



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

Autor:

D. Iván Viloría Vázquez

Tutor:

Dr. Juan Carlos Aguado Manzano

Valladolid, Septiembre de 2022

TÍTULO: Integración de sistemas de posicionamiento *indoor* para un testbed de Edge Computing para el soporte de vehículos conectados

AUTOR: D. Iván Viloría Vázquez

TUTOR: Dr. Juan Carlos Aguado Manzano

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática

TRIBUNAL

PRESIDENTE: Dr. Ignacio de Miguel Jiménez

VOCAL: Dr. Ramón de la Rosa Steinz

SECRETARIO Dr. Ramón J. Durán Barroso

FECHA: Septiembre de 2022

CALIFICACIÓN:

Resumen

Con la realización de este Trabajo de Fin de Máster, se quiere proporcionar algunas de las herramientas necesarias para poder llegar a realizar el *platooning* en la maqueta utilizada en este trabajo. Teniendo este objetivo en mente, se ha desarrollado un odómetro para obtener tanto la velocidad como la distancia recorrida por los vehículos utilizados en la maqueta, buscando con esto, obtener un control más preciso en el movimiento de estos al manejarse de forma autónoma. Además, se han integrado dos sistemas de posicionamiento *indoor* para obtener la posición de los vehículos dentro de la maqueta, en primer lugar, se integró un sistema de posicionamiento compuesto por módulos UWB junto con el que se desarrolló un *software* para calcular la posición de los vehículos de la maqueta, en segundo lugar, se ha integrado un sistema de posicionamiento compuesto por módulos de ultrasonidos. Integrados todos los sistemas, se han realizado las pruebas pertinentes de cada uno de ellos evaluando la precisión de cada sistema.

Abstract

With the realization of this Final Master Project, the aim is to provide some tools to reach the objective of performing platooning in the testbed used in this work. With his objective in mind, an odometer has been developed to measure the speed and the distance that the vehicle used in the testbed has traveled through, searching a more accurate control of the vehicle which is driven autonomously. Moreover, two indoor positioning systems have been integrated into the testbed, on the first place, an indoor positioning system based on UWB modules has been integrated combined with a self-developed *software* for measuring the position of the vehicles in the testbed, on the second place, an indoor positioning system based on ultrasound modules has been integrated too. With all the systems integrated, several tests have been realized to evaluate the accurate of each system individually.

Keywords

Platooning, Indoor Positioning System, CPM

Agradecimientos

En primer lugar, agradecer a Juan Carlos la infinita paciencia que ha tenido conmigo y toda la ayuda y apoyo que me ha brindado, durante la realización de este Trabajo de Fin de Máster y durante los cursos anteriores a su realización.

También quería hacer mención a todos los compañeros que han pasado por el laboratorio 28: Nacho, Carlota, Raúl, Luis, Iñigo, Adrián, Lidia y Alvar. Sin vosotros todos esos “breves” descansos y nuestras “profundas” conversaciones en el laboratorio y fuera de él, no habrían sido lo mismo y todo este proceso habría sido mucho menos divertido y no tendría tan buen recuerdo.

No podría finalizar este trabajo, sin mencionar a personas sin las cuales los viernes después de salir de la universidad no hubieran sido lo mismo, Ana, Víctor y el pequeño Lucas, muchas gracias por todo el apoyo que me habéis brindado durante todo este tiempo.

Muchas gracias a Dani y Frey por haberme animado a seguir adelante en momentos no tan buenos, donde la motivación era escasa y razones para dejarlo no faltaban y por último no podría finalizar los agradecimientos sin agradecer a mis padres por acompañarme durante todo este proceso desde el principio hasta el final que es este trabajo.

Este trabajo ha sido financiado por la Consejería de Educación de la Junta de Castilla y León y el Fondo Europeo de Desarrollo Regional (proyecto VA231P20), y el Ministerio de Ciencia e Innovación y la Agencia Estatal de Investigación (proyecto PID2020-112675RB-C42 financiado por MCIN/AEI/10.13039/501100011033)

Este trabajo ha sido realizado gracias a Vodafone, que autorizó de manera gratuita a la UVa a utilizar algunas de las frecuencias que tiene licenciadas para servicios de telefonía móvil para uso propio en actividades de investigación y docencia.

Índice

Índice iii

Índice de figuras.....	v
Capítulo 1. Introducción.....	1
1.1. Motivación.....	1
1.2. Objetivos.....	2
1.3. Etapas y métodos.....	2
1.4. Recursos.....	3
Capítulo 2. Estado del Arte.....	4
2.1. Sistemas de posicionamiento <i>indoor</i>	4
2.2. Proyecto ETSI ITS-G5.....	5
2.3. Platooning.....	7
2.3.1. Objetivos del <i>platooning</i>	7
2.3.2. Proyectos <i>platooning</i>	9
2.3.3. Casos de uso a resolver por una formación del <i>platooning</i>	18
2.4. Collective Perception Messages.....	24
Capítulo 3. Odómetro en el vehículo AWS Deepracer.....	33
3.1. Introducción.....	33
3.2. Montaje y desarrollo odómetro.....	33
3.2.1. Selección de materiales y diseño del circuito.....	33
3.2.2. Pruebas funcionales del prototipo.....	34
3.3. Resultados odómetro.....	39
Capítulo 4. Posicionamiento <i>Indoor</i>: Módulos UWB M5Stack.....	44
4.1. Introducción.....	44
4.2. Análisis de los módulos UWB.....	44
4.3. Configuración de los módulos UWB.....	49
4.4. Despliegue de los módulos UWB.....	51
4.5. Resultados módulos UWB.....	53
Capítulo 5. Posicionamiento <i>Indoor</i>: Módulos Marvelmind Indoor Positioning System	55
5.1. Introducción.....	55
5.2. Análisis Módulos Marvelmind.....	55
5.3. Configuración de las balizas de ultrasonidos Marvelmind.....	58
5.4. Despliegue de las balizas de ultrasonidos Marvelmind.....	59

5.5. Resultados balizas de ultrasonidos Marvelmind	60
Capítulo 6. Conclusiones y Líneas Futuras.....	63
Referencias.....	64

Índice de figuras

Figura 1. Pila del protocolo ETSI-ITS G5 [21]	6
Figura 2. Niveles de los objetivos de platooning [3]	8
Figura 3. Ilustración de un <i>Platoon</i> [27].....	9
Figura 4. Componentes principales (<i>hardware</i> y <i>software</i>) para el sistema de comunicación V2V del proyecto SARTRE [27]	10
Figura 5. Configuración de un camión automatizado para Energy ITS [30].....	10
Figura 6. Jerarquía de planificación y control de COMPANION [31].....	12
Figura 7. Sistema implementado en vehículo experimental de un vehículo de KONVOI [29].....	14
Figura 8. Capas del sistema ENSEMBLE [33]	16
Figura 9. Módulos para un vehículo en ENSEMBLE [34]	17
Figura 10. Caso de uso Formación del <i>platoon</i> de forma planeada [33].....	18
Figura 11. Caso de uso Unirse al <i>platoon</i> desde la parte trasera [33].....	19
Figura 12. Caso de uso Unirse al <i>platoon</i> entre los vehículos que lo conforman [33].....	19
Figura 13. Caso de uso <i>Platooning</i> en estado de espera [33]	20
Figura 14. Caso de uso coordinación para una frenada [33]	20
Figura 15. Caso de uso de una frenada de emergencia [33]	21
Figura 16. Caso de uso intrusión de un vehículo en el <i>platoon</i> [33]	22
Figura 17. Caso de uso desacoplo del <i>platoon</i> por la parte trasera [33].....	22
Figura 18. Caso de uso desacoplo del <i>platoon</i> por la parte delantera [33].....	23
Figura 19. Caso de uso desacoplo del <i>platoon</i> por el centro [33].....	23
Figura 20. Estructura de un CPM [35].....	25
Figura 21 Información dentro <i>Sensor Information Container</i> proporcionada por un vehículo [35]	26
Figura 22. Información proporcionada <i>dentro de Sensor Information Container</i> por una RSU [35]	27
Figura 23 Información de un objeto detectado por un vehículo [35]	27
Figura 24. Información incluida en el contenedor de espacio libre [35]	28
Figura 25. Modelos de sensores implementados en Artery [35]	28
Figura 26. Mapa de Luxemburgo sobre el que se realiza una simulación en SUMO [35].....	29
Figura 27. Escenario de simulación "araña". Vista general (izquierda) y zoom detallado (derecha) [35]	29
Figura 28. PDR en función de la distancia para los CPM	31
Figura 29. Gráfica funcionamiento sensor <i>hall</i> A1120EUA-T	34
Figura 30. Circuito recomendado para el sensor A1120EUA-T	34
Figura 31. Circuito controlador motor de pasos PM55L.....	35
Figura 32. Circuito control sensor A1120	36
Figura 33. Rueda con imanes en circuito de pruebas	38
Figura 34. Montaje odómetro en el coche AWS Deepracer	38
Figura 35. Rotonda utilizada para las pruebas.....	39
Figura 36. Traslación del punto de giro al centro del coche en ambos sentidos del giro	40

Figura 37. Esquema módulo Ai-ThinkerBU01	44
Figura 38. Conexión entre el módulo UWB y el adaptador UART-USB	45
Figura 39. Elementos de una trama en el módulo DW1000.....	46
Figura 40. Secuencia de transmisión del módulo DW1000	47
Figura 41. Secuencia de recepción del módulo DW1000.....	48
Figura 42. Montaje módulo UWB (tag) en el coche AWS Deepracer	51
Figura 43. Montaje módulos UWB (<i>anchors</i>).....	51
Figura 44. Método de <i>multilateration</i> utilizado para el cálculo de las posiciones con los módulos UWB.....	52
Figura 45. Imagen de las pruebas con los módulos UWB.....	53
Figura 46. Resultado de la vuelta con las balizas UWB.....	54
Figura 47. Componentes del sistema Marvelmind Indoor Positioning System..	56
Figura 48. Esquema de altavoces de una baliza Super Beacon en el sistema Marvelmind.....	56
Figura 49. Montaje baliza Marvelmind en el coche AWS Deepracer.....	59
Figura 50. Imagen de las pruebas para las balizas Marvelmind.	60
Figura 51. Resultados obtenidos al realizar el recorrido con las balizas Marvelmind.....	61

Capítulo 1. Introducción

1.1. Motivación

El mercado del sector de la automoción actual demanda innovaciones electrónicas y de *software*. Debido a esta demanda, los vehículos actuales contienen redes complejas de unidades electrónicas capaces de analizar el medio que los rodea, tener una conectividad en un área amplia y actuar en consecuencia a la información recibida. El nivel de complejidad que han adquirido los vehículos actuales es muy alto, llegando a incorporar hasta 70 unidades electrónicas de control (ECUs) o más, capturando una gran cantidad de señales a la vez [1].

Toda esta información obtenida por el vehículo no es únicamente útil para el vehículo en sí, también puede ser utilizados para construir sistemas de transporte inteligentes con el uso de estos datos, transmitiéndolos a alta velocidad y a largas distancias tanto entre vehículos (V2V) como a infraestructura (V2I) de las carreteras por las que circulan estos vehículos [1], esto es, se puede utilizar esta información para crear un tipo de cooperación entre los diferentes vehículos. La iniciativa de desarrollar estos sistemas de comunicación llevó a su estandarización, dando lugar a diferentes alternativas, como ARIB en Japón, WAVE en USA e ITS-G5 en Europa [2].

El desarrollo de estos estándares ha llevado al desarrollo de diferentes casos de uso, todos ellos con unos objetivos similares, como el aumento de la seguridad en las carreteras, un aumento de la eficiencia en el consumo de combustible o un uso más eficiente de las carreteras evitando atascos y obstrucciones al planificar las rutas y viajes [3]. Uno de los casos de uso de estos estándares es el *platooning*, el cual pretende implementar estos sistemas de transporte inteligente pero centrándolo sobre todo en el uso de camiones, para aumentar su eficiencia en el consumo de combustible, ya que estos vehículos tienen unos consumos muy elevados debido a su gran tonelaje y la alta resistencia al aire que ofrecen, y sobre todo la seguridad durante sus tránsitos por las carreteras porque, por norma general, son vehículos que circulan a velocidades más reducidas y con unas dimensiones mucho mayores que el resto de vehículos, lo que hace que sean más propensos a producir accidentes o verse involucrados en los mismos.

Una funcionalidad que se añade al estándar ITS-G5, que pretende mejorar la percepción del entorno de los vehículos conectados al sistema, son los mensajes de percepción colectiva (CPM) [4], gracias a los cuales los vehículos conectados pueden conocer la localización de potenciales peligros en la vía o en las proximidades de esta. Esta funcionalidad es muy útil para todos los vehículos, pero es muy interesante que se aplique a los camiones, ya que al ser vehículos más pesados si tuvieran una anticipación mayor ante los peligros se reducirían los accidentes en los que se ven involucrados, por lo tanto, sería muy útil integrar estos mensajes dentro de los sistemas que realizan *platooning* y utilizan los estándares para sistemas de transporte inteligentes.

Considerando estos dos elementos, tanto el *platooning* como los CPM, se buscaron los elementos necesarios para llevarlos a cabo dentro de la maqueta desarrollada en el Trabajo

de Fin de Máster de Ignacio Royuela González, y prepararla para realizar en el futuro demostraciones y test sobre estos sistemas utilizando redes 4G y 5G.

La maqueta desarrollada consiste en una pista lo suficientemente grande como para simular situaciones reales de tráfico, además incluye no solo zonas de rectas sino intersecciones, cruces y zonas de cambio de carril para poder simular un mayor número de situaciones que se puedan dar en una carretera real.

Además de la pista, la maqueta incorpora una serie de modificaciones al vehículo AWS Deepracer a nivel de software y hardware para poder realizar un guiado del mismo a través de la cámara y gestionar el control a través de un ordenador conectado a la red 5G desplegada.

Los elementos necesarios que se decidieron desarrollar en este Trabajo de Fin de Máster fueron: un método para medir la velocidad del vehículo y un método para poder determinar la posición del vehículo dentro de la maqueta.

1.2. Objetivos

Los principales objetivos de este Trabajo de Fin de Máster son, estudiar el funcionamiento de los sistemas de cooperativos centrándonos especialmente en el de *platooning* por ser el más apropiado para nuestra maqueta y ser el más desarrollado en la literatura. Por otro lado, el segundo objetivo es preparar la maqueta para poder soportar este tipo de sistemas, integrando diversos sistemas de *ego-perception* en el vehículo que incluían sistemas de posicionamiento *indoor*.

Dentro de estos objetivos generales, los objetivos específicos de este trabajo son los siguientes:

- Desarrollo de un sistema de odometría para el vehículo AWS Deepracer, incluyendo *software* y *hardware*.
- Estudio de diferentes sistemas de posicionamiento *indoor*.
- Integración de al menos un sistema de posicionamiento *indoor* en el vehículo y tests de funcionamiento.

1.3. Etapas y métodos

Para conseguir los objetivos mencionado se llevaron a cabo las siguientes fases:

- i. Estudio de los sistemas cooperativos y de los casos de uso de *platooning*
- ii. Estudio de los diferentes sistemas de posicionamiento *indoor* disponibles actualmente en el mercado.
- iii. Diseño e implementación de un odómetro basado en sensores de tipo *hall*.
- iv. Montaje en el vehículo del sistema del odómetro y obtención de resultados respecto a la precisión del mismo.
- v. Montaje e integración de los sistemas de posicionamiento *indoor* elegidos durante la fase de estudio.
- vi. Realización de test y pruebas de posicionamiento del vehículo.

1.4. Recursos

Los elementos utilizados para llevar a cabo la realización de este proyecto fueron los siguientes:

- 1 vehículo Amazon Deepracer con el pack EVO [5].
- 1 arduino Nano [6].
- 1 motor de pasos NMB PM55L [7].
- 1 sensor *hall* A1120EUA-T [8].
- 1 controlador de motor de pasos TB6600 [9]
- 1 Set Super-MP-3D de balizas de ultrasonidos [10].
- *Software dashboard* de la empresa Marvelmind [11].
- 5 balizas UWB M5-Stack [12].
- 1 adaptador UART-USB [13].
- Maqueta desarrollada en el Trabajo de Fin de Máster de Ignacio Royuela González.

Capítulo 2. Estado del Arte

En el estado del arte de este Trabajo de Fin de Máster, se revisará en primer lugar los diferentes tipos de sistemas de posicionamiento *indoor*, continuando con una revisión del proyecto ETSI ITS-G5 como el referente europeo para los sistemas de transporte inteligente, ya que es uno de los elementos más importantes para llegar a realizar cualquier tipo de conducción autónoma. A continuación, se realizará una revisión de uno de los casos de uso de la conducción autónoma, el *platooning*, tanto de los proyectos, objetivos y casos de usos a resolver por una formación de *platooning*. Por último, se realizará una revisión de la propuesta de la ETSI sobre la introducción de unos mensajes de percepción colectiva en el proyecto ETSI ITS-G5. Todos estos elementos que se analizarán en el estado del arte se incluyen porque se consideran el caso de uso final a alcanzar con los desarrollos llevados a cabo en este.

2.1. Sistemas de posicionamiento *indoor*

Los sistemas de posicionamiento *outdoor*, como GPS (*Global Positioning System*), son ampliamente usado en el día a día, sin embargo, si estos sistemas se utilizan para intentar localizar vehículos u objetos en interiores, se pierde mucha precisión al tener obstáculos como puertas, techos y paredes se produce una degradación de la señal.

Debido a esta degradación fue necesaria la implementación de los sistemas IPS (*Indoor Positioning System*), en el caso planteado en este Trabajo de Fin de Máster será necesaria la inclusión de estas tecnologías, ya que, todos los sistemas de transporte inteligente (*Intelligent Transportation System*, ITS) requieren GPS o un sistema de localización, por lo tanto, en el caso planteado es necesario emularlo en un interior, por lo que se hará uso de las tecnologías IPS para conseguir este objetivo. Los sistemas IPS utilizan un amplio abanico de tecnologías para realizar esta localización en interiores [14]:

- *Wireless local area network*: La tecnología WLAN se utiliza en el intercambio de datos de forma inalámbrica a unas distancias de entre 20 y 100 metros como máximo desde el punto de acceso a la red, las cuales pueden funcionar en frecuencias de 2,4 y 5 GHz. Los sistemas ILS que se basan en la tecnología WLAN utilizan varios puntos de acceso para determinar la posición de un objetivo dentro de un área determinada en la zona que abarca la cobertura de la red. La precisión de estos sistemas suele rondar entre 1 y 3 metros.
- *Radio-frequency identification* (RFID): Esta tecnología utiliza ondas de radio para intercambiar datos entre un lector y una etiqueta (*tag*). Hay dos tipos dentro de la tecnología RFID, los sistemas RFID pasivos donde el *tag* que se quiere localizar es un receptor, el rango de estos sistemas es pequeño, entre 1 o 2 metros, trabajando en varias bandas de frecuencia (125-134 KHz, 13,56 MHz y 865-960 MHz) y con un precio bajo. En segundo lugar, se encuentran los sistemas RFID activos, donde el objeto que se quiere localizar también actúa como un transmisor, ofreciendo unos rangos mayores de cobertura entre 10 y 100 metros, operando habitualmente en las bandas de frecuencia de 433 y 915 MHz. El sistema LANDMARC [15], que fue un sistema pionero que usó la tecnología

RFID para la localización *indoor*, conseguía precisiones de aproximadamente 1 metro en una habitación de 40 m² usando 4 receptores y 16 tags de referencia.

- **Ultrasonidos:** Se utilizan las frecuencias de ultrasonidos, normalmente alrededor de los 40 KHz, para determinar la posición de un objeto utilizando el tiempo de viaje de la señal de un transmisor a un receptor. La razón de utilizar las frecuencias de ultrasonidos es que no es intrusivo con la audición de las personas que se encuentren cerca del sistema, aunque presenta la desventaja de necesitar el despliegue de una infraestructura para su uso en la zona donde se requiera. El sistema de posicionamiento *Active Bat* [16], consigue precisiones de hasta 3cm en un área de 1000 m².
- **Bluetooth:** Es una tecnología que se utiliza para la transmisión de datos en un rango corto, entre 10 y 100 metros con *Bluetooth 2.0*, está definido en [] como una red inalámbrica de área personal (*Wireless Personal Area Network*, WPAN) trabajando en la banda de frecuencia 2,4 GHz, el objetivo es crear un sistema que no requiera línea de visión entre transmisor y receptor. El sistema de posicionamiento denominado Topaz [17] consigue precisiones de hasta 2 metros.
- **Inertial:** Los sensores de inercia, tales como acelerómetros o giroscopios, en combinación con otros sensores se pueden utilizar para determinar la velocidad y dirección de un objeto, teniendo conocimiento del punto de partida del objeto se puede predecir la próxima posición de este. Beaurergard et al [18] propone un sistema que obtiene precisiones de hasta 2,5 metros.
- **Ultra-wideband:** Las ondas electromagnéticas se emiten en pulsos cortos, menos 1 ns de duración, en una banda de frecuencias ancha, mayor a 500 MHz, permitiendo filtrar la señal original de las reflejadas. La precisión del sistema propuesto por la compañía Ubisense [19] alcanza valores de hasta 15 cm.

Los sistemas elegidos para este proyecto una vez realizado un estudio de mercado con las diferentes alternativas, han sido, el sistema de posicionamiento *indoor* basado en la tecnología de ultrasonidos de la empresa Marvelmind, como la opción que ofrece una mayor precisión, una gran cantidad de documentación, pero a la vez un mayor coste y como otra opción, un sistema de posicionamiento *indoor* basado en la tecnología UWB con balizas adquiridas a la empresa M5Stack teniendo una opción más económica, pero con unas prestaciones menores y una documentación menos exhaustiva.

Vistos los diferentes tipos de sistemas de posicionamiento *indoor*, se procederá a revisar el proyecto ETSI ITS-G5.

2.2. Proyecto ETSI ITS-G5

El proyecto más relevante en Europa es el ETSI ITS-G5, desarrollado por la ETSI (*European Telecommunications Standards Institute*) el CEN (*Comité Europeo de Normalisation*) junto con sus respectivos comités técnicos. Aunque también cabe mencionar otros estándares como el IEEE WAVE desarrollado en Norte América y el ARIB T109 desarrollado en Japón [20].

El proyecto ETSI ITS-G5, cuya pila de protocolos vemos en la Figura 1 está dividido de la siguiente forma por las distintas capas [21]:

- Capa física: Hace uso del protocolo IEEE 802.11p.
- Capa de acceso: (*Decentralized Congestion Control*) DCC
- Capa de Transporte: Geonetworking
- Capa de red: (*Basic Transport Protocol*) BTP
- Capa de aplicaciones: (*Cooperative Awareness Message*) CAM y (*Decentralized Environmental Message*) DEMN

El proyecto dispone de 4 bandas de frecuencia dentro de la banda de 5 GHz para la comunicación dentro de los sistemas ITS (Intelligent Transportation System) [22]:

- ITS-G5A: Se utiliza para aplicaciones relacionadas con la seguridad en la carretera.
- ITS-G5B: Para el resto de las aplicaciones que no estén relacionadas con la seguridad.
- ITS-G5C: Se utiliza para *broadband radio access networks (BRAN)*, *radio local area network (RLAN)* y *wireless local area network (WLAN)*.
- ITS-G5D: Futuras aplicaciones para sistemas ITS.

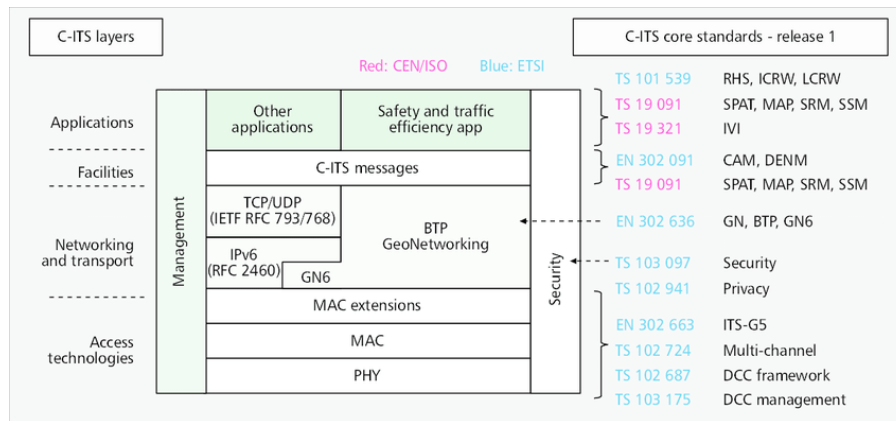


Figura 1. Pila del protocolo ETSI-ITS G5 [21]

El estándar ITS-G5 introduce un nuevo esquema para minimizar la probabilidad de congestión en el canal de radio (DCC). Se utiliza DCC para controlar la carga del canal, modificando dinámicamente las condiciones de acceso al mismo. [2]

Se definen en las capas superiores unos nuevos mensajes denominados CAM y DENM. CAM manda mensajes periódicos con el estado del vehículo a los nodos cercanos de la red, con estos mensajes los nodos de la red son conscientes tanto de la posición, movimientos y características relevantes de las estaciones cercanas.

DENM también manda mensajes periódicos al igual que CAM, solo que en este caso manda el estado de las carreteras o el tráfico de esta, si suponen una amenaza o un peligro para las estaciones vecinas [23].

En la capa de transporte se implementa, como alternativa a TCP, el protocolo *GeoNetworking* [24], se controla el transporte de los paquetes desde un origen a un destino, pero el encaminamiento de estos paquetes se realiza en función de la posición de los nodos de la red en lugar de una dirección IP convencional. Todas las estaciones del sistema contienen una tabla con las estaciones vecinas conocidas, que se utiliza para saber

en función del tipo de mensaje a que estaciones mandarlo y a cuáles no, dentro de estos envíos podemos encontrar diversos tipos:

- *GeoUnicast*: Envío de la información a un único nodo.
- *GeoBroadcast*: Se envía la información a todos los nodos de una zona determinada.
- *GeoAnyCast*: Se envía a cualquier nodo.
- *GeoMulticast*: El envío de la información se realiza únicamente a una serie de nodos determinados de una zona concreta.

El protocolo BTP se encarga de proporcionar una comunicación punto-a-punto en las redes ad-hoc, aquellas que no dependen de una infraestructura preexistente ya sean *routers* o puntos de acceso. En la estación de origen se identifica el protocolo que quiere iniciar la comunicación, y en el destino marca el protocolo que recibirá la información [25].

En la capa física, el estándar 802.11p [26] se basa en una modificación del estándar 802.11 del IEEE expresamente para acceso inalámbrico en entornos con vehículos (*Wireless Access in Vehicular Environment*, WAVE) modificando ligeramente tanto la capa física como la capa MAC para conseguir una conexión robusta en este tipo de entornos con vehículos en movimiento [27].

Revisados los distintos elementos que componen el proyecto ETSI ITS-G5, se pasará a revisar los proyectos de investigación relacionados con el *platooning*, además de una definición de objetivos que busca cumplir y una serie de posibles casos de uso que debería resolver el *platooning*.

2.3. Platooning

Platooning se define como el enlace entre dos o más vehículos dentro de un convoy, usando tecnología de conectividad y sistemas de soporte para la conducción autónoma. Estos vehículos mantienen una distancia predeterminada entre ellos en función del tipo de vía por el que circulen o las circunstancias de esta [28].

2.3.1. Objetivos del *platooning*

Una vez introducidos algunos de los proyectos más importantes y relevantes, pasaremos a evaluar los objetivos que se buscan conseguir aplicando el *platooning* dentro de las carreteras.

Dentro de los objetivos de *platooning* se pueden distinguir diferentes niveles representados en la Figura 2, objetivos a nivel de vehículo (*Vehicle Level*) centrados en optimizar el rendimiento deseado del vehículo. A nivel de *platoon* (*Platoon Level*) buscan optimizar, por ejemplo, la seguridad del conjunto de vehículos o el ahorro de combustible y energía [3].

Por último, a nivel global (*Global Level*) se incorporan objetivos relacionados con la presencia de múltiples *platoons* en circulación y el tráfico de la vía.

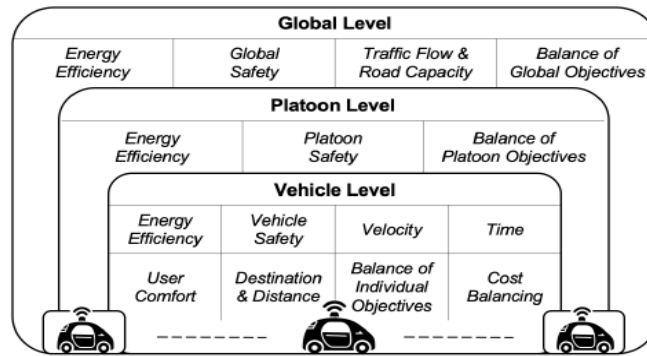


Figura 2. Niveles de los objetivos de platooning [3]

Varios de los objetivos son comunes a todos los niveles como por ejemplo la eficiencia energética o la seguridad tanto a nivel de vehículo como a nivel de *platoon*. A continuación, se comentarán los más relevantes:

- **Eficiencia energética:** Se busca la reducción del consumo de combustible y la emisión de gases de cada uno de los vehículos que forman un *platoon*. En Japón en 2011 el consumo de energía por parte de los camiones fue aproximadamente de un 8% del total del país. Mediante la conducción en *platoons*, con una distancia entre vehículos pequeña, se consigue una reducción grande en cuanto a la resistencia al aire de los vehículos que preceden al LV, cuanto mayor sea la velocidad del *platoon*, mayor será el porcentaje de ahorro de combustible en los vehículos que lo componen [29].
- **Seguridad:** Se pueden distinguir tres niveles de seguridad como vemos en la Figura 2. Manteniendo una distancia mejora la seguridad, en caso de ser necesaria una frenada de emergencia desde el punto de vista de un vehículo individual (*Vehicle Safety*). Si hablamos a nivel del *platoon* (*Platoon Safety*), aumenta la seguridad al mantener de manera coordinada esta distancia entre los vehículos que lo conforman, todos estos factores influyen en el nivel más alto de los tres mencionados (*Global Safety*), ya que al prevenirse los accidentes entre los vehículos del *platoon* y vehículos externos a este, aumentará la seguridad global [29].
- **Flujo de tráfico y capacidad de las carreteras:** Un aumento de la cantidad de tráfico en las carreteras y un mejor flujo de este se puede conseguir gracias al *platooning*, ya que el flujo de tráfico estaría más controlado y sincronizado al viajar una parte de los vehículos, camiones en su mayoría, en *platoons* de forma sincronizada y compacta. Aunque no todo es positivo, ya que el uso de demasiados camiones de una misma carretera, aunque formen *platoons*, podría suponer un colapso de la vía si no es de forma controlada [29].
- **Velocidad:** Cada vehículo individualmente busca llevar la velocidad más adecuada, por ejemplo, la máxima velocidad legal de la vía si fuera posible o una velocidad deseada cualquiera. Implementar *platooning* podría llevar a una velocidad más baja, como resultado de una homogenización entre las velocidades de todos los vehículos del *platoon*, por lo tanto, es importante al planificar la ruta elegir una serie de vehículos para formar el *platoon* que tengan unas características y especificaciones similares [29].

- **Tiempo:** La planificación de las rutas óptimas, en cuanto al tráfico y la elección de la vía, ayuda a que se minimicen los tiempos necesarios para completar la ruta designada. Sin embargo, el *platooning* puede llevar a aumentar los tiempos de viaje, ya que alguno de los vehículos que lo conforman puede necesitar llevar una velocidad menor, por lo tanto, el resto se adaptarán a este, pero esto se vería compensado con el resto de los beneficios, como la seguridad añadida de circular dentro de un *platoon* o la eficiencia en el consumo de combustible del vehículo [29].

De los objetivos descritos, los más relevantes o los que más se busca conseguir son la eficiencia energética, la seguridad, el flujo de tráfico y la capacidad de la carretera y la velocidad. Esto se debe a que son los objetivos que al cumplirlos pueden tener un impacto más visible y relevante en el resto de los usuarios de una carretera, por ejemplo, con un menor número de accidentes en la vía o los propietarios de una empresa de transporte al ver reducidos sus costes en combustible realizando las mismas funciones.

2.3.2. Proyectos *platooning*

En esta sección, se introducirán los diversos proyectos de investigación y desarrollo relacionados con *platooning*:

- **SARTRE:** Es un proyecto co-financiado por la Comisión Europea cuyo objetivo es desarrollar e integrar soluciones para permitir a los vehículos circular en *platoons*, desarrollado entre los años 2007 a 2013. De acuerdo con este proyecto se entiende el *platoon* como un conjunto de vehículos, liderados por uno conducido de forma manual seguido por el resto de forma autónoma tanto longitudinal como lateralmente como vemos en la Figura 3 [27].

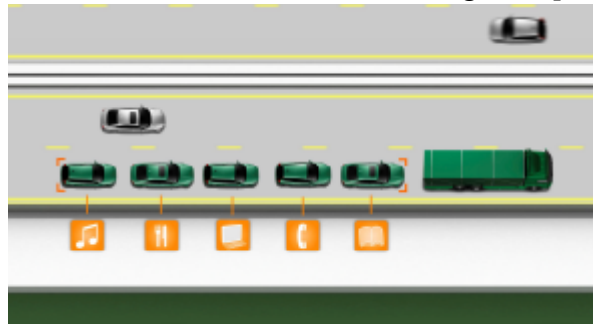


Figura 3. Ilustración de un *Platoon* [27]

Los *platoons* forman un sistema cooperativo donde cada vehículo es un subsistema de este. Dentro del sistema la información de los sensores, algoritmos de control y actuación se distribuye por todo el *platoon* mediante las comunicaciones vehículo a vehículo (V2V). Es muy importante para la seguridad dentro del *platoon* que las comunicaciones V2V sean fiables en cuanto a los retardos del sistema o posibles errores en la transmisión [27].

El bus CAN es un bus de comunicaciones diseñado para permitir las comunicaciones entre los distintos dispositivos y controladores (*Engine Control Unit*, ECU) del vehículo. Un mismo vehículo puede contener varios buses CAN

para las diferentes áreas del vehículo, como por ejemplo el tren motriz o sistemas multimedia [27].

En el proyecto SARTRE se añade un bus CAN dedicado para interconectar un clúster de ordenadores y ECUs. Estos nodos son responsables de funciones como los algoritmos de control del vehículo, la interfaz humano-máquina (HMI), fusión de sensores y las comunicaciones V2V. El sistema de comunicación V2V en este proyecto utiliza IEEE 802.11p [26] para las comunicaciones inalámbricas.

En la Figura 4 vemos un esquema de un sistema de comunicación V2V en el proyecto SARTRE, la principal función de este sistema es encaminar mensajes del bus CAN de un vehículo perteneciente a un *platoon* hasta otro bus CAN de un vehículo diferente perteneciente al *platoon*. La mayoría de las comunicaciones se realizan entre el vehículo que lidera el *platoon* (*Leading Vehicle, LV*) mediante *broadcast* a todos los demás vehículos que le preceden (*Following Vehicles, FVs*).

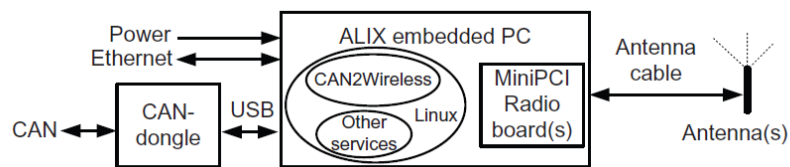


Figura 4. Componentes principales (*hardware* y *software*) para el sistema de comunicación V2V del proyecto SARTRE [27]

- **Energy ITS:** Un programa que se inició en el año 2008 por el Ministerio de Economía, Comercio e Industria de Japón. El proyecto tiene dos objetivos principales: un *platoon* automatizado compuesto por camiones y una evaluación de la efectividad de los sistemas de transporte inteligentes (*Intelligent Transport Systems, ITS*) en cuanto al ahorro de energía con el uso de estos [29].

El *platoon* desarrollado en este proyecto consistía en 3 camiones pesados (25 toneladas) y un camión ligero, llegando a conducir a 80 Km/h con separaciones de 10 metros y 4 metros en una prueba. Las funciones del *platoon* son el mantenimiento del carril, control de la velocidad de los vehículos, evitar colisiones y mantener la distancia entre los vehículos establecida para el *platoon*. En la Figura 5 se muestra el equipamiento necesario para estos camiones que formarán el *platoon*, todo esto sin ser necesarios cambios en la infraestructura de las carreteras. Cada camión está equipado con dos cámaras o LiDARs en cada lateral, una en la parte delantera y otra en la trasera, estas cámaras se utilizan para detectar los límites del carril en ambos lados del camión [29].

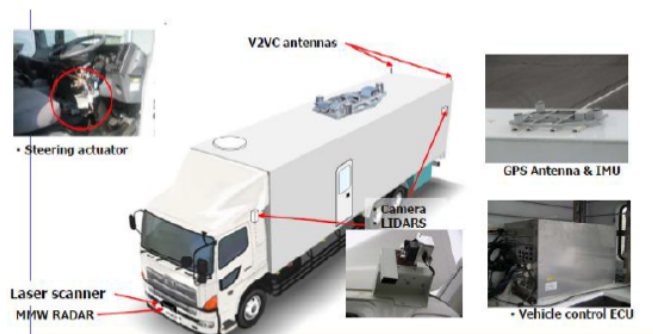


Figura 5. Configuración de un camión automatizado para Energy ITS [30]

Cada camión también dispone de un radar funcionando a 76 GHz y un radar bidimensional en el primer camión del *platoon* para la detección de obstáculos y para el mantenimiento de la distancia entre vehículos establecida en el resto de los camiones que forman el *platoon*. El uso de estos dos dispositivos se debe a la necesidad de robustez en el sistema, además debido a que no tiene la precisión necesaria el *platoon* utiliza las comunicaciones V2V [29].

Estas comunicaciones V2V utilizan dos medios, comunicaciones DSRC (*Dedicated Short-Range Communication*) a 5,8 GHz e infrarrojos también para darle robustez al sistema. Los datos de cada uno de los camiones se comparten con el resto en tiempo real, tales como la velocidad de referencia, la aceleración, la velocidad de cada vehículo, la señal de frenado y datos de gestión necesarios para el funcionamiento del *platoon* [29].

El canal de comunicación V2V a través de DSRC tiene dos canales independientes, además del canal de infrarrojos para añadir más redundancia al sistema. El sistema de giro tiene dos motores independientes al igual que el sistema de freno tiene dos unidades de control, todo esto con el mismo objetivo que lo anteriormente mencionado [29].

- **COMPANION** (*Cooperative dynamic formation of platoons for safe energy-optimized goods transportation*): El objetivo principal de este Proyecto perteneciente a la Comisión Europea, era desarrollar tecnologías de movilidad cooperativa para realizar el *platooning*, con el foco principal en mejorar la eficiencia en el consumo de combustible y la seguridad en el transporte y movilidad de los vehículos. [31]

Este proyecto, desarrollado entre los años 2013 a 2016, para desarrollar estas tecnologías busca cumplir 5 objetivos:

- Desarrollo y validación de una plataforma externa al vehículo para la coordinación de los *platoons*,
- Desarrollo y validación un sistema interno al vehículo para la coordinación de los *platoons*.
- Desarrollo y validación una interfaz para el usuario usable tanto dentro como fuera de los vehículos.
- En cuanto al marco legislativo y de estandarización del *platooning*, se ha analizado el marco legal de la Unión Europea para encontrar los posibles vacíos respecto a este tema. Por lo tanto, se proponen soluciones legales para la correcta incorporación de estas tecnologías.
- El último objetivo es conseguir demostrar el funcionamiento del sistema, en carreteras europeas de varios países y mostrar las capacidades de la plataforma externa al vehículo en una simulación a gran escala.

En cuanto al funcionamiento del sistema propuesto, en primer lugar, se introduce el inicio, destino y tiempo deseado de llegada para el reparto. El sistema externo al vehículo calcula la ruta óptima para el transporte, al mismo tiempo busca otros vehículos que tomen la misma ruta o una similar para formar un *platoon* [31].

Una vez iniciada la ruta, el vehículo se une a los más adecuados para formar el *platoon* y se coloca a la distancia más adecuada de estos y el conductor activará

el modo *platooning* dentro del sistema. El nivel de autonomía que se busca en el proyecto COMPANION es únicamente el control longitudinal del vehículo, por lo tanto, el control lateral es responsabilidad del conductor. Mientras los vehículos se encuentren al alcance con las comunicaciones V2V se intercambiarán datos como la localización actual, velocidad y más datos necesarios, por ejemplo, una señal que se activa en caso de una frenada de emergencia de uno de los vehículos del *platoon*.

En el proyecto COMPANION tal y como vemos en la Figura 6, la planificación y el control se divide en tres capas, la capa superior (*Strategic Layer*) se encarga tanto de la planificación de la ruta como de su optimización, una vez realizada esta tarea, la segunda capa (*Tactical Layer*) se encarga de optimizar lo anterior en función del perfil de velocidad necesario para el vehículo, no seguiría la misma planificación en caso de planificarse la ruta para un camión o un vehículo utilitario. Toda la comunicación entre los vehículos y esta plataforma externa al vehículo se realizará mediante la red móvil.

La última capa (*Operational Layer*), se encarga del control del *platoon*, manteniendo una distancia adecuada entre los vehículos que lo forman, también realizará las acciones necesarias para que nuevos vehículos se incorporen al *platoon* o alguno lo abandone. Para las comunicaciones V2V utiliza una versión del estándar ITS-G5.

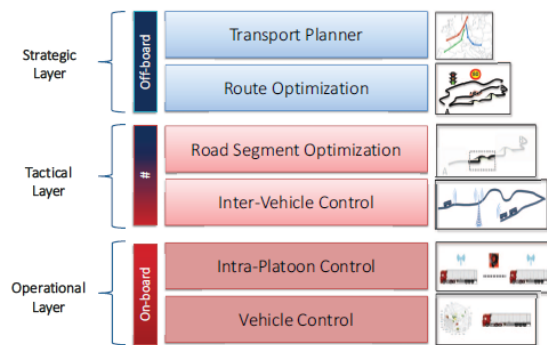


Figura 6. Jerarquía de planificación y control de COMPANION [31]

En cuanto a la arquitectura del sistema utilizado en COMPANION en la Figura vemos los distintos componentes [31]:

- *HMI Web Interface*: Permite al usuario crear y gestionar rutas de transporte y asignarlas, tanto la ruta creada, el estado del vehículo, datos de localización y las ventajas de la ruta se muestran en esta interfaz.
- *Route Calculation Engine (RCE)*: Calcula la ruta para un único vehículo teniendo en cuenta el tráfico de las vías, características del vehículo y propiedades de la carretera. Toda esta información, genera como resultado una ruta que contiene una ventana de velocidades posibles, utilizada posteriormente por el motor de optimización (*Optimization Engine*) para optimizar todo el sistema global de *platooning*.
- *Monitoring and Optimization Engine (MOE)*: Calcula la velocidad óptima y los puntos de unión y separación a un *platoon* óptimos para un vehículo

concreto, dando como resultado una serie de rutas, limitaciones de tiempo y velocidad además de aproximaciones de consumo para las rutas estimadas. Las rutas proporcionadas por la RCE son recibidas como una secuencia de enlaces que dividen la ruta en pequeños fragmentos para seleccionar en qué momento es más adecuado unirse o separarse de un *platoon*.

- *Fleet Manager*: Teniendo una organización en *platoons* dada por el MOE e información del tráfico dada por el RCE, se generan planes concretos de asignación para cada vehículo. Todos estos planes, son compartidos con el usuario a través de la HMI.
- *Vehicle and Fuel Consumption Model Service*: Calcula posibles configuraciones de velocidad para el vehículo en función de las características dinámicas del mismo.
- *Vehicle Gateway*: Gestiona las comunicaciones entre los componentes internos y externos al vehículo conectados al sistema propuesto por COMPANION, almacenando los datos para un vehículo concreto en caso de no ser posible compartirlos en un momento determinado.

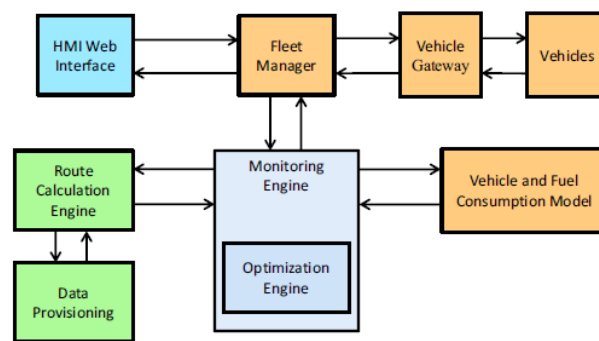


Figura. Esquema de componentes de la arquitectura COMPANION [31]

- **KONVOI**: El objetivo de este proyecto es el desarrollo y evaluación de la circulación de camiones en *platoons*. Este proyecto fue financiado por el Ministerio Federal de Economía y Tecnología de Alemania. Los *platoons*, tal y como se conciben en este proyecto, se componen de 5 vehículos, el primero de ellos conducido de forma manual mientras que el resto se controlan de forma totalmente autónoma. Está pensado para ser utilizado en el sistema de autopistas, donde los camiones se encuentran ocasionalmente viajando en la misma dirección durante un largo periodo de tiempo [29].

Los componentes principales para la implementación de este proyecto son actuadores, en el tren motriz y el sistema de giro, sensores, detección de objetos y detección del carril, el sistema de comunicaciones V2V (WLAN), la unidad de automatización encargada de la coordinación de los diferentes estados de los vehículos, la unidad de control encargada del control de la velocidad a través del control de crucero adaptativo (*Adaptive Cruise Control*, ACC) y el guiado lateral, por último el sistema de información para el conductor (*Driver Information System*, DIS) encargado de la HMI, GPS y comunicaciones móviles. En cuanto al control de crucero adaptativo, se encarga de proporcionar asistencia al conductor en la tarea del control longitudinal del vehículo. El sistema del ACC

se encarga de gestionar el acelerador, el sistema del tren motriz además de los frenos para mantener una distancia determinada con el vehículo que circula delante [32].

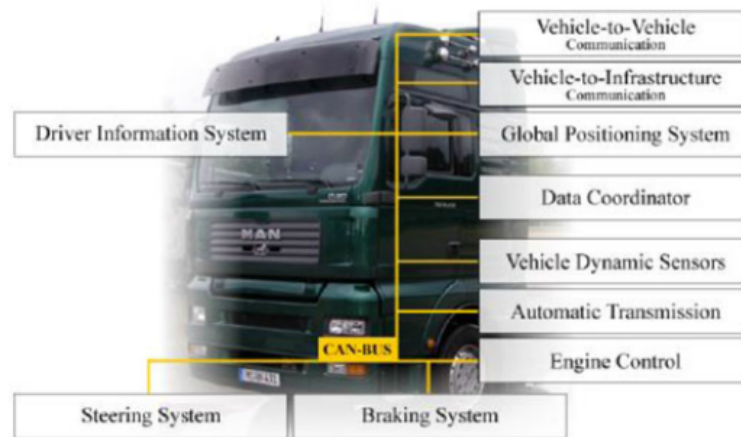


Figura 7. Sistema implementado en vehículo experimental de un vehículo de KONVOI [29]

El guiado del control lateral se calcula en función de la posición del vehículo respecto a las líneas del carril y el *offset* lateral del vehículo que lidera el *platoon*. Todo este guiado lateral se efectúa a través de un motor eléctrico que modifica la dirección del vehículo para mantener la trayectoria deseada [29].

El control longitudinal del vehículo se encarga a un sensor LiDAR, una cámara y un sensor radar. Los sensores determinan la distancia al vehículo de delante y el *offset* lateral con el vehículo que lidera el *platoon*, con esta información el sistema V2V transfiere los datos necesarios del vehículo a todos los miembros del *platoon* que se requieren para el ACC con el objetivo de mantener una distancia de 10 metros de separación entre los vehículos [29].

Con ayuda del DIS el conductor puede planear su ruta, selecciona participantes del *platoon* que sean eficientes y confirmar las maniobras del *platoon*, tanto como para construirlo como para disolverlo en caso de ser necesario [29].

- **ENSEMBLE:** Se trata de un proyecto co-fundado por la Comisión Europea, DG Research e Innovation (Directorate-General for Research and Innovation) dentro del programa HORIZON 2020. El objetivo principal de ENSEMBLE es la adopción del *platooning* en camiones en Europa para mejorar el consumo de combustible, aumentar la seguridad en el tráfico y el flujo de este en las carreteras europeas [33].

Dentro de este objetivo de implementar el *platooning* se definen los siguientes objetivos para llegar a esta implementación [33]:

- *Platooning* interoperable: Cuando se forma un *platoon* escalable y multimarca, los vehículos deben ser compatibles entre sí para asegurar un funcionamiento seguro y adecuado.
- *Platooning* seguro: La seguridad es uno de los objetivos claves para asegurar la correcta aceptación del *platooning*. Dentro de ENSEMBLE esto se quiere alcanzar mediante los siguientes puntos:
 - Diseñando un sistema a prueba de fallos y con mecanismos que le permitan ser tolerante a los mismos en caso de suceder alguno,

incluyendo interacciones con el resto de los miembros del *platoon* como con el resto de los usuarios de la carretera. Todo esto se quiere alcanzar mediante unas comunicaciones inalámbricas seguras y robustas.

- Se planeará con las autoridades pertinentes la definición de los requisitos para el *platooning* en las carreteras abiertas, teniendo en cuenta el impacto de los *platoons* en la carretera y en la infraestructura de esta.
- *Platooning* en la vida real: Las pruebas que se quieren realizar en pistas de pruebas y en las carreteras, servirán para conseguir unas metas:
 - Evaluar el impacto en el tráfico, infraestructura y logística que funcione por la zona de las pruebas.
 - Promover un *platooning* multimarca con el objetivo de probarlo y corroborar su funcionamiento
- *Platooning* integrado: El concepto de *platooning* debería permitir una integración sin fisuras en la cadena de valor de la logística, por lo tanto, este objetivo se completará diseñando una interfaz con el fin de conseguirlo.

El sistema de *platooning* se dividirá en una serie de capas definidas en la Figura 8:

- Capa de servicios (*Services Layer*): Representa la plataforma donde se pueden realizar las operaciones de logística.
- Capa de estrategia (*Strategic Layer*): Es responsable de las decisiones que afectan a la decisión de formar los *platoons* en función de la compatibilidad de los vehículos y el nivel de *platooning* implementado, la optimización en cuanto al consumo de combustible, tiempos de viaje, destino y el impacto en el flujo del tráfico de la carretera que se fuera a utilizar y la infraestructura correspondiente. Para realizar esta tarea son necesarias comunicaciones inalámbricas de largo alcance, mediante tecnologías móviles ya existentes.
- Capa de táctica (*Tactical Layer*): Coordina la formación de los *platoons*, tanto desde la cola del *platoon* como uniéndose en otra parte de este, y la disolución de estos. Además, se asegura de la cohesión del *platoon* en carreteras conflictivas, marcando la velocidad del *platoon* deseada, distancia entre los vehículos que lo conforman y el *offset* lateral para reducir el desgaste de la carretera. Todo esto se lleva a cabo mediante un protocolo de interacción usando comunicaciones intravehiculares de corto alcance.
- Capa operacional (*Operational Layer*): Esta capa se encarga del control del vehículo (por ejemplo, la aceleración, frenado y giro) y la ejecución de estas maniobras para conseguir que las tareas necesarias para el *platoon* se puedan realizar de forma automática. La tarea principal es regular la distancia entre vehículos y la velocidad, dependiendo del nivel del *platooning* la posición lateral relativa al carril o al vehículo que circula delante.

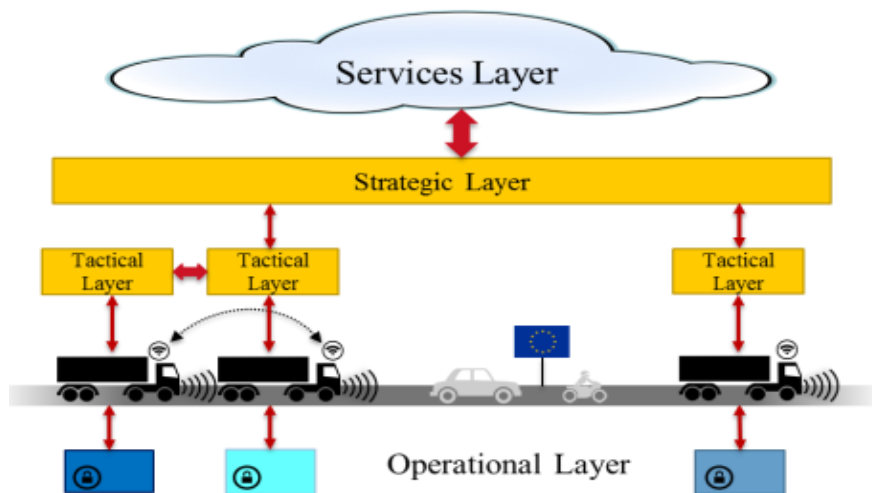


Figura 8. Capas del sistema ENSEMBLE [33]

En cuanto a los niveles del *platooning* mencionados en las capas de ENSEMBLE, en este proyecto se tienen en cuenta tres aspectos para definir los niveles de *platooning*:

- El conductor como respaldo: Significa dar suficiente tiempo al conductor para reaccionar, en caso de ser necesario.
- Estimación del rendimiento de frenado: Para poder calcular la distancia mínima segura para seguir al vehículo de delante, cada miembro del *platoon* necesita conocer su rendimiento de frenado y el del resto de vehículos que lo rodean.
- Localización: Para poder identificar al compañero con el que se comunica, al menos la localización junto con la precisión a la hora de mantener el carril es necesaria para el *platooning*.

Teniendo estos tres retos en cuenta, se diferencian dos niveles en el *platooning* con la responsabilidad del conductor como la clave principal de ambos:

- *Platooning Support Function* (PSF): El conductor se encarga de la conducción, por lo tanto, es responsable de elegir la distancia de seguridad y monitorizar el sistema. Para dar suficiente tiempo al conductor en caso de ser necesario, se deben respetar espacios en tiempo de unos 1,5 segundos, es decir, un espacio equivalente al que recorrería el vehículo en 1,5 segundos a la velocidad que se esté desplazando en ese instante. El PSF se encarga del control longitudinal, pero se pueden incluir asistencias para el control lateral, como por ejemplo el centrado del carril.
- *Platooning Autonomous Function* (PAF): En el LV debe haber un conductor responsable de la conducción, pero los camiones que lo preceden están totalmente automatizados. Teniendo estos vehículos automatizados los espacios en tiempo se podrían reducir hasta los 0,5 segundos, dependiendo del tiempo de respuesta del sistema y la precisión del sistema.

En la Figura 9 vemos los módulos que tienen que incorporar los vehículos del *platoon* para realizar las funciones comentadas en los distintos niveles [34]:

- *V2X communication*: Conjunto de *software* y *hardware* que establece las comunicaciones entre vehículos necesarias para *platooning* utilizando el protocolo ETSI ITS-G5.
- *Platoon status and platoon vehicle property collection and sharing*: Este módulo se encarga de recolectar la información relevante del *platoon* y de los vehículos que lo conforman, pudiendo ser necesaria esta información para otros miembros del *platoon* o para el propio sistema.
- *Platoon manoeuvre coordination*: Coordina maniobras específicas que necesitan una aproximación cooperativa entre varios miembros del *platoon*.
- *Platoon cohesion mechanism*: Contiene las estrategias para mantener la cohesión del *platoon*, por ejemplo, en una carretera de montaña o con mucha pendiente. Aquí interviene la capa de estrategia, proporcionando la información necesaria a la capa de operación para mantener esta cohesión mencionada en el *platoon*.
- HMI: Proporciona la interfaz necesaria para que el conductor pueda realizar las operaciones necesarias durante la conducción.
- *Sensors*: Este módulo contiene todos los sensores necesarios que se monten en el vehículo para el control longitudinal, además de usarse para recibir la información relativa a la percepción de los vehículos y entorno que lo rodean.
- *Longitudinal control*: Ejecuta el algoritmo de control longitudinal para la correcta ejecución de las aceleraciones y frenadas, necesarias para mantener la distancia marcada con el vehículo que circula delante.

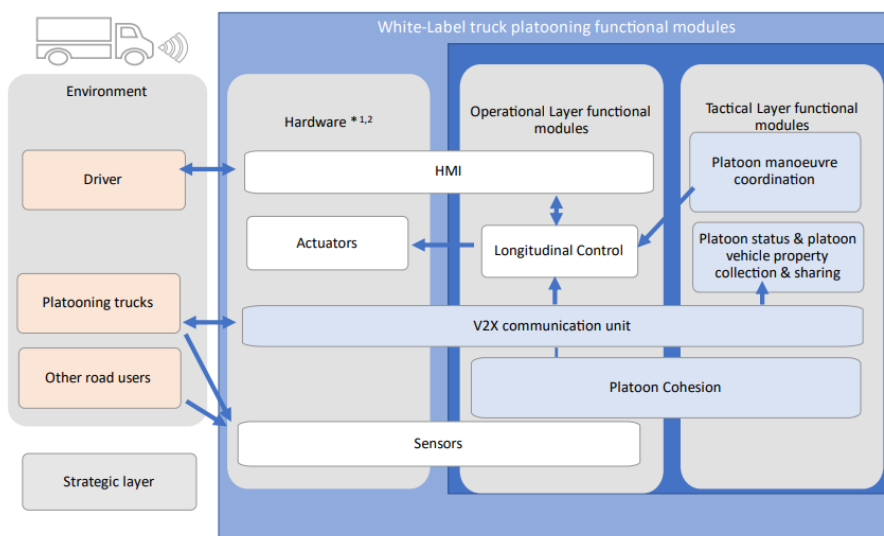


Figura 9. Módulos para un vehículo en ENSEMBLE [34]

2.3.3. Casos de uso a resolver por una formación del *platooning*

Descritos los objetivos de *platooning* y algunos proyectos que han trabajado en ello, pasaremos a describir los diferentes casos de uso que podemos encontrar útil el uso de *platooning*, algunos de ellos son propios de *platooning* como la formación de un *platoon*, pero otros son más generales como puede ser realizar un adelantamiento a un vehículo [33].

2.3.3.1. Formación del *platoon*

- **Formación del *platoon* de forma planeada** (Figura 10): Se daría cuando dos camiones, suscritos al servicio de *platooning* y con la tecnología necesaria para realizarlo (Paso 1), se encontrarán en un punto concreto de una ruta, ya marcada por el servicio de *platooning* (Paso 2), y se considera adecuado que se unan en un *platoon*. Esta planificación se manda desde el servicio de *platooning* a los vehículos implicados y, una vez se encuentren los vehículos, los conductores de estos autorizarán la maniobra y se procederá a la formación del *platoon* (Paso 3).

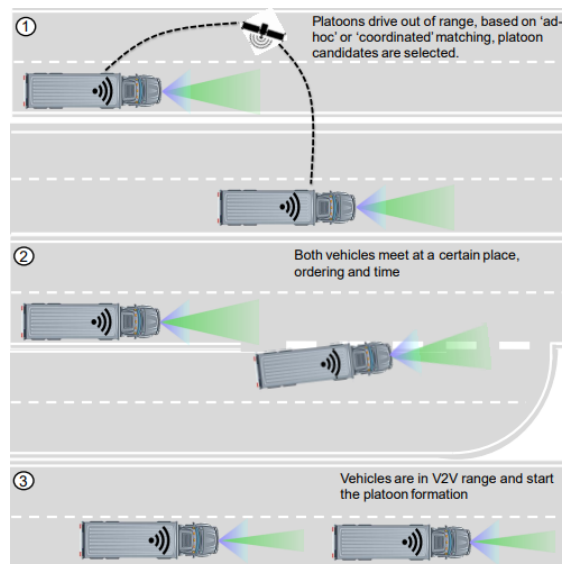


Figura 10. Caso de uso Formación del *platoon* de forma planeada [33]

- **Formación del *platoon* en tiempo real:** Las condiciones para este caso son muy similares al caso anterior, la diferencia es que el punto en el que ambos vehículos se deberían encontrar no es fijo, sino que varía en tiempo real y se comunica a los vehículos constantemente para que lo tengan en cuenta y tomen las consideraciones oportunas.

2.3.3.2. Unirse al *platoon*

En este caso se considera que el *platoon* ya existe y un tercer vehículo decide unirse a él. Se consideran dos posibilidades:

- **Unirse desde la parte trasera del *platoon*** (Figura 11): Mediante el uso de las comunicaciones V2V, un candidato (un único vehículo o un *platoon*) manda una solicitud para unirse al *platoon* que circula delante (Paso 1). Una vez se acepta

esta solicitud y se alcanzan las condiciones óptimas se empieza a reducir la distancia entre el *platoon* y el candidato hasta alcanzar la distancia entre vehículos marcada (Paso 2).

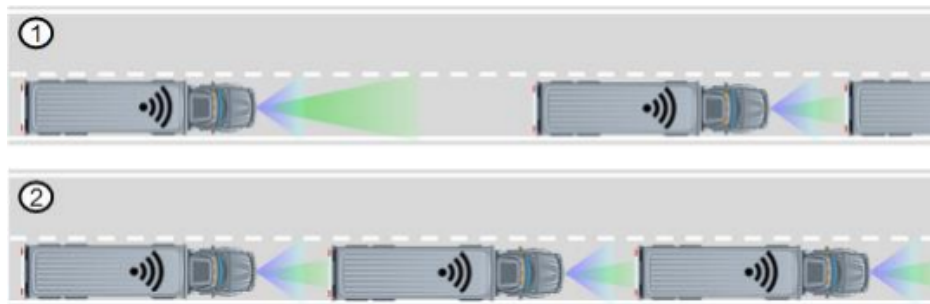


Figura 11. Caso de uso Unirse al *platoon* desde la parte trasera [33]

- **Unirse al *platoon* entre los vehículos que lo conforman** (Figura 12): Un vehículo se incorpora entre dos miembros de un *platoon* ya establecido (Paso 1) provocando una separación mayor de la original. Posteriormente, este vehículo que se ha incorporado se une al vehículo que tiene delante (Paso 2) provocando que el vehículo que tiene detrás se sincronice con el nuevo vehículo del *platoon* (Paso 3). Tras realizar esta sincronización, ambos vehículos continúan circulando de la forma determinada por el *platoon* de forma sincronizada (Paso 4).

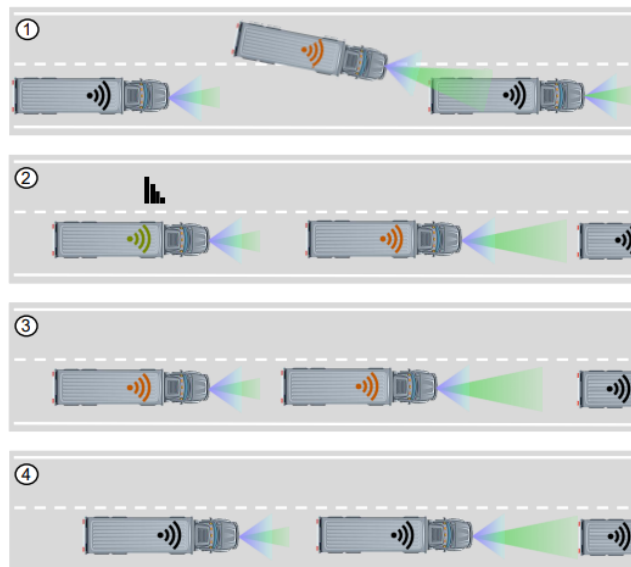


Figura 12. Caso de uso Unirse al *platoon* entre los vehículos que lo conforman [33]

2.3.3.3. *Platooning*

Estos casos de uso se desarrollan suponiendo que el *platoon* ya está formado y que circulan a una velocidad constante gracias al ACC, manteniendo una distancia determinada entre ellos.

- **platooning en estado de espera** (Figura 13): Un grupo de dos o más vehículos se mantienen en formación, con una distancia pequeña entre ellas gracias al ACC usando las comunicaciones V2V aumentando la seguridad (Paso 1).



Figura 13. Caso de uso *Platooning* en estado de espera [33]

- **Coordinación para una frenada** (Figura 14): Con un *platoon* ya formado y circulando de forma estable (Paso 1), el LV reduce la velocidad por un evento normal (tráfico en la vía, peaje...), por lo tanto, el resto de los vehículos del *platoon* seguirán esta deceleración iniciada por el LV (Paso 2). El sistema al volver a detectar la aceleración de nuevo del vehículo que circula delante, solicita la aprobación del conductor para iniciarla de la misma forma (Paso 3), para finalmente volver al estado estable anterior a la frenada (Paso 4).

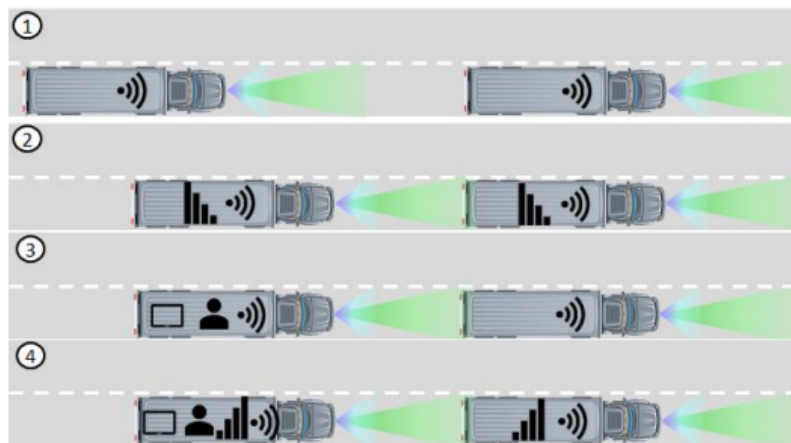


Figura 14. Caso de uso coordinación para una frenada [33]

- **Frenada de emergencia** (Figura 15): Todos los vehículos del *platoon* deberían incorporar un sistema de frenada de emergencia automático (*Automatic Emergency Braking System*, AEBS). Este sistema proporciona una señal de emergencia no más tarde de 1,4 segundos tras el inicio de la maniobra de frenada de emergencia. Cuando se detecta que es necesaria una frenada con una deceleración de más de 4 m/s^2 , se considera una frenada de emergencia (Paso 2), cuando se detecta se manda al resto de vehículos del *platoon* mediante las comunicaciones V2V, una señal de alerta de posible colisión (Paso 3). El vehículo comienza una deceleración tras recibir esta señal de máximo $3,5 \text{ m/s}^2$ y alerta al conductor de esta posible colisión (Paso 4). El sistema AEBS del vehículo anterior al que ha iniciado la frenada, está evaluando constantemente la necesidad de pasar a realizar la frenada de emergencia usando los sensores que incorpora el vehículo (Paso 5). Una vez el sistema AEBS detecta que es necesaria la frenada de emergencia, se inicia la secuencia de frenada de emergencia que dura o hasta la detención total del vehículo o la situación de alerta desaparezca (Paso 6).

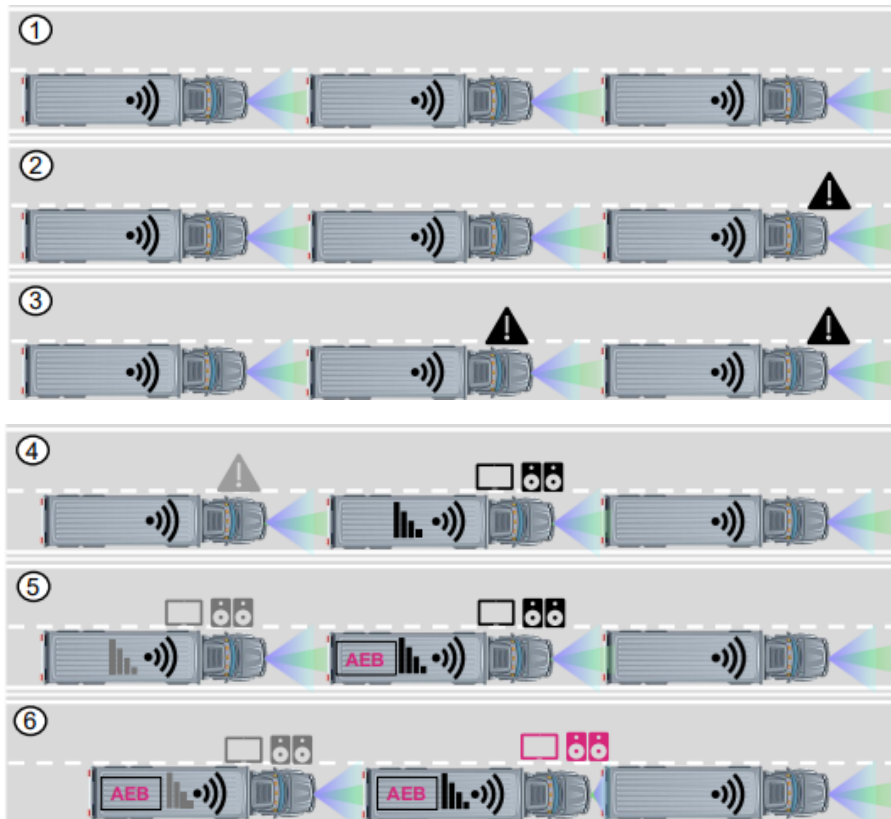


Figura 15. Caso de uso de una frenada de emergencia [33]

- **Intrusión de un vehículo en el platoón** (Figura 16): Si el *platoon* circula de forma estable (Paso 1) y un vehículo se incorpora entre dos miembros del *platoon* (Paso 2), el miembro del *platoon* que circula detrás del vehículo que se ha incorporado, incrementa la distancia con este hasta alcanzar una distancia segura (Paso 3). Una vez el vehículo que no pertenece al *platoon* abandona el carril en el que se encuentra (Paso 4), el miembro del *platoon* que circulaba detrás vuelve a alcanzar la distancia de seguridad establecida con el siguiente miembro del *platoon* (Paso 5).

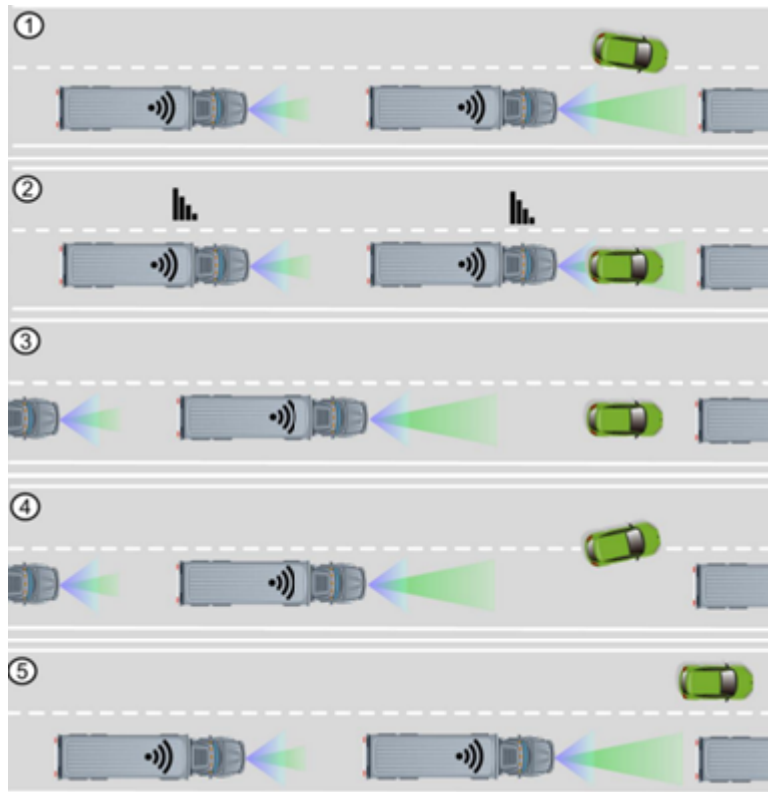


Figura 16. Caso de uso intrusión de un vehículo en el *platoon* [33]

2.3.3.4. Desacoplarse del *platoon*

Cuando un vehículo abandona el *platoon* pueden darse diferentes situaciones que generan cada una un caso de uso distinto:

- **Desacoplo del *platoon* por la parte trasera** (Figura 17): El vehículo situado en la última posición del *platoon* empieza el procedimiento de desacoplo (Paso 1), este proceso comienza mandando una señal al resto de miembros para notificar del desacoplo mediante comunicaciones V2V y aumenta la distancia con el *platoon* (Paso 2). Cuando se alcanza una distancia determinada, se da por terminado el proceso de desacoplo y se termina la comunicación con el resto de los miembros del *platoon* (Paso 3).

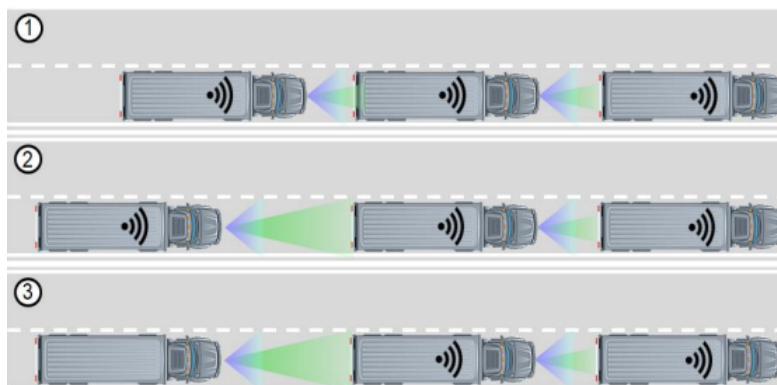


Figura 17. Caso de uso desacoplo del *platoon* por la parte trasera [33]

- **Desacoplo del *platoon* por la parte delantera** (Figura 18): El LV (*Leading Vehicle*) comienza el proceso de desacoplo (Paso 1), manda una señal al resto de miembros del *platoon* mediante las comunicaciones V2V y comienza a aumentar la distancia de seguridad con el resto de los miembros (Paso 2). Una vez ha alcanzado una distancia de seguridad, se interrumpen las comunicaciones con el resto de los miembros del *platoon* y estos deben verificar si pueden seguir conformando el *platoon* con un nuevo LV (Paso 3).

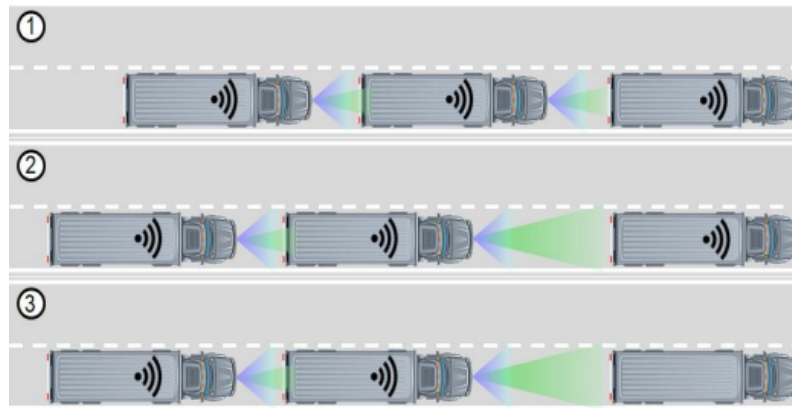


Figura 18. Caso de uso desacoplo del *platoon* por la parte delantera [33]

- **Desacoplo del *platoon* por el centro** (Figura 19): Cuando un vehículo que se encuentra en el centro del *platoon* quiere abandonarlo comienza el proceso de desacoplo, manda una señal al resto de miembros para indicarlo mediante comunicaciones V2V (Paso 1). Los miembros del *platoon* que circulan delante y detrás del que quiere abandonar el *platoon*, aumentan la distancia con este para que la maniobra sea más segura (Paso 2). El vehículo realiza la maniobra de cambio de carril y termina las comunicaciones V2V (Paso 3), posteriormente los miembros restantes reducen la distancia a la ya determinada (Paso 4).

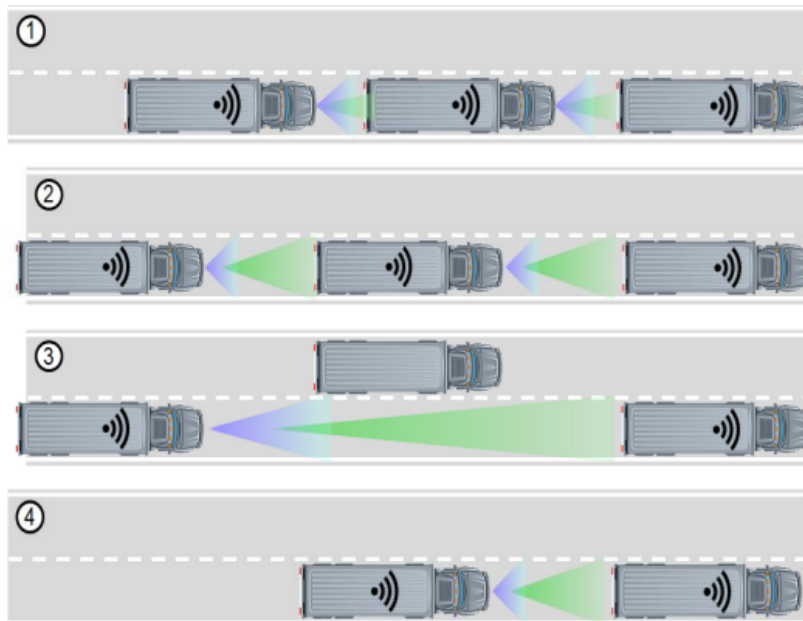


Figura 19. Caso de uso desacoplo del *platoon* por el centro [33]

Estos casos de uso son los más relevantes a mencionar para el *platooning*, vistos los proyectos, los objetivos y los casos de uso de *platooning*, es importante mencionar también la necesidad de introducir unos mensajes, que permitan comunicar a los vehículos los distintos obstáculos que perciban durante la circulación, para que se tomen las consideraciones pertinentes tanto por el sistema como por los propios conductores.

2.4. Collective Perception Messages

Los mensajes de percepción colectiva (*Collective Perception Messages*, CPM), se basan en compartir las detecciones de objetos o usuarios del entorno del vehículo o estación a través de los sensores y sistemas incorporados, estos mensajes están incluidos dentro del servicio de percepción colectiva (*Collective Perception Service*, CPS) [35].

Estos mensajes pueden ser utilizados en diversos casos que pueden beneficiar tanto a los usuarios de las carreteras, como a los propios peatones, por ejemplo. Algunos de esos casos son los siguientes:

- **Detección de usuarios no conectados:** Los usuarios de las vías que no están conectados a estos sistemas C-ITS, solamente pueden ser detectados por otros usuarios o infraestructura que sí están conectados a estos sistemas mediante los sensores y cámaras que pueden llevar incorporados. En este caso, el límite de detección viene dado por las capacidades de los sensores y cámaras equipados en los vehículos e infraestructura, siendo especialmente crítica en zonas donde hay muchos edificios y ángulos muertos. Lo que se pretende con los CPM, es aumentar la percepción de estos usuarios u objetos y que se comparta mediante los sistemas C-ITS [35].
- **Detección de objetos críticos para la seguridad:** Además de los objetos y usuarios que no estén conectados a los sistemas C-ITS, puede haber objetos no deseados en la vía que presenten un potencial peligro. Por ejemplo, podría tratarse de un árbol caído, un remolque que se ha desprendido de un vehículo... etc. Compartir la información de estos obstáculos, permite tomar las precauciones necesarias de parte de los usuarios conectados a los C-ITS y aumentar la seguridad en las carreteras [35].

En algunas situaciones, puede ser beneficioso utilizar la información recibida en los mensajes CAM para generar los CPM. Esas situaciones podrían darse si se tiene la necesidad de aumentar la percepción del entorno, para aumentar la seguridad en el mismo. En una intersección, por ejemplo, la infraestructura podría mandar mensajes CPM de forma periódica a los usuarios de la vía, estos mensajes podrían provenir de la información obtenida por los propios sensores de la infraestructura o de otros vehículos que lo hayan compartido con anterioridad.

Otra situación en la que podría ser beneficioso sería el caso de que haya personas conectadas al sistema ITS, a través de comunicaciones persona-vehículo (*Person-to-vehicle*, P2V), persona-infraestructura (*Person-to-infrastructure*, P2I) o persona-red (*Person-to-network*, P2N) que compartan, tanto su ubicación como otra información relevante para el resto de los usuarios de la vía. De esto podrían verse beneficiados, por ejemplo, trabajadores que realicen reparaciones en la carretera, agentes de policía

realizando un corte de tráfico u otro personal de emergencia, entre otros, para aumentar la seguridad en la vía al estar conectados al sistema. Para realizar las comunicaciones punto-multipunto necesarias para los CPM, sería conveniente utilizar *Geonetworking* según se especifica en [36].

Un CPM está compuesto por una cabecera ITS estándar y múltiples contenedores como se ve en la Figura 20. La cabecera contiene la información de la versión del protocolo utilizado, el tipo de mensaje y un ID perteneciente a la estación de origen.

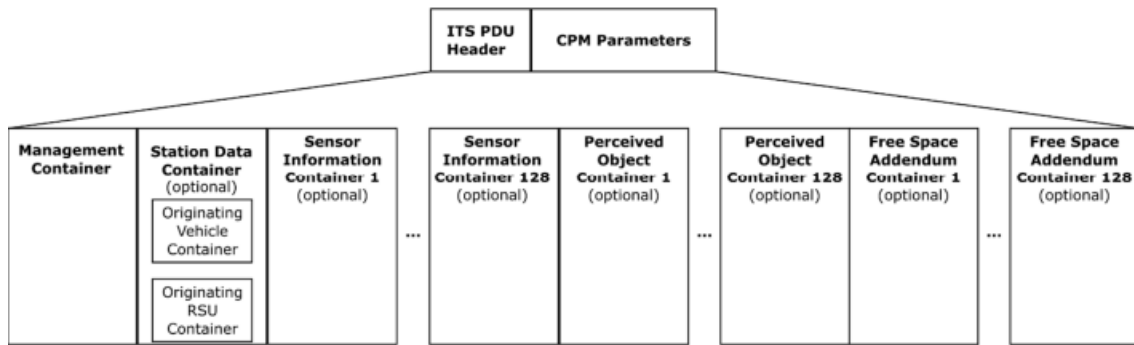


Figura 20. Estructura de un CPM [35]

El mensaje estaría compuesto por: un contenedor de datos de la estación (*Station Data Container*), un contenedor denominado como información del sensor (*Sensor Information Container*), junto con el contenedor de un objeto percibido (*Perceived Object Container*) y por último mencionar el contenedor que describe el espacio libre alrededor de la estación (*Free Space Addendum Container*).

El contenedor de datos de la estación (*Station Data Container*) proporciona información más específica de la estación que ha generado el mensaje. Esta parte variará en función de qué tipo de estación ha generado el mensaje, si lo ha generado un vehículo (*Originating Vehicle Container*), contiene información sobre la dinámica del vehículo que será utilizada, junto con la información del contenedor de un objeto percibido, en un punto de referencia fijo para el resto de las estaciones con las que se comparta el mensaje. Esta información sería el ángulo de orientación del vehículo, ángulo de inclinación, ángulo de balanceo, altura del vehículo e información del tráiler en caso de llevarlo.

Si en lugar de un vehículo, el mensaje es generado por una estación fija (*Originating RSU Container*) a pie de carretera (*Road Side Unit, RSU*), este contenedor incluye una serie de parámetros que ayudan a identificar la localización exacta y la intersección en la que se encuentra la estación generadora.

El contenedor de información del sensor (*Sensor Information Container*) contiene la información correspondiente a las características de cada uno de los sensores montados, ya sea en un vehículo o en una RSU. Cada uno de los sensores que incluyen información en este paquete tienen asignado un id que es utilizado posteriormente por otros componentes del mensaje (*Perceived Object Container*) para relacionar qué sensor ha detectado un objeto concreto. Además del id para cada sensor, se incluye una clasificación del sensor para saber de qué tipo es cada uno, por ejemplo, si el sensor es un radar, LiDAR o cámara.

Este contenedor también diferencia entre la información que proviene de una estación fija y de un vehículo en movimiento. Si es un vehículo el que proporciona la información, se indica la posición de montaje del sensor en el vehículo respecto a un punto de referencia, el rango, el ángulo de apertura horizontal y opcionalmente el vertical (Figura 21). En caso de tener varias áreas de detección para el mismo sensor, por ejemplo, proporcionando información sobre percepción a corta y larga distancia, se pueden llegar a incluir hasta 10 áreas distintas por cada sensor.

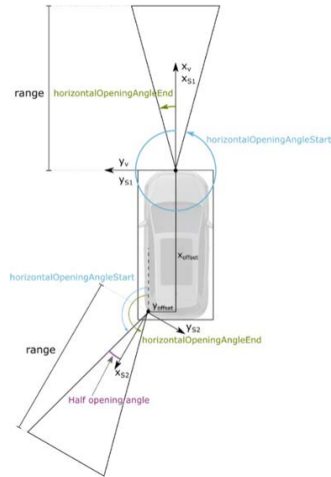


Figura 21 Información dentro *Sensor Information Container* proporcionada por un vehículo [35]

Si la información proviene de una RSU se proporciona una información similar, un punto de referencia con el cual se calcula la posición deseada del objeto a través del offset respecto a ese punto, la posición del sensor respecto a ese punto de referencia, y los ángulos de apertura tanto horizontal como vertical, además de la altura a la que se encuentra el sensor en caso de ser necesario (Figura 22).

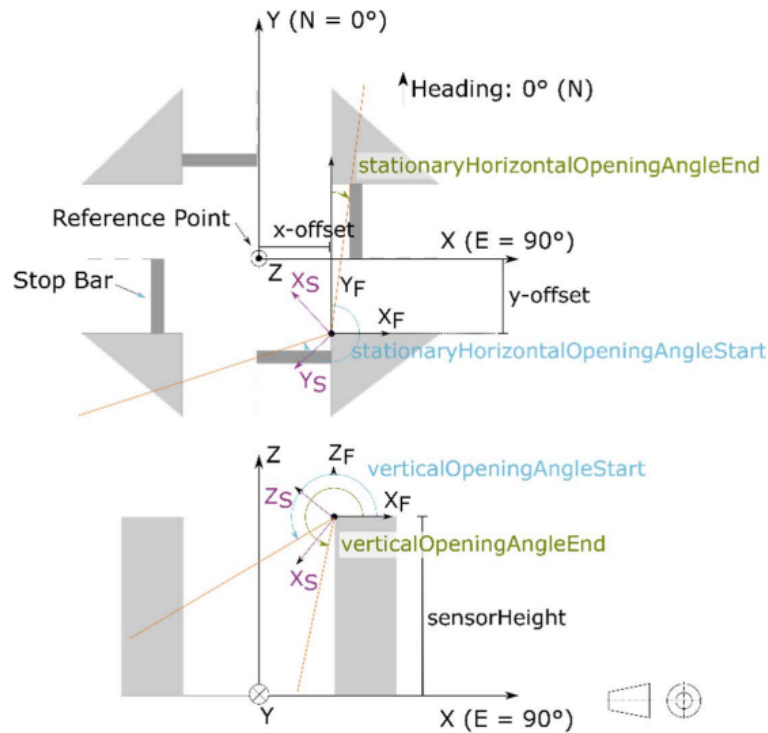


Figura 22. Información proporcionada dentro de *Sensor Information Container* por una RSU [35]

Cada vez que se detecte un objeto debería añadirse un contenedor de objeto percibido (*Perceived Object Container*) al CPM. Dentro de este contenedor se encuentra una descripción del estado dinámico del objeto detectado (Figura 23) y sus propiedades, con al menos la distancia y la velocidad del objeto respecto a la referencia tomada para la medición. A cada objeto se le asigna un ID entre 0 y 255 manteniéndolo hasta que se deje de detectar ese objeto.

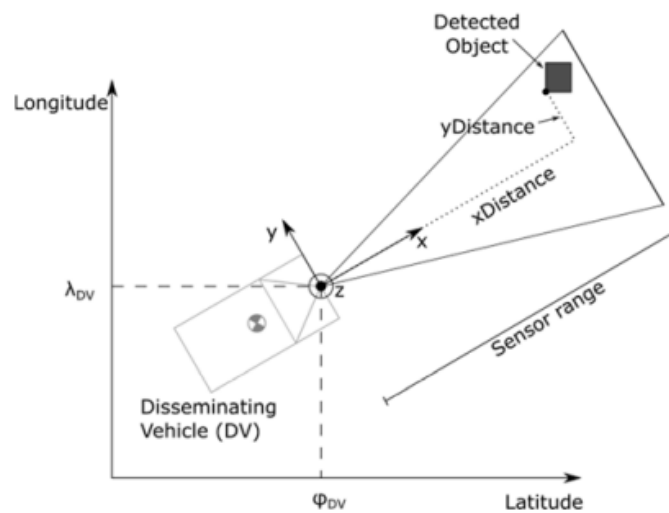


Figura 23 Información de un objeto detectado por un vehículo [35]

El contenedor de espacio libre (*Free Space Addendum Container*) se puede utilizar para proporcionar información sobre el espacio libre alrededor de alguno de los sensores equipados. Si se combina esta información de espacio libre con la información sobre los objetos detectados, el receptor puede obtener una descripción muy completa de toda el

área, sabiendo cuales son las que están libres (azul en la Figura 24) y sabiendo la localización de cada objeto detectado determinar las zonas de sombra (marrón en la Figura 24) [4].

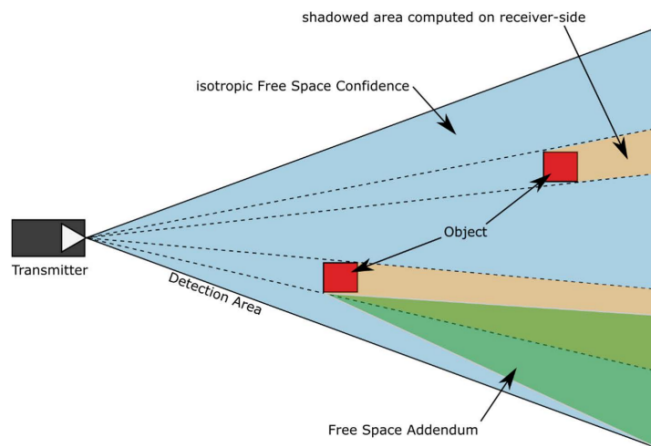


Figura 24. Información incluida en el contenedor de espacio libre [35]

Para maximizar la efectividad del servicio de percepción colectiva, es necesario entender cómo funcionan sus mecanismos de generación de mensajes en las diferentes situaciones. En el artículo [4] se proponen dos casos de simulación, donde se comprueba el rendimiento mediante la medición de diferentes parámetros como la carga del canal y la percepción generada con el uso de los CPM.

El primer caso utiliza el framework de simulación *Artery*, un *software* gratuito bajo la licencia GNU GPLv2. *Artery* incorpora el simulador de redes OMNeT++/OMNEST junto con el simulador de tráfico SUMO (*Simulation of Urban Mobility*), que incorpora la pila completa del protocolo ETSI ITS-G5 y unos modelos realistas para la propagación de la señal.

Se han implementado dos configuraciones diferentes de sensores para este primer estudio tal y como se ve en la Figura 25, el primero implementa una combinación de mediciones a media distancia (65 metros) con una apertura de $\pm 40^\circ$ y larga distancia (150 metros) con una apertura menor de $\pm 5^\circ$, mientras que el segundo utiliza una visión de 360 grados en un radio de 150 metros.

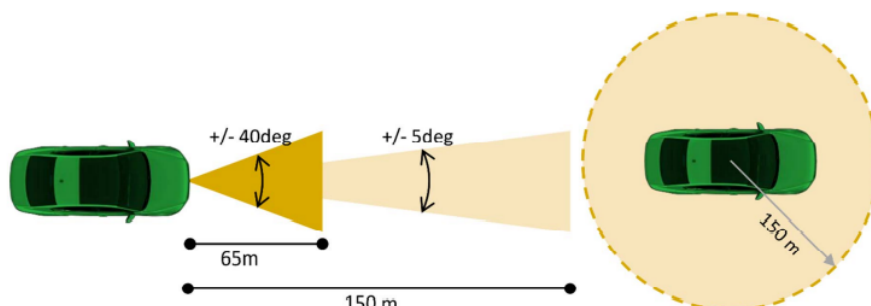


Figura 25. Modelos de sensores implementados en Artery [35]

Los escenarios elegidos han sido:

1. **Escenario Luxemburgo en SUMO** (Figura 26): Simula el tráfico que se produce durante 24 horas en Luxemburgo, incluye diferentes situaciones reales de tráfico tanto en áreas urbanas como en suburbanas y autopistas. Además, incluye obstáculos (zonas rojas en Figura 26) a lo largo de las carreteras implementadas en el modelo, que junto con los sensores equipados en los vehículos generarán zonas de sombra llevando a un modelo más realista en cuanto a la percepción de cada vehículo.



Figura 26. Mapa de Luxemburgo sobre el que se realiza una simulación en SUMO [35]

2. **Escenario Araña** (Figura 27): Se eligió este modelo porque supone un reto en cuanto a la carga en el canal, compuesto por 20 anillos concéntricos con una distancia entre ellos de 50 metros, sin obstáculos entre medias, cada anillo con un carril para cada sentido y los vehículos circulan a 140 km/h de forma constante. Cada vehículo tiene equipado un radar de 360° de los anteriormente mencionados, por lo tanto, la carga de mensajes generada será muy alta ya que al no haber obstáculos junto con el alcance del radar habrá una percepción muy amplia del escenario.

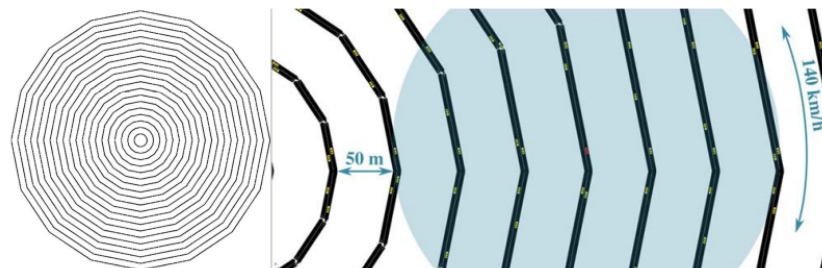


Figura 27. Escenario de simulación "araña". Vista general (izquierda) y zoom detallado (derecha) [35]

Los resultados se han centrado en analizar una serie de parámetros concretos, para entender mejor en conjunto el funcionamiento de los CPM y el efecto que tendría su implementación:

1. **Tamaño y tasa de generación de los mensajes:** El número de CPMs generados por cada vehículo depende del número de objetos detectados. Un vehículo genera

un CPM tan pronto como detecta un objeto nuevo o uno ya percibido cambia su posición o sus condiciones como para generar un nuevo mensaje. Las políticas de generación de mensajes implementadas aumentan el tamaño de los ya que incluyen un gran número de objetos detectados, cuando el tráfico aumenta o el rango de visión de los sensores, aumenta el tamaño ya que se incluyen más objetos en cada CPM.

Por lo tanto, se puede deducir que lo más eficiente es seleccionar los objetos percibidos en los CPM de forma dinámica, es decir, comparar los objetos que se consideran nuevos con los ya detectados en anteriores CPMs, y verificar que es necesaria la generación de un nuevo mensaje y no se trata de un objeto ya detectado con anterioridad.

2. **Carga del canal:** La regla dinámica elegida para la generación de los mensajes, que se comentó en el apartado anterior, requiere de un cierto ancho de banda para tener un funcionamiento adecuado, por lo tanto, es importante prestar atención a este parámetro a la hora de implementar estos sistemas. Para analizar la carga del canal se hace uso del parámetro CBR (*Channel Busy Ratio*), un valor alto de este indica que el canal está muy congestionado y por lo tanto se degradará el funcionamiento del sistema.

La política de generación de mensajes fija, generar mensajes con una frecuencia determinada independientemente de las circunstancias y objetos detectados, genera poca carga si la frecuencia de generación es baja, unos 2 Hz, y una carga muy alta si la frecuencia es alta, unos 10 Hz. La política dinámica genera unos resultados de carga intermedios comparándola con la política fija, con unos entre 3 y 10 CPMs generados por segundo, mientras que la fija genera una mayor cantidad pudiendo provocar congestión en el canal.

La carga no solo se ve afectada por la política de generación elegida, también se ve afectado por la densidad de tráfico del escenario. Un aumento en el flujo de tráfico aumenta el parámetro CBR.

El parámetro CBR afecta a otro parámetro importante, el PDR (*Packet Delivery Ratio*) que se define como la probabilidad de que un CPM se reciba correctamente. En la Figura 28 se puede ver cómo es afectado este parámetro PDR en función tanto de la distancia, de la política de generación utilizada, la densidad de tráfico y del tipo de sensor utilizado para construir el CPM. Como es lógico, cuanto mayor es la distancia entre los vehículos mayor es la degradación que sufre este parámetro debido a los efectos de la propagación, especialmente cuando la carga es muy alta tal y como se aprecia en las líneas señaladas con flechas.

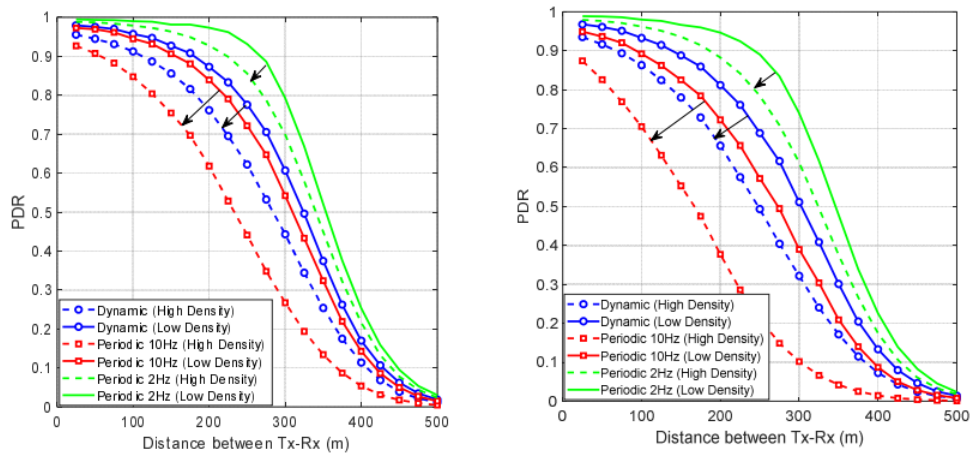


Figura 28. PDR en función de la distancia para los CPM

3. **Percepción:** Las políticas de generación dinámica mencionadas, pueden hacer un uso más eficiente del espectro y el ancho de banda no incluyendo todos los objetos detectados en cada CPM, pudiendo afectar a la percepción de la situación del entorno para el receptor. Para medir la percepción se utilizará el parámetro OAR (*Object Awareness Ratio*), se considerará que un objeto se ha detectado correctamente cuando al menos un CPM incluye información acerca de este objeto, este ratio se calculará en instantes de tiempo espaciados entre 0,1 y 1 segundos.

Este parámetro OAR se degrada de manera muy considerable con la distancia entre el objeto y receptor, afectando especialmente a la política periódica. Otro aspecto a tener en cuenta es la densidad de tráfico, cuanto mayor sea esta densidad menor se degrada el parámetro OAR por haber menos cambios de posición en los vehículos.

En la propuesta de la ETSI [4] se introducen dos métodos de generación de los CPM en dos escenarios concretos, aunque estas no son las únicas propuestas para la generación de mensajes. En [37] proponen unas nuevas reglas de generación de mensajes en los mismos escenarios que la propuesta de la ETSI, la propuesta es simple, el vehículo comprueba si es necesario generar un nuevo CPM cada vez que transcurre un intervalo de tiempo determinado T_{GenCpm} , siguiendo esta norma, para cada objeto detectado se calcula la diferencia respecto a la posición (ΔP), velocidad (ΔS) y tiempo transcurrido (ΔT) desde la última vez que se incluyó el objeto detectado en un CPM. Se generará un nuevo CPM si al menos uno de estos tres parámetros varía dentro de unos márgenes determinados. La variación presentada en este artículo es la siguiente, se incluye un algoritmo que cuando se cumple la condición propuesta por la ETSI para generar un CPM, predice si alguno de los objetos detectados que no están incluidos en el CPM actual, será necesario incluirlo en el siguiente CPM generado.

Para conseguir esto, la predicción se basa en considerar que el objeto mantiene su aceleración siguiendo las siguientes fórmulas, donde S es la velocidad actual y A la aceleración actual:

$$\text{Next } \Delta P = \Delta P + S \cdot T_{\text{GenCpm}} + 0.5 \cdot A \cdot T_{\text{GenCpm}}^2$$

$$\text{Next } \Delta S = \Delta S + A \cdot T_{\text{GenCpm}}$$

$$\text{Next } \Delta T = \Delta T + T_{\text{GenCpm}}$$

Seguendo esto, un objeto se incluirá en el siguiente CPM si $\text{Next } \Delta P > 4$ metros, $\text{Next } \Delta S > 0.5$ m/s o $\text{Next } \Delta T > 1$ segundo. Utilizando esta modificación, los resultados obtenidos en cuanto al generación de los CPM y el número de objetos detectados son los siguientes:

- En un entorno urbano, la generación de CPMs se reduce de 6,1 Hz a 4,1 Hz, es decir un 32,8% menos con una densidad de tráfico baja, mientras que una densidad de tráfico alta se reduce de 5,7 Hz a 3,9 Hz, un 31,6% menos. Consigue esta reducción en la generación de CPMs además de aumentar el número de objetos detectado en cada CPM, de 1,7 a 2,9 con una densidad de tráfico baja y de 1,9 a 3.1 con una densidad de tráfico alta.
- En un entorno extra-urbano, la generación de CPMs también se reduce de 9,8 Hz a 6,0 Hz, una reducción de un 38,8% en una situación de baja densidad de tráfico y de 9,6 Hz a 5,4 Hz, una reducción de un 43,8% en una situación de alta densidad de tráfico. Todo esto, al igual que en el caso anterior, se consigue aumentando el número de objetos percibidos en cada CPM, con un aumento de 6,1 a 11,9 en situaciones de baja densidad de tráfico y de 5,1 a 10,7 en situaciones de alta densidad de tráfico.

Capítulo 3. Odómetro en el vehículo AWS Deepracer

3.1. Introducción

Como ha quedado claro en el capítulo 2, el vehículo debe tener un tipo específico de propiocepción para poder implementar los servicios de transporte inteligente cooperativo. Particularmente, debe ser capaz de establecer su posición geográfica dentro de un mapa con gran precisión y por otro lado debe ser capaz de medir los valores de velocidad de aceleración a los que está sometido, bien sea como resultado indirecto de un cálculo hecho a partir de la posición geográfica o bien porque incluye algún tipo de sensor de odometría.

El vehículo Deepracer incluye una pequeña IMU con muy baja precisión que podría dar información sobre la aceleración instantánea, pero no es capaz de dar valores fiables de velocidad y, por lo tanto, tampoco de posición. Estos dos problemas tienen que ser resueltos para poder implementar servicios de transporte inteligente.

En primer lugar, se decidió desarrollar un odómetro para el vehículo AWS Deepracer. El odómetro debería resolver varios problemas al mismo tiempo, primero permite un mejor control de la velocidad y por lo tanto del control de vehículo. Aunque es posible fijar una velocidad al vehículo, esto se realiza no fijando una velocidad concreta, sino un nivel de corriente al motor, por lo que la velocidad no se conoce. Es evidente que esto tiene consecuencias en el algoritmo de control. Por otra parte, se esperaba que a partir de la información obtenida hubiera sido posible saber la distancia recorrida y por ende la posición dentro de la maqueta. Este segundo objetivo demostró ser imposible de conseguir.

3.2. Montaje y desarrollo odómetro

3.2.1. Selección de materiales y diseño del circuito

En este apartado se expondrá el proceso de desarrollo del odómetro para el vehículo AWS Deepracer, tanto de la parte de *hardware* como de *software*, además de los resultados obtenidos en cuanto a su precisión en las medidas.

El odómetro es un sistema que permite medir la distancia recorrida. En el caso de los vehículos a veces también sirve para medir la velocidad. En este trabajo se ha construido un odómetro clásico, esto es, medir el número de vueltas que giran las ruedas del vehículo, para lo cual se utilizarán imanes dispuestos longitudinalmente en la circunferencia de una de las ruedas y un sensor de tipo *hall*.

Las condiciones que debe cumplir el odómetro es que debe dar suficiente precisión para saber la posición del vehículo con un margen de error pequeño (se buscaba que fuera de alrededor de 2 cm) y que fuera capaz de medir velocidades de hasta 0,8 m/s

Por lo tanto, para el desarrollo del odómetro, fue necesario en primer lugar seleccionar un sensor *hall* adecuado para la tarea y unos imanes, de un tamaño adecuado, para colocar en la rueda del vehículo AWS Deepracer.

Se eligió el sensor *hall* A1120EUA-T, ya que incluye un ciclo de histéresis que permite evitar falsas lecturas con los pasos de los imanes. El funcionamiento del sensor (Figura 29) es el siguiente, mientras no reciba ningún campo magnético perpendicular al sensor la señal de salida V_{OUT} se mantendrá en alta, mientras que si comienza a incidir un campo magnético perpendicular con un valor suficiente B_{OP} conmuta a una señal en baja. Una vez ese campo magnético se reduce hasta el valor B_{RP} vuelve a conmutar a un valor en alta.

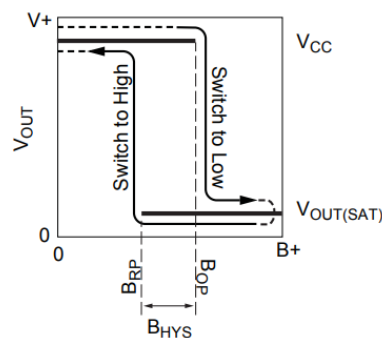


Figura 29. Gráfica funcionamiento sensor *hall* A1120EUA-T

El circuito recomendado para utilizar el sensor es el que vemos en la Figura 30, se incluye un condensador C_{BYP} de valor $0,1 \mu F$ entre la alimentación y la tierra del dispositivo, para reducir el ruido que la alimentación podría introducir en el dispositivo y perturbar las mediciones. Además, se añade una resistencia R_L de $1 K\Omega$ según las especificaciones del sensor. El voltaje de alimentación del circuito es 5V.

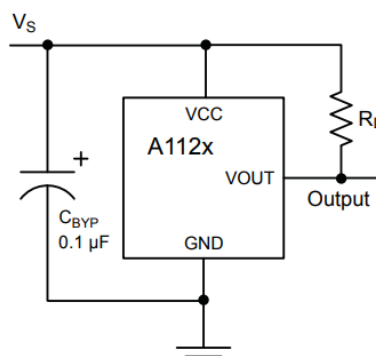


Figura 30. Circuito recomendado para el sensor A1120EUA-T

3.2.2. Pruebas funcionales del prototipo

Una vez caracterizado el sensor se montó un circuito (Figura 31) para realizar las pruebas necesarias para ver si se podía medir la velocidad deseada para la maqueta, $0,8 \text{ m/s}$, y cuantos imanes se podían incorporar en la rueda para la medición de la velocidad y la distancia recorrida de cara tener la mejor precisión posible. Para realizar las pruebas se

utilizó un motor de pasos PM55L-048 unipolar, junto con un controlador de motor de pasos TB6600.

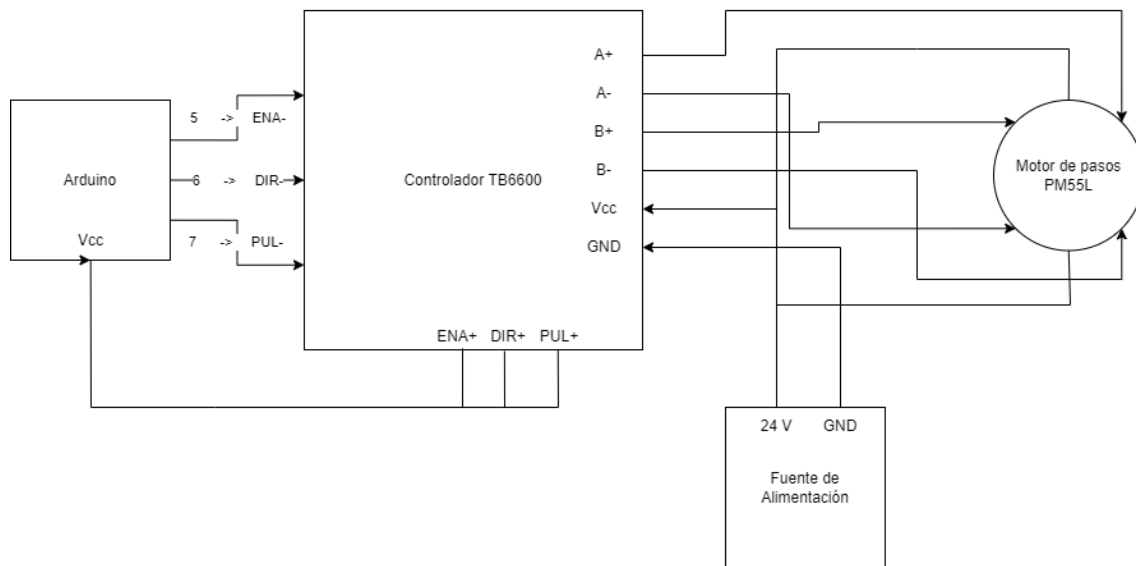


Figura 31. Circuito controlador motor de pasos PM55L

El funcionamiento del circuito es el siguiente, el Arduino utilizado manda tres señales al controlador TB6600, la primera ENA a través del PIN 5 del Arduino, con esta señal si se mandase un "1" se pararía el motor. La señal DIR a través del PIN 6 marcaría la dirección de giro del motor, si se mantiene en baja girará a derechas, y por último la señal PUL a través del pin 7, con esta señal se marcará la velocidad de giro del motor, ya que dependiendo de la frecuencia de conmutación entre alta y baja que le demos a esta señal, más rápido conmutarán las señales A y B para hacer girar el motor más rápidamente. El valor asignado para la conmutación de la señal PUL es de 5ms entre cambios de estado, para poder llegar a alcanzar las 300 rpm.

La fuente de alimentación proporcionará un voltaje de 24 V y 4 A para poder utilizar la mayor velocidad de giro posible del motor ya que, si no, no se alcanzarían las revoluciones deseadas para hacer las pruebas, unas 300 rpm para emular la velocidad deseada de 0,8 m/s.

Para las pruebas se acopló al motor de pasos unas piezas de madera circula a modo de ruedas a las que estaban adheridos los imanes que posteriormente se acoplarán a la llanta del coche para la maqueta, el tamaño de la rueda de madera es el mismo que el interior de la llanta del coche, 3,16 cm de radio y 19,85 cm de circunferencia. Los imanes elegidos son de diferentes tamaños, para comprobar cuál es el más adecuado para funcionar con el sensor *hall* seleccionado. Se ha encontrado que dependiendo del tamaño de los imanes el sensor *hall* debe encontrarse a una distancia mínima diferentes, que son las siguientes:

1. Imán de 5x2 mm => Distancia requerida para medir correctamente la pasada del imán: 7 mm.
2. Imán de 4x1 mm => Distancia requerida para medir correctamente la pasada del imán: 3 mm.
3. Imán de 3x1 mm => Distancia requerida para medir correctamente la pasada del imán: 2 mm.

4. Imán de 2x1 mm => Distancia requerida para medir correctamente la pasada del imán: < 1 mm.
5. Imán de 1x1 mm => Distancia requerida para medir correctamente la pasada del imán: < 1 mm.

Viendo las distancias necesarias para medir correctamente el paso del imán con el sensor, el más adecuado para el montaje es el imán de tamaño 4x1 mm, ya que ofrece una distancia al sensor razonable y es lo suficientemente pequeño para montar 8 o más imanes en la rueda del vehículo, que nos daría una precisión de alrededor de 2,5 cm en la medida de distancia recorrida.

Además del circuito para el motor, en las pruebas se montó otro circuito para recoger las medidas del sensor y calcular la velocidad como se ve en la Figura 32. Este circuito es el mismo que el de la Figura 30 llevado sobre un Arduino.

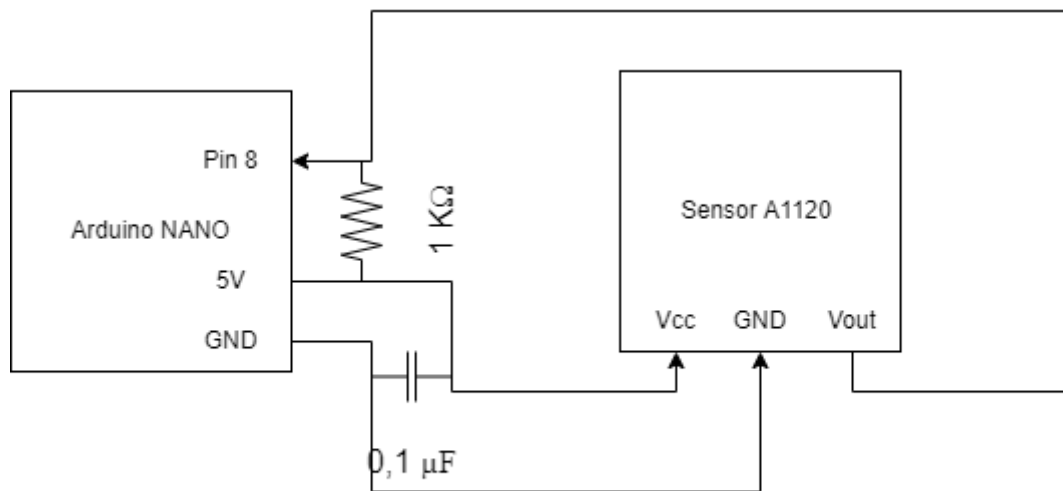


Figura 32. Circuito control sensor A1120

Con este segundo circuito, en el pin 8 del Arduino se detectan las pasadas de los imanes como cambios de estado en la señal del sensor y se les asigna un *timestamp*. La diferencia de tiempos entre cambios de estado nos permite calcular de manera simple la velocidad del vehículo, y con el número de cambios de estado se calcula la distancia recorrida. Las ecuaciones a aplicar son sencillas:

$$\text{Distancia recorrida} = \frac{\text{Número imanes detectados}}{\text{Número imanes}} 2\pi \cdot 19,85 \text{ [cm]}$$

$$\text{Velocidad instantánea} = \frac{0,1985}{\text{Número imanes}} \cdot \frac{1}{\text{Tiempo entre dos imanes}} \left[\frac{\text{m}}{\text{s}} \right]$$

Toda esta información, velocidad, diferencia de tiempo entre medidas, número de imanes detectados y distancia recorrida se mandan a través del puerto serie al coche AWS Deepracer al que estará conectado el Arduino.

Recibida esta información en el coche, se publica la información en un nodo ROS, para poder acceder a este valor de velocidad calculada mediante el nodo de control externo del

vehículo desarrollado en el Trabajo de Fin de Máster de Ignacio Royuela González [38] con un subscritor de ROS.

ROS (*Robot Operating System*) es un *framework* de código abierto, diseñado para la programación de robots proporcionando una capa de abstracción, que permite no preocuparse del *hardware* del sistema. La base de ROS son los nodos, mensajes, tópicos y servicios. Los nodos se definen como los procesos que realizan los cálculos u operaciones necesarias para una funcionalidad, los mensajes son utilizados por los nodos para comunicarse entre sí con unos datos previamente definidos, un nodo manda estos mensajes publicándolos en un tópico que es visible para el resto de los nodos del sistema y de la misma forma, si un nodo quiere obtener una información concreta se suscribiría al tópico donde el otro nodo ha publicado la información. En el caso de necesitar estas comunicaciones de manera síncrona, se utilizarían servicios, con un mensaje definido para la respuesta y otro para la solicitud [20].

La información recogida por el nodo subscritor de ROS, se envía utilizando UDP en el campo de datos de este protocolo, para identificar los distintos datos enviados, se añadió un carácter de identificación para saber el tipo de dato que se ha recibido en la recepción de la trama. El carácter de identificación de la velocidad dentro de la trama es el carácter V. Esta información relativa a la velocidad no es el único dato que se manda utilizando UDP, también se manda información de la cámara marcada con el carácter de identificación I o R, en función de si es la cámara derecha (R) o izquierda (I), además se manda información del LIDAR marcada con el carácter L e información de la batería (B) o parámetros de control (C).

Una vez montado el circuito completo y habiendo comprobado, que el tamaño de los imanes seleccionado es el adecuado con la distancia a la que se puede realizar la medición. Se procedió a comprobar cuantos imanes se podía incorporar en la rueda, para obtener la máxima precisión posible en la medida. Se probaron las configuraciones con 1, 2, 4, 8 y 16 imanes (configuraciones elegidas por la facilidad para trazar los ángulos). Se comprobó que el sistema funciona con hasta 8 imanes. En el caso de 16 imanes el sensor no conmuta correctamente debido a la proximidad entre imanes debido a que el campo magnético global de todos los imanes es demasiado potente. Por ello se decidió que el número ideal para nuestro sistema es de 8 imanes (Figura 33). El test de prueba consistió en forzar al motor a girar un número concreto de vueltas, que se corresponden, según el número de imanes colocados a un número concreto de cambios de polaridad en el sensor. En el caso de los 8 imanes, se hizo girar el motor a 300 rpm durante un minuto exacto midiendo el número de vueltas totales de la rueda, el valor esperado era de 300 revoluciones y se obtuvo un resultado de 302 revoluciones, por lo tanto, si deberían haberse leído un total de 2400 imanes en el cálculo teórico, se leyeron 2416 imanes. Sin embargo, en el caso de utilizar 16 imanes en la rueda, se repitió el mismo caso que con 8, obteniendo sobre el resultado esperado de 300 revoluciones 205 revoluciones medidas, sobre 4800 imanes se han leído únicamente 3280, obteniendo un 31,66% de pérdidas en la lectura de los imanes. En cuanto a la precisión, con 8 imanes se consigue una presión de $\pm 2,48$ cm, suficiente para los objetivos planteados.



Figura 33. Rueda con imanes en circuito de pruebas

Comprobado que con esta configuración se adapta correctamente a las condiciones planteadas de velocidad y medida de longitud, se procedió al montaje en el coche AWS Deepracer en la rueda trasera izquierda. Se montó en esta rueda ya que no es una rueda directriz y, por lo tanto, no habría problemas en cuanto a perder pasadas de imanes, dado que la rueda al girar junto con sus imanes adheridos y el sensor *hall* se mantendrán siempre en el mismo eje como se observa en la Figura 34. En dicha figura vemos en la parte superior el Arduino, junto con el resto de los componentes soldados a una placa perforada, acoplada al único hueco que tiene disponible el coche y en la parte inferior izquierda vemos los imanes adheridos a la rueda y el sensor sujeto al eje de la misma.

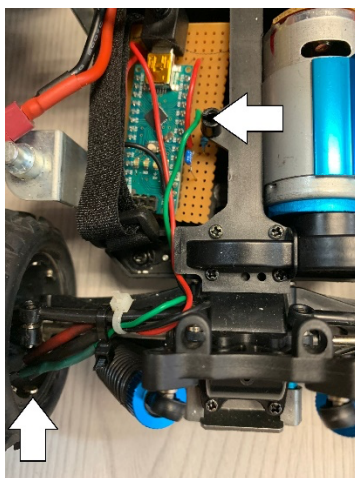


Figura 34. Montaje odómetro en el coche AWS Deepracer

Las pruebas para medir la precisión del odómetro se realizaron en la maqueta fabricada tanto por Juan Carlos Aguado como por Ignacio Royuela dentro del proyecto de investigación Artemis (VA231P20). En este caso se montó por un lado una rotonda de 80 cm de radio y por lo tanto 5 metros de circunferencia en la línea amarilla, utilizada para medir la precisión en los giros en ambas direcciones, (Figura 35) y por otro lado una recta de 25 metros de longitud para medir la precisión en línea recta.



Figura 35. Rotonda utilizada para las pruebas

En la siguiente sección, se pasarán a analizar los resultados de precisión obtenidos durante las pruebas con el odómetro en ambos escenarios, la rotonda y la recta.

3.3. Resultados odómetro

Inicialmente, se comenzaron las pruebas realizando 10 vueltas a la rotonda en ambas direcciones y comprobando, tanto la distancia recorrida como la velocidad obtenida.

El valor de radio de la rueda utilizado para estas mediciones fue de 3,2 cm, además hay que tener otras consideraciones en cuenta, el coche al realizar los giros no los realiza totalmente centrado sobre la línea amarilla, como se observa en la Figura 35. Por lo tanto, la distancia total recorrida no será la de 5 metros que se ha calculado para la circunferencia descrita por la línea, sino que en el giro a derechas el vehículo va por la parte exterior de la carretera, aumentando la distancia recorrida por vuelta, y en el giro de izquierdas ocurre justo lo contrario.

Sabiendo que el coche no realiza la rotonda centrado, se considerará que la distancia recorrida al realizar el giro, no es exactamente 5 metros por vuelta, sino que se supondrá que el coche está desplazado la totalidad de su ancho hacia dentro o hacia afuera, por lo tanto las distancias que recorrerá serán las siguientes:

- Odometro en el exterior: 0,8 metros (radio original) + (2* 0,07) metros (Mitad del ancho del coche) + = 5,9 metros por vuelta.
- Odometro en el interior: 0,8 metros (radio original) – (2* 0,07) metros (Mitad del ancho del coche) = 4,14 metros por vuelta.

Las medidas obtenidas en esta primera prueba fueron las siguientes:

Tabla 1. Resultados odómetro primera prueba

	Velocidad Medida	Velocidad Teórica	Distancia Medida	Distancia Teórica
Interior	0,70m/s	0,68 m/s	39,3 metros	4,14 metros
Exterior	0,48 m/s	0,50 m/s	62,60 metros	59 metros

Después de la prueba en la rotonda se realizó la medida de la distancia en la recta de 25 metros, obteniendo un valor de 23,2 metros, por lo tanto, se ajustó el valor del radio de la

rueda a 3,6 cm para obtener la medida exacta de 25 metros y ayudar también en la media de la distancia en las curvas.

Viendo los resultados obtenidos en esta primera prueba, la medida de la velocidad es bastante precisa, pero fluctúan mucho los valores a la hora de visualizarlo en pantalla, por lo tanto, para conseguir un valor más estable, en lugar de utilizar el valor de velocidad instantánea obtenido del Arduino, se introdujeron los valores en un array de 10 posiciones y con estos valores almacenados en el array, se calcula la media para obtener un valor final de la velocidad.

Se fijó un tamaño de 10 en el array porque un valor mayor, aunque otorgaría una mayor estabilidad, ralentizaría la llegada a un punto de estabilidad a la hora de visualizar la velocidad ya que, al aumentar el tamaño del array, el tiempo necesario para llenar todos esos valores es mayor, obteniendo con un tamaño de 10 un buen compromiso entre el tiempo de visualización y la estabilidad.

El error en el cálculo de la distancia es mayor que en el caso de la velocidad, por lo tanto, se decidió calcular la distancia recorrida de una forma alternativa. El sensor *hall* está situado en la rueda trasera izquierda, por lo tanto, la medida de la distancia se realiza desde ese punto, así que se decidió realizar la traslación del punto de medida de la rueda trasera izquierda al centro del coche, para realizar esta traslación era necesario conocer el ángulo al que estaba girando el coche.

La información del ángulo se extrae de un nodo ROS del coche, en el cual un nodo publica el valor del ángulo al que orienta el coche el motor de la dirección.

Conocido este ángulo se realizaron los cálculos necesarios para trasladar este punto de media al centro del coche como se ve en la Figura 36.

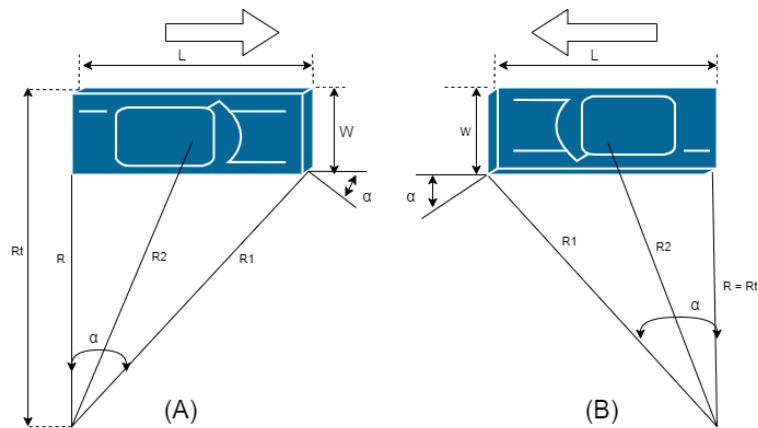


Figura 36. Traslación del punto de giro al centro del coche en ambos sentidos del giro

A través del radio de la circunferencia que describe la rueda con el odómetro R_t , el valor del radio en el centro del coche R_2 , el radio en la rueda delantera R_1 , el radio de giro α y el ancho W y largo L del coche, se realiza la traslación de la distancia recorrida desde el punto donde se realiza la medida hasta el centro del coche mediante las siguientes ecuaciones:

- **Caso A** (Giro a derechas), considerando cada paso de imán al tener 8 imanes un recorrido de $(\pi/4 * r)$, siendo r el radio de la rueda:

$$R_t = R_1 * \cos \alpha + W$$

$$R = R_1 * \cos \alpha = L \frac{\cos \alpha}{\sin \alpha}$$

$$R_2 = \sqrt{\left(R + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}$$

$$d = \frac{\pi}{4} * r \Rightarrow \beta * R_t = \frac{\pi}{4} * r$$

$$\begin{aligned} D = \beta * R_2 &= \frac{\pi}{4} * \frac{r}{R_t} * R_2 = \frac{\pi}{4} * r * \frac{\sqrt{\left(R + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}}{R + W} \\ &= \frac{\pi}{4} * r * \frac{\sqrt{\left(L \frac{\cos \alpha}{\sin \alpha} + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}}{L \frac{\cos \alpha}{\sin \alpha} + W} \\ &= \frac{\pi}{4} * r * \frac{\sqrt{\left(L * \cos \alpha + \frac{W}{2} * \sin \alpha\right)^2 + \left(\frac{L}{2}\right)^2 * (\sin \alpha)^2}}{L * \cos \alpha + W * \sin \alpha} \end{aligned}$$

- **Caso B** (Giro a izquierdas), considerando cada paso de imán al tener 8 imanes un recorrido de $(\pi/4 * r)$, siendo r el radio de la rueda:

$$R_t = R_1 * \cos \alpha$$

$$R = R_1 * \cos \alpha = L \frac{\cos \alpha}{\sin \alpha}$$

$$R_2 = \sqrt{\left(R + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}$$

$$d = \frac{\pi}{4} * r \Rightarrow \beta * R_t = \frac{\pi}{4} * r$$

$$\begin{aligned} D = \beta * R_2 &= \frac{\pi}{4} * \frac{r}{R_t} * R_2 = \frac{\pi}{4} * r * \frac{\sqrt{\left(R + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}}{R} \\ &= \frac{\pi}{4} * r * \frac{\sqrt{\left(L \frac{\cos \alpha}{\sin \alpha} + \frac{W}{2}\right)^2 + \left(\frac{L}{2}\right)^2}}{L \frac{\cos \alpha}{\sin \alpha}} \\ &= \frac{\pi}{4} * r * \frac{\sqrt{\left(L * \cos \alpha + \frac{W}{2} * \sin \alpha\right)^2 + \left(\frac{L}{2}\right)^2 * (\sin \alpha)^2}}{L * \cos \alpha} \end{aligned}$$

Para realizar este cálculo alternativo de la distancia era necesario mandar información adicional desde el Arduino, ya que este cálculo lo realizará el coche. La información

necesaria es el número de imanes leídos entre cada mensaje recibido, ya que la fórmula calculada se aplica una vez por cada imán leído entre mensajes, es decir, si entre cada mensaje recibido se han leído 5 imanes se aplicará la fórmula para cada uno de estos 5 imanes, además se envía desde el Arduino el tiempo entre medidas de los imanes para además de poder calcular la distancia se pueda obtener un valor de velocidad alternativo y comprobar cuál es más preciso entre ambos.

Por último, se observó que el valor recibido del nodo ROS del ángulo de giro oscilaba mucho, lo que causaba distorsión en la medida, por lo que se introdujo un array como en el caso de la velocidad para tener un valor de ángulo de giro con un valor más estable mediante el cálculo de la media.

Dentro del programa se consideraba que el coche tenía el mismo ángulo de giro máximo en ambas direcciones (izquierda y derecha), pero se observó que el ángulo de giro máximo es mayor cuando gira a izquierdas que a derechas, obteniendo además trayectorias más cerradas en la rotonda, por lo tanto, en las últimas pruebas se consideraron dos casos: en el primero el ángulo de giro es igual a izquierdas y derechas con un valor máximo de 15°, en el segundo el ángulo es diferente, a izquierdas el valor máximo es 20° y a derechas el valor máximo es 10°.

Teniendo todo lo anterior en cuenta, se muestran en la Tabla 2 los diferentes resultados en cada caso, teniendo por un lado los resultados obtenidos con la fórmula teórica (Teóricos), los resultados obtenidos de las lecturas del Arduino (Leídos) y por último los resultados obtenidos del cálculo alternativo mediante la fórmula desarrollada anteriormente (Fórmula). Los resultados fueron los siguientes:

Tabla 2. Resultados finales odómetro

	Posición del odómetro	Imanes leídos		Distancia			Velocidad		
		Teóricos	Leídos	Teórico	Odómetro	Fórmula	Teórico	Odómetro	Fórmula
$\alpha = 15^\circ$ en A y B	Exterior	1940	2139	54,6 metros	60,45 metros	56,01 metros	0,478 m/s	0,52 m/s	0,49 m/s
	Interior	1625	1629	45,8 metros	46,03 metros	50,72 metros	0,67 m/s	0,68 m/s	0,75 m/s
$\alpha = 10^\circ$ en B y $\alpha = 20^\circ$ en A	Exterior	1940	2100	54,6 metros	58,61 metros	54,76 metros	0,723 m/s	0,77 m/s	0,73 m/s
	Interior	1625	1637	45,8 metros	46,26 metros	52,70 metros	0,64 m/s	0,65 m/s	0,74 m/s

Viendo los resultados obtenidos, vemos que la precisión al ajustar el radio de la rueda, por la imprecisión obtenida previamente en la medida en la recta a 3,6 cm, aumenta y se reduce el error en la medida de la distancia y la velocidad. Considerando en primer lugar el caso que $\alpha = 15^\circ$. para los giros a derechas (A) o los giros a izquierdas (B), los resultados con el odómetro en la posición exterior mejora la precisión, respecto a las primeras pruebas, tanto en la medida de la velocidad como de la distancia. En caso de la distancia, se observa que, respecto al resultado teórico esperado 54,6 metros, se obtiene un resultado de 60,45 metros con el odómetro y 56,01 metros utilizando la fórmula, por lo tanto, se obtiene un mejor resultado en este caso utilizando la fórmula al igual que ocurre con la velocidad, esperando un valor de 0,478 m/s se obtienen 0,52 m/s con el odómetro,

mientras que con la fórmula los resultados son más precisos obteniendo un valor de 0,49 m/s.

Sin embargo, cuando el odómetro se encuentra en la posición interior, ocurre lo contrario, los resultados mejoran con las medidas del odómetro, obteniendo una medida de 46,03 metros con el odómetro respecto a los 45,8 metros esperados teóricamente. Esto se debe a que las oscilaciones del valor del ángulo, utilizado en el cálculo de la fórmula, que se obtienen del vehículo mediante un nodo ROS, tienen unas oscilaciones demasiado grandes debido a la propia fabricación del coche. Esto se debe a que el vehículo no gira igual hacia ambos sentidos, al girar a izquierdas el servo que maneja la dirección necesita ejercer una fuerza mayor para conseguir el mismo ángulo que si girase a derechas.

Teniendo esto en cuenta, se repitieron las pruebas, pero considerando dos valores de ángulo de giro máximo distintos en función de si gira a izquierdas o a derechas, $\alpha = 10^\circ$ en los giros a izquierdas y $\alpha = 20^\circ$ en los giros a derechas. Con esta suposición, los resultados, obtenidos con la fórmula, tanto en la medida de la velocidad como de la distancia son más próximas al valor teórico esperado en el caso de la posición del odómetro en el exterior (giro a derechas), sin embargo, al tener el odómetro en la posición exterior la precisión empeora ya que se obtiene un resultado de 52,7 metros respecto a los 50,72 metros del caso anterior.

Como conclusión, cada uno de los métodos utilizados ofrece mejores resultados en cada uno de los escenarios, en caso de tener el odómetro en la posición interior (giro a izquierdas, B) se obtienen mejores resultados con la medida del odómetro, mientras que si el odómetro está en la posición exterior (giro a derechas, A) ofrece mejores resultados la fórmula, esto concuerda con los resultados obtenidos en el número de imanes medidos por el odómetro, ya que se miden más imanes por las oscilaciones que sufre el vehículo al girar en este sentido, se producen errores en la lectura de los imanes.

Capítulo 4. Posicionamiento *Indoor*: Módulos UWB M5Stack

4.1. Introducción

Este sistema de posicionamiento *indoor* se implementa debido a la necesidad de conseguir la posición del vehículo dentro de la maqueta.

Como se explicó en el capítulo 2, se propuso la implementación del sistema *indoor* mediante dos sistemas que por ser económicos estaban al alcance del proyecto. En este capítulo se explican los resultados de integración y uso de los módulos UWB M5Stack. Este primer sistema se plantea como la opción más económica de los dos sistemas planteados, y utiliza la tecnología UWB para conseguir la posición del vehículo, pero con una precisión menor según el fabricante que el otro sistema propuesto.

A continuación, se procederá a realizar un análisis del funcionamiento de los módulos UWB, siguiendo con la configuración y despliegue de los mismos finalizando con los resultados de precisión obtenidos en las pruebas realizadas.

4.2. Análisis de los módulos UWB

Los módulos UWB de M5Stack integran el protocolo de comunicaciones UWB (*Ultra Wide Band*) que utiliza pulsos de nanosegundos para localizar objetos y definir su posición. El diseño de esta unidad utiliza el transmisor-receptor Ai-ThinkerBU01 cuyo diseño se basa en el del módulo DW1000 de Decawave, cuyo chip interno STM32 utiliza el algoritmo para el cálculo de la distancia consiguiendo precisiones de $\pm 10\text{cm}$, soportando también comandos AT para la configuración de los módulos. El esquema del módulo Ai-ThinkerBU01 se puede ver en la Figura 37.

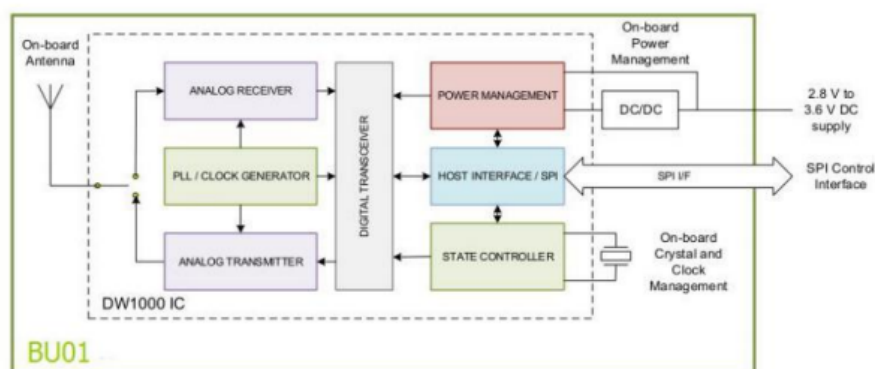


Figura 37. Esquema módulo Ai-ThinkerBU01

Este módulo incorpora un puerto HY2.0-4P que permite conectar los módulos M5Stack al coche mediante un adaptador USB-UART para la posterior configuración como vemos en la Figura 38.



Figura 38. Conexión entre el módulo UWB y el adaptador UART-USB

Los módulos UWB M5Stack solo permiten la utilización de un módulo configurado como una estación móvil y otros 4 como estaciones fijas, la estación móvil devolverá la distancia que calcula a las estaciones fijas que calcula por la interfaz UART al equipo al que esté conectado. Es importante señalar que esta limitación no es del protocolo en sí, sino que parece que se debe al diseño del propio sistema. Por lo tanto, sistemas basados en la misma tecnología de otros fabricantes podrían admitir más de una estación móvil.

El módulo DW1000 [12] contiene un transmisor, un receptor y un *back-end* digital que sirve de interfaz para el host al que esté conectado, acepta los datos del procesador-host para la transmisión y proporciona los datos recibidos al procesador host mediante una interfaz SPI (*Serial Peripheral Interface*). El sistema host, el coche AWS DeepRacer o un Arduino como veremos, lee y escribe del DW1000 mediante el SPI.

Los distintos estados en los que se puede encontrar el módulo DW1000 se describen a continuación [39]:

- Estado **OFF**: En este estado el módulo se encuentra completamente apagado, no tiene ningún voltaje o corriente aplicado en los pines de alimentación-
- Estado **WAKEUP**: El oscilador de cristal se habilita en cuanto se cambia a este estado y después de aproximadamente 4 ms se habilitará el LDO (*Low Drop-Out Voltage Regulator*) y la salida del DW1000 se desbloqueará habilitando el paso al siguiente estado.
- Estado **INIT**: El circuito comienza a funcionar siguiendo la frecuencia marcada por el oscilador a 19,2 MHz, a este reloj se le llama XTI.
- Estado **IDLE**: Se mantiene en espera en este estado con la interfaz SPI habilitada por si recibe instrucciones del host, una vez las recibe pasa al estado TX o RX según sea necesario.
- Estado **TX**: El módulo transmite una trama con todo el contenido que se encuentre en el buffer, en el canal de RF indicado y con los parámetros requeridos. Una vez se completa la transmisión de la trama, el módulo puede pasar a tres estados, lo normal sería que pasara al estado IDLE y se mantuviera en espera, pero si hay un

bit (ATXSLP, After TX Automatically Sleep) concreto activado pasaría al estado **SLEEP** o **DEEPSLEEP**.

- Estado **RX**: En recepción, espera la llegada de un preámbulo (*Preamble*) seguido del SFD (*Start of Frame Delimiter*) y una vez lo recibe a continuación llega el PHR (*PHY Header*). A continuación, decodificará toda la información posterior al PHR hasta que finalice la trama o haya algún problema. En cualquiera de los dos casos, pasaría posteriormente al estado IDLE de nuevo excepto, como pasaba en el estado RX, si está activado el bit ARXSLP (*After RX Automatically Sleep*) el módulo pasaría al estado SLEEP o DEEPSLEEP.
- Estado **SLEEP**: Si se entra en este estado, se mantiene únicamente un oscilador funcionando por si fuera necesario pasar al estado WAKEUP, por lo tanto, el consumo del módulo mientras esté en este estado es muy bajo.
- Estado **DEEPSLEEP**: Todos los dispositivos internos del módulo se apagan en este estado, menos la memoria *always-on* para mantener la información necesaria en caso de ser necesario una restauración o volver a despertar el módulo. Este estado, es válido para mantener el dispositivo en un reposo casi absoluto y conseguir un consumo de energía muy bajo.
- Estado **SNOOZE**: Es similar al estado INIT con la excepción de que un contador se mantiene activo, para poner el módulo de nuevo en el estado RX cuanto este expire.

Vistos los estados en los que se puede encontrar el módulo, pasaremos a explicar cómo se realiza la transmisión y la recepción:

- Transmisión (TX): Correspondiente a lo que se realiza en el estado TX, y esta es una de las funciones básicas del módulo DW1000. La trama que se transmite tiene los elementos que vemos en la Figura 39:
 - El preámbulo (*Preamble*), con una longitud entre 64, 1024 y 4096 símbolos en función de la frecuencia de funcionamiento del sistema, se encarga de sincronizar el transmisor y el receptor.
 - SFD, cuya longitud puede ser 8 o 64 símbolos en función de la tasa de transmisión deseada, indica el final del preámbulo y marca los símbolos en los que la trama transmitida contendrá información o un intervalo de guarda.
 - PHR, con una longitud de 19 bits, indica la tasa de transmisión al receptor, la longitud de la trama, duración del preámbulo, alcance del paquete y un código que puede ser utilizado para corrección de errores.
 - Por último, contiene los datos guardados en el buffer para su transmisión (Data), con una longitud hasta de 127 octetos.



Figura 39. Elementos de una trama en el módulo DW1000

La transmisión empieza en el estado IDLE como se ve en la Figura 40, una vez se escriben los datos que se necesiten en el buffer de transmisión y se han configurado los parámetros necesarios, como la longitud del preámbulo o la tasa de transmisión, se inicializa la transmisión poniendo el bit TXSTRT (*Transmit*

Start) a 1. Se manda la trama completa descrita anteriormente y una vez se finaliza la transmisión se indica poniendo a 1 el bit TXFRS (*Transmit Frame Sent*) y el módulo volvería al estado IDLE.

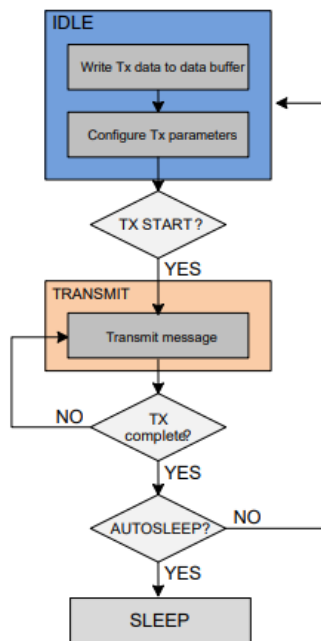


Figura 40. Secuencia de transmisión del módulo DW1000

- **Recepción (RX):** Antes del inicio de la recepción se está buscando constantemente detectar un preámbulo, mediante el análisis de los símbolos recibidos, todo esto tras haber configurado los parámetros necesarios para detectar los preámbulos en el estado IDLE como vemos en la Figura 41. La detección del preámbulo, ya en el estado RX, se realiza calculando la correlación cruzada de algunos trozos del propio preámbulo recibido, mediante esta correlación se determina la longitud del preámbulo recibido para poder separar correctamente el preámbulo del SFD que vendrá a continuación. Todo este proceso debe realizarse en un tiempo previamente determinado, ya que si no se detecta el final del preámbulo en ese tiempo se descartará la recepción de la trama y se volverá al estado IDLE.

Una vez detectado el SFD, mediante una secuencia final introducida en el preámbulo además de la longitud anteriormente estimada, marca el cambio al dispositivo entre la modulación del preámbulo a la modulación BPM/BPSK utilizada para el resto de la trama. También se incluye un temporizador para esta parte de la recepción, si no finaliza la recepción en un tiempo establecido se volvería al estado IDLE como con el preámbulo.

Tras finalizar la recepción del SFD, se obtiene del PHR la información relativa a la longitud de los datos introducidos en la trama, y la tasa de datos necesaria para la demodulación de los mismos. Tras obtener esta información se comenzaría la decodificación de los datos mediante un decodificador de Viterbi, para posteriormente pasar estos datos decodificados por otro decodificador Reed-Solomon para aplicar alguna corrección a mayores, si fuera posible.

Finalmente, si se ha terminado correctamente la recepción, se indicaría al host poniendo a 1 los bits RXDFR (*Receiver Data Frame Ready*) y RXFCG (*Receiver Frame Check Sequence Good*), en caso negativo si ha vencido el temporizador previamente marcado para esta decodificación se cancelaría la recepción y se volvería al estado IDLE como en los casos anteriores.

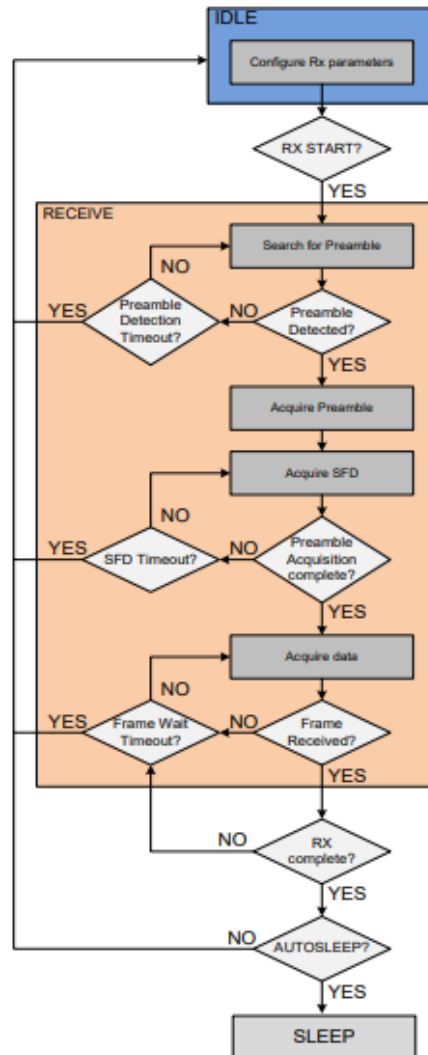


Figura 41. Secuencia de recepción del módulo DW1000

El sistema utiliza la transmisión y recepción de mensajes para calcular la posición de la estación móvil. Para ello implementa algoritmos que son *Time Difference of Arrival* (TDOA) y *Time of Flight* (TOF). Antes de pasar a explicarlos se mencionarán algunas consideraciones necesarias para la localización. Para localizar un objeto móvil, se necesita un número mínimo de estaciones fijas: en caso de ser en dos dimensiones serían necesarias tres estaciones fijas. Además, es necesario que estas estaciones fijas estén lo suficientemente cerca del objeto móvil como para comunicarse; el aumento de la distancia y la tasa de transmisión elegida, pueden aumentar el tamaño del preámbulo en el mensaje. Todas estas consideraciones se deben tener en cuenta a la hora de colocar las estaciones fijas.

El algoritmo TOF, requiere comunicación bidireccional desde el objeto móvil a las estaciones fijas, mediante la cual se intercambian mensajes periódicos utilizados para

medir el retardo durante la propagación entre el origen y el destino. Este tiempo multiplicado por la velocidad de la luz, nos indicaría la distancia a la que se encuentra la estación fija de la estación móvil.

En el algoritmo TDOA la estación móvil manda mensajes (*blinks*) de forma periódica a las estaciones fijas, con estos mensajes todas las estaciones sincronizan sus relojes de funcionamiento para que las mediciones del tiempo de llegada de los mensajes sean comparables. Tras realizar la sincronización de los relojes, por cada par de estaciones fijas la diferencia de tiempo en la llegada de los mensajes (*blink*), define una superficie en la que debería encontrarse el objeto móvil que ha enviado el mensaje.

Desde el punto de vista de consumo energético es mejor TDOA, ya que la estación tiene que mandar un único mensaje para ser localizado, mientras que con TOF tiene que mandar y recibir varios mensajes con varias de las estaciones además de necesitar conocer que estaciones se encuentran cerca para identificarlas y direccionar los mensajes correctamente. TDOA sería más adecuado para situaciones con una gran densidad de estaciones móviles en funcionamiento, mientras que TOF es bueno para situaciones que requieran una detección a distancias más cortas, especialmente cuando las estaciones involucradas son todas móviles, aunque en nuestro caso concreto solo se admite una estación móvil por la configuración de los módulos adquiridos, además según el fabricante los módulos adquiridos utilizarán TDOA.

4.3. Configuración de los módulos UWB

El módulo M5Stack se configura mediante comandos AT, los cuales se componen de una cadena de datos, que comienzan con el prefijo AT y se mandan a un módulo de comunicación para la programación o configuración del mismo. Estos comandos se pueden mandar al módulo gracias al adaptador USB-UART mencionado anteriormente, los comandos necesarios para configurar los módulos son los siguientes:

- *AT*: Es el comando que ejecuta la función de test, el módulo devuelve un OK en caso de tener un funcionamiento normal.
- *AT+Switchids*: Activa la medida de la distancia si se manda un 1, en caso de mandar un 0 el módulo dejaría de devolver la distancia.
- *AT+Interval*: Indica cuantas veces se deben obtener datos de las estaciones fijas, para comenzar a calcular el valor de la distancia, cuanto más grande sea este valor, más lenta es la frecuencia de refresco.
- *AT+Version*: Si se manda este comando, el módulo devuelve el fabricante, número de serie del módulo y el número de versión.
- *AT+RST*: El módulo se resetea completamente al recibir este comando, necesario mandárselo cada vez que se quiera poner en marcha el sistema para un correcto funcionamiento.
- *AT+Anchor_tag*: Se pueden mandar dos valores en este comando, un 1 y se configuraría como una estación fija (*anchor*), en el caso de las estaciones fijas además de mandar el modo en el que funcionan hay que mandar el ID que se le asignará con un valor entre 0 y 3, ya que solo puede haber 4 estaciones fijas en el sistema. Si se manda un 0, se configurará como una estación móvil (*tag*), como

solo se puede configurar una estación móvil en el sistema no es necesario incluir un ID.

Vistos los posibles comandos AT que soportan los módulos, a continuación, se indicarán los comandos utilizados para configurar las estaciones fijas y la estación móvil en nuestro caso:

1. Estaciones fijas (*anchor*):

```
s.write(bytes("AT+RST\r\n".encode()))
time.sleep(1)

for i in range(1):
    time.sleep(0.05)
    s.write(bytes("AT+anchor_tag=1,0\r\n".encode()))
    time.sleep(0.05)
time.sleep(2)
```

En primer lugar, se reinicia el módulo para eliminar la configuración anterior y dejar el módulo en “blanco” para la nueva. Tras el reinicio y esperar un segundo, se manda dos veces el comando *AT+anchor_tag*, se manda dos veces ya que el fabricante indica que es conveniente hacerlo así para una configuración correcta del módulo. Finalmente, se esperan 2 segundos a que el módulo finalice su calibración y comience a funcionar correctamente.

2. Estación móvil (*tag*):

```
s.write(bytes("AT+RST\r\n".encode()))
time.sleep(1)

for i in range(1):
    time.sleep(0.05)
    s.write(bytes("AT+anchor_tag=0\r\n".encode()))
    time.sleep(0.05)
    s.write(bytes("AT+interval=5\r\n".encode()))
    time.sleep(0.05)
    s.write(bytes("AT+switchdis=1\r\n".encode()))
    time.sleep(0.05)
time.sleep(2)
```

Con la estación móvil el procedimiento es similar, aunque es necesaria una configuración más compleja. En primer lugar, se reinicia el módulo para dejar el módulo en “blanco”, posteriormente se marca con *AT+anchor_tag* el valor 0 indicando que es la estación móvil, por lo tanto, es necesario indicar dos comandos más, con *AT+Interval=5* le indica el intervalo de refresco del valor de la distancia que calcula de las estaciones fijas, además de incluir el comando *AT+switchdis=1* para activar la medida de la distancia de este módulo. Se espera dos segundos para finalizar la configuración del módulo.

4.4. Despliegue de los módulos UWB

El despliegue de los módulos para las pruebas se realizó de la siguiente manera: Dado que el sistema solo soporta una estación móvil (*tag*) se fabricó un soporte en el coche AWS Deepracer para poder colocar el módulo en una posición vertical, favoreciendo la propagación de la señal de la antena, como se ve en la Figura 42, ya que, según la hoja de especificaciones del módulo, la posición más adecuada para el mejor funcionamiento de la antena es una posición vertical [23].

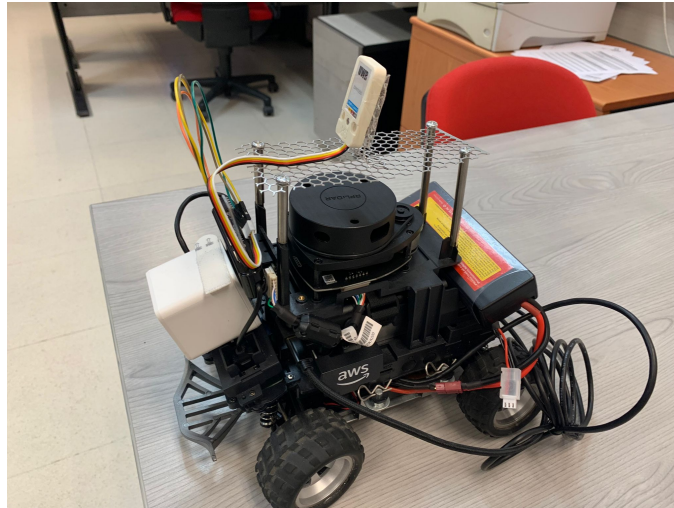


Figura 42. Montaje módulo UWB (*tag*) en el coche AWS Deepracer

En cuanto a las estaciones fijas (*anchors*), para poder posicionarlas donde fuera necesario, se realizó el montaje que se observa en la Figura 43. Se realizó el montaje de los módulos en conjunto con un Arduino UNO, que implementa el script de configuración visto anteriormente para las estaciones fijas, todo esto junto con un adaptador de dos pilas en serie que proporcionarían 7,4 voltios y 9800 mAh, suficiente para alimentar el Arduino que a su vez alimenta el módulo UWB.

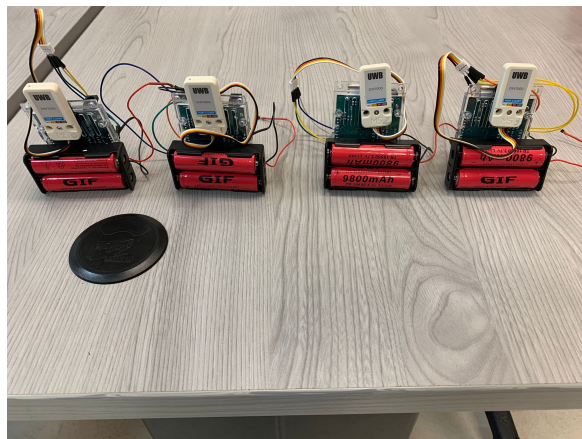


Figura 43. Montaje módulos UWB (*anchors*)

La alimentación del Arduino se hace a través del pin *Vin* de la placa, ya que acepta valores entre los 6 y 12 voltios, sin embargo, debe realizarse cuidadosamente esta conexión ya

que en caso de una caída de tensión no hay un diodo de protección para evitar daños en la placa, como si ocurriría si alimentásemos el Arduino por la entrada de jack o el puerto USB.

La información obtenida por la estación móvil es leída por el coche AWS Deepracer mediante el adaptador UART/USB, esta información se publica mediante un publicador en un nodo ROS con un formato de array de tamaño 4 con las distancias a cada una de las estaciones fijas.

La publicación en el nodo ROS se recoge mediante un suscriptor en otro nodo ROS, que al recibir la información la introduce en un socket UDP, que manda la información al equipo al que se encuentra conectado el coche al igual que se hacía con la información relativa a la velocidad en el apartado anterior.

Una vez recibida la información respectiva a las distancias a las estaciones obtenidas por el módulo UWB, se calcula la posición de la estación móvil (*tag*) mediante un script en Python que aplica un algoritmo de *multilateration* mediante el cual, habiendo recibido las distancias a las estaciones fijas, calcula cual es la posición más probable en la que se encuentra la estación móvil, realizando la intersección de las circunferencias formadas con centro las posiciones de las estaciones fijas (*anchors*) y radio la distancia a la estación móvil (*tag*) como se ve en la Figura 44.

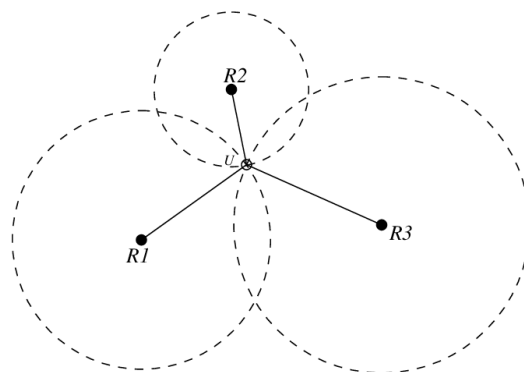


Figura 44. Método de *multilateration* utilizado para el cálculo de las posiciones con los módulos UWB

Las estaciones fijas se colocaron en las esquinas de una maqueta, con dimensiones 3,5 metros x 3,5 metros que se puede ver en la Figura 45. Las estaciones fijas como se mencionó anteriormente son identificadas con un ID de 0 a 3. La estación con ID 0 se encontraría en la posición, que denominaremos origen de coordenadas, [0,0] en la esquina inferior izquierda, la estación con ID 1 se encontrará en el punto [3.5,0] en la esquina inferior derecha, mientras que las dos estaciones restantes con IDs 2 y 3 se encontrarán en las esquinas superiores en los puntos [3.5,3.5] en caso de la estación con ID 2 y [0,3.5] para la estación con ID 3.



Figura 45. Imagen de las pruebas con los módulos UWB

4.5. Resultados módulos UWB

Una vez colocadas las estaciones fijas (Cuadrados azules en la Figura 45), se realizó una vuelta con el coche siguiendo el óvalo exterior (Línea roja punteada en la Figura 45), como se ve en la Figura 45. Durante la vuelta se fue calculando la posición del coche, en función de las distancias a las estaciones fijas, cada vez que se recibió información del módulo UWB, en la Figura 46 podemos observar que la precisión a lo largo de la vuelta es insuficiente, esta imprecisión a la hora de medir la posición viene determinada por el error que marca el fabricante de aproximadamente 10 cm en cada uno de los módulos, es decir, cada una de las medidas de las distancias a las estaciones fijas (*anchors*), puede introducir un error de 10 cm, en un escenario en el que manejamos un circuito con un ancho de la línea de guiado para el vehículo (azul en este caso) es de 2,5 cm y el carril en el que se sitúa esta línea es de 60 cm de ancho.

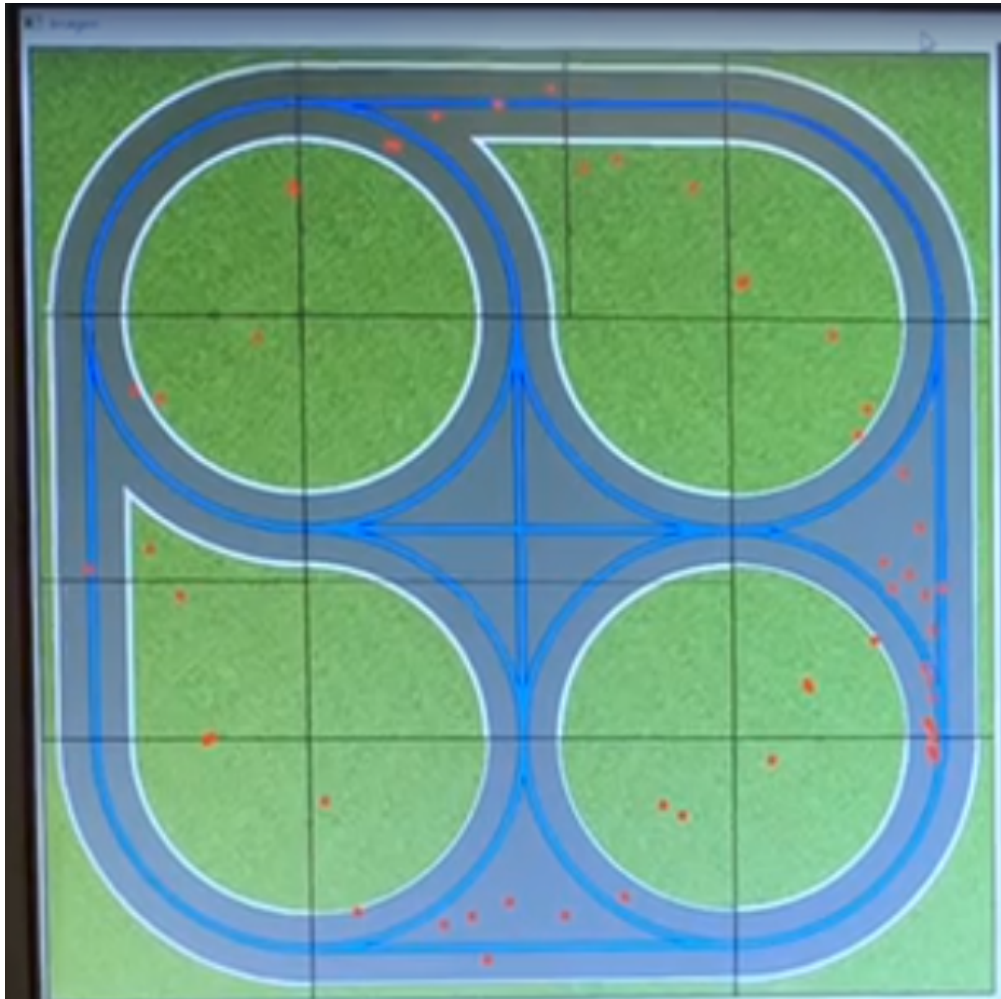


Figura 46. Resultado de la vuelta con las balizas UWB

Otro problema añadido es la frecuencia de recepción de información de las estaciones fijas (*anchors*) en la estación móvil (*tag*). Esta frecuencia es baja ocasionando que la posición que se calcula en el servidor vaya con retraso con respecto a la posición real del vehículo. Esta frecuencia no se puede aumentar más, ya que, el valor $AT+Interval$ es el que marca esta frecuencia y está fijado en un valor de 5 que es el menor valor posible que se le puede dar, es decir, la frecuencia a la que trabaja es la más alta posible.

Además, hay que mencionar que este sistema solo permite localizar una estación móvil a la vez en el mismo lugar, es decir, únicamente podemos tener funcionando a la vez 4 estaciones fijas (*anchors*) y una única estación móvil (*tag*) ya que el *firmware* que traen de fábrica estos módulos, no permite un mayor número de estaciones simultáneas por utilizar el mismo canal de comunicación para todos los módulos, por lo tanto, si se pusieran en funcionamiento más de una estación móvil a la vez se producirían interferencias y el sistema no funcionaría. Este sería otro punto negativo, ya que, aunque las pruebas se hayan realizado con único vehículo, en un futuro se harán pruebas con múltiples vehículos, teniendo que estar estos localizados dentro de la maqueta.

Capítulo 5. Posicionamiento *Indoor*: Módulos Marvelmind Indoor Positioning System

5.1. Introducción

Este sistema de posicionamiento *indoor* se plantea como una opción menos económica, pero que ofrece unas prestaciones mucho mejores, según el fabricante, al utilizar la tecnología de ultrasonidos en lugar de la UWB. Al igual que los módulos UWB, el objetivo de estas balizas es conseguir la posición del vehículo dentro de la maqueta utilizada.

La estructura de este capítulo es similar al anterior, comenzando con un análisis del funcionamiento de las balizas de ultrasonidos, siguiendo con la configuración y despliegue de las mismas, finalizando con los resultados obtenidos y una breve comparativa con las balizas UWB.

5.2. Análisis Módulos Marvelmind

En este apartado analizaremos el sistema de posicionamiento *indoor* de Marvelmind, el segundo de los sistemas de posicionamiento integrados en el vehículo AWS DeepRacer en este Trabajo de Fin de Máster.

El sistema de posicionamiento *indoor* diseñado por Marvelmind Robotics ofrece precisiones de $\pm 2\text{cm}$ y su uso principal es en la localización de robots autónomos, vehículos guiados de forma autónoma (AGV) y otros dispositivos móviles dentro de naves industriales y recintos de características similares.

El despliegue de este sistema está compuesto por una serie de balizas fijas interconectadas vía radio en una banda de acceso libre, junto con una o más balizas móviles instaladas en los objetos que van a ser localizados. Para realizar la localización las balizas incorporan 5 altavoces funcionando en frecuencias de ultrasonidos. Además, el sistema incorpora un *router* que se conecta a un PC para visualizar la información de la posición en el *software* proporcionado por el fabricante. En la Figura 47 podemos ver tanto el *router*, esquina inferior izquierda de la imagen, como las balizas utilizadas para la implementación del sistema.



Figura 47. Componentes del sistema Marvelmind Indoor Positioning System

Este sistema calcula la posición en función del retardo de propagación de los pulsos emitidos por los altavoces entre las balizas móviles y fijas (Time-Of-Flight, TOF) utilizando un algoritmo de trilateración.

A continuación, analizaremos los distintos componentes del sistema y explicaremos la arquitectura utilizada durante las pruebas.

- **Balizas *Super Beacon*:** Todas las balizas pueden funcionar como balizas fijas y balizas móviles, esto puede configurarse en el *software* del fabricante de forma muy sencilla. Las que funcionen como balizas fijas, debería montarse en un punto de mayor altura respecto a la que se encuentre el móvil, el coche en nuestro caso, intentando maximizar la cobertura de la señal de ultrasonidos emitida por los 5 altavoces que incorpora cada baliza. Las balizas soportan una serie de bandas para la comunicación vía radio: la banda libre destinada a ISM (*Industrial, Scientific and Medical*) a 915 MHz en Estados Unidos, la banda SRD (*Short Range Device*) a 868 MHz y la banda ISM a 433 MHz, ambas dos últimas en Europa. Las balizas trabajan con frecuencias de ultrasonidos entre 19 KHz y 45 KHz emitidas por los 5 altavoces que incorporan como vemos en la Figura 48. Las balizas son capaces de recibir varias frecuencias de ultrasonidos al mismo tiempo, permitiendo implementar distintas arquitecturas.

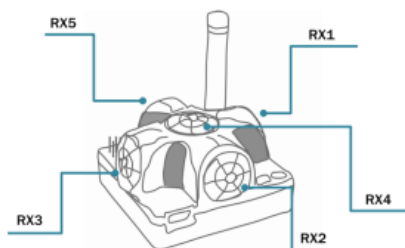


Figura 48. Esquema de altavoces de una baliza Super Beacon en el sistema Marvelmind

- **Balizas móviles *Super Beacon*:** Las balizas móviles también se nombran *hedgehog* y como se ha mencionado anteriormente, cualquier baliza puede actuar tanto como fija como móvil. Estas balizas serían las que colocaríamos en el vehículo o robot a monitorizar, que se recomiendan colocar en una posición horizontal para proporcionar una cobertura para la señal de ultrasonidos óptima.

La tasa a la que la baliza móvil actualiza su posición al sistema depende del número de balizas móviles utilizadas y de la arquitectura utilizada.

La información de la posición de las balizas móviles se puede obtener de dos formas, mediante una conexión usb a la baliza móvil o mediante otra conexión usb al *router* del sistema. La información de la posición se transmite en formato GPS (NMEA 0183).

- Modem HW v5.1: El modem es el controlador central del sistema, se debe conectar al equipo que incorpore el *software* del *dashboard* para monitorizar el sistema y configurarlo. La localización del modem se debe elegir de forma que tenga cobertura vía radio respecto a alguna de las balizas, pudiendo tener un rango de alcance en las frecuencias de 915 MHz de hasta 400 metros en condiciones ideales, utilizando una tasa de transmisión baja de 38 kbps.

El modem utiliza un protocolo de radio propio para la sincronización del modem y las balizas, pudiendo funcionar, al igual que las balizas, tanto a 915 MHz como a 433 MHz y 868 MHz.

Utilizando estos componentes, se pueden utilizar diferentes arquitecturas en función de los requisitos del sistema a implementar. Las arquitecturas son las siguientes:

- *Non-Inverse Architecture* (NIA): Se pueden monitorizar entre 1 y 4 robots pudiendo tener en el sistema, entre balizas móviles y fijas, hasta 250 en total. En esta arquitectura la frecuencia de refresco de la posición depende del número de estaciones móviles con una relación de f/n , siendo n el número de estaciones móviles y f la frecuencia de refresco de la posición, para lo cual utiliza TDMA (*Time-Division Multiple Access*). En esta arquitectura en concreto, las únicas balizas que transmiten ultrasonidos a la hora de realizar la localización son las móviles, las balizas fijas reciben estas señales y transmiten al modem vía radio los datos necesarios para el cálculo de la posición. Las balizas fijas solo transmiten ultrasonidos en el momento de inicio del sistema y de la construcción del mapa. Una de las consideraciones importantes a tener en cuenta, es que debe haber línea de visión directa entre la estación móvil y dos o más estaciones fijas, a una distancia de unos 30 metros entre la fija y la móvil.

La frecuencia de ultrasonidos utilizada en este caso es de 31 KHz tanto en transmisión como en la recepción, mientras que vía radio se puede utilizar cualquiera de las frecuencias disponibles (433 MHz, 868 MHz y 915 MHz). En esta configuración se pueden alcanzar tasas de refresco de la posición de 25 Hz.

- *Inverse Architecture* (IA): Esta arquitectura permite monitorizar un mayor número de robots/drones, aunque el número máximo de balizas sigue siendo de 250 entre balizas móviles y fijas. En este caso, la tasa de refresco no depende del número de balizas móviles utilizadas, ya que no se transmite en una única frecuencia de ultrasonidos, pero esta arquitectura es más sensible a interferencias en entornos más ruidosos. Todas las balizas del sistema transmiten y reciben frecuencias de ultrasonidos en todo momento, menos la baliza móvil que únicamente recibe las señales de ultrasonidos, por lo tanto, funciona de manera opuesta a las balizas móviles en la arquitectura NIA. El resto de las características de la arquitectura son los mismos que en la arquitectura NIA, con una tasa de refresco de la posición de 25 Hz, una distancia entre balizas fijas de hasta 30

metros y necesidad de visión directa de la baliza móvil con mínimo dos balizas fijas estando a un máximo de 30 metros de cada una de ellas.

En cuanto a la comunicación vía radio, utiliza las mismas frecuencias que en el caso anterior (433 MHz, 868 MHz y 915 MHz) en función de las necesidades del despliegue.

Vistas las diferentes arquitecturas, la seleccionada para el proyecto ha sido la NIA, ya que es más sencilla en su configuración y en nuestro caso únicamente vamos a utilizar una baliza móvil para las pruebas de este sistema de localización.

5.3. Configuración de las balizas de ultrasonidos Marvelmind

Para realizar la configuración de las balizas y el modem para funcionar con la arquitectura NIA, es necesario el *software* que proporciona la empresa Marvelmind denominado *dashboard*. Con este *software* se instala el *firmware* necesario en los componentes del sistema, ya que dependiendo de la arquitectura se instala un *firmware* diferente.

Una vez instalado el *firmware*, se debe configurar cada una de las balizas introduciendo una dirección concreta del dispositivo, seleccionando un número entre 1 y 254 para identificar la baliza en el sistema. Además, se debe introducir manualmente la altura a la que se encontrará la baliza fija para lograr un posicionamiento correcto dentro del mapa generado. Podemos seleccionar manualmente la frecuencia por la que queremos que se comuniquen las balizas y el móvil vía radio y la frecuencia de ultrasonidos que utilizará para el posicionamiento. También es importante configurar todas las balizas para que trabajen con la misma tasa de transmisión, cuanto mayor sea este valor, menor será la distancia que podemos separar las balizas. Los valores disponibles se encuentran entre 38 Kbps y 500 Kbps (38 Kbps, 153 Kbps y 500 Kbps), ya que si se configuran erróneamente estos valores el sistema no funcionará correctamente. Por último, hay que seleccionar la tasa de refresco de la posición con valores entre 25 Hz y 0.05 Hz (0.05 Hz, 0.1 Hz, 0.2 Hz, 0.5 Hz, 1 Hz, 2 Hz, 4 Hz, 8 Hz, 12 Hz, 16 Hz y 25 Hz).

En el *software dashboard* el mapa se puede dividir en submapas, cada uno de estos submapas puede contener entre 1 y 4 balizas fijas, estas balizas fijas funcionan en una zona de servicio (*service zone*) propia dibujada por el usuario. Si hay algunas áreas de servicio que se crucen en algún punto se denomina área de *handover*. Esta zona marca la transición para que las balizas móviles pasen a formar parte de otro submapa, realizando esta transición de forma progresiva y necesitando que estas zonas de *handover* sean lo suficientemente grandes como para que a la baliza móvil le dé tiempo a actualizar la posición por lo menos 4 veces.

El caso que se estudiará en el apartado de resultados únicamente requerirá un submapa, ya que las pruebas se realizarán en una habitación pequeña y no se requiere una cobertura mayor del sistema.

Una vez configuradas correctamente tanto las balizas como el modem, se procedió a la instalación de la baliza móvil en el vehículo AWS Deepracer tal y como se ve en la Figura 49.



Figura 49. Montaje baliza Marvelmind en el coche AWS Deepracer

Para la lectura de los datos de la baliza móvil, la empresa facilita un paquete de ROS que permite leer esta información ya decodificada y con un formato de coordenadas con posiciones en el eje X, Y y Z. El paquete de ROS instalado en el coche AWS Deepracer, no permite introducir ningún mensaje ni información además de la que venga configurada de serie, por lo tanto, fue necesario desplegar otro sistema ROS en el coche para poder ejecutar el paquete de ROS para las balizas Marvelmind. Entonces el coche tiene dos paquetes ROS corriendo en paralelo, pero aunque estos paquetes no pueden utilizar los mensajes que tengan diferentes entre sí, por ejemplo, el paquete ROS del coche no puede utilizar el mensaje publicado por el paquete ROS de las balizas Marvelmind que lleva la información de la posición, sin embargo, sí que puede utilizar los mensajes que tienen ambos sistemas en común, estos mensajes son los de uso genérico como mensajes cuyos datos son un entero (*int*), una cadena (*string*) o una matriz. Por lo tanto, para poder facilitar la información del paquete de ROS de las balizas al paquete ROS del coche, se creó un subscriptor al tópico *hedge_pos* que contiene un mensaje con la información de la posición en formato [posición_eje_x, posición_eje_y, posición_eje_z].

Tras leer esta información con el subscriptor, se publica con el mismo nodo en un tópico en el paquete ROS del coche, utilizando un mensaje genérico *Float64MultiArray* manteniendo el formato anteriormente mencionado, como un vector con las posiciones en los diferentes ejes. Una vez publicada esta información ya en el paquete de ROS del coche, se realizó el mismo proceso que con los dos sistemas anteriores, se añadió esta información al socket UDP para enviarlo al ordenador que gestiona el sistema del vehículo.

5.4. Despliegue de las balizas de ultrasonidos Marvelmind

La prueba realizada con este sistema ha sido la siguiente: utilizando la misma pista que en el caso de los módulos UWB, se han colocado 4 balizas fijas en las paredes de la habitación a una altura de 2,4 metros, que junto con la baliza móvil colocada en el coche como se ve en la Figura 49 y el *router* conectado al ordenador que ejecuta el *software* del *dashboard* para la gestión del sistema. Todo este esquema se puede ver en la Figura 50,

en círculos azules se ven las balizas fijas y con un círculo rojo se ve el modem que controla las balizas Marvelmind, además de la baliza móvil montada en el vehículo y el circuito utilizado en el resto de pruebas.



Figura 50. Imagen de las pruebas para las balizas Marvelmind.

La ruta establecida para el coche AWS DeepRacer es la misma que en el caso de las balizas UWB, seguir el óvalo exterior del circuito para poder realizar la comparativa entre ambos sistemas. (Incluir imágenes de los resultados de las balizas de ultrasonidos) Los parámetros utilizados en las balizas en las pruebas han sido los siguientes:

- Frecuencia de ultrasonidos: 31 KHz, al utilizar la arquitectura NIA, únicamente es necesaria una frecuencia de ultrasonidos que transmitirá la baliza móvil a las balizas fijas para el cálculo de la posición.
- Frecuencia de radio: 868 MHz, en nuestro caso utilizamos esta banda ya que es la destinada a este tipo de sistemas en Europa.
- Tasa de transmisión: 38 Kbps ya que al utilizar únicamente una baliza móvil con esta tasa de transmisión es suficiente, a medida que se fueran incluyendo más balizas móviles y aumentando la tasa de refresco de la posición sería necesario aumentar también este valor.
- Tasa de refresco de la posición: Seleccionamos el valor de +16Hz ya que se busca controlar el sistema en tiempo real, por lo tanto, buscamos la mayor tasa de refresco posible que permite el sistema.

5.5. Resultados balizas de ultrasonidos Marvelmind

Configurados todos los parámetros necesarios en las balizas del sistema y colocadas en las posiciones establecidas en la habitación, para poder rastrear la posición de la baliza

móvil es necesario congelar el submapa en el que se encuentra la estación móvil, lo cual establece las distancias entre las balizas fijas como válidas y permite despertar la baliza móvil estableciendo el modo *hedgehog* en la baliza móvil, lo que permite el movimiento de la misma en el submapa “congelado”.

Como podemos ver, la precisión en el caso de las balizas de ultrasonidos es mucho más adecuada para el circuito utilizado, ya que las desviaciones son mínimas y permitiría localizar adecuadamente el vehículo en los cruces y las intersecciones del circuito.

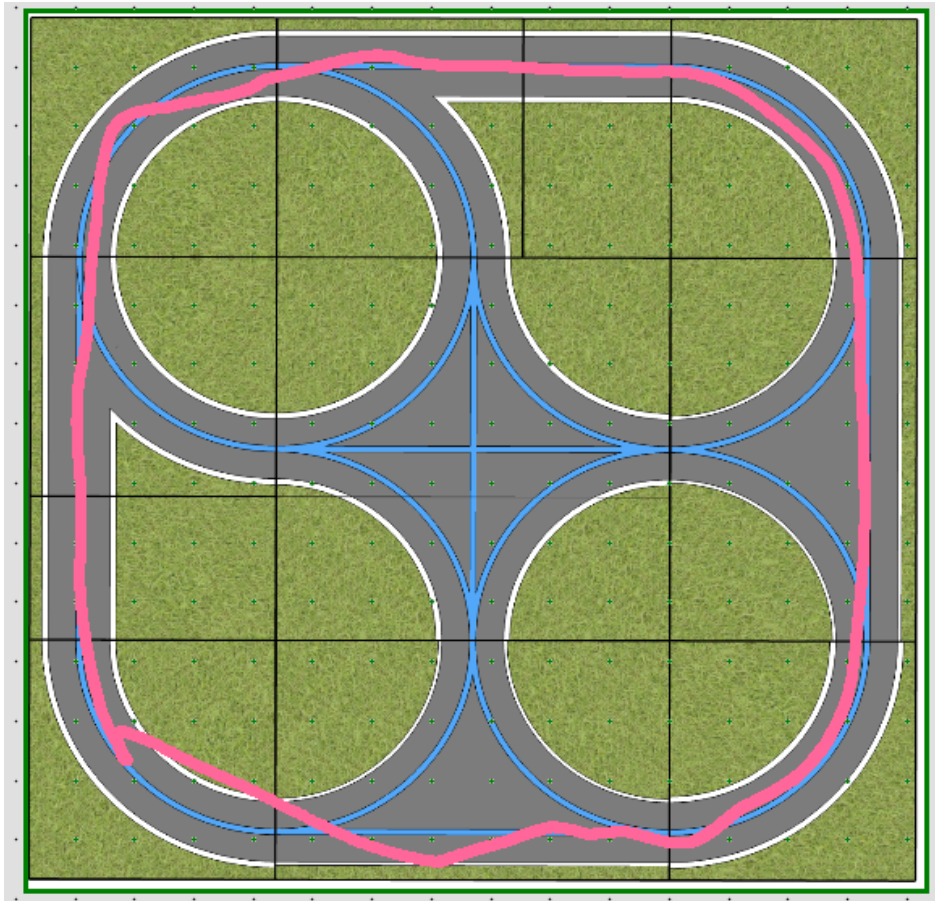


Figura 51. Resultados obtenidos al realizar el recorrido con las balizas Marvelmind

Comparándolo con el sistema de localización con las balizas UWB, podemos mencionar los siguientes puntos:

- **Precisión del sistema:** Como se ve a simple vista comparando las trayectorias obtenidas con ambos sistemas, las balizas Marvelmind otorgan una precisión mucho mayor, además de una tasa de actualización de la posición mayor. Este resultado era previsible, ya que las precisiones que podían otorgar ambos sistemas son muy diferentes, ± 2 cm en caso de las balizas Marvelmind y ± 10 cm en caso de las balizas UWB.
- **Configuración del sistema:** A pesar de ser un sistema más complejo en su funcionamiento, las balizas Marvelmind se configuran de una manera mucho más sencilla que los módulos UWB, el motivo principal es que el *software dashboard* te otorga todas las funcionalidades necesarias para el sistema y permite una configuración sencilla del mismo, además de contar con una documentación muy

extensa y simple de seguir, mientras que los módulos UWB cuentan con una documentación más complicada de utilizar y la configuración no viene facilitada por ningún *software*, requiere la programación de los módulos manualmente mediante scripts de python.

- Despliegue del sistema: Las balizas Marvelmind se despliegan de manera muy rápida, requiriendo únicamente encenderlas y pegarlas en la pared a una altura determinada en caso de las fijas o fijarlas al vehículo en caso de las móviles. Las balizas UWB fijas (*anchors*) por otro lado, requieren colocarlas siempre en las mismas posiciones, ya que el script que calcula la posición de la estación móvil (*tag*) requiere que se introduzcan previamente las posiciones de las estaciones manualmente indicando las distancias entre ellas, lo que otorga una flexibilidad al sistema muy baja. Además, con las balizas UWB es necesario ser muy cuidadoso con la colocación en vertical de las balizas, ya que la precisión se verá gravemente afectada si no están en una posición completamente vertical.
- Integración de las balizas/estaciones con el vehículo AWS DeepRacer: El caso de las balizas UWB la integración ha sido más compleja ya que además de ser necesario el adaptador USB-UART para su funcionamiento, fue necesario realizar la conexión y la lectura de datos mediante un script de Python, además de realizar la configuración de ROS a través de un nodo publicador y un nodo subscriber con su correspondiente tópico. Para el caso de las balizas Marvelmind únicamente fue necesario la creación de los nodos ROS, ya que el paquete ROS que gestionaba y transformaba la información venía facilitado por la empresa, por lo tanto, su integración fue mucho más sencilla, además para su conexión al coche solo hacía falta un cable USB-USB microB sin necesidad de añadir ningún adaptador extra al sistema.

Capítulo 6. Conclusiones y Líneas Futuras

En este TFM se han implementado tres sistemas diferentes en el vehículo AWS Deepracer, un odómetro realizado con un sensor *hall* y unos imanes acoplados a la rueda del vehículo, un sistema de posicionamiento *indoor* compuesto por una serie de balizas UWB y otro sistema de posicionamiento *indoor* compuesto por las balizas de ultrasonidos Marvelmind.

En cuanto al odómetro, se ha realizado la selección de un sensor *hall* adecuado para nuestro sistema y unos imanes adecuados para el tamaño de la rueda del vehículo, tras ver cuántos imanes eran viables de incorporar en la rueda se realizó el montaje y la integración en el vehículo.

Realizada esta integración se observó que la distancia y la velocidad calculada incluían un error demasiado grande, por lo tanto, se decidió calcular tanto la distancia y la velocidad de una forma alternativa, obteniendo unos resultados lo suficientemente precisos para su uso en la maqueta usada para las pruebas. El sistema demostró no ser lo suficientemente preciso para localizar el vehículo en el circuito, pero sí servirá para controlar la velocidad y mejorar, por lo tanto, la estabilidad del algoritmo de guiado.

Los módulos de localización *indoor* UWB se integraron en el vehículo para realizar la localización en la maqueta, para realizar esta integración fue necesario realizar una serie de scripts en python tanto para la configuración de los mismos, la lectura de los datos que proporcionaban de la distancia y el cálculo de la posición del módulo instalado en el coche. Una vez integrados, se comprobó que la precisión que ofrecían es insuficiente para el caso estudiado en este Trabajo Final de Máster.

En segundo lugar, se realizó la integración de los módulos de ultrasonidos Marvelmind también para la localización *indoor* del vehículo AWS Deepracer. Estos módulos, junto con el *software dashboard* proporcionado por la empresa presentaron unos resultados mucho mejores, tanto a la hora de obtener una precisión mayor en la medida de la posición, como en su instalación e integración que ha sido mucho más sencilla que con los módulos UWB. Por lo tanto, para la maqueta utilizada en este Trabajo de Fin de Máster, son más adecuadas las balizas de ultrasonidos Marvelmind.

Una vez finalizado el análisis e integración de estos tres sistemas, se proponen las siguientes líneas futuras:

- Fusionar la información de la IMU del vehículo con la rueda, mejorando la información de velocidad y distancia recorrida.
- Utilizar el valor de la velocidad obtenido por el odómetro para realizar un control longitudinal del vehículo con este parámetro, en lugar de hacerlo con el valor del “acelerador” como se realiza actualmente.
- Implementar una arquitectura IA con las balizas Marvelmind, al ser necesario introducir un mayor número de balizas móviles en el sistema.
- La información de las balizas Marvelmind, se podría utilizar para implementar cualquiera de los protocolos del estándar ITS-G5, como por ejemplo el CAM o el DENM.

Referencias

- E. D. S. S. Siegel. Josh, «A Survey of the Connected Vehicle Landscape--
- 1] Architectures, Enabling Technologies, Applications, and Development Areas,» Octubre 2017. [En línea]. Available: https://www.researchgate.net/publication/320220885_A_Survey_of_the_Connected_Vehicle_Landscape--Architectures_Enabling_Technologies_Applications_and_Development_Areas. [Último acceso: 10 07 2022].
 - 2] A. Abunei, C.-R. Comşa y I. Bogdan, «Implementation of ETSI ITS-G5 based inter-vehicle communication embedded system,» 14 Septiembre 2017. [En línea]. Available: <https://ieeexplore.ieee.org/document/8034921>. [Último acceso: 3 Junio 2022].
 - 3] T. Sturm, C. Krupitzer, M. Segata y C. Becker, «A Taxonomy of Optimization Factors for Platooning,» 29 Mayo 2020. [En línea]. Available: <https://ieeexplore.ieee.org/document/9103935>. [Último acceso: 10 Junio 2022].
 - 4] ETSI, «Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2,» Diciembre 2019. [En línea]. Available: https://www.etsi.org/deliver/etsi_tr/103500_103599/103562/02.01.01_60/tr_103562v020101p.pdf. [Último acceso: 12 Junio 2022].
 - 5] Amazon, «AWS Deep Racer,» [En línea]. Available: <https://aws.amazon.com/es/deepracer/>.
 - 6] Arduino, «Arduino Nano,» [En línea]. Available: <https://store.arduino.cc/products/arduino-nano>.
 - 7] NMB-Technologies, «Motor de pasos NMB PM55L,» [En línea]. Available: https://www.jameco.com/z/PM55L-048-ZXX3-NMB-Technologies-2-Phase-Unipolar-Stepper-Motor-24VDC_2209254.html.
 - 8] Allegro Microsystems, «Sensor Hall A1120EUA-T,» [En línea]. Available: <https://es.rs-online.com/web/p/circuitos-integrados-de-sensores-de-movimiento/6807245>.
 - 9] ARCELI, «Controlador Motor de pasos TB6600,» [En línea]. Available: https://www.amazon.es/ARCELI-Controlador-Segmentos-Adecuado-trif%C3%A1sico/dp/B07QDSC2W4/ref=sr_1_2?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=b6600&qid=1632774649&qsid=262-9464675-9018938&sr=8-2&sres=B00BMPEZ08%2CB07QDSC2W4%2CB09DCHK6WY.

Marvelmind Robotics, «Starter Set Super-MP-3D,» [En línea]. Available:
10] <https://marvelmind.com/product/starter-set-super-mp-3d/>.

Marvelmind Robotics, «Manual Marvelmind,» [En línea]. Available:
11] https://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf.

M5 Stack, «Módulos UWB para posicionamiento indoor,» [En línea].
12] Available: <https://shop.m5stack.com/products/ultra-wideband-uwb-unit-indoor-positioning-module-dw1000>.

WINGONEER, «Adaptador UART-USB,» [En línea]. Available:
13] https://www.amazon.es/WINGONEER%C2%AE-CP2104-convertidor-M%C3%B3dulo-compatible/dp/B01CYBHM26/ref=sr_1_1?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=2THZY00NIUDQ9&keywords=CP2104&qid=1636108156&srefix=cp2104%2Caps%2C152&sr=8-1.

L. Batistić y M. Tomic, «Overview of indoor positioning system
14] technologies,» 2 Julio 2018. [En línea]. Available:
<https://ieeexplore.ieee.org/abstract/document/8400090/citations#citations>. [Último
acceso: 27 Julio 2022].

L. Ni, Y. Liu, Y. C. Lau y A. Patil, «LANDMARC: indoor location sensing
15] using active RFID,» 8 Abril 2003. [En línea]. Available:
<https://ieeexplore.ieee.org/document/1192765>. [Último acceso: 26 Julio 2022].

A. Ward, A. Jones y A. Hopper, «A new location technique for the active
16] office,» Octubre 1997. [En línea]. Available:
<https://ieeexplore.ieee.org/document/626982>. [Último acceso: 26 Julio 2022].

Topaz, «Topaz,» 2018. [En línea]. Available: <http://www.tadlys.co.il/>.
17] [Último acceso: 26 Julio 2022].

S. Beauregard, Widyawan y M. Klepal, «Indoor PDR performance
18] enhancement using minimal map information and particle filters,» 3 Septiembre
2008. [En línea]. Available: <https://ieeexplore.ieee.org/document/4570050>.
[Último acceso: 26 Julio 2022].

Ubisense, 2018. [En línea]. Available: [https://ubisense.com/enquiry-
19\] success/](https://ubisense.com/enquiry-success/). [Último acceso: 26 Julio 2022].

S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer y
20] F. Dressler, «Demo: OpenC2X — An open source experimental and prototyping
platform supporting ETSI ITS-G5,» 30 Enero 2017. [En línea]. Available:
<https://ieeexplore.ieee.org/document/7835955/authors#authors>. [Último acceso: 11
Junio 2022].

H. L. & R. K. Jean Cassou-Mounat, «Simulation of Cyberattacks in ITS-
21] G5 Systems,» de *Simulation of Cyberattacks in ITS-G5 Systems*, 2020, p. 3–14.

ETSI, «Intelligent Transport Systems (ITS); Access layer specification for
22] Intelligent Transport Systems operating in the 5 GHz frequency band,» 3 Mayo
2013. [En línea]. Available:
https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.01_30/en_302663v010201v.pdf. [Último acceso: 12 06 2022].

F. P. A. M. A. F. José Santa, «Experimental evaluation of CAM and DENM
23] messaging services in vehicular communications,» 10 Junio 2014. [En línea].
Available:
https://www.sciencedirect.com/science/article/abs/pii/S0968090X14001193?fr=R-R-2&ref=pdf_download&rr=730476bade1369d2. [Último acceso: 15 Junio 2022].

ETSI, «Intelligent Transport Systems (ITS); Vehicular Communications;
24] GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-
multipoint communications; Sub-part 1: Media-Independent Funcionalidad,» 20
Mayo 2014. [En línea]. Available:
https://www.etsi.org/deliver/etsi_en/302600_302699/3026360401/01.02.01_30/en_3026360401v010201v.pdf. [Último acceso: 16 Junio 2022].

ETSI, «TS 102 636-5-1 - V1.1.1 - Intelligent Transport Systems (ITS);
25] Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part
1: Basic Transport Protocols; Sub-part 1: Basic Transport Protocol,» Febrero 2011.
[En línea]. Available:
https://www.etsi.org/deliver/etsi_ts/102600_102699/1026360501/01.01.01_60/ts_1026360501v010101p.pdf. [Último acceso: 17 Junio 2022].

D. Jiang y L. Delgrossi, «IEEE 802.11p: Towards an International Standard
26] for Wireless Access in Vehicular Environments,» 20 Mayo 2008. [En línea].
Available:
https://ieeexplore.ieee.org/abstract/document/4526014?casa_token=3YAPQkr2kRoAAAAA:G13u-EZxxG8QdlbiXpPPhE_cX8P2D6WzS0AG5yL1B5380lvoS0wZi4gLhNcgC3lZR7qQkbcSBA. [Último acceso: 18 Junio 2022].

E. H. D. S. Carl Bergenheim, «Vehicle-to-Vehicle Communication for a
27] Platooning System,» 14 Julio 2012. [En línea]. Available:
<https://www.sciencedirect.com/science/article/pii/S1877042812028340>. [Último
acceso: 22 Mayo 2022].

ACEA, «Platooning Roadmap,» [En línea]. Available:
28] https://www.acea.auto/files/Platooning_roadmap.pdf . [Último acceso: 20 Mayo
2022].

S. Tsugawa, S. Jeschke y S. E. Shladover, «A Review of Truck Platooning
29] Projects for Energy Savings,» Marzo 2016. [En línea]. Available:
<https://ieeexplore.ieee.org/document/7497531>. [Último acceso: 12 Mayo 2022].

- 30] S. Tsugawa, S. Kato y K. Aoki, «An automated truck platoon for energy saving,» 5 Diciembre 2011. [En línea]. Available: <https://ieeexplore.ieee.org/document/6094549>. [Último acceso: 25 Mayo 2022].
- 31] S. Eilers, J. Mårtensson, H. Pettersson, M. Pillado, D. Gallegos, M. Tobar, K. H. Johansson, X. Ma, T. Friedrichs, S. S. Borojeni y M. Adolfson, «COMPANION -- Towards Co-operative Platoon Management of Heavy-Duty Vehicles,» 02 Noviembre 2015. [En línea]. Available: <https://ieeexplore.ieee.org/document/7313300/authors#authors>. [Último acceso: 21 Mayo 2022].
- 32] M. M. M. B. Greg Marsden, «Towards an understanding of adaptive cruise control,» 14 Septiembre 2000. [En línea]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0968090X0000022X>. [Último acceso: 26 Mayo 2022].
- 33] ENSEMBLE, «ENSEMBLE-D2.3_V2-Platooning-use-cases,-scenario-definition-and-Platooning-Levels,» 16 Febrero 2022. [En línea]. Available: https://platooningensemble.eu/storage/uploads/documents/2022/03/14/ENSEMBLE-D2.3_V2-Platooning-use-cases,-scenario-definition-and-Platooning-Levels_FINAL.pdf. [Último acceso: 27 Mayo 2022].
- 34] ENSEMBLE, «ENSEMBLE_D2.5_Final_Version_Functional_specification_for_white-label_truck,» 21 Febrero 2022. [En línea]. Available: https://platooningensemble.eu/storage/uploads/documents/2022/03/14/ENSEMBLE_D2.5_Final_Version_Functional_specification_for_white-label_truck_FINAL.pdf. [Último acceso: 27 Mayo 2022].
- 35] «TR 103 562 - V2.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2,» Diciembre 2019. [En línea]. Available: https://www.etsi.org/deliver/etsi_tr/103500_103599/103562/02.01.01_60/tr_103562v020101p.pdf. [Último acceso: 1 Junio 2022].
- 36] ETSI, «EN 302 636-3 - V1.1.2 - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture,» 18 Marzo 2014. [En línea]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263603/01.01.02_20/en_30263603v010102a.pdf. [Último acceso: 25 Julio 2022].
- 37] G. Thandavarayan, M. Sepulcre y J. Gozalvez, «Generation of Cooperative Perception Messages for Connected and Automated Vehicles,» 6 Noviembre 2020. [En línea]. Available: https://ieeexplore.ieee.org/abstract/document/9250667?casa_token=utQG1QNFQh4AAAAA:7QMtjV0hOhwfd-GOWadXmK3fww6FXzZUQ3ig7UK38FZX5yBY9KBU6IZxHROU-piSvuYOWibRrA. [Último acceso: 25 Julio 2022].

I. R. González, *Diseño e implementación de un testbed de Edge computing para el soporte de vehículos conectados*, Trabajo de Fin de Master, Pendiente de presentación, Universidad de Valladolid, 2022.

Decawave, «DW1000 Datasheet,» Decawave, 31 Marzo 2016. [En línea].
39] Available: <https://uniteng.com/wiki/lib/exe/fetch.php?media=uwb:dw1000-datasheet-v2.09.pdf>. [Último acceso: 15 06 2022].