



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Máster Universitario
en Tecnologías, Sistemas y
Redes de Comunicaciones

Despliegue y Evaluación del Servicio de Conducción Remota sobre una Red Piloto 5G Standalone

Autora: Maritza Elizabeth Palacios Morocho

Director: Jose Francisco Monserrat del Río

Fecha de comienzo: 01/02/2019

Lugar de trabajo: Grupo de Comunicaciones Móviles del iTEAM

Objetivos — Esta Tesina de Máster tiene como objetivo demostrar que gracias a la característica de baja latencia que ofrecen las redes móviles de Quinta Generación (5G) se pueden brindar servicios de conducción remota. Las especificaciones se basarán en el *Release 15* del sistema del 3GPP (*Third Generation Partnership Project*) para redes 5G NR (*New Radio*) *Standalone*. Posteriormente, se realizarán pruebas de campo y se llevarán a cabo ajustes de código para el correcto funcionamiento del control de coche. Estos objetivos generales se pueden desarrollar en más profundidad en los siguientes objetivos secundarios:

- Estudiar las tecnologías de red de 5G.
- Desplegar una red NR, basada en las especificaciones de la *Release 15* del sistema 3GPP que completa el sistema 5G del IMT-2020.
- Evaluar las capacidades de las tecnologías 5G en una red pre-comercial.
- Desarrollar el caso de uso de conducción remota.
- Integrar el control desarrollado dentro de la red pre-comercial 5G desplegada.
- Validar la integración del control del coche y el desarrollo de este.
- Comparar la experiencia de usuario de conducción remota empleando la tecnología 5G con la 4G.

Metodología — Para llevar a cabo los objetivos expuestos, en primer lugar se realizó un estudio de las redes de 5G definidas en la *Release 15* del 3GPP y de los servicios de la capa de aplicación de Vehículo a Todo (V2X), que están especificados en la *Release 16* cuya finalización está planificada para marzo del 2020. De la misma manera se realizó un estudio del sistema operativo robótico (ROS) y de su código fuente para poder desarrollar el caso de uso. Una vez realizado el control del coche de manera remota se procedió a la instalación de dos cámaras IP Hikvision para tener una visión del escenario en el que se realizarían las pruebas. Además, se realizaron pruebas de campo para ajustar los parámetros de la cámara y del código de control. Se incluyeron pruebas de esfuerzo para determinar el consumo energético y asegurar la inexistencia de fallos en el software o hardware tras un largo período de trabajo. Por último, se utilizó un simulador que posee el iTEAM, que permite variar los tiempos de latencia, para encontrar el punto de trabajo en que la experiencia de tele-conducción es impracticable.

Desarrollos teóricos realizados — No existe un desarrollo teórico en esta Tesina de Máster, ya que la misma es claramente experimental. Es importante destacar que el prototipo se presentó en el *Global 5G Event* y EUCNC2019. Sin embargo, sí que hubo que estudiar los fundamentos de las redes de 5G, las redes V2X y de ROS, tecnologías fundamentales en esta Tesina de Máster.

Desarrollo de prototipos y trabajo de laboratorio — Una vez realizada la revisión teórica y el despliegue de la red, se realizaron pruebas de comunicación entre dos máquinas que tenían instalado ROS. Una vez conseguido esto se procedió a conectar un *joystick* PS3 de Linux genérico con ROS. Terminada esta etapa de interconexión se empezaron las pruebas de campo. Por último, se instalaron dos cámaras empleadas para la visualización de la parte frontal y trasera del coche. También se realizaron pruebas para determinar la autosuficiencia energética del coche durante largas horas de uso.

Resultados — Una vez implementada la red, se realizaron las pruebas necesarias para el caso de uso, con lo cual se obtuvo una implementación exitosa del mismo. Con esto se demostró que es posible brindar servicios de conducción remota empleando redes móviles de quinta generación, gracias a su baja latencia y alta fiabilidad.

El caso de uso fue expuesto en el *Global 5G Event* y EUCNC2019 junto a un simulador de realidad virtual perteneciente al iTEAM, en el que se recreaba el mismo escenario físico pero con distintas tecnologías móviles. Se realizó una comparativa entre ellos, mediante una encuesta y se obtuvo como resultado que la experiencia de los usuarios al conducir con la tecnología 4G les resultaba muy complicada, ya que el coche tardaba mucho en reaccionar a los movimientos realizados. En cambio, con la tecnología 5G la respuesta del coche era inmediata.

Cabe mencionar, que en la demostración física de conducción remota se podía observar una latencia mínima en la imagen mostrada en el televisor y se comprobó que era causada netamente por el procesamiento de imagen, el mismo que estaba determinado por la cámara empleada.

Líneas futuras — Para que el usuario tenga una mejor experiencia de los beneficios que ofrecen las características de baja latencia y alta fiabilidad de las redes 5G NR, se propone trabajar en el procesado de imagen de las cámaras. Otra propuesta muy interesante sería trabajar ya no en redes V2N si no en redes V2V de manera que la conducción pase de ser remota a ser totalmente autónoma con comunicación entre vehículos.

Publicaciones — Hasta la fecha, no existe una publicación asociada con el trabajo de esta tesina de máster, aunque se está trabajando en la redacción de un artículo para un congreso científico, resumiendo la experiencia adquirida.

Abstract — Once the network was implemented, we carried out the necessary tests in order to get a successful implementation. This showed that it is possible to provide remote driving services with fifth generation mobile networks, due to their low latency and high reliability.

This case was exposed in the Global 5G Event and EUCNC2019 together with a virtual reality simulator that belongs to iTEAM, in which the same physical scene was simulated but with a number of technologies. A comparison between them, was made by means of a survey and it was obtained as a result that the users' experience when driving with the 4G technology was very complicated, because the car took a long time to react to the steering wheels movements, whereas with the 5G technology the response of the car was immediate and satisfactory.

It is worth mentioning that in the physical demonstration of remote driving, a minimal latency could be observed in the image shown on the screens and it was proved that it was caused purely by the image processing, which relied on the camera used in our experiments.

Autora: Maritza Elizabeth Palacios Morocho email: mapamo3@teleco.upv.es

Director: Jose F. Monserrat del Río, email: jomondel@dc.com.upv.es

Fecha de entrega: 10-09-19

ÍNDICE

I. INTRODUCCIÓN	4
I.1. MOTIVACIÓN	4
I.2. OBJETIVOS	5
I.3. PRINCIPALES RESULTADOS	6
II. ESTADO DEL ARTE	7
II.1. ESTADO ACTUAL DE LAS REDES 5G	7
II.2. ARQUITECTURA Y BANDAS DE FRECUENCIA.....	9
II.3. CASOS DE USO DE LAS REDES 5G.....	11
II.4 SISTEMA OPERATIVO ROBÓTICO	12
II.5. UNITY	13
III. IMPLEMENTACIÓN	14
III.1. ENTORNO DE DESARROLLO DEL SOFTWARE	15
III.2. PROGRAMACIÓN.....	16
III.3. INTERCONEXIÓN DE LA RED CON EL COCHE	23
III.4. CONEXIÓN DE LA CÁMARA AL SISTEMA.....	29
III.5 INSTRUCCIONES AL USUARIO	30
IV. RESULTADOS	31
V. CONCLUSIONES Y LINEAS FUTURAS	37
AGRADECIMIENTOS	38
REFERENCIAS	38

I. INTRODUCCIÓN

A punto de finalizar la segunda década del siglo XXI, los países desarrollados han adoptado las comunicaciones móviles en el centro de sus vidas, en sus relaciones laborales y personales [1].

Las comunicaciones en el mundo han ido evolucionando en función de las necesidades de la población, por lo que el tráfico va incrementando ya que aumenta el número de dispositivos conectados. Esto, obliga a las redes a tener versatilidad, escalabilidad y sobre todo adaptabilidad, de manera que no exista un aumento de latencia al aumentar el número de dispositivos conectados, o que el alto consumo de batería afecte las prestaciones de las mismas [2].

Es por eso, que las telecomunicaciones han ido cambiando cada 10 años. La primera generación (1G) era una tecnología analógica que solo permitía llamadas de voz entre terminales. En cambio, la segunda generación (2G) empleó la tecnología GSM que es digital debido a que se tenía que solventar las necesidades de enviar datos y voz por el mismo dispositivo móvil. Por su parte, la tercera generación (3G), también conocida como UMTS, permitió la navegación en internet usando el dispositivo móvil, además de poder visualizar vídeos en línea. Por último, la cuarta generación (4G - LTE) proporciona una potente tecnología de acceso radio que permite la movilidad y alta transferencia de archivos con una velocidad de más de 100 Mbps y una latencia de unos 50 ms.

Actualmente las redes de 5G se encuentran en su fase final de desarrollo. Se tiene previsto que todos los operadores desplieguen las mismas en el año 2020, aunque actualmente se encuentran desplegadas pequeñas redes pre-comerciales. La nueva generación de telecomunicaciones 5G, representan la revolución tanto de los sistemas inalámbricos como de las telecomunicaciones, abriendo paso a las ciudades inteligentes, facilitando la vida de las personas, permitiendo aumentar la productividad en la industria, y en los sectores públicos gracias a que ofrece velocidades de datos elevadas en comparación con el actual 4G (LTE). En este caso, la 5G es 100 veces más rápida que la 4G, aunque la característica más importante es su baja latencia que teóricamente tendría que ser de 1 ms pero hasta el momento solo se ha podido llegar a 10 ms. A su vez, la 5G garantiza una conectividad más fiable permitiendo que miles de dispositivos se conecten de forma simultánea a una célula. Con 5G podemos dar la bienvenida a la era del Internet de las cosas que nos llevará a un mundo automatizado e inteligente.

I.1. MOTIVACIÓN

Para el 2020 se busca un despliegue total de las redes de quinta generación (5G) que son infraestructuras de comunicaciones, redes de extremo a extremo altamente flexibles, las mismas que proporcionan un mayor rendimiento, latencia, confiabilidad, capacidad y movilidad. Al mismo tiempo satisfacen los requisitos diversificados de múltiples servicios de banda ancha móvil mejorada (*eMBB*), que aborda casos de uso centrados en el acceso al contenido multimedia, servicios y datos. También permite las comunicaciones ultra fiables de baja latencia (*URLLC*), en las que podemos mencionar los casos de uso que a mi parecer son los que más beneficios

conlleven, como lo es la conducción autónoma y la industria automatizada. Así mismo da paso a las comunicaciones masivas de tipo máquina (*mMTC*), que conectan un gran volumen de dispositivos, los cuales transmiten una tasa de datos muy baja. Podemos mencionar entre sus casos de uso la medición, logística y agricultura inteligente.

El *3GPP (3rd Generation Partnership Project)* se encuentra trabajando desde 2016 en la estandarización de la tecnología 5G NR (*New Radio*), que es el estándar global para 5G que garantizará la calidad, el mayor rendimiento y la interoperabilidad de los dispositivos y redes 5G. En junio del 2018 se finalizó la versión para el estándar 5G en redes SA (*Standalone*) en lo que se conoce como la *Release 15*. Actualmente el 3GPP se encuentra trabajando en la *Release 16* que completará el marco del IMT-2020 (*International Mobile Telecommunication for 2020*). La *Release 16*, trata sobre una variedad de temas entre ellos los servicios de la capa de aplicación de vehículo a todo (*V2X*). Dentro del mismo se puede identificar diferentes modos de comunicación como: vehículo a vehículo (*V2V*), vehículo a persona (*V2P*), vehículo a infraestructura (*V2I*) y vehículo a red (*V2N*).

También existen asociaciones que trabajan en conjunto para desarrollar soluciones de extremo a extremo para futuros servicios de movilidad y transporte. Entre ellas podemos citar a la Asociación Automotriz 5G (*5GAA*) que representa la organización mundial de empresas, de las industrias de la automoción, la tecnología y las telecomunicaciones (*TIC*) que apoya la idea, de que 5G será la plataforma definitiva para habilitar los Sistemas Cooperativos de Transporte Inteligente (*C-ITSO*) y la provisión de *V2X*.

Es así que la motivación de esta tesina es demostrar que se puede implementar servicios de conducción remota gracias a las características de baja latencia que ofrece las redes 5G. Cabe mencionar que este caso uso entra en el campo de las redes *V2N* y que esta demostración se realizó dentro del Grupo de Comunicaciones Móviles (*GCM*) del Instituto de Telecomunicaciones y Aplicaciones Multimedia (*iTEAM*) de la Universitat Politècnica de València (*UPV*) en el primer campus 5G de España, ubicado en la misma.

1.2. OBJETIVOS

Esta Tesina de Máster tiene como objetivo demostrar que gracias a la característica de baja latencia que ofrecen las redes móviles de Quinta Generación (*5G*) se puede brindar servicios de conducción remota. Las especificaciones se basarán en la *Release 15* del sistema del 3GPP para redes 5G NR *Standalone*. Posteriormente, se realizarán pruebas de campo y se llevarán a cabo ajustes de código para el correcto funcionamiento del control de coche. Estos objetivos generales se pueden desarrollar en más profundidad en los siguientes objetivos secundarios:

- Estudiar las tecnologías de red de 5G.
- Desplegar una red NR, basada en las especificaciones de la *Release 15* del sistema 3GPP que completa el sistema 5G del IMT-2020.

- Evaluar las capacidades de las tecnologías 5G en una red pre-comercial.
- Desarrollar el caso de uso de conducción remota.
- Integrar el control desarrollado dentro de la red pre-comercial 5G desplegada.
- Validar la integración del control del coche y el desarrollo de este.
- Comparar la experiencia de usuario de conducción remota empleando la tecnología 5G con la 4G.

I.3. PRINCIPALES RESULTADOS

Una vez implementada la red, se realizaron las pruebas necesarias para el caso de uso y se obtuvo una implementación exitosa del mismo. Con esto se demostró que es posible brindar servicios de conducción remota empleando las redes móviles de quinta generación, gracias a su baja latencia y alta fiabilidad.

El caso de uso fue expuesto en el *Global 5G Event* y EUCNC2019 junto a un simulador de realidad virtual perteneciente al iTEAM, en el que se recreaba el mismo escenario físico, pero con variación de tecnología. Se realizó una comparativa entre ellos, mediante una encuesta y se obtuvo como resultado que la experiencia de los usuarios al conducir con la tecnología 4G les resultaba muy complicada, ya que el coche tardaba mucho en reaccionar tras girar el volante, mientras que con la tecnología 5G la respuesta del coche era inmediata.

Cabe mencionar, que en la demostración física de conducción remota se podía observar una latencia mínima en la imagen mostrada en el televisor. Se comprobó que era causada netamente por el procesamiento de imagen, el mismo que estaba determinado por la cámara empleada.

I.4. ORGANIZACIÓN DEL DOCUMENTO

En la Sección II se resume el estado del arte. En la Subsección II.1, II.2 y II.3 se analiza el estado actual de las redes de 5G, sus características principales, así como sus distintos tipos de arquitectura, sus bandas de frecuencias utilizadas y casos de uso. En la Subsección II.4 se realiza una pequeña descripción del Sistema Operativo Robótico y sus características principales enfocándonos en Ros-Indigo-Igloo. En la Subsección II.5 se explica el software comercial desarrollado por Unity Technologies, ya que en este entorno se encuentra desarrollado el simulador del iTEAM de conducción remota, el mismo que nos servirá para realizar la comparación entre tecnologías. En la Sección III se detalla la implementación del caso de uso, empezando por describir el entorno de desarrollo del software, seguido de la interconexión de la red con el coche y la conexión de la cámara al sistema, además se indican las instrucciones brindadas al usuario para un correcto control del coche. Para finalizar, en la Sección IV se discuten los resultados obtenidos, mientras que en la Sección V se encuentran las conclusiones obtenidas de la actual investigación y se exponen las líneas futuras de investigación.

II. ESTADO DEL ARTE

En este capítulo realizaremos una revisión del estándar 5G, específicamente los contenidos expuestos por la *Release 15* y *16* del sistema 3GPP.

Además, se revisará el Sistema Operativo Robótico como herramienta de programación para el control del coche. Obviamente la descripción está enfocada a las redes V2N, sobre las que se desarrolla nuestro caso de uso.

II.1. ESTADO ACTUAL DE LAS REDES 5G

Actualmente, las redes 5G han gobernado el interés de los operadores de redes móviles (MNO) en la industria automotriz. Es por eso que recientemente se consolidó la formación de *5G Automotive Association*, que reúne a los principales fabricantes de automóviles y actores de las TIC con el objetivo de promover soluciones interoperables para el V2X celular, basado en 5G.

Uno de los propulsores las redes 5G es la necesidad de dar soporte a una extensa diversidad de mercados verticales, ya que estos producen casos de uso considerablemente diferentes por lo que tendrán requisitos diferentes.

Las características que se pretenden conseguir con las redes 5G [3] son:

1. Una latencia de 1 ms. Para lograr esto se requiere se necesitan intervalos de tiempo de transmisión más pequeños. La latencia en términos simples se refiere al tiempo en que tarda en llegar la información desde la fuente hasta el destino.
Las aplicaciones que requieren tales latencias bajas son la conducción autónoma, cirugía remota, control remoto de drones y realidad virtual (VR) entre otras.
2. Una alta eficiencia energética. Esto representa un bajo consumo de energía cuando no haya datos para transmitir. En el espectro de ondas milimétricas varía de 10x hasta 100x.
3. Aumentar la fiabilidad de la red hasta el 99.999%.
4. La velocidad de datos experimentada por el usuario cambiará de 10 Mbps ofrecida en LTE a 100 Mbps o hasta 1 Gbps.
5. Aumento de la eficiencia espectral 3x en comparación el ofrecido en LTE.
6. Aumento de densidad de los dispositivos hasta 10^6 por km^2 .

La Fig. 1, muestra todas las características que ofrece la red 5G en comparación con LTE.

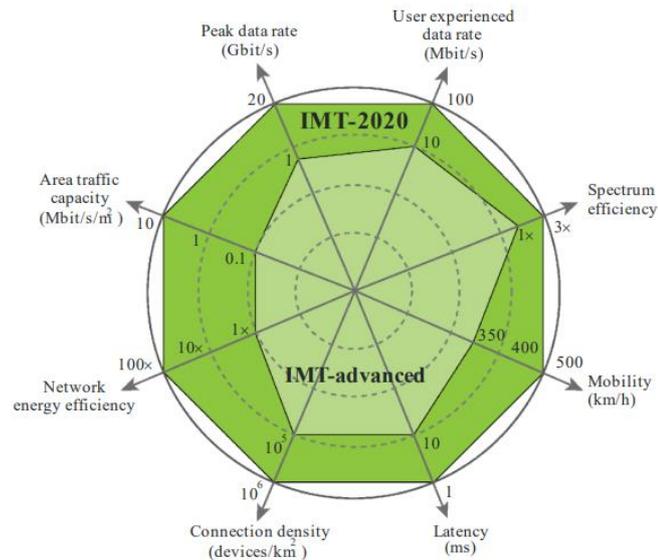


Fig. 1 Comparativa de las características de 5G con LTE [4]

Las redes de 5G se enfocan en tres servicios genéricos:

1. Banda ancha móvil mejorada (eMBB) [5] la misma que busca mejorar significativamente la velocidad de datos, la latencia, la densidad de usuarios, la capacidad y la cobertura de acceso de banda ancha móvil para permitir la transmisión en vivo, incluso en los entornos más concurridos.
2. Comunicaciones de baja latencia ultra confiables (URLLC) [5] permite a los usuarios y dispositivos comunicarse bidireccionalmente con otros dispositivos al tiempo que genera una latencia mínima y garantizando alta disponibilidad de la red.
3. Comunicación masiva tipo máquina (mMTC) [5], hace posible el bajo consumo de energía, dispositivos de larga duración para soportar aplicaciones tales como sensores embebidos en carreteras, sensores de aparcamiento, y medidores de servicios inteligentes.

Dentro de las comunicaciones ultra fiables de baja latencia encontramos las redes V2X. Las redes V2X, se fundamentan en el conocimiento cooperativo con el fin de poder brindar más servicios a los usuarios.

Las ventajas que nos ofrecen las redes V2X son enormes. Entre las mismas se puede mencionar que aumentará la seguridad vial, por lo que la tasa de muertes en la carretera será menor. Otro punto clave es que mejorarán la movilidad, por lo que el tráfico será más eficiente.

Las comunicaciones V2X en 3GPP [6] se dividen en cuatro ramas: V2V, V2I, V2N y V2P. V2V se refiere a la comunicación entre vehículos; V2I a la comunicación entre los vehículos y la infraestructura cercana, como por ejemplo los dispositivos de control de tráfico; V2N a la comunicación de vehículos con redes basadas en Internet y V2P al intercambio de información entre vehículos y los peatones.

En la Fig. 2 se observa las cuatro ramas en que se dividen las comunicaciones V2X

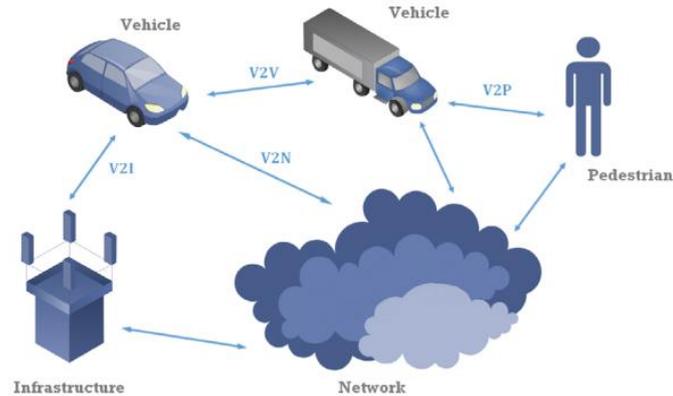


Fig. 2 Redes V2X [7]

Específicamente dentro de las redes V2N se encuentra el caso de uso de conducción remota, en el que se requiere que el funcionamiento de los sensores sea de alta fiabilidad. Cuando hablamos de sensores nos referimos a una cámara, un radar o un sensor de ultrasonidos. Es por eso que los servicios de conducción remota dependen tanto del rápido procesamiento de los datos de la red como de los sensores, ya que una latencia alta o ambigüedades en el procesamiento de los mismos podrían causar un grave problema de seguridad.

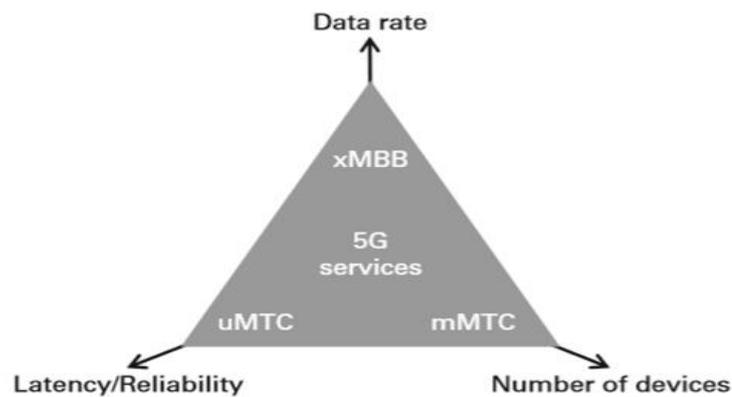


Fig. 3 Servicios de las redes 5G [5]

II.2. ARQUITECTURA Y BANDAS DE FRECUENCIA

Existen 12 propuestas diferentes sobre la interconexión de la 5G NR con el núcleo de red [5]. En sus dos extremos se conoce la variación NSA (*Non Standalone*) y la SA (*Standalone*).

En el caso NSA, o no independientes, los móviles NR se conectan a la estación base NR (*gNB*), y el *gNB* se enlaza a través de una estación base LTE (*eNB*) a la red central del LTE denominada EPC. La Fig. 3 muestra lo descrito anteriormente. En cambio, la Fig. 4 muestra prácticamente la misma arquitectura con la diferencia que tanto el *eNB* como el *gNB* no se conectan a la red central EPC si no a la red central NGCN (*NextGen Core*) que es 5G.

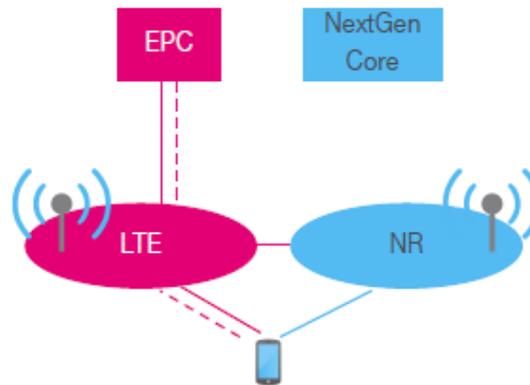


Fig. 4 Non-Standalone/"LTE Assisted", EPC conectado [5]

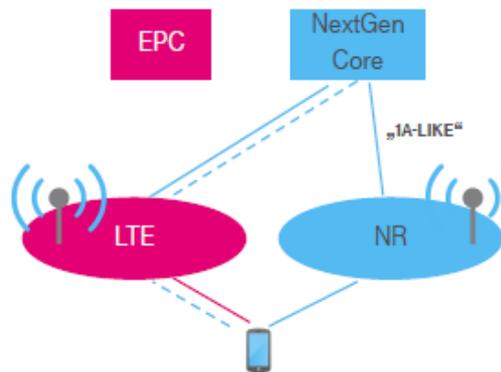


Fig. 5 Non Standalone/"LTE Assisted", NGCN conectado [5]

En el caso SA, o Independiente, los móviles NR se conectan al NR gNB, y los gNB se conectan directamente al NGCN. Es la solución 5G completa.

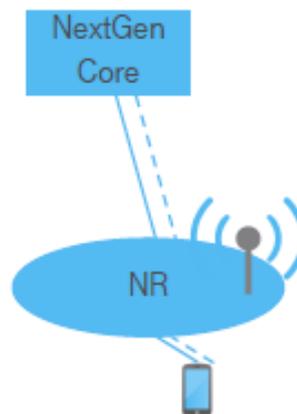


Fig. 6 Standalone, EPC conectado [5]

Según el Ministerio de Energía, Turismo y Agencia Digital [8], las bandas del espectro asignadas para el despliegue de servicios 5G en la Unión Europea son las que se recogen en la Tabla 1. Además, la tabla muestra el estado de dichas bandas de frecuencia 5G en España.

Banda de frecuencia		Situación en territorio Español
3.4 GHz - 3.8 GHz	3.4 GHz - 3.6 GHz	Empleada para desplegar servicios 5G
	3.6 GHz - 3.8 GHz	Empleada para desplegar servicios 5G, anteriormente era usada para los radioenlaces de señal de televisión. La operadora Orange tiene adjudicada 100 MHz de la banda de frecuencia pero en el rango de 3.600 Mhz a 3.800 Mhz
700 MHz		Usada para la difusión de TDT
26 GHz		Se encuentra aún pendiente de definir las condiciones técnicas de utilización.

Tabla 1. Bandas de frecuencias para el despliegue de servicios 5G en la Unión Europea [8]

II.3. CASOS DE USO DE LAS REDES 5G

Como se mencionó anteriormente, el objetivo principal de esta tesina es demostrar el potencial de las redes 5G para desplegar nuevas aplicaciones y servicios. En este caso, específicamente se enfoca en los servicios de conducción remota.

Y si bien los servicios ofrecidos actualmente se centran primordialmente en las comunicaciones personales, como lo es la voz, la mensajería instantánea, las redes sociales, la transmisión multimedia y la navegación en Internet, en cambio 5G se abre a nuevos campos como se comenta a continuación [9]:

- Automatización industrial: en la cual se realiza integración de sensores, robótica y control remoto.
- Digitalización del entorno en el que vivimos: en este caso hablamos ciudades inteligentes, al decir ciudades inteligentes no nos referimos tan solo a casas inteligentes, sino también a redes de energía inteligente, distribución agua y gas inteligente, etc.
- Servicios de seguridad pública: son sistema de información que permite al gremio policial mediante sistemas de alerta una mejor gestión en caso de desastres naturales.
- Servicios de tele-presencia: podemos mencionar, realidad virtual, hologramas y oficinas virtuales.
- Servicios de interés público: como la salud, educación y colaboración ciudadana.
- Diagnóstico remoto y gestión: empleado en los centros de diagnóstico de vehículos debido a que pueden recopilar información, la misma que es enviada periódicamente por los vehículos en modo V2N. De esta manera permiten rastrear su estado con fines de diagnóstico remoto [9], también puede ser usada para el rastreo de flotas usando la posición del vehículo y de esa manera evitar fraudes a seguro.
- Conducción autónoma: pensada para escenarios con una alta densidad de vehículos, estos pueden conducir muy cerca uno del otro, para esto se requiere cobertura total de la red en las carreteras, además de un intercambio de video y datos altamente fiable para una mayor eficiencia y seguridad en la conducción autónoma [10].

- Conducción remota: en la que el conductor sigue siendo un humano, pero ubicado físicamente en cualquier parte. En este caso, el control del vehículo lo realiza asistido mediante el uso de cámaras y datos provenientes de diferentes sensores.

Al ser un tipo de comunicaciones en tiempo real requiere estrictos requisitos para que la red garantice un rápido control del vehículo. Este tipo de conducción se puede usar en diferentes escenarios como para recoger a alguien de un sitio determinado, o cuando se deben transportar cosas en entornos peligrosos por ejemplo: en zonas donde existió un accidente nuclear o un terremoto, en la construcción de carreteras o el arado de nieve, e etc. En resumen, existen un sin número de servicios que se pueden ofrecer con la conducción remota [9].

II.4 SISTEMA OPERATIVO ROBÓTICO

ROS [11] proporciona librerías y herramientas que permiten a los desarrolladores de software realizar aplicaciones para robots. También proporciona controladores de dispositivos, herramientas de visualización, comunicación por mensajes, administración de paquetes, entre otras. Se encuentra con licencia *open source*, BSD.

Existen diferentes distribuciones de ROS, todas con soporte LTS y compatibles con una sola versión de Ubuntu LTS. La Fig. 7 muestra las diferentes distribuciones existentes [12].



Fig. 7 Distribuciones ROS [12]

Para un mayor entendimiento de ROS se debe tener claro los siguientes conceptos [13]:

- Nodo: corresponde a un ejecutable que usa ROS para comunicarse con otros nodos mediante la biblioteca denominada cliente ROS. Los nodos tienen dos posibles funciones, publicar o suscribirse a un *topic*. Cuando publican están ofreciendo un servicio mientras que cuando se suscriben están solicitando uno.

- Mensajes: con estructuras de datos sencillas empleadas cuando se suscriben o publican en un *topic*.
- *Topics*: son los canales sobre los que los nodos pueden intercambiar mensajes entre ellos, tanto para la publicación como para la suscripción.
- *Master*: proporciona los servicios de nombres y registro de los nodos dentro de la distribución ROS, con el fin de tener siempre un conocimiento actual de todos los suscriptores como publicadores. Este permite que los nodos se localicen entre sí.
- *roscout*: es una herramienta de informe que permite imprimir información en la consola.
- *roscore*: aquí se encuentran los nodos y librerías que representan un requisito para que se ejecute correctamente la distribución de ROS y es indispensable para la comunicación entre nodos.

En esta tesina de máster se implementó la versión *ROS-INDIGO-IGLOO* ya que las librerías para RBCAR se encuentran desarrolladas en la misma, por la empresa fabricante Robotnik. Para su correcta instalación hay que seguir los pasos expuestos en [13]. A continuación, se mencionan los paquetes más relevantes para el desarrollo de esta Tesina:

1. *rbcар_common*

Este paquete [14] contiene los diferentes controladores y los archivos de inicio para el robot RBCAR, compartidos para la simulación y el robot real.

2. *rbcар_pad*

Este paquete [14] permite controlar el robot mediante un *joystick* o un panel de juego. Envía los mensajes recibidos usando la entrada del *joystick*, correctamente adaptados, al tema de comando correcto.

3. *robotnik_msgs*

Este paquete [15] define los mensajes y servicios utilizados por algunos paquetes de Robotnik.

4. *Catkin*

Es un sistema de compilación multiplataforma y de código abierto. Para instalarlo se ha de hacer uso de la guía expuesta en [15].

II.5. UNITY

Unity es un software comercial desarrollado por Unity Technologies, el cual está diseñado para soportar y desarrollar videojuegos en 2D y 3D. Unity se puede usar para simulaciones de computadoras, realidad virtual, consolas y dispositivos móviles [16].

Unity tiene tres componentes importantes:

1. Un motor de juego que permite que los juegos sean desarrollados, probados y jugados en varios ambientes.
2. Una interfaz de usuario.
3. Un editor de código: el IDE proporciona un editor de texto para escribir código.

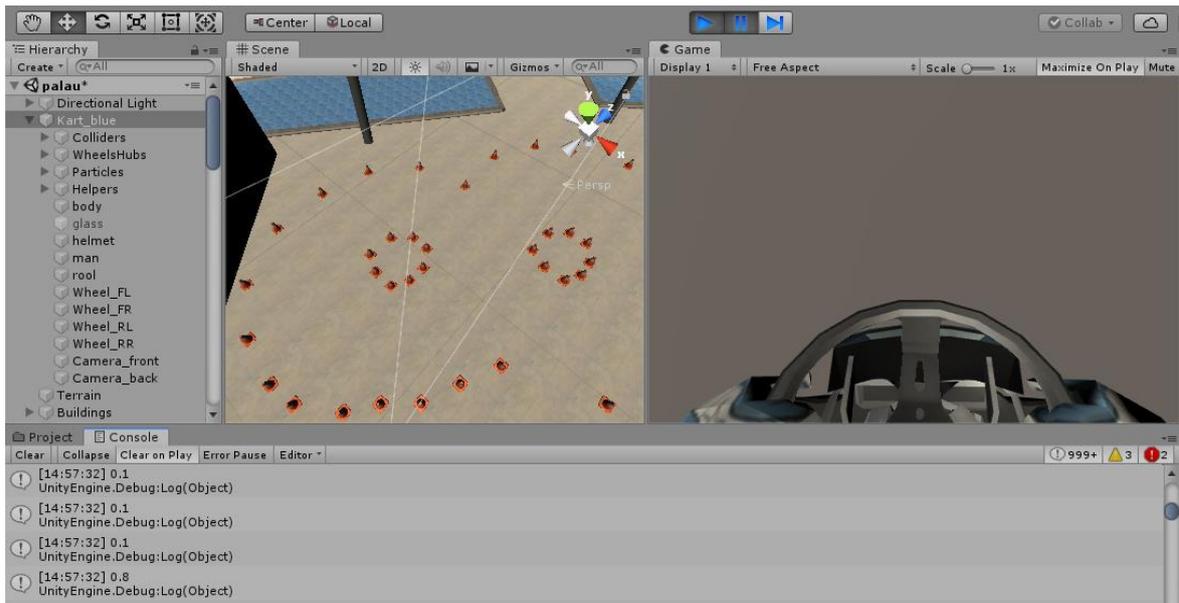


Fig. 8 Simulador en Unity del iTEAM

III. IMPLEMENTACIÓN

Esta tesina se desarrolló en el GCM del iTEAM de la UPV, dentro del campus 5G desplegado por la UPV en colaboración con Orange y ZTE.

Para la implementación se empezó desarrollando el caso de uso. Para esto lo primero que se mostrará a continuación, es cómo se realiza la comunicación del *joystick* con el sistema operativo Linux. A continuación se describirá el proceso para establecer la comunicación entre dos ordenadores, los cuales cuentan con el mismo sistema operativo robótico instalado, uno de ellos actuando como máster y el otro como cliente. De esta manera se enviará la información desde el ordenador que desempeña las funciones de controlador que se encuentra conectado al CPE_1 de 5G, hacia el CPE_3 conectado al ordenador que posee el coche, el mismo que realiza las funciones de receptor.

Posteriormente se estudió las tecnologías 5G NR, para proseguir con la interconexión de la arquitectura del caso de uso con la arquitectura general de la red 5G desplegada. Más adelante se mostrará la parte física del montaje, tanto en la UPV como en el Palacio del Congreso.

Una vez validado el control del coche, se colocaron dos cámaras IPs, las mismas que operaban con la configuración por defecto. Se mostrará más adelante la aplicación empleada para la visualización de las mismas.

A continuación se realizaron pruebas de campo para ajustar los parámetros de la cámara y del código de control. Se incluyeron pruebas de esfuerzo para determinar el consumo energético o detectar algún tipo de fallo tanto en el software como en el hardware tras un largo periodo de trabajo.

Para finalizar se realizó el montaje del escenario en el que se expondría el caso de uso y se realizaría una comparación entre la conducción remota en tiempo real en una red móvil 5G, con la

conducción mediante un simulador que posee el iTEAM que permite variar los tiempos de latencia y de esa manera se puede encontrar el punto de trabajo en el que la experiencia de tele-conducción es impracticable.

III.1. ENTORNO DE DESARROLLO DEL SOFTWARE

El programa se desarrolló en la distribución Ros Indigo Igloo, compatible principalmente en la versión Ubuntu 14.04 LTS, que es la empleada en este caso. Esta distribución usa el lenguaje C++.

El proyecto se encuentra estructurado mediante tres carpetas principales, que se observan en la Fig. 9: la primera es *config*, en la cual se encuentra el archivo de extensión *.yaml*, que contiene el mapeo de botones y ejes del volante de carreras PS3. La segunda corresponde a *src*, que contiene el archivo de extensión *.cpp* correspondiente al código principal de control del coche, y la tercera es *launch* que contiene el archivo de extensión *.launch*, que permite lanzar fácilmente múltiples nodos ROS de forma local o remota.



Fig. 9 Carpetas principales

Cada vez que se realiza una modificación del código principal se debe realizar un `catkin_make`, para volver a compilar cualquier paquete ubicado dentro del directorio `~/catkin_ws`. De esta manera, en el momento de ejecutarlo se verán reflejados los cambios realizados. Este proceso se lo realiza de la siguiente manera [18]:

1. Entrar en la carpeta `catkin_ws`.

```
$ cd/ros/indigo/catkin_ws/
```

2. Construir el paquete.

```
$ catkin_make
```

Después de construir correctamente el paquete (Fig. 10), las carpetas *build* y *devel* se actualizan. La primera carpeta es en la que se invocan *cmake* y *make*, y la segunda contiene todos los archivos y objetos generados, además de los archivos de configuración.

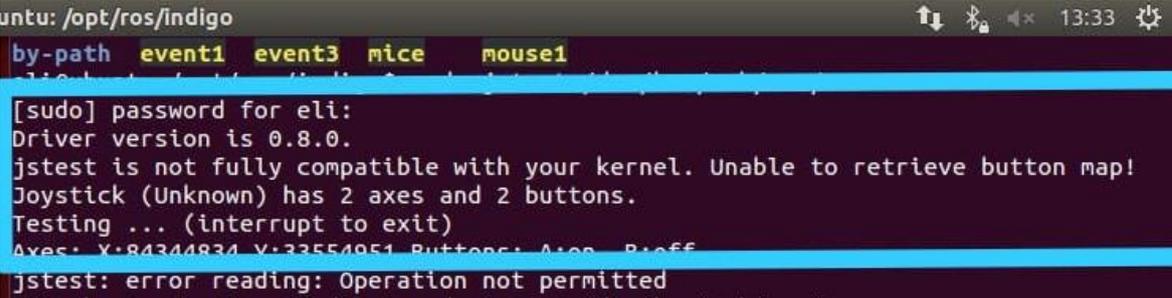
```
[ 97%] Generating C++ code from robotnik_msgs/SetNamedDigitalOutput.srv
[ 97%] Generating C++ code from robotnik_msgs/QueryAlarms.srv
[ 98%] Generating C++ code from robotnik_msgs/set_analog_output.srv
[ 98%] Generating C++ code from robotnik_msgs/GetBool.srv
[ 99%] Generating C++ code from robotnik_msgs/enable_disable.srv
[ 99%] Generating C++ code from robotnik_msgs/set_odometry.srv
[ 99%] Generating C++ code from robotnik_msgs/SetMotorMode.srv
[ 99%] Built target robotnik_msgs_generate_messages_cpp
Scanning dependencies of target robotnik_msgs_generate_messages
Scanning dependencies of target rbcар_pad_node
[ 99%] Built target robotnik_msgs_generate_messages
[100%] Building CXX object rbcар_common/rbcар_pad/CMakeFiles/rbcар_pad_node.dir/src/rbcар_pad.cpp.o
Linking CXX executable /opt/ros/indigo/catkin_ws/devel/lib/rbcар_pad/rbcар_pad_node
[100%] Built target rbcар_pad_node
root@iteam:/opt/ros/indigo/catkin_ws#
```

Fig. 10 Construcción correcta de paquetes

Hay que tener en cuenta que mediante la utilización del *bash scripts*, se puede realizar automáticamente la comunicación de los dos ordenadores con la distribución ROS. Esto es debido a que cualquier comando que se pueda ejecutar desde el terminal puede ser incluido dentro de este archivo.

III.2. PROGRAMACIÓN

Lo primero a realizar es la configuración y reconocimiento del *joystick*. Es requisito indispensable que éste sea compatible con Linux. Si no cumple este requisito nos mostrará lo siguiente.



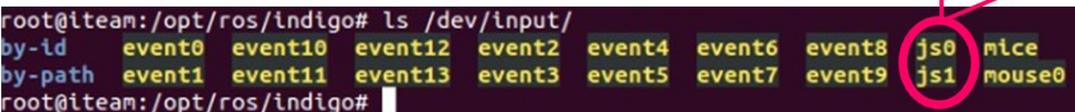
```
untu: /opt/ros/indigo
by-path event1 event3 mice mouse1
[sudo] password for eli:
Driver version is 0.8.0.
jstest is not fully compatible with your kernel. Unable to retrieve button map!
Joystick (Unknown) has 2 axes and 2 buttons.
Testing ... (interrupt to exit)
Axes: X:84344834 Y:33554951 Buttons: A:on B:off
jstest: error reading: Operation not permitted
```

Fig. 11 Joystick no compatible

En el entorno de ROS se debe ejecutar los siguientes pasos [19] para realizar la tarea antes mencionada:

1. Instalar el paquete *joy*, cuya función es reconocer el dispositivo *joystick* conectado.
\$ sudo apt-get install ros-indigo-joy
2. Configurar el *joystick*, para esto se debe conectar el dispositivo al ordenador y comprobar que lo reconoce. Para esto se ejecuta:
\$ ls/dev/input /

En la pantalla aparecerá un listado con dispositivos como se observa en la Fig. 12. El que corresponde al *joystick* aparecerá de la siguiente forma “jsX”, siendo “X” el número asignado como etiqueta al *joystick*. Pueden existir varios, en el caso que se reconocieran otros dispositivos anteriormente.



Joysticks reconocidos

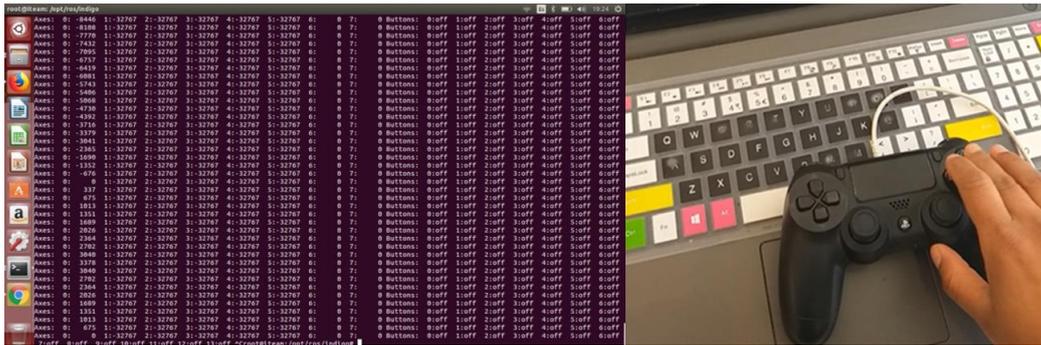
```
root@iteam:/opt/ros/indigo# ls /dev/input/
by-id event0 event10 event12 event2 event4 event6 event8 js0 mice
by-path event1 event11 event13 event3 event5 event7 event9 js1 mouse0
root@iteam:/opt/ros/indigo#
```

Fig. 12 Joysticks reconocidos

3. Para asegurarnos del correcto funcionamiento del *joystick* ejecutamos el siguiente comando:

\$ sudo jstest/dev/input/jsX

Aparecerá en la pantalla los datos correspondientes tanto a los ejes y a los botones del *joystick* y variaran cuando presionemos los botones del dispositivo.

Fig. 13 Correcto funcionamiento del *joystick*

4. Luego se requiere hacer accesible el *joystick* para cualquier nodo ROS. Esto se hace de la siguiente forma:

```
$ ls -l / dev / input / jsX
```

El comando anterior nos arrojará en el terminal lo siguiente:

```
crw-rw-XX- 1 root dialout 188, 0 209-02-14 10:15 /dev/input/jsX
```

Hay que tener en cuenta que si XX es rw: el dispositivo jsX está configurado correctamente.

Pero si XX es --: el dispositivo jsX no está configurado correctamente y se necesita ejecutar la siguiente orden para corregir el error:

```
$ sudo chmod a + rw / dev / input / jsX
```

Una vez realizada correctamente la configuración del *joystick*, se procede a la comunicación entre los dos ordenadores con ROS. Para esto se realiza lo siguiente:

1. El ordenador en el que se encuentra conectado el *joystick*, se debe declarar como nodo cliente y como nodo máster al ordenador del coche. Para eso se debe abrir terminal y luego el archivo `bashrc` de la siguiente manera:

```
$ gedit ~/.bashrc
```

IMPORTANTE: Se configura estos comandos en el archivo `bashrc` para que se ejecuten automáticamente cada vez que se abra el terminal. Hay que tener en cuenta que siempre que se cambie la IP del `bashrc`, se debe cerrar todas las terminales y volver a ejecutar el programa para que se actualice el mismo.

2. Una vez en el archivo se colocó al final del mismo lo siguiente:

```
export ROS_MASTER_URI=http://dirección_IP_coche:11311
```

```
export ROS_IP=dirección_IP_del_ordenador_del_joystick
```

```
export ROS_MASTER_URI=http://172.23.3.1:11311
export ROS_IP=10.0.13.21
```

Fig. 14 Ejemplo de configuración de IPs dentro del `bashrc`

3. En el archivo `bashrc` del ordenador del coche se declara la IP del nodo cliente, en este caso se ingresa de la misma manera y se coloca:

```
export ROS_IP=dirección_IP_del_ordenador_del_joystick
```

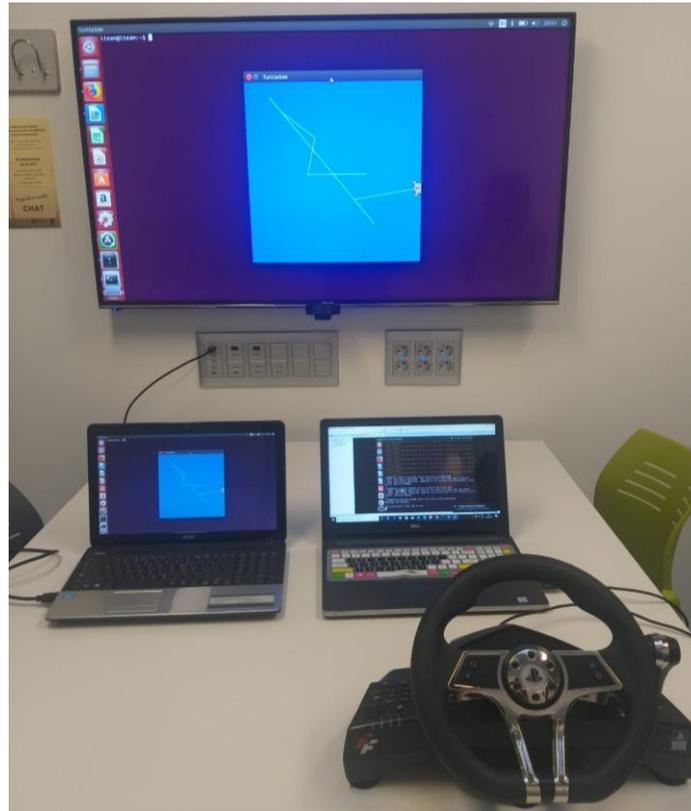


Fig. 15 Comunicación entre dos ordenadores con Ros Indigo Igloo

4. Dentro de este archivo también se debe colocar las siguientes líneas de código para configurar el entorno de ROS.

```
source /opt/ros/indigo/setup.bash
```

```
source /opt/ros/indigo/catkin_ws/devel/setup.bash
```

Una vez conseguido esto, a continuación comenzamos con el mapeo del *joystick*. En un principio se utilizó el *joystick* ps3 como se observa en la Fig. 16 pero se evidenció una inestabilidad en el momento de la conducción generada por el acelerómetro interno que posee el mismo, por lo que se efectuó el cambio del *joystick* por un volante de carreras para PS3 (Fig. 17), ya que este ofrece mayor estabilidad al momento de determinar el posicionamiento del volante del coche real.

Fig. 16 Pruebas con *joystick* PS3

Fig. 17 Conducción con volante de carreras para PS3

Al cambiar de elemento de control se necesitó un nuevo mapeo de los botones, como de los ejes, a demás de la modificación del código central del coche para que trabaje en función de nuestros nuevos requerimientos.

Para el nuevo mapeo se determina la etiqueta de cada elemento del volante de la siguiente manera:

```
$ cd /opt/ros/indigo
```

```
$ sudo jstest / dev / input / jsX
```

Los comandos anteriores nos abrirán una ventana en la que nos mostrará la etiqueta del botón o eje al que pertenece cada botón, palanca o pedales del volante. Las nuevas etiquetas se observan en la Fig. 18.

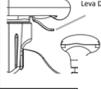
button_ go		1	button_ speed_ down		3
button_ stop		0	button_ dead_m an		5
button_ speed_ up		4	button_ reverse		2

Fig. 18 Nuevas etiquetas para el mapeo

Para realizar el cambio en el *script* correspondiente al mapeo, se debe acceder al mismo como se detalla a continuación:

```
$ sudo nautilus
```

Se abrirá una ventana con permisos de super usuario. Una vez ahí, se debe ingresar al siguiente directorio:

```
Equipo/ros\indigo\catkin_ws\src\rbcar_common\rbcar_pad\config  
rbcar_pad_config
```

En este caso el mapeo dentro del *script* queda definido de la siguiente forma como lo muestra la Fig. 19.

```
*rbcar_pad_config.yaml x  
axis angular position: 0 #2 volante  
button_go: 1 #7 marcha hacia adelante  
button_stop: 0 #6 freno  
button_speed_up: 4 # asignado para aumentar la velocidad  
button_speed_down: 3 # asignado para disminuir la velocidad  
button_dead_man: 5 #11 palanca para que se activen todos los botones  
volante: true #true funciona el volante, false funciona el joystick  
speed_step: 1.0 # velocidad constante con el volante de PS4 para marcha hacia adelante  
button_reverse: 2 # boton para marcha atras  
speed_step_1: -0.75 # velocidad constante con el volante de PS4 para marcha atras
```

Fig. 19 Mapeo del volante de carreras

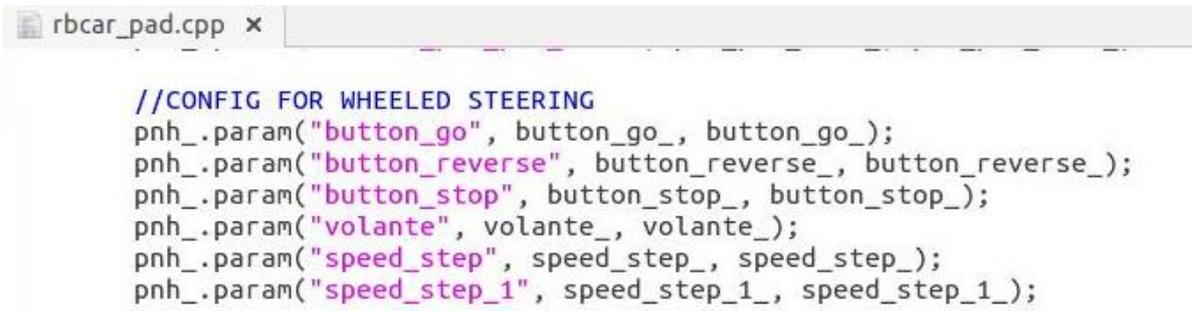
Una vez realizado el mapeo, se procede a escribir las sentencias en el código principal, que permite que el volante del coche refleje los movimientos del volante de carreras.

El primer paso es definir las variables del mapeo dentro del código principal, en la parte que corresponde al volante, tal como lo muestra Fig. 20.

```
rbcar_pad.cpp x  
int button_kinematic_mode_  
//! boton para dar marcha hacia adelante  
int button_go_  
//! boton para dar marcha hacia atras  
int button_reverse_  
//! boton para frenar  
int button_stop_  
//! opcion para cambiar entre volante y joystick  
bool volante_  
//! velocidad inicial del volante  
double speed_step_  
//! velocidad inicial del volante retro  
double speed_step_1_  
...
```

Fig. 20 Declaración de las variables

La segunda acción a realizar, es la configuración de las variables creadas anteriormente como se observa en la Fig. 21.



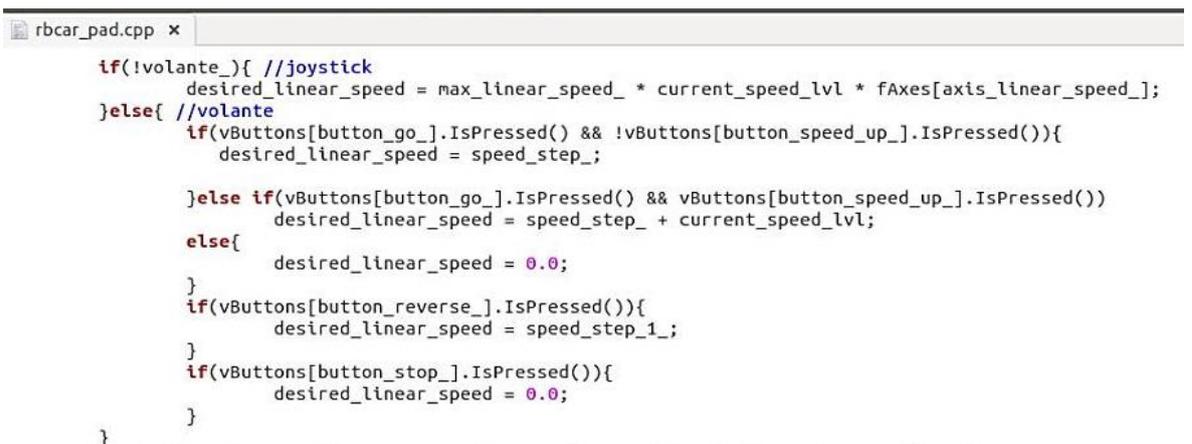
```

//CONFIG FOR WHEELED STEERING
pnh_.param("button_go", button_go_, button_go_);
pnh_.param("button_reverse", button_reverse_, button_reverse_);
pnh_.param("button_stop", button_stop_, button_stop_);
pnh_.param("volante", volante_, volante_);
pnh_.param("speed_step", speed_step_, speed_step_);
pnh_.param("speed_step_1", speed_step_1_, speed_step_1_);

```

Fig. 21 Configuración de las variables

Una vez configuradas las variables, en la Fig. 22 se muestra las sentencias que permiten que el coche realice la acción de ir hacia delante con el pedal derecho, que frene con el pedal izquierdo y que dé marcha atrás presionando el botón correspondiente al círculo del volante de carreras para PS3. También se estableció la opción de ir aumentando la velocidad al presionar al mismo tiempo la palanca derecha con el pedal derecho, y para disminuirla se debe presionar al mismo tiempo tanto el botón correspondiente al círculo como el pedal derecho.



```

if(!volante_){ //joystick
    desired_linear_speed = max_linear_speed_ * current_speed_lvl * fAxes[axis_linear_speed_];
}else{ //volante
    if(vButtons[button_go_].IsPressed() && !vButtons[button_speed_up_].IsPressed()){
        desired_linear_speed = speed_step_;
    }else if(vButtons[button_go_].IsPressed() && vButtons[button_speed_up_].IsPressed()){
        desired_linear_speed = speed_step_ + current_speed_lvl;
    }else{
        desired_linear_speed = 0.0;
    }
    if(vButtons[button_reverse_].IsPressed()){
        desired_linear_speed = speed_step_1_;
    }
    if(vButtons[button_stop_].IsPressed()){
        desired_linear_speed = 0.0;
    }
}
}

```

Fig. 22 Código para las marchas

En la Fig. 23 se muestra los valores establecidos con el que se va ir aumentando la velocidad y el rango máximo que esta alcanzará.

Debido a que el escenario de conducción era un circuito pequeño cuya área era de 110m², la velocidad inicial está configurada a 3km/h por seguridad en el momento de conducir. Además, en la demostración el vehículo podría ser conducido por el público en general, por lo que no era conveniente ir a velocidades altas.

Finalmente en la Fig. 24 muestra la manera correcta de usar ROS_INFO para imprimir en la terminal información de las variables.

```

if(vButtons[button_speed_up_].IsReleased()){
    current_speed_lvl += 0.2;
    if(current_speed_lvl > 6.0)
        current_speed_lvl = 1.0;
}

if(vButtons[button_speed_down_].IsReleased()){
    current_speed_lvl -= 0.2;
    if(current_speed_lvl < 0.0)
        current_speed_lvl = 0.0;
}

```

Fig. 23 Valores de velocidad

```

ROS_INFO("axis_angular_position_ %d  desired_angular_position=%5.2f desired_linear_vel=%5.2f
current_speed_lvl=%5.2f" , axis_angular_position_, desired_angular_position, desired_linear_speed,
current_speed_lvl);

```

Fig. 24 ROS_INFO

Es importante tener en cuenta que el archivo con la extensión `.launch` ubicado en el directorio `ros/indigo/catkin_ws/src/rbcar_common/rbcar_pad/launch`, es en el que se debe modificar la etiqueta del `joystick jsX`, porque éste ejecutará tanto el nodo del joystick como el del coche.

```

<?xml version="1.0"?>
<launch>

  <!-- joy node -->
  <node respawn="false" pkg="joy" type="joy_node" name="joy" >
    <param name="dev" type="string" value="/dev/input/js1" />
    <param name="deadzone" value="0.12" />
  </node>

```

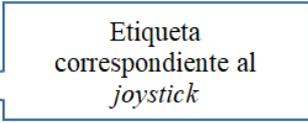


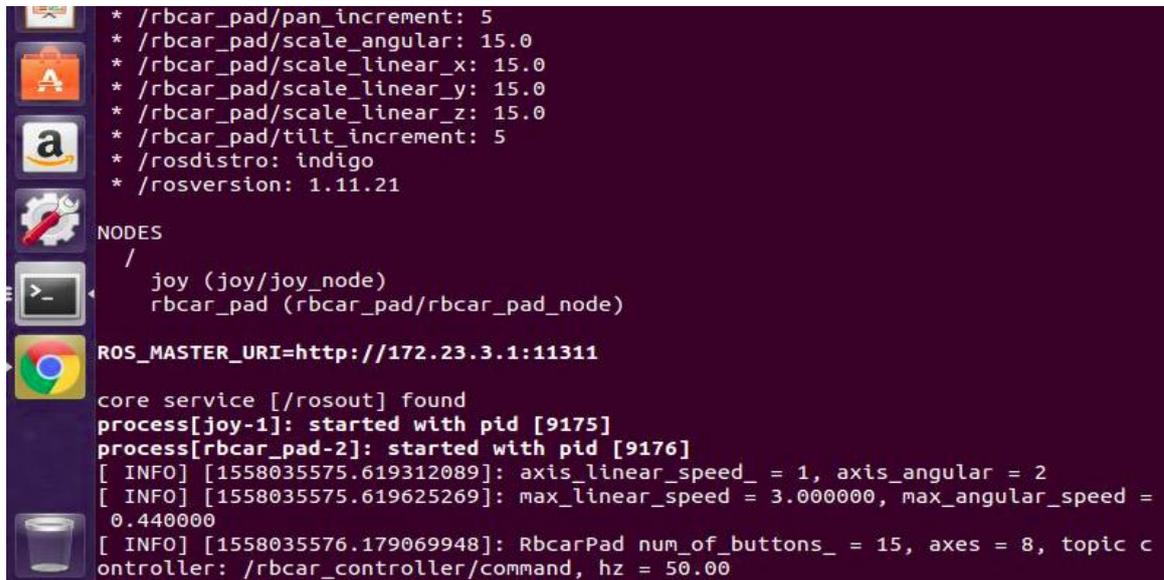
Fig. 25 Script en el que se establece que joystick usar

Una vez establecido el `joystick` a usar se debe ejecutar las siguientes sentencias para inicializar el control remoto del coche:

1. Ingresar como super usuario ya que el programa fue instalado como super usuario.
`sudo -i`
2. Ingresar a la carpeta donde se encuentra instalado ROS.
`cd /opt/ros/indigo`
3. Ejecutar el programa `.launch` que llama a todos los otros `scripts`.
`roslaunch rbcar_pad rbcar_pad.launch`

Si se ejecutó correctamente debe observarse lo que se muestra en la Fig. 26, donde se evidencia que en el terminal se imprime el mapeo configurado. Los nodos que están conectados en este caso muestra que son dos: el primero es el nodo `joy`, que se corresponde al ordenador que controla el volante, y el segundo es el nodo `rbcar_pad`, correspondiente al ordenador que se encuentra ubicado dentro del coche.

También muestra que se ha realizado una conexión con el ROS_MASTER_URI empleando la IP 172.23.3.1. Además, usa el puerto 11311 para la comunicación y se encuentran inicializados ambos nodos. Los datos que imprime al final son datos informativos.



```

* /rbcар_pad/pan_increment: 5
* /rbcар_pad/scale_angular: 15.0
* /rbcар_pad/scale_linear_x: 15.0
* /rbcар_pad/scale_linear_y: 15.0
* /rbcар_pad/scale_linear_z: 15.0
* /rbcар_pad/tilt_increment: 5
* /roѕdistro: indigo
* /roѕversion: 1.11.21

NODES
/
  joy (joy/joy_node)
  rbcар_pad (rbcар_pad/rbcар_pad_node)

ROS_MASTER_URI=http://172.23.3.1:11311

core service [/roѕout] found
process[joy-1]: started with pid [9175]
process[rbcар_pad-2]: started with pid [9176]
[ INFO ] [1558035575.619312089]: axis_linear_speed_ = 1, axis_angular = 2
[ INFO ] [1558035575.619625269]: max_linear_speed = 3.000000, max_angular_speed = 0.440000
[ INFO ] [1558035576.179069948]: RbcарPad num_of_buttons_ = 15, axes = 8, topic controller: /rbcар_controller/command, hz = 50.00

```

Fig. 26 Comunicación con el Ros Máster exitosa

III.3. INTERCONEXIÓN DE LA RED CON EL COCHE

La red pre-comercial 5G *standalone*, fue instalada y configurada para las pruebas iniciales en el campus 5G por el equipo de trabajo de ZTE en el edificio 8G, acceso D. Los equipos físicos se encuentran en la cubierta y la antena en el patinillo como se muestra en la Fig. 27.



Fig. 27 Instalación de la red pre-comercial 5G en el campus 5G

La antena es de 8x8 y tiene capacidad de conectarse en una banda por debajo de 6 GHz. Además, es de banda ancha y multiplexa hasta 4 usuarios.

El equipo central se encuentra conectado por una fibra desde el Hotel AC que se encuentra en la avenida Francia y de ahí se conecta al MSC (*Mobile Switching Centre*) de Orange.

El sistema de control tiene una arquitectura cliente – servidor, por lo que se tendrá dos escenarios para realizar las pruebas. El escenario correspondiente al cliente, se muestra en la Fig. 28. En la parte izquierda se observan los equipos 5G interconectados, mientras que la parte derecha muestra los componentes para el control que constan de un volante conectado a un ordenador que envía los datos a través del CPE (*Customer premise equipment*) al vehículo. Existe otro ordenador en este escenario para las cámaras, que enviarán las imágenes a un monitor remoto para visualizar el entorno del vehículo.



Fig. 28 Escenario correspondiente al cliente



Fig. 29 Instalación del servidor o *máster*

El escenario correspondiente al servidor o *máster* se observa en la Fig. 29. En la parte izquierda se observan las dos cámaras en el momento de su instalación y en la parte derecha se evidencian los equipos que iban dentro del coche, el cual constaba de un CPE con su respectiva batería. El CPE da conectividad a un *router* que, a su vez, conecta tanto al ordenador interno del coche como a las dos cámaras IP.

Para el *Global 5G Event* y EUCNC2019 se realizó el despliegue de otra red en el Palacio del Congreso que, al igual que en el campus 5G, tenía dos partes, servidor y cliente. La Fig. 30 muestra la zona destinada para la exposición del caso de uso. Esta zona fue escogida de manera que tuviera un nivel bueno de RSRP (*Reference Signals Received Power*), fijado en -69 dBm y también que tuviera visión directa del coche, de manera que se pueda comprobar el efecto de la conducción remota tanto de forma directa como de forma remota empleando las cámaras. Además tenía que ser un área que permitiera al conductor maniobrar de forma segura.

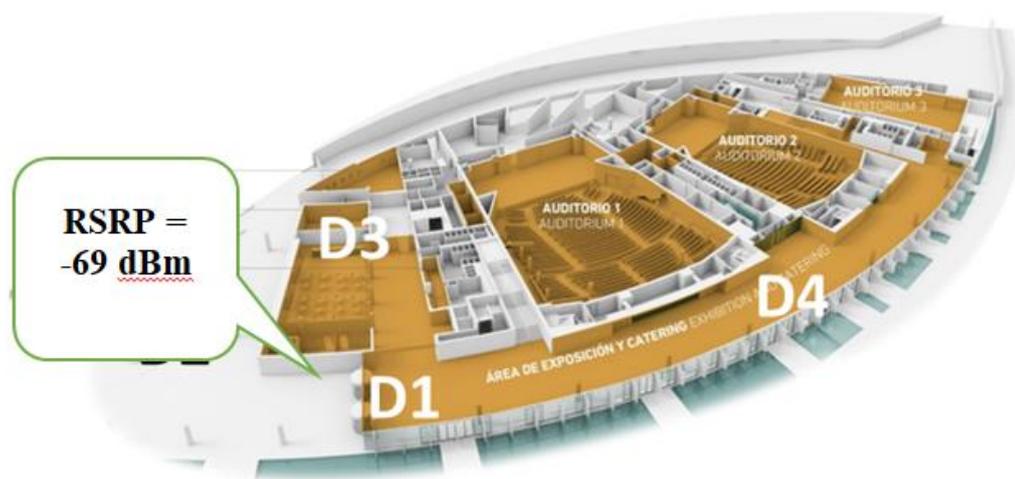


Fig. 30 RSRP de la zona de prueba

La Fig. 31 muestra un esquemático de la solución tecnológica montada. En la parte izquierda se muestra la parte del *stand* que corresponde al cliente, siendo este escenario la solución *uplink* en la que los ordenadores se conectan al *router*, el mismo que se encuentra conectado al CPE_1. Este se registra en la celda 5G del sector 3 que se encuentra ubicado en el edificio Géminis a unos 400 m del Palacio de Congreso. Desde la estación base, la información se traslada al núcleo de red de Orange que se encuentra ubicado en el polígono de Vara de Quart de Valencia. En el *downlink* la información regresa hacia la celda 5G del sector 3 del edificio Géminis y se conecta con el CPE_3 instalado en el coche. Este CPE se conecta a otro *router* que tiene conexión hacia las dos cámaras IPs y el ordenador interno del coche.

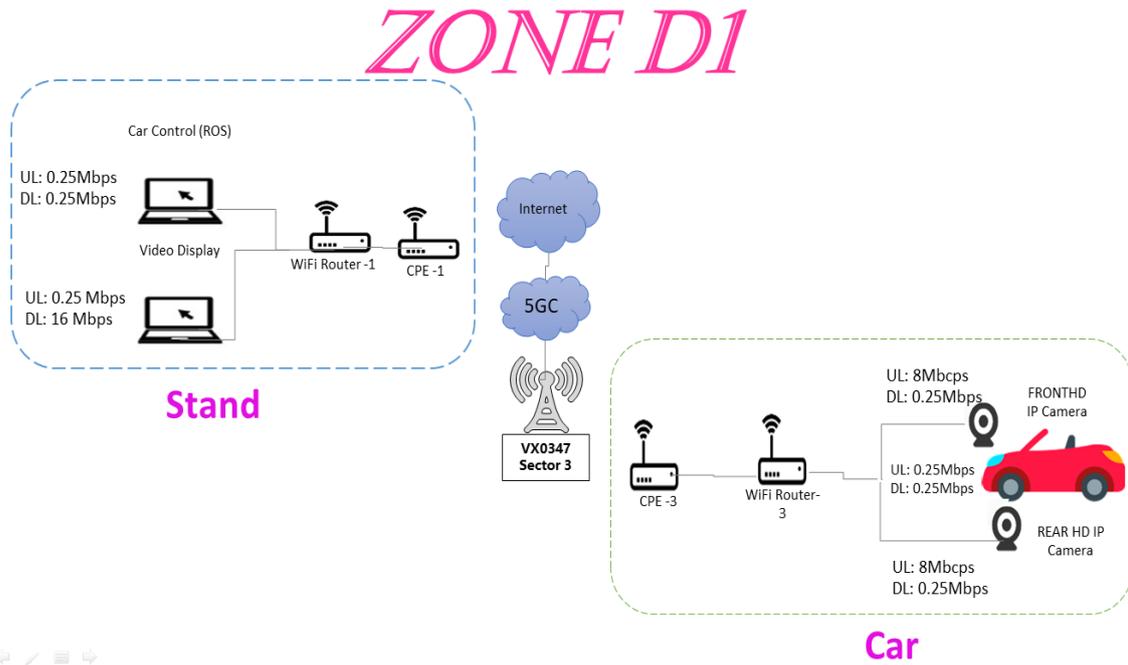


Fig. 31 Esquema de la red



Fig. 32 Escenario cliente - Palacio del Congreso

La Fig. 32 muestra el escenario real correspondiente al cliente en el Palacio del Congreso, en la parte izquierda se observa el edificio Géminis con las antenas ubicadas en la azotea y a la derecha los equipos 5G. La Fig. 33 muestra el que corresponde al servidor. El circuito se montó en un espacio de 10 m de ancho por 11 m de largo, usando balizas de plástico adheridas al piso con velcro y cinta doble cara. Además, se colocaron banderines para tener un mejor control de la zona.



Fig. 33 Escenario servidor - Palacios del Congreso

Cada CPE tenía que ser conectado a un ordenador destinado a realizar una sola función. En el momento de empezar las pruebas se debía realizar un *Attach* y un *Detach* finalizadas las mismas.

El proceso de *Attach* se realizaba de la siguiente manera:

1. Se asigna una IP fija al ordenador, como se observa en la Fig. 34.

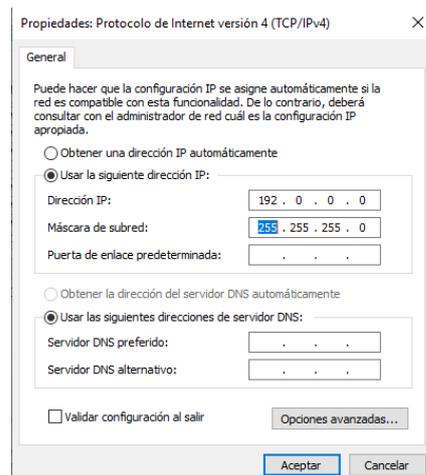


Fig. 34 Asignación de IP estática

2. Se ejecuta como administrador el programa LMT del CPE, desarrollado por ZTE como lo muestra la Fig. 35.

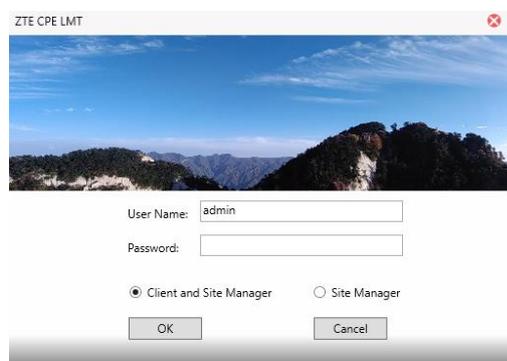


Fig. 35 Programa LMT

- Una vez dentro de la aplicación, ir a UE1, hacer clic derecho y seleccionar la opción *Signal Trace*.

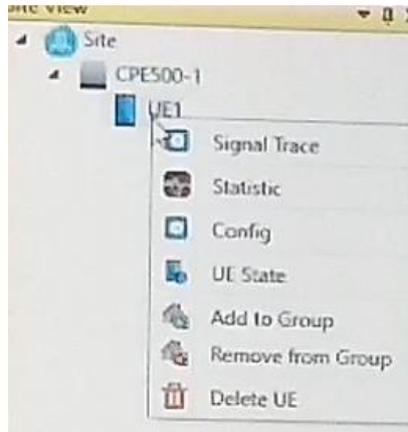


Fig. 36 Signal Trace

- Luego, hacer clic en *Reset y Start Signal Trace*.

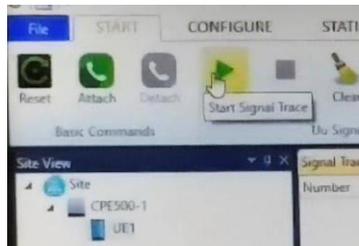


Fig. 37 Reset y Start Signal Trace

- Por último, seleccionar *Attach*, con lo cual nos mostrará lo que se observa en la Fig. 38. Una vez que se encuentre la IP asignada en la parte inferior se encontrará vinculado el CPE a la red.

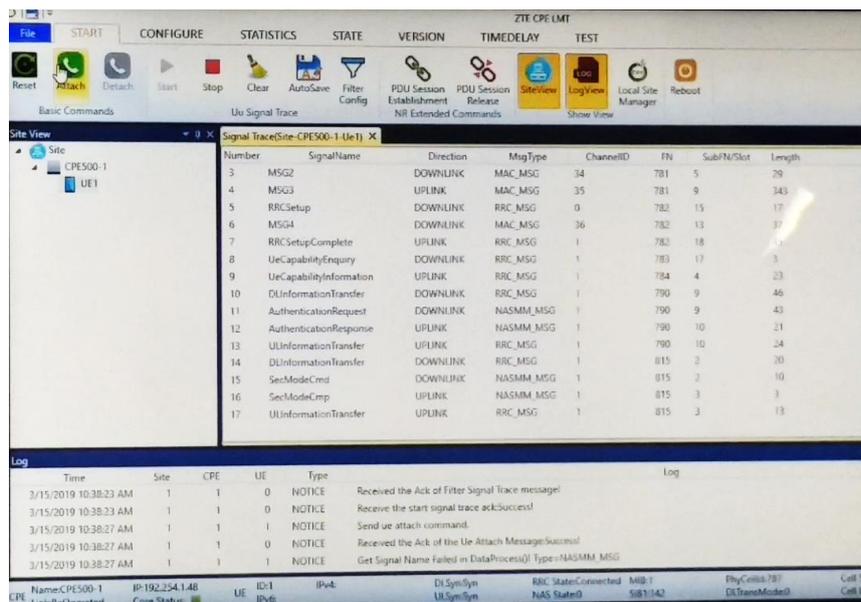


Fig. 38 Vinculación del CPE con la red móvil

Para realizar el *detach*, lo que vendría a ser la desvinculación del CPE con la red se debe hacer clic en stop.



Fig. 39 Desvinculación del CPE de la red móvil

III.4. CONEXIÓN DE LA CÁMARA AL SISTEMA

Para visualizar las imágenes transmitidas por las cámaras se usó la aplicación *VSPlayer V7.4.2-Hikvision*, solamente disponible para el sistema operativo Windows. Se probaron diferentes tipos de cámaras variando tanto el ángulo de visión como la velocidad de procesamiento de imágenes de las mismas y esta resultó ser la mejor opción.

La Fig. 40 en la parte izquierda muestra que la cámara va conectada a un *router* y este al CPE_3. Se emplea un *router* ya que el CPE solo tiene un puerto Ethernet para el usuario y se necesita 3 puertos para dar conectividad al servidor. Dos destinados para las cámaras y el tercero para el ordenador con el controlador del coche. En la parte derecha se observa la aplicación para acceder a la imagen de las cámaras. Se accedía a ellas mediante las IPs asignadas a cada una de las mismas y utilizando como modo de conexión el protocolo TCP (*Transmission Control Protocol*).

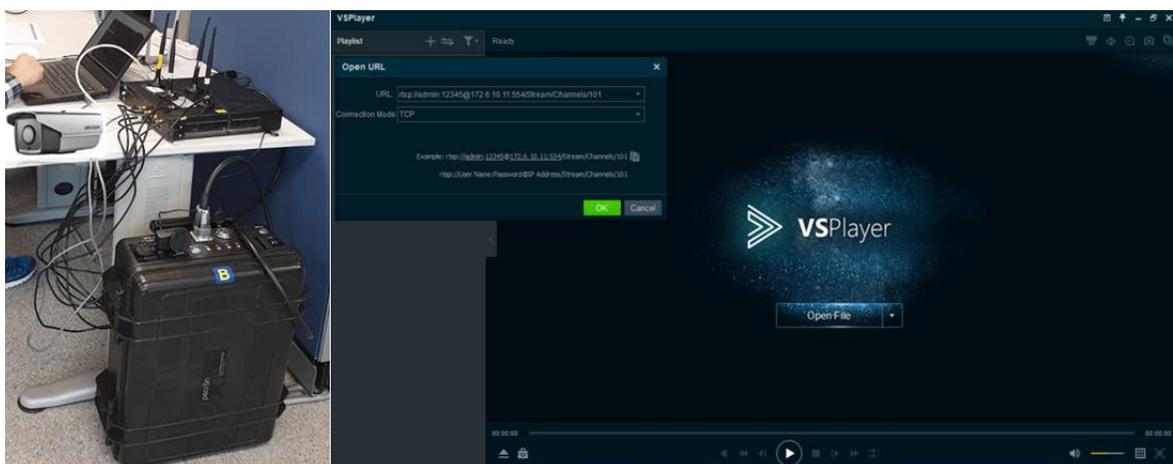


Fig. 40 Conexión de las cámaras IPs

En la Fig. 41 se puede observar el ángulo de visión de la cámara usada que es de 98.5° , con una resolución de flujo principal y velocidad de fotogramas de 25fps (1920×1080) y con una tasa de transmisión de datos de 16 Mbps.



Fig. 41 Imagen obtenida por la cámara IP

III.5 INSTRUCCIONES AL USUARIO

Ya que el coche podía ser operado por público general, en el evento se les proporcionaban las siguientes instrucciones, explicando la función de los botones, palancas y pedales en el volante de carreras PS3 (Fig. 42).

1. Mantenga siempre presionado la palanca R-wings para mover el volante y dar marcha hacia delante o hacia atrás.
2. Para dar marcha hacia delante, presione al mismo tiempo el pedal correspondiente al acelerador y el ala derecha.
3. Para dar marcha atrás, presione al mismo tiempo el botón con un círculo y el ala derecha.
4. Si desea frenar, presione al mismo tiempo el pedal correspondiente al freno y el ala derecha.



Fig. 42 Volante de carreras para PS3

IV. RESULTADOS

Una vez concluido el *Global 5G Event* y EUCNC2019, se procesaron los resultados obtenidos de distintas encuestas que se pasaron a los participantes en el evento. Cabe destacar la gran experiencia que este evento supuso para mí, ya que pude compartir con expertos en el sector de la 5G provenientes de todos los rincones del mundo.

Fueron varios meses de arduo trabajo con el grupo técnico de ZTE y del iTEAM (Fig. 43), en el que he podido consolidar amistades con personas que en un futuro podrían ser mis compañeros de trabajo y sobre todo de las que he aprendido tanto a nivel científico como personal.



Fig. 43 Grupo técnico de ZTE y del iTEAM

Un evento de estas características, que consolidó a Valencia como ciudad pionera en la tecnología 5G me ha permitido desarrollar fortalezas y notar carencias que debo corregir. Con respecto a la demostración de coche conducido en remoto, esta tuvo gran impacto, ya que se acercaron cientos de personas para informarse sobre su funcionamiento y para ser parte de la experiencia de conducción remota.

Existieron un sin número de reportajes tanto televisivos como de prensa escrita y medios digitales sobre el caso de uso y la participación de la UPV en el mismo, tal como se observa en la Fig. 44. Estos reportajes no solo fueron a nivel nacional, también existieron a nivel internacional, teniendo por ejemplo una importante repercusión en China

Recibimos innumerables muestras de afecto y felicitaciones y varias empresas se interesaron en el desarrollo realizado, e incluso se llegó a firmar un convenio de colaboración entre la UPV y la Asociación Española de Vehículos Autónomos como se evidencia en la Fig. 45. Sin duda, una experiencia personal y profesional inolvidable.

La tecnología 5G en robótica y coche conectado se muestra en Valencia, que acoge dos encuentros europeos

Publicado: 17/06/2019



Robótica y aplicaciones de coche conectado y conducción autónoma basadas en la tecnología 5G son algunos de los casos de uso que ha presentado este jueves la Universidad Politécnica de Valencia (UPV), Orange y ZTE como avance de los encuentros europeos "Global 5G Event" y "TuCNC 2019" que se celebran en Valencia a partir de este mismo lunes.



La Politècnica desarrolla un coche autónomo dirigido a distancia con tecnología 5G

La UPV demuestra el potencial industrial de las nuevas redes con un robot mecánico que reproduce al instante los movimientos del brazo humano

Ramón Ferrando | 13.06.2019 | 23:45

La Universitat Politècnica de Valencia (UPV) y Orange, con la colaboración de ZTE, presentaron ayer el potencial de la nueva tecnología 5G para usos industriales. El Instituto ITEAM y el instituto ai2 de la UPV han desarrollado un coche autónomo dirigido a distancia con tecnología 5G y han diseñado el sistema de control de un brazo robótico que reproduce al instante los movimientos del brazo humano. Narcís Cardona, director del Instituto ITEAM, explicó que los dos proyectos que han completado con éxito tras meses de trabajo muestran «los primeros resultados del VLC».



La Politècnica desarrolla un coche autónomo dirigido a distancia con tecnología 5G

Más vídeos

Motor
Los coches necesitan 5G (y se lo van a dar)

Por Sergio Amador

La Universidad Politécnica de Valencia investiga qué tiene que aportar la transmisión de datos a alta velocidad a la conducción automática. Puede ser clave para mejorar su rendimiento y abre la puerta al telepilotaje.



Con este vehículo, el ITEAM de la Universidad Politécnica de Valencia ha demostrado que la teleconducción a distancia es posible y segura.

Fig. 44 Reportajes



Fig. 45 Firma de acuerdo de colaboración entre la UPV y la Asociación Española de Vehículos Autónomos

En resumen, el trabajo desarrollado en esta tesina de máster dejó muy en alto el nombre de la UPV, como un centro de investigación puntero a nivel Internacional. Fue tanta su acogida, que el Presidente de la Generalitat, el excelentísimo señor D. Ximo Puig, y el Rector Magnífico de nuestra Universidad, el Dr. Francisco Mora, también se unieron a los cientos de visitantes que quisieron ser parte de esta magnífica experiencia de conducción remota como se observa en la Fig. 46.



Fig. 46 Demostraciones de servicios de conducción remota

Durante toda la semana del evento se realizó una encuesta comparativa entre la conducción remota del coche real y la reproducción del mismo escenario en un simulador perteneciente al iTEAM.

Para esta comparativa se tomó en cuenta la latencia que se registraba durante las pruebas y el evento, en la red móvil de 5G en comparación con diferentes latencias definidas en el simulador.

La Fig. 47 muestra el mecanismo utilizado para calcular la latencia existente en la red de 5G. Se calcularon valores a lo largo de una hora, manteniendo todas las demos de 5G funcionando al mismo tiempo en la red móvil. Se seleccionó este como el peor escenario para el caso de uso, debido a que al aumentar el número de dispositivos conectados en la red podría existir un aumento en la latencia.

Este estudio del peor caso permitió concluir que la conducción remota es perfectamente viable con 5G, por lo que en casos con menor latencia también se puede ejecutar exitosamente.

```

NetworkPinger>ping 172.23.3.1

Pinging 172.23.3.1 with 1 byte of data:
[15:02:36] Reply from 172.23.3.1: byte=1 time=12ms TTL=62
[15:02:37] Reply from 172.23.3.1: byte=1 time=10ms TTL=62
[15:02:38] Reply from 172.23.3.1: byte=1 time=11ms TTL=62
[15:02:39] Reply from 172.23.3.1: byte=1 time=9ms TTL=62
[15:02:40] Reply from 172.23.3.1: byte=1 time=10ms TTL=62
[15:02:41] Reply from 172.23.3.1: byte=1 time=10ms TTL=62
[15:02:42] Reply from 172.23.3.1: byte=1 time=10ms TTL=62
[15:02:43] Reply from 172.23.3.1: byte=1 time=9ms TTL=62
[15:02:44] Reply from 172.23.3.1: byte=1 time=9ms TTL=62
[15:02:45] Reply from 172.23.3.1: byte=1 time=11ms TTL=62
[15:02:46] Reply from 172.23.3.1: byte=1 time=9ms TTL=62
[15:02:47] Reply from 172.23.3.1: byte=1 time=10ms TTL=62
[15:02:48] Reply from 172.23.3.1: byte=1 time=9ms TTL=62
[15:02:49] Reply from 172.23.3.1: byte=1 time=14ms TTL=62
[15:02:50] Reply from 172.23.3.1: byte=1 time=12ms TTL=62
[15:02:51] Reply from 172.23.3.1: byte=1 time=14ms TTL=62
[15:02:52] Reply from 172.23.3.1: byte=1 time=12ms TTL=62

```

Fig. 47 Muestra de la latencia en la red móvil de 5G

En la Fig. 48 se observa la tabulación de los diferentes tiempos de latencia que se experimentaron en la hora cargada. En la parte de la derecha se muestran dos columnas. La primera representa el tiempo de latencia mientras que la segunda columna representa la probabilidad de ocurrencia correspondiente a cada una de las latencias en función del número total de la muestra.

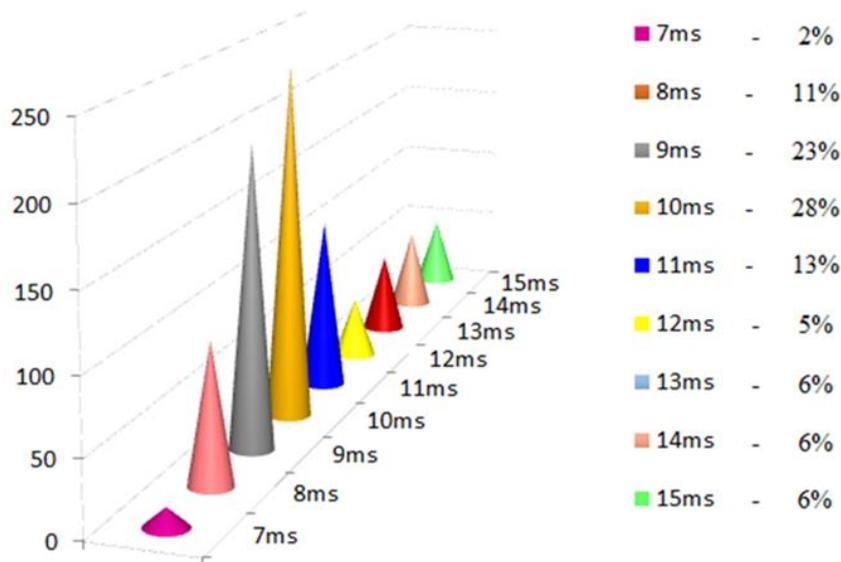


Fig. 48 Tabulación de la latencia experimentada

La Fig. 49 muestra un ejemplo de la variación temporal de dicha latencia. Se puede observar la variabilidad que existe, así como que existe un valor medio cercano a 10 ms, suficiente para satisfacer las necesidades de comunicación del caso de uso.

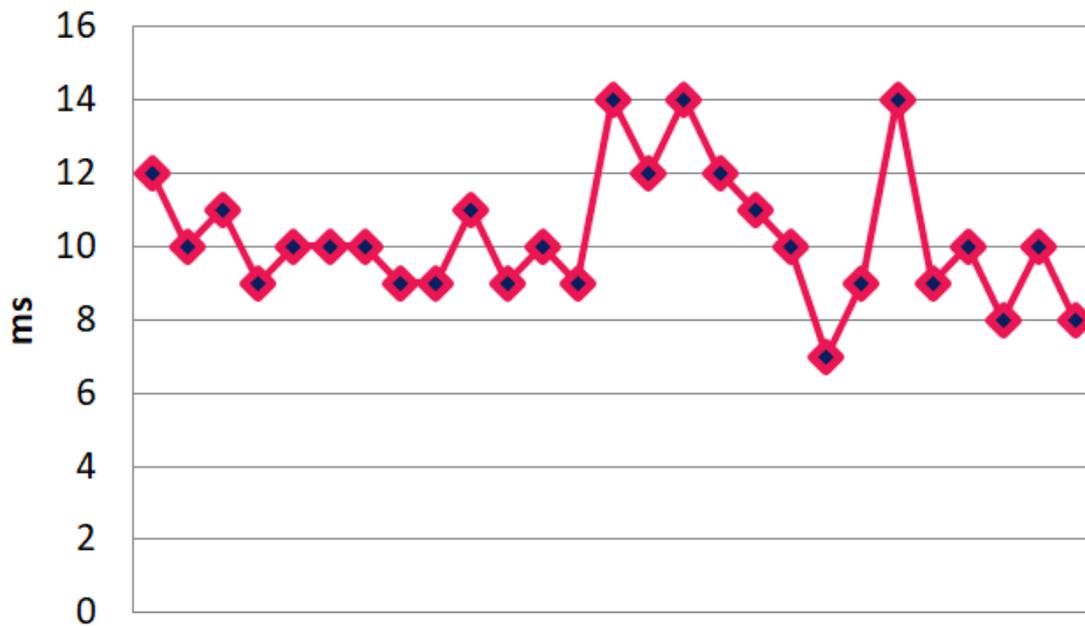


Fig. 49 Latencia a la que tiende la red móvil

Para poder realizar la comparativa entre las experiencias de conducción a diferentes latencias, se realizó una encuesta verbal a los usuarios que condujeron en el simulador.

La metodología fue primero probar la conducción del coche en la demo y luego pasar al simulador y variar las latencias como se observa en la Fig. 50.



Fig. 50 Escenario del simulador

Los resultados obtenidos de las pruebas en el simulador se muestran en la Fig. 51 y Fig. 52. En dicho simulador se establecieron 10 diferentes latencias:

- 10 ms
- 20 ms
- 40 ms
- 50 ms, latencia típica de LTE.
- 70 ms
- 100 ms, latencia característica de 3G.
- 400 ms
- 600 ms

- 800 ms
- 1 s, latencia experimentada en una red 4G cuando el conductor y vehículo se conectan con distintos operadores.

El orden de las mismas fue aleatorio con cada usuario. De esta manera se mejoraba la percepción del momento exacto de degradación por latencia en el momento de conducir. Ya que si a un usuario se le somete a una conducción con un cambio progresivo en las latencias, este aprende a ir conduciendo con latencia.

El usuario calificaba cada experiencia de conducción con un nivel de dificultad, los cuales eran: fácil, medio, complicado e imposible. La Fig. 51 muestra la tabulación de la respuesta de los usuarios en la que se puede observar que la mayoría está de acuerdo que es muy fácil conducir con una latencia de 10ms, pero en cambio con una latencia de 1s se les hace imposible la misma.

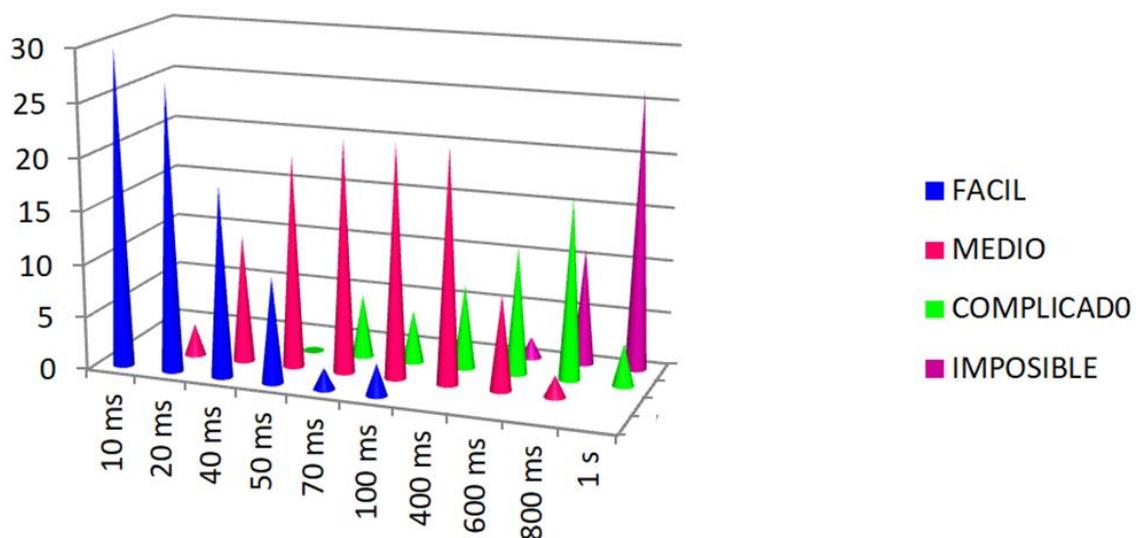


Fig. 51 Tabulación de encuesta a usuarios

De la Fig. 52 podemos deducir un rango aproximado para los diferentes niveles de dificultad al momento de conducir, la intersección de las áreas cubiertas por cada nivel. Estos rangos serían los siguientes:

- De 10ms hasta 40 ms la experiencia de conducción remota es fácil.
- El nivel medio se encuentra comprendido desde los 41 ms hasta los 600 ms.
- El nivel correspondiente a complicado en cambio se encuentra de los 600 ms hasta los 800 ms.
- El nivel en que la conducción remota es imposible es a partir de los 800 ms.

Estos resultados muestran que la única tecnología hasta la fecha que nos ofrece el rango de latencia para una conducción remota fácil y segura es la red móvil de 5G, y que es necesaria una velocidad de transmisión de datos rápida, sin pérdida de paquetes y con el menor retardo.

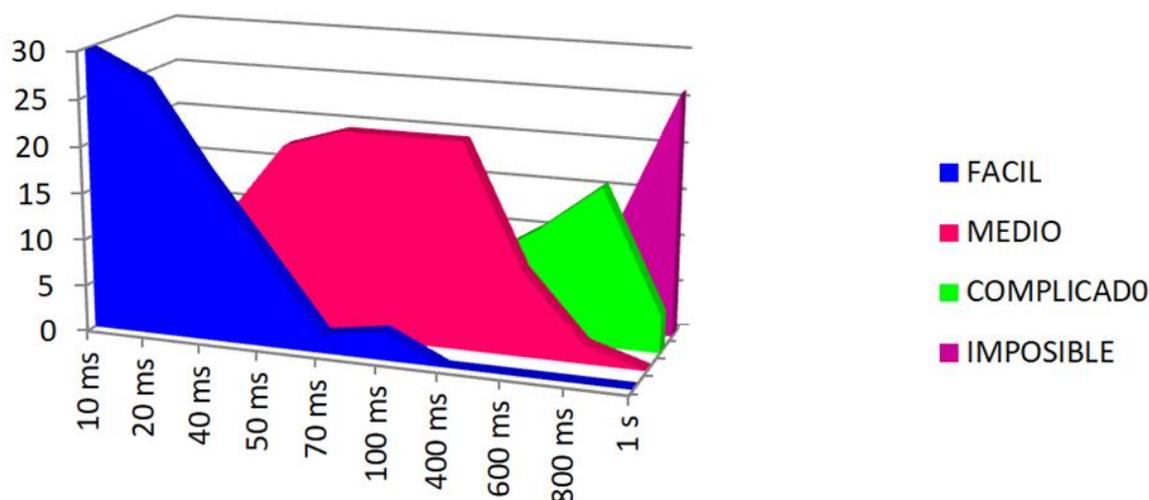


Fig. 52 Rango de niveles de complejidad

V. CONCLUSIONES Y LINEAS FUTURAS

Finalizado el proceso de desarrollo, pruebas y corrección de errores, se consiguió brindar los servicios de conducción remota de manera exitosa, empleando una red móvil de 5G. Para esto se desplegó el escenario físico necesario para realizar la exposición del caso de uso, demostrando el potencial que ofrecen estas redes móviles en la industria automotriz. Además, gracias a la comparativa entre tecnologías que se realizó empleando el simulador del iTEAM, se pudo comprobar que la única tecnología que ofrece en la actualidad las características requeridas como baja latencia y alta confiabilidad para que exista una conducción fácil y segura para el usuario, es la 5G.

También se pudo observar durante el evento que conto con más de 800 expertos a nivel internacional, que esta nueva tecnología es la base para un sin número de proyectos y que las investigaciones que hasta la fecha se presentan con esta tecnología son complementarias. Tal es el caso de Nokia, que tiene un proyecto enfocado en obtener un procesado de imagen más rápido, compatible con una de las líneas futuras de esta tesina de máster.

Como líneas futuras se plantea automatizar completamente los servicios de conducción, o que exista un sistema híbrido donde exista tanto la opción de activar la conducción de manera remota o la conducción totalmente autónoma. Otra proyección es realizar a partir de la latencia óptima para conducción un estudio de la velocidad máxima en la que se puede conducir de manera segura, tanto de forma remota como autónoma. Además, para que el usuario tenga una mejor experiencia de los beneficios que ofrecen las características de baja latencia y alta fiabilidad estas redes, se propone trabajar en el procesado de imagen de las cámaras de manera que la latencia que fue introducida por ellas en el desarrollo de este caso de uso sea eliminada. Esto se puede realizar utilizando cámaras que no contengan ningún tipo de procesado interno y se emplee un software para realizar el mismo de manera más eficiente. Otra propuesta muy interesante sería trabajar ya no en redes

V2N si no en redes V2V, de manera que la conducción pase de ser remota a ser cooperativa y totalmente autónoma con comunicación entre vehículos.

AGRADECIMIENTOS

Agradezco en primer lugar a mi director, Jose Francisco Monserrat por haberme guiado en el desarrollo de mi TFM en el Grupo de Comunicaciones Móviles, ya que este trabajo ha representado una de las más grandes experiencias de mi vida, debido a que me ha permitido trabajar con una tecnología totalmente nueva y, sobre todo, ser partícipe de unos de los más grandes eventos que fueron el *Global 5G Event* y EUCNC2019. En estos eventos pude conocer a muchas personas y proyectos tecnológicos de gran relevancia.

También agradezco a Gerardo Martínez, quien ha estado presente durante todo el desarrollo e implementación del caso de uso de conducción remota, estando presto a ayudarme a solucionar cualquier problema que se me presentaba tanto logístico como experimental.

Doy sobre todo las gracias a mi madre, hermanas, hermano, sobrinos que se encuentran en Ecuador, por apoyarme siempre en mis sueños y darme su amor; también a mis tíos que se encuentran aquí en España porque a pesar de no estar en la misma ciudad se han preocupado mucho por mí, haciendo sentir una parte importante de sus vidas y sobre todo querida.

REFERENCIAS

- [1] J.F. Monserrat, M. García, J.J. Olmos y N. Cardona, *3GPP LTE-Advanced y su evolución hacia la 5G móvil*. MARCOMBO S.A., 2017.
- [2] IJMTTER, *Evolution of Mobile Generation Technology: 1G to 5G and Review of Upcoming Wireless Technology 5G*. Octubre, 2015. [En línea]. Disponible: <https://pdfs.semanticscholar.org/4dfd/40cc3a386573ee861c5329ab4c6711210819.pdfs>
- [3] ITU-R, *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*. September, 2015. [En línea]. Disponible: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf
- [4] 3GPP, *Technical Specification Group Services and System Aspects; Release description; Release 15*. Noviembre, 2017. [En línea]. Disponible: <https://www.3gpp.org/release-15>
- [5] P. Marsch, Ö. Bulakci, O. Queseth, & M. Boldi, (Eds.). *5G system design: architectural and functional considerations and long term research*. John Wiley & Sons, 2018.
- [6] ETSI, *Service requirements for V2X services*. Marzo, 2017. [En línea]. Disponible: https://www.etsi.org/deliver/etsi_ts/122100_122199/122185/14.03.00_60/ts_122185v140300p.pdf
- [7] J. Flores de Vargas, D.Martín. Sacristán y J.F. Monserrat, *5G New Radio, Numerologies and their Impact on V2X Communications*. WAVES., 2018
- [8] Ministerio de Energía, Turismo y Agenda Digital., *Plan Nacional 5G*. Enero, 2018. [En línea]. Disponible: https://avancedigital.gob.es/5G/Documents/plan_nacional_5g.pdf
- [9] IEEE, *5G as Enabler for Industrie 4.0 Use Cases: Challenges and Concepts*. September 2018. [En línea]. Disponible:

- https://www.researchgate.net/publication/328995487_5G_as_Enabler_for_Industrie_40_Use_Cases_Challenges_and_Concepts
- [10] 3GPP, *Technical Specification Group Services and System Aspects; Feasibility Study on New Services and Markets Technology Enablers, Stage 1 Release 14*. September, 2016. [En línea]. Disponible: <https://www.3gpp.org/release-14>
- [11] ROS. org, Documentation. Julio, 2014. [En línea]. Disponible: <http://wiki.ros.org/es>
- [12] ROS. org, *Distributions*. Julio, 2014. [En línea]. Disponible: <http://wiki.ros.org/Distributions>
- [13] ROS. org,, *Understanding Nodes*. Julio, 2014. [En línea]. Disponible: <http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>
- [14] ROS. org, *Installation*. Julio, 2014. [En línea]. Disponible: <http://wiki.ros.org/indigo/Installation/Ubuntu>
- [15] ROS. org, *Rbcar_common*. Julio, 2014. [En línea]. Disponible: https://github.com/RobotnikAutomation/rbcar_common
- [16] Unity, 2017. Unity Documentation - 2D or 3D projects. [En línea]. Disponible: <https://docs.unity3d.com/> [Cited: June 10, 2017].
- [17] ROS. org, *Robotnik_msgs*. Julio, 2014. [En línea]. Disponible: https://github.com/RobotnikAutomation/robotnik_msgs
- [18] ROS. org, *Package Summary*. Julio, 2014. [En línea]. Disponible: http://wiki.ros.org/catkin#Installing_catkin
- [19] ROS. org, *Configuring and Using a Linux-Supported Joystick with ROS*. Julio, 2014 [En línea]. Disponible: <http://wiki.ros.org/joy/Tutorials/ConfiguringALinuxJoystick>