizados y definidos y la solución propuesta detallada a partir de él, es posible realizar un análisis de costes a partir de los sensores requeridos para cada nivel SAE que correspondería con el quinto paso.

En las siguientes secciones se detallan los modelos de dispositivos que se han analizado detalladamente y el por qué de su selección. Este paso corresponde con el 5.1.

6.5.1 LiDAR

Como se ha detallado en la sección 6.4, se recomienda el uso de sensores LiDAR de estado sólido, más en concreto ocho dispositivos colocados según la Figura 6.5. Una de las principales causas de utilizar un dispositivo LiDAR de estado sólido es su reducido coste en comparación a otros dispositivos aun a expensas de perder ángulo de *visión* ya que estos dispositivos cubren 120 grados solamente aunque para ello se ha planteado una colocación que ofrece un solapamiento similar (redundancia de tres dispositivos) a la solución estándar de cuatro dispositivos LiDAR de 360 grados. Los dos dispositivos LiDAR de estado sólido más remarcables son:

- **Quanergy modelo s3:** es un LiDAR de estado sólido con un arco de emisión de **120 grados** horizontal y 10 grados vertical. Según las especificaciones procesa **320.000 de puntos por segundo**. Tiene un **alcance de 150m** con un 80 % de reflexión y un rango mínimo de 5cm¹³. Coste esperado: \$250¹⁴.
- **Velodyne Velarray:** es un LiDAR de estado sólido con un arco de emisión de **120 grados** horizontal y 35 grados vertical. Tiene un **alcance de 200m**. Se prevé que tendrá un coste de unos cientos de dólares.

¹³<http://sensata.com/events/iaa/media/Tech%20Sheet%20-%20LiDAR%20S3.ENGLISH.pdf>

¹⁴<https://www.spar3d.com/news/hardware/vol14no2-quantenergy-solid-state-lidar-details/>

Aunque el modelo de *Velodyne* tiene mejores características en los apartados conocidos, actualmente está en desarrollo por lo que el modelo de **Quanergy** tiene el punto a favor de la experiencia ya que actualmente está en uso. Por ello, aunque su alcance sea menor y su arco de emisión vertical sea más reducido, se ha elegido el modelo **S3** de *Quanergy* para el desarrollo de este vehículo.

6.5.2 Radar

Para la solución propuesta se han seleccionado siete dispositivos de radar, tres delanteros, dos traseros laterales y dos traseros que cubren los puntos ciegos y la mayoría de las maniobras. Entre los dispositivos existentes tenemos:

- **Bosch Mid-Range Radar (MRR)**: este dispositivo radar tiene un rango de distancia máxima que oscila entre **160m con una apertura de 6 grados** y **12m con una apertura de 42 grados** cuando se posiciona al frente del vehículo y de 70m con una apertura de 5 grados cuando se posiciona en la parte trasera del vehículo¹⁵.
- **Delphi Electronically Scanning RADAR (ESR) 2.5**: con un rango medio de 60m con una apertura de 45 grados y 174m con una apertura de 10 grados¹⁶.

Aunque el modelo de *Delphi* tiene un rango superior, el modelo de *Bosch* se ajusta mejor al rango que se desea ya que el principal uso de este vehículo se realizará en túneles donde no existe gran separación entre paredes y el rango máximo no superara los 12m. También cabe destacar que las tareas se realizarán en un entorno de baja velocidad. Para el vehículo propuesto se ha seleccionado el modelo de **Bosch MRR**. Se estima que el coste de todos los sensores ronde los \$10.000¹⁷.

6.5.3 Cámaras infrarrojas

Para el modelo planteado se han seleccionado ocho cámaras infrarrojas posicionadas alrededor del coche. Han sido seleccionadas las cámaras **Viper** de **Adasky** ya que proporcionan una visión total en cualquier situación además de que están enfocadas a vehículos con un nivel 3 o superior. Gracias a estas cámaras infrarrojas es posible detectar balizas lumínicas y personas en total oscuridad. Se estima un coste alrededor \$12.000 por el conjunto total de cámaras.

6.5.4 GPS e INS

Dado que el vehículo propuesto debe realizar tareas en túneles donde no, por cuestiones lógicas, no alcanza la señal GPS es necesario que el sistema posea un INS.

Se ha seleccionado el modelo **Spatial** que engloba un sistema de navegación inercial asistido por GPS en miniatura reforzado y Attitude Heading Reference System (AHRS) que pro-

¹⁵<http://www.bosch-mobility-solutions.com>

¹⁶<https://autonomoustuff.com/product/delphi-esr-2-5-24v/>

¹⁷<https://qz.com/924212/what-it-really-costs-to-turn-a-car-into-a-self-driving-vehicle/>

porciona una posición, velocidad, aceleración y orientación precisas en las condiciones más exigentes. Combina acelerómetros calibrados por temperatura, giroscopios, magnetómetros y un sensor de presión con un receptor Global Navigation Satellite System (GNSS) avanzado. Estos se combinan en un sofisticado algoritmo de fusión para ofrecer una navegación y orientación precisa y confiable.

6.5.5 Controlador

Para el controlador, o cerebro principal, del vehículo se ha elegido el modelo **NVIDIA Drive XP** ya que otorga un SDK de desarrollo que permite controlar diversas tareas del vehículo. Este controlador permite además añadir todos los sensores y procesar la información que obtiene de ellos. El modelo *Pegasus* permite un control de nivel SAE 5 del vehículo. Se calcula que el coste de un kit de desarrollo alcanza los \$15.000¹⁸.

6.5.6 Solución completa

Una vez se ha realizado el paso 5.1 del esquema es posible terminar el paso 5 donde se detallan los gastos por nivel, por ello en la Tabla 6.7 se pueden ver los elementos necesarios según la Tabla 6.6 y los requisitos especificados de cantidad y localización.

Dispositivo	Modelo	Cantidad	Precio/unidad
LiDAR	Quanergy S3	8	~\$250
Radar	Bosch Mid-Range Radar (MRR)	7	~\$1.100*
Cámara Infrarroja	Adasky Viper	8	~\$1.500*
GPS e INS	Spatial	1	\$3.500
Controlador	NVIDIA Drive PX Pegasus	1	~\$15.000

Cuadro 6.7: Tabla de sensores con sus unidades requeridas y precio por unidad. *El precio de este componente ha sido calculado en base al coste medio de implementar el conjunto de sensores según los precios estimados en <https://qz.com/924212/what-it-really-costs-to-turn-a-car-into-a-self-driving-vehicle/>

A partir de la Tabla 6.6 y gracias a la Tablas 6.2 - 6.5 realizadas siguiendo el esquema propuesto, se desglosan los siguientes costes según el nivel SAE del vehículo (véase Tablas 6.8 - 6.10).

Dispositivo	Cantidad	Precio/unidad	Total	Gasto nivel
Radar	7	~\$1.100	~\$7.700	~\$7.700
Cámara Infrarroja	8	~\$1.500	~\$12.000	~\$12.000
Controlador	1	~\$15.000	~\$15.000	~\$15.000
			~\$34.700	~\$3.500

Cuadro 6.8: Costes de los sensores de un vehículo de nivel 1 y 2

¹⁸<https://www.anandtech.com/show/11913/nvidia-announces-drive-px-pegasus-at-gtc-europe-2017-feat-nextgen-gpus>

Dispositivo	Cantidad	Precio/unidad	Total	Gasto nivel
Radar	7	~\$1.100	~\$7.700	\$0
Cámara Infrarroja	8	~\$1.500	~\$12.000	\$0
GPS e INS	1	\$3.500	\$3.500	\$3.500
Controlador	1	~\$15.000	~\$15.000	\$0
			~\$38.200	~\$3.500

Cuadro 6.9: Costes de los sensores de un vehículo de nivel 3.

Dispositivo	Cantidad	Precio/unidad	Total	Gasto nivel
LiDAR	8	~\$250	~\$2.000	~\$2.000
Radar	7	~\$1.100	~\$7.700	\$0
Cámara Infrarroja	8	~\$1.500	~\$12.000	\$0
GPS e INS	1	\$3.500	\$3.500	\$0
Controlador	1	~\$15.000	~\$15.000	\$0
			~\$40.200	~\$2.000

Cuadro 6.10: Costes de los sensores de un vehículo de nivel 4.

Al coste de los sensores se le debe añadir el coste del camión con volquete y los actuadores. Esto puede añadir un sobrecoste de ~\$150.000 al precio final del proyecto. Por lo que el precio total de un camión *Volvo FMX* automatizado a un nivel SAE 4 sería de: ~\$190.000.

Conclusiones

EN este capítulo se expondrán las conclusiones obtenidas a lo largo del estudio realizado y se detallara como han sido alcanzados los objetivos propuestos. Para ello se repasara el objetivo general propuesto en este documento y los objetivos parciales derivados de el y se detallaran las conclusiones de llevar a cabo dicho objetivo.

7.1 Conclusiones

Como resultado final de la realizado del estudio detallado de los dispositivos hardware de sensorización, prototipos de camiones autónomos existentes en la actualidad, y técnicas y algoritmos relativos a la percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias, toma de decisiones y control del movimiento se ha obtenido un esquema de trabajo que establece los pasos recomendados desde que se comienza a diseñar una solución hasta que el camión autónomo se implanta realmente, además de la realización de una solución conceptual de un camión autónomo para la resolución de problemas LHD.

En este documento se ha planteado un objetivo principal y una serie de objetivos parciales que desglosan el principal. A continuación se detallaran las conclusiones que se han obtenido tras la realización de este estudio, la realización del esquema de trabajo y el diseño del camión conceptual.

7.1.1 Solución conceptual de un camión autónomo para la resolución de problemas LHD

El objetivo principal que este proyecto tiene es el *diseño de una solución conceptual de un camión autónomo para la resolución de problemas LHD*. Este objetivo se ha cumplido satisfactoriamente ya que se ha elaborado, siguiendo el esquema propuesto en este documento, una solución conceptual de camión autónomo. En dicho objetivo se plantearon una serie de puntos a cumplir:

- **Se ha definido una solución escalable que sigue los niveles SAE.** Se ha creado una solución que itera sobre ella misma para ir añadiendo complejidad al sistema y así ir escalando niveles SAE. Por lo que siguiendo ese esquema se ha detallado una solución

escalable.

- **Se han detallado los pasos a seguir para implementar los distintos niveles SAE.** Dado que se ha utilizado el esquema definido en el punto anterior se puede decir que la solución del vehículo, sus tareas DDT han sido detallados según el nivel.
- **Se han detallado el conjunto de sensores HW que permite cubrir todas las necesidades de cada nivel SAE.** Dado que el esquema utilizado define una serie de tareas a realizar según el nivel SAE y posteriormente se identifican los sensores requeridos para realizar dicha tarea, se puede confirmar que se han detallado los sensores a utilizar por la solución.
- **El SW necesario para implementar cada nivel** se ha visto reducido a un SDK que otorga la empresa *NVIDIA* para sus controladores. Dado que se ha determinado que el SW está ligado al modelo del sensor y estos a las tareas que a su vez están ligados a los niveles, se ha determinado que el esquema definido no puede recomendar sobre que SW elegir.

Como conclusión de estos puntos se puede extraer que la definición de una solución conceptual de un camión autónomo se ha llevado a cabo siguiendo el esquema definido en este documento y que ha sido elaborada correctamente dado que ha cumplido con todos los puntos de forma correcta.

7.1.2 Objetivos parciales

En este documento, a partir del objetivo principal, o general, se establecieron una serie de objetivos parciales. Dichos objetivos parciales han sido elaborados correctamente sin problema alguno. A continuación se detallaran las conclusiones de cada objetivo parcial, tal y como se ha realizado en la sección anterior.

- **La identificación y definición:** de los dispositivos y técnicas aplicadas a la conducción autónoma en general, que pueden extrapolarse para la resolución de problemas LHD, se ha realizado mediante un **estudio minucioso y sistemático** de los dispositivos hardware de sensorización, prototipos de camiones autónomos existentes en la actualidad, y técnicas y algoritmos relativos a la percepción y modelado del entorno, localización, construcción de mapas, planificación de trayectorias.
- **El esquema:** se ha realizado siguiendo la **taxonomía** definida por el **estándar SAE** y los niveles definidos en dicho estándar. Dicho esquema define una serie de **pasos** que de ser seguidor generaran una serie de datos útiles para la implementación de los distintos niveles SAE de un vehículo autónomo.
- **Escalabilidad:** el esquema ha sido realizado siguiendo **los niveles SAE**. Dicho esquema define una serie de pasos que se realizan de **forma iterativa** para así alcanzar dicha escalabilidad dado que es posible definir un esquema para nivel 1 o para nivel 4 con

solo repetir los pasos 1 y 2.

- **La estimación de costes** ha sido realizada a partir del **conjunto de dispositivos** necesarios para implementar los distintos niveles SAE. Estos dispositivos han sido seleccionados según su tipo a partir de una comparación entre distintos dispositivos similares.
- **La usabilidad** se ha implementado mediante la inclusión de la **taxonomía** de SAE en el esquema y la descripción de los términos a utilizar se ha realizado de manera clara y sencilla para así hacerla más amigable a alguien neófito en la materia. Además la solución planteada se ha realizado lo más actual posible, utilizando dispositivos recientes o futuros, con la intención de que dicha solución no se quede obsoleta de aquí a unos años por lo que puede ser utilizada por empresas que quieran implementar un vehículo autónomo en la actualidad.

7.1.3 Trabajo futuro

Una vez planteada la solución conceptual de un camión autónomo se deben plantear los siguientes pasos a tomar, por eso como trabajo futuro basado en este documento se propone:

- **Implementar la solución planteada de forma escalable.** Para ello se realizaran pruebas de concepto donde se utilizaran pequeños vehículos como por ejemplo carretilla o toros mecánicos capaces de transportar mercancías o la versión pequeña de los volquetes, los vehículos dumper. La implementación en pequeños vehículos abaratará el coste inicial de \$150.000 del precio del camión y gracias a su pequeño tamaño permitirá trabajar en zonas más reducidas y controladas para mantener las pruebas bajo un estricto control.
- Como se definió en el último apartado del método de trabajo, una de las tareas futuras sería **mantener en observación las tecnologías implicadas** para así permitir adaptar el esquema a futuros estándares y dispositivos.

7.1.4 Reflexiones personales

Este estudio no habría podido ser realizado sin los conocimientos adquiridos en el **Máster de Ing. Informática** de la **UCLM** ya que gracias al máster he adquirido los conocimientos necesarios para poder elaborar este documento de la manera correcta.

Como mención especial he de remarcar que gracias al máster he podido:

- **Aplicar estándares a situaciones reales.** Lo aprendido en las asignaturas relacionadas con los estándares ISO me ha dado las bases necesarias para entender el estándar SAE y poder aplicarlo al esquema propuesto ya que me han otorgado la capacidad de análisis, síntesis y evaluación necesaria. Estas asignaturas también me han otorgado la capacidad para analizar las necesidades que se plantean en un entorno.

- **Realizar una investigación tecnológica.** Las asignaturas del máster me han enseñado la capacidad para argumentar y justificar lógicamente las decisiones tomadas, como también la capacidad de resolución de problemas aplicando técnicas de ingeniería.

Por último destacar que el máster ha desarrollado cualidades útiles para el desarrollo de este documento como es el aprendizaje autónomo, mi motivación por la calidad y el razonamiento crítico.

Referencias

- [Abe07] D. Abeijón. Fusion de Datos. <https://upcommons.upc.edu/bitstream/handle/2099.1/4376/03.pdf>, 2007.
- [ACO15] R.M. Eustice A.G. Cunningham, E. Galceran y E. Olson. MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. En *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, páginas 1670–1677. IEEE, 2015.
- [ADIEM03] J.M. Armingol A. De la Escalera y M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and vision computing*, 21(3):247–258, 2003.
- [AGU12] P. Lenz A. Geiger y R. Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. En *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12*, páginas 3354–3361, Washington, DC, USA, 2012. IEEE Computer Society. url: <http://dl.acm.org/citation.cfm?id=2354409.2354978>.
- [AV13] O. Adiyatov y H.A. Varol. Rapidly-exploring random tree based memory efficient motion planning. En *Mechatronics and Automation (ICMA), 2013 IEEE International Conference on*, páginas 354–359. IEEE, 2013.
- [BC04] T.D. Barfoot y C.M. Clark. Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46(2):65–78, 2004.
- [BL91] J. Barraquand y J-C. Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991.
- [BLG99] K. Cornelis B. Leibe, N. Cornelis y L. Van Gool. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,, 1999.
- [BLG07] K. Cornelis B. Leibe, N. Cornelis y L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1–8, 2007.

- [Blo96] I. Bloch. Information combination operators for data fusion: a comparative review with classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 26(1):52–67, 1996.
- [Cas12] G. Castillo. Navegación Integrada INS-GPS: Aplicación a un SAR aerotransportado. Master's thesis, Universidad de Buenos Aires. Facultad de Ingeniería, 2012. url: <http://psic.fi.uba.ar/Publicaciones/Tesis/Castillo/Castillo.pdf>.
- [Cas13] Federico Castanedo. A Review of Data Fusion Techniques. *The Scientific World Journal*, 2013(704504):19, 2013.
- [CBS92] J. Leonard B. Rao C. Brown, H. Durrant-Whyte y B. Steer. Distributed data fusion using Kalman filtering: A robotics application. *Data Fusion in Robotics and Machine Intelligence*, páginas 267–309, 1992.
- [CFP12] M. Pearson C. Fox, M. Evans y T. Prescott. Tactile slam with a biomimetic whiskered robot. En *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, páginas 4925–4930. IEEE, 2012.
- [Cha85] B. Chazelle. Approximation and decomposition of shapes. *Algorithmic and Geometric Aspects of Robotics*, 1:145–185, 1985.
- [CL09] C.G. Cassandras y S. Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.
- [CN08] M. Cummins y P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. url: <https://doi.org/10.1177/0278364908090961>.
- [Cso98] M. Csorba. Simultaneous localisation and map building. Master's thesis, Robotic Recherche Group Department of Engineering of Oxford, 1998.
- [CWH07] S. Thrun C. Wang, C. Thorpe y M. Hebert. Simultaneous Localization, Mapping and Moving Object Tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007. url: <https://doi.org/10.1177/0278364907081229>.
- [Das97] B. V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.
- [Dea13] J.P. Desai y et al. *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, volume 88. Springer, 2013.
- [DGO10] F. Cuesta D.A. García, F Bravo y A. Ollero. Planificación de Trayectorias con el Algoritmo RRT. Aplicación a Robots No Holónomos. 3, 01 2010.

- [DPDW98] E.M. Nebot D. Pagac y H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, páginas 623–629, 1998.
- [DW88] Hugh F. Durrant-Whyte. Sensor Models and Multisensor Integration. *The International Journal of Robotics Research*, 7(6):97–113, 1988. url: <http://dx.doi.org/10.1177/027836498800700608>.
- [EBL12] E. Bossé E. Blasch y D. Lambert. *High-Level Information Fusion Management and System Design*. Artech House Publishers, 2012.
- [EGO17] R.M. Eustice E. Galceran, A.G. Cunningham y E. Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, páginas 1–16, 2017.
- [Elf90] A. Elfes. *Sonar-Based Real-World Mapping and Navigation*, páginas 233–249. Springer New York, New York, NY, 1990. url: https://doi.org/10.1007/978-1-4613-8997-2_18.
- [Elh15] A. Elhassan. Autonomous driving system for reversing an articulated vehicle, 2015.
- [FIAH12] U. Malik F. Islam, J. Nasir, Y. Ayaz, y O. Hasan. RRt-smart: Rapid convergence implementation of RRT towards optimal solution. En *Mechatronics and Automation (ICMA), 2012 International Conference on*, páginas 1651–1656. IEEE, 2012.
- [FS89] K. Fujimura y H. Samet. A hierarchical strategy for path planning among moving obstacles (mobile robot). *IEEE transactions on robotics and Automation*, 5(1):61–69, 1989.
- [FV11] A. Furda y L. Vlacic. Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. *IEEE Intelligent Transportation Systems Magazine*, 3(1):4–17, 2011.
- [Gar12] D. A. López García. *Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos*, volume 124. Universidad de Huelva, 2012.
- [GD12] T. Gu y J.M. Dolan. On-road motion planning for autonomous vehicles. En *International Conference on Intelligent Robotics and Applications*, páginas 588–597. Springer, 2012.

- [GDB00] H. Durrant-Whyte G. Dissanayake y T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. *IEEE International Conference on Robotics and Automation*, 2:1009–1014, 2000.
- [Gea08] T. Gindele y et al. Design of the planner of Team AnnieWAY’s autonomous vehicle used in the DARPA Urban Challenge 2007. En *Intelligent Vehicles Symposium, 2008 IEEE*, páginas 1131–1136. IEEE, 2008.
- [Gie04] J. Giesbrecht. Global Path Planning for Unmanned Ground Vehicles. página 56, 12 2004.
- [Hay86] V. Hayward. Fast collision detection scheme by recursive decomposition of a manipulator workspace. En *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, páginas 1044–1049. IEEE, 1986.
- [HBL14] D. Hsu H. Bai y W.S. Lee. Integrated perception and planning in the continuous space: A POMDP approach. *The International Journal of Robotics Research*, 33(9):1288–1302, 2014.
- [Her86] M. Herman. Fast, three-dimensional, collision-free motion planning. En *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, páginas 1056–1063. IEEE, 1986.
- [HKL08] D. Hsu H. Kurniawati y W.S. Lee. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. En *Robotics: Science and systems*, volume 2008. Zurich, Switzerland., 2008.
- [HL97] D. L. Hall y J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [HM04] D. L. Hall y S. A. H. McMullen. *Mathematical techniques in multisensor data fusion*. Artech House, 2004.
- [HWA02] J. Kearney H. Wang y K. Atkinson. Robust and efficient computation of the closest point on a spline curve. *Curve and Surface Design*, página 397–406, 10 2002.
- [Int08] SAE International. Vehicle Dynamics Terminology, 2008. url: <http://standards.sae.org/j670.200801/>.
- [Int17] SAE International. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2017. url: <http://standards.sae.org/j3016.201401/>.

- [JCS09] D. Ferguson J. Carsten, A. Rankin y A. Stentz. Global planning on the Mars Exploration Rovers: Software integration and surface testing. *Journal of Field Robotics*, 26(4):337–357, 2009. url: <http://dx.doi.org/10.1002/rob.20287>.
- [JGB14] S.S. Srinivasa J.D. Gammell y T.D. Barfoot. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. En *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, páginas 2997–3004. IEEE, 2014.
- [JHU06] R. Motwani J.E. Hopcroft y J.D. Ullman. Automata theory, languages, and computation. *International Edition*, 24, 2006.
- [JMGC12] J.L. Martinez J. Morales, A. Mandow y A.J. Garcia-Cerezo. Driver assistance system for backward maneuvers in passive multi-trailer vehicles. En *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, páginas 4853–4858. IEEE, 2012.
- [JMV11] M. D. Adams J. Mullane, B.-N. Vo y B.-T. Vo. A random-finite-set approach to Bayesian SLAM. *IEEE Transactions on Robotics*, 27(2):268–282, 2011.
- [JMvH13] E. v.d.Houwen J.W. Marck, A. Mohamoud y R. van Heijster. Indoor radar SLAM A radar application for vision and GPS denied environments. En *Radar Conference (EuRAD), 2013 European*, páginas 471–474. IEEE, 2013.
- [JRSWC00] P.I. Corke J.M. Robertsthan, E.S. Duff, P. Sikka, G.J. Winstanley, y J. Cunningham. Autonomous control of underground mining vehicles using reactive navigation. En *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 4, páginas 3790–3795. IEEE, 2000.
- [JSD07] M. Zollner J. Schroder, M. Hoffmann y R. Dillmann. Behavior decision and path planning for cognitive vehicles using behavior networks. En *Intelligent Vehicles Symposium, 2007 IEEE*, páginas 710–715. IEEE, 2007.
- [JT80] C.L. Jackins y S.L. Tanimoto. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3):249–270, 1980.
- [JWL11] J.M. Snider J. Wei, J.M. Dolan y B. Litkouhi. A point-based mdp for robust single-lane autonomous driving behavior under uncertainties. En *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, páginas 2586–2592. IEEE, 2011.

- [JZea14] P. Bender J. Ziegler y et al. Making Bertha Drive—An autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, 2014.
- [Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [KCCBS86] C. Y. Chong K. C. Chang y Y. Bar-Shalom. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Transactions on Automatic Control*, 31(10):889–897, 1986.
- [KCS12] M. Lee K. Chu y M. Sunwoo. Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles. *Trans. Intell. Transport. Sys.*, 13(4):1599–1616, Diciembre 2012. url: <http://dx.doi.org/10.1109/TITS.2012.2198214>.
- [KF10] S. Karaman y E. Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *Robotics Science and Systems VI*, 104, 2010.
- [KF11] S. Karaman y E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [KL16] J. Kim y S. Lee. A vehicular positioning with GPS/IMU using adaptive control of filter noise covariance. *ICT Express*, 2(1):41–46, 2016.
- [Kle04] L. A. Klein. *Data and Sensor Fusion*, página 51. SPIE Press, 2004.
- [KNH15] J. Rajamäki K. Naderi y P. Hämmäläinen. RT-RRT*: a real-time path planning algorithm based on RRT. En *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, páginas 113–118. ACM, 2015.
- [Lat12] J-C. Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.
- [LaV98] S.M. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.
- [LEKO96] J-C. Latombe L. E Kavraki, P. Svestka y M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [LJ00] S.M. LaValle y J.J. Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. 2000.

- [LJ09] R. Li y L. Jia. On the layout of fixed urban traffic detectors: an application study. *IEEE Intelligent Transportation Systems Magazine*, 1(2):6–12, 2009.
- [LJS08] J. Cortés L. Jaillet y T. Siméon. Transition-based RRT for path planning in continuous cost spaces. En *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, páginas 2145–2150. IEEE, 2008.
- [LK16] T. Lee y Y.J. Kim. Massively parallel motion planning algorithms under uncertainty using POMDP. *The International Journal of Robotics Research*, 35(8):928–942, 2016.
- [LPA16] S. Koenig L. Palmieri y K.O. Arras. RRT-based nonholonomic motion planning using any-angle path biasing. En *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, páginas 2775–2781. IEEE, 2016.
- [LR14] Y. Li y Y. Ruichek. Occupancy Grid Mapping in Urban Environments from a Moving On-Board Stereo-Vision System. *Sensors*, 14:10454–10478, 2014.
- [LW10] L. Li y F.Y. Wang. *Advanced motion control and sensing for intelligent vehicles*. Springer, 2010.
- [MB13] M. Magnabosco y T. P. Breckon. Cross-spectral visual simultaneous localization and mapping (SLAM) with sensor handover. *Robotics and Autonomous Systems*, 61(2):195–208, 2013. url: <https://doi.org/10.1016/j.robot.2012.09.023>.
- [MBD00] E. Haritaoglu M. Betke y L.S. Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, 12(2):69–83, 2000.
- [MBS13] B. Brüggemann M. Brunner y D. Schulz. Hierarchical rough terrain motion planning using an optimal sampling-based method. En *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, páginas 5539–5544. IEEE, 2013.
- [ME85] H. Moravec y A. Elfes. High Resolution Maps from Wide Angle Sonar. En *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, páginas 116 – 121, March 1985.
- [Mea08a] I. Miller y et al. Team Cornell’s Skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, 25(8):493–527, 2008.
- [Mea08b] M. Montemerlo y et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

- [MLL12] D. L. Hall M. Liggins y J. Llinas. *Multisensor Data Fusion, Second Edition: Theory and Practice*. CRC, 2012.
- [MMD10] M. Mahlisch M. Munz y K. Dietmayer. Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems. *IEEE Intelligent Transportation Systems Magazine*, 2(1):6–17, 2010.
- [MML⁺97] MM Mitler, JC Miller, JJ Lipsitz, JK Walsh, y CD Wylie. The sleep of long-haul truck drivers. *The New England journal of medicine*, 337(11):755—761, September 1997. url: <http://europepmc.org/articles/PMC2430925>.
- [Mon03] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.
- [MWS05] T. Rentschler M. Weilkes, L. Burkle y M. Scherl. Future vehicle guidance assistance combined longitudinal and lateral control. *Walter de Gruyter GmbH*, 53(1):4–10, 2005.
- [MWT10] S. Kammel M. Werling, J. Ziegler y S. Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. En *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, páginas 987–993. IEEE, 2010.
- [NA] J. Nilsson y S. Abraham. Trailer Parking Assist (TPA).
- [OMC03] S. Hanks O. Madani y A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003.
- [Pet81] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.
- [PT87] C.H. Papadimitriou y J.N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
- [PU08] C. Pradalier y K. Usher. Robust trajectory tracking for a reversing tractor trailer. *Journal of Field Robotics*, 25(6-7):378–399, 2008. url: <http://dx.doi.org/10.1002/rob.20241>.
- [RB96] Robert F Randolph y Christine MK Boldt. Safety analysis of surface haulage accidents. Technical report, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA (United States). Dept. of Mining and Minerals Engineering, 1996.

- [RC03] P. Ridley y P. Corke. Load haul dump vehicle kinematics and control. *Journal of dynamic systems, measurement, and control*, 125(1):54–59, 2003.
- [RCLS02] C.-C. Yih R. C. Luo y K. L. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, 2002.
- [Rea08] F. W Rauskolb y et al. Caroline: An autonomously driving vehicle for urban environments. *Journal of Field Robotics*, 25(9):674–724, 2008.
- [RK04] A. Ranganathan y S. Koenig. PDRRTs: Integrating graph-based and cell-based planning. En *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, páginas 2799–2806. IEEE, 2004.
- [RKB09] D. Dolgov R. Kummerle, D. Hahnel y W. Burgard. Autonomous driving in a multi-level parking structure. *2009 IEEE International Conference on Robotics and Automation*, páginas 3395–3400, 2009.
- [RLF16] O. Garcia R. Lemos y J.V. Ferreira. Local and Global Path Generation for Autonomous Vehicles Using Splines. *Ingeniería*, 21:188 – 200, 12 2016. url: <http://www.scielo.org.co/scielo.php?script=sci.arttext&pid=S0121-750X2016000200006&nrm=iso>.
- [Ros97] J. K Rosenblatt. DAMN: A distributed architecture for mobile navigation. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):339–360, 1997.
- [RSC90] M. Self R. Smith y P. Cheeseman. Autonomous Robot Vehicles. En *Autonomous Robot Vehicles*, capítulo Estimating Uncertain Spatial Relationships in Robotics, páginas 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. url: <http://dl.acm.org/citation.cfm?id=93002.93291>.
- [RSS11] I. Nourbakhsh R. Siegwart y D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. The MIT Press, edición 2nd, 2011.
- [Sam80] H. Samet. Region Representation: Quadrees from Boundary Codes. *Commun. ACM*, 23(3):163–170, Marzo 1980. url: <http://doi.acm.org/10.1145/358826.358836>.
- [SBD11] T. Gindele S. Brechtel y R. Dillmann. Probabilistic MDP-behavior planning for cars. En *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, páginas 1537–1542. IEEE, 2011.

- [SBD14] T. Gindele S. Brechtel y R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. En *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, páginas 392–399. IEEE, 2014.
- [SCH10] J. Davidson S. Candido y S. Hutchinson. Exploiting domain knowledge in planning for uncertain robot systems modeled as POMDPs. En *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, páginas 3596–3603. IEEE, 2010.
- [SCS13] S. Scherer S. Choudhury y S. Singh. RRT*-AR: Sampling-based alternate routes planning with applications to autonomous emergency landing of a helicopter. En *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, páginas 3947–3952. IEEE, 2013.
- [Set96] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [SGMA07] M. Muñoz S. Garrido, D. Blanco, L. Moreno, y M. Abderrahim. *Sensor-based Global Planning for Mobile Manipulators Navigation Using Voronoi Diagram and Fast Marching*. 02 2007.
- [SM09] J. Stergiopoulos y S. Manesis. Anti-jackknife state feedback control law for nonholonomic vehicles with trailer sliding mechanism. *International Journal of Systems, Control and Communications*, 1(3):297–311, 2009.
- [SSDW99] E.M. Nebot S. Scheduling, G. Dissanayake y H. Durrant-Whyte. An experiment in autonomous navigation of an underground mining vehicle. *IEEE Transactions on robotics and Automation*, 15(1):85–95, 1999.
- [SV10] D. Silver y J. Veness. Monte-Carlo planning in large POMDPs. En *Advances in neural information processing systems*, páginas 2164–2172, 2010.
- [TD09] D.Q. Tran y M. Diehl. An application of sequential convex programming to time optimal trajectory planning for a car motion. En *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, páginas 4366–4371. IEEE, 2009.
- [TEFS80] Y. Bar-Shalom T. E. Fortmann y M. Scheffe. Multi-target tracking using joint probabilistic data association. *Proceedings of the IEEE Conference on Decision and Control*, 19:807–812, 1980.

- [Thr00] S. Thrun. Monte carlo pomdps. En *Advances in neural information processing systems*, páginas 1064–1070, 2000.
- [THS08] R. German T. Herpel, C. Lauer y J. Salzberger. Multi-sensor data fusion in automotive applications. *2008 3rd International Conference on Sensing Technology*, 2008.
- [Uea08] C. Urmson y et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- [UM13] S. Ulbrich y M. Maurer. Probabilistic online POMDP decision making for lane changes in fully automated driving. En *Intelligent Transportation Systems- (ITSC), 2013 16th International IEEE Conference on*, páginas 2063–2067. IEEE, 2013.
- [WDK07] C. Rizos W. Ding, J. Wang y D. Kinlyside. Improving Adaptive Kalman Estimation in GPS/INS Integration. *Journal of Navigation*, 60(3):517–529, 2007.
- [WMGA00] Andrew W. Moore y Christopher G. Atkeson. The Parti-game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State-spaces. 21, 05 2000.
- [WXZJZ12] J.M. Dolan W. Xu, J. Wei, H. Zhao, y J-Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. En *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, páginas 2061–2067. IEEE, 2012.
- [XS02] N. Xiong. y P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186, 2002.
- [Yav05] Y. Yavin. Modelling the motion of an underground mining vehicle. *Mathematical and computer modelling*, 42(9-10):1123–1130, 2005.
- [YH95] K.P. Yoon y C-L. Hwang. *Multiple attribute decision making: an introduction*, volume 104. Sage publications, 1995.
- [YKKFH09] G. Fiore Y. Kuwata, J. Teo, S. Karaman, E. Frazzoli, y J.P. Howhan. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.
- [Zea09] H. Zhao y et al. Sensing an intersection using a network of laser scanners and video cameras. *IEEE Intelligent Transportation Systems Magazine*, 1(2):31–37, 2009.

[Zea14] J. Ziegler y et al. Making Bertha drive—An autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, 2014.

Este documento fue editado y tipografiado con \LaTeX empleando la clase **esi-tfm** (versión 0.20171204) que se puede encontrar en:
https://bitbucket.org/arco_group/esi-tfg

[respeta esta atribución al autor]

