NEXT GENERATION INTERNET

Grant Agreement No.: 957216
Call: H2020-ICT-2018-2020

Topic: ICT-56-2020
Type of action: RIA

inGenious

D3.4 Bio-haptic and XR-enabled IoT devices

Revision: v.1.0

Work package	WP 3
Task	Task 3.4
Due date	31/07/2022
Submission date	29/07/2022
Deliverable lead	NOK
Version	1.0
Editors	Ignacio Benito Frontelo (NOK), Jaime Ruiz Alonso (NOK)
Authors	Jaime Ruiz Alonso (NOK), Ignacio Benito Frontelo (NOK), Manuel Fuentes (5CMM), Miguel Cantero (5CMM), Manuel Pasamontes (NED), Raúl Lozano (UPV)
Reviewers	Carsten Weinhold (BI), Nuria Molner (UPV), Efstathios Katranaras (SEQ)
Abstract	In this deliverable, we describe development and integration of immersive devices such as Head-Mounted Displays and haptic gloves. All these advancements will be gathered in this document as well as demonstrated and validated in WP6.
Keywords	Internet-of-Things, 5G, IoT Connectivity, C-IoT, Edge IoT Sensors, Immersive services

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	29/07/2022	EC version	See author list

Disclaimer

This iNGENIOUS D3.4 deliverable is not yet approved nor rejected, neither financially nor content-wise by the European Commission. The approval/rejection decision of work and resources will take place at the Final Review Meeting planned in July 2023, after the monitoring process involving experts has come to an end.

The information, documentation and figures available in this deliverable are written by the "Next-Generation IoT solutions for the universal supply chain" (iNGENIOUS) project's consortium under EC grant agreement 957216 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	R*	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to iNGENIOUS project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

Executive Summary

This document describes how immersive displays and haptic devices are being used in iNGENIOUS project. These IoT devices are closer to the end user and provide and enhanced experience for the AGV use case described and proposed in the project.

Such devices play a key role in enabling 5G IoT scenarios that cover the autonomous and supervised modes of operation in the case of AGVs. Two complementary implementations of controlling remote vehicles in both autonomous and manual modes are presented to narrate a single story in which 5G infrastructure and immersive glasses equipped with haptic controllers can make life safer and easier to people managing industrial AGV activities in the environment of a seaport.

Head Mounted Display applications are described in detail as well as their interactions with the rest of the surrounding network and architecture so that the reader is able to understand the big picture and get a clear view of what is possible in terms of user experience, and which are the benefits of using this technology.

The haptic gloves and the mixed reality technology provide a new level of immersivity by combining 360-degree environments, real-time video and feedback in different senses than the sight or hearing.

Describing the process to collect both objective KPI measurements and subjective results are also an important part of the objectives of this deliverable.

The relationship with the AGV use case in the project as well as the means which are required to integrate all the pieces with a single demonstration in the environment of a mmW 5G network are explained.

Finally, an exploitation view of the technologies in the scope of the ~~scope of~~ this project~~ed~~ and beyond is briefly depicted so that future industrial work and initiatives can arise from the ideas of the project.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Table of Contents

Co-funded by the Horizon 2020
Framework Programme of the European Union

List of figures

Co-funded by the Horizon 2020
Framework Programme of the European Union

Abbreviations

ACK	Acknowledge message
ACR	Absolute Category Rating
AGV	Automated Guided Vehicle
AMF	Access and Mobility Management Function
AR	Augmented Reality
DL	Downlink
E2E	End-to-end
HMD	Head Mounted Display
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IoT	Internet of Things
IP	Internet Protocol
ITU	International Telecommunication Union
KPI	Key Performance Indicator
LTE	Long-Term Evolution
M2M	Machine-to-Machine
MEC	Multi-access Edge Computing
MIMO	Multiple-Input Multiple-Output
mmW	Millimetre Wave
MR	Mixed Reality
NAT	Network Address Translation
NR	New Radio
OS	Operating System
POE	Power Over Ethernet
QoS	Quality of Service
RAN	Radio Access Network
RTP	Real-time Transport Protocol
RTT	Round Trip Time
Rx	Receive
SA/NSA	Stand Alone / Non-Stand Alone
SW	Software
ToD	Tele-operation Driving
Tx	Transmit
UC	Use Case
UDP	User Data Protocol
UE	User Equipment
UL	Uplink
UPF	User Plane Function
VR	Virtual Reality
XR	Extended Reality

1 Introduction

1.1 Objective of this Deliverable

This deliverable will describe in detail the Haptic and Head Mounted Display (HMD) devices used in this project which are part of the IoT ecosystem in the scope of the iNGENIOUS use cases.

Both local and End-to-End (E2E) architecture of the involved software and hardware will be depicted together with the integration activities required to interact with the rest of technologies and components in the whole picture.

This document is organized in seven main chapters. The first one is the introduction. The second one is focused on the HMDs which will offer a mixed reality view from the user perspective and their adaptation to the specific utility that they are given. The third one will describe the haptic devices in detail, and which is their purpose in the use cases. Then, the fourth chapter will explore a system vision taking the User Story paradigm as reference for analysing the integration steps in depth. Finally, section five will pay attention to the exploitation, business opportunities as well as the lessons learned during the project while the seventh chapter will explore the conclusion and summary of this analysis.

A continuous single use case will be developed as a blended interaction of two technologies and two situations that may and will happen with autonomous vehicles: autonomous mode driving and Tele-operation driving. Chapter 3 will focus on the case in which the Automatic Guided Vehicle (AGV) is working in autonomous mode. In this part of the story, haptic devices combined with HMD technology will be presented to help the end user to control and supervise the movement over a predefined pre-programmed track. Chapter 2 will focus on the point in time in which a human being needs to take control remotely of the AGV to drive it outside the predefined track to avoid some obstacle or whatever not foreseen programmed event. Both systems are simultaneously connected to the AGV even though only one of them is performing control actions. The coordination of both activities will not be via a single MR application integration but by using a simple interaction between them so that it is clear who is the master of the control.

1.2 Role of MR and Haptic devices in the AGV Use Case

The AGVuse case is devoted to have AGVs which are supervised and controlled in an advanced fashion using 5G to take benefit of the better specifications of these networks in terms of latency and throughput.

It is critical to have control of AGVs in industrial environments with almost 100% of accuracy and reliability. Using Wi-Fi or other radio networks different than 5G will result in an unacceptable variation of latencies and network throughput that might lead to undesired accidents costing money and, in some cases, potential injuries to people. This results in cost issues that can be

Co-funded by the Horizon 2020 Framework Programme of the European Union

avoided by having a reliable protocol running on top of a network deployment which ensures latencies and throughputs suitable for the use case, even making use of advanced slicing or quality of service priority schemas that are only available under 5G deployments.

The end user perspective plays a fundamental role in this use case as while most of the time, the AGVs will behave autonomously, in some cases they will need to be operated by humans. The use of classical displays and input devices such as mouse or keyboards is the most typical way to address the end user needs to communicate with AGVs and the rest of the system. However, it lacks simplicity and immersivity from whatever point of view we may think. So, simpler, and more immersive ways of exploring the environment are required. In the AGVs use case implementation this is a key point of differentiation from other possible implementations.

The use of such devices will influence and condition the design and the protocols adaptation of the rest of the components in the use case (network, AGVs, UEs, etc). This means that we need to reflect and explain how these devices are adapting to the rest of components, but also the other way round, as the network might be required to be configured in a certain way or even the AGVs upper layer of communications and protocols will require slight changes to be operated using these new devices.

One clear example is the need for enabling Edge infrastructure so that video flows can reach the HMDs almost in real time. It is not acceptable having delays of 1 second if we are driving a small vehicle through a port in which there are valuable and sensible materials and human workers which should be able to work in a safe environment.

1.3 End-to-end Description

1.3.1 Architecture overview

In this section an overall view of the Architecture from and End-To-End perspective will be described. To understand which is the whole picture it is important to start by first highlighting the use case in which the devices will be used In the AGV use case, an AGV has a predefined route, and it is controlled using haptic gloves to move from point A to point B. Then, the vehicle finds and object in its way so that it cannot follow the programmed track. A remote driver takes control of the AGV and uses a steering wheel and pedals altogether with an HMD in which a merged VR and real-time video representation is projected in that device. Finally, the remote driver returns control to the AGV so that the rest of the predefined track can be completed using the haptic gloves and the autonomous capabilities of the robot. This is depicted in Figure 1 below.

Co-funded by the Horizon 2020
Framework Programme of the European Union

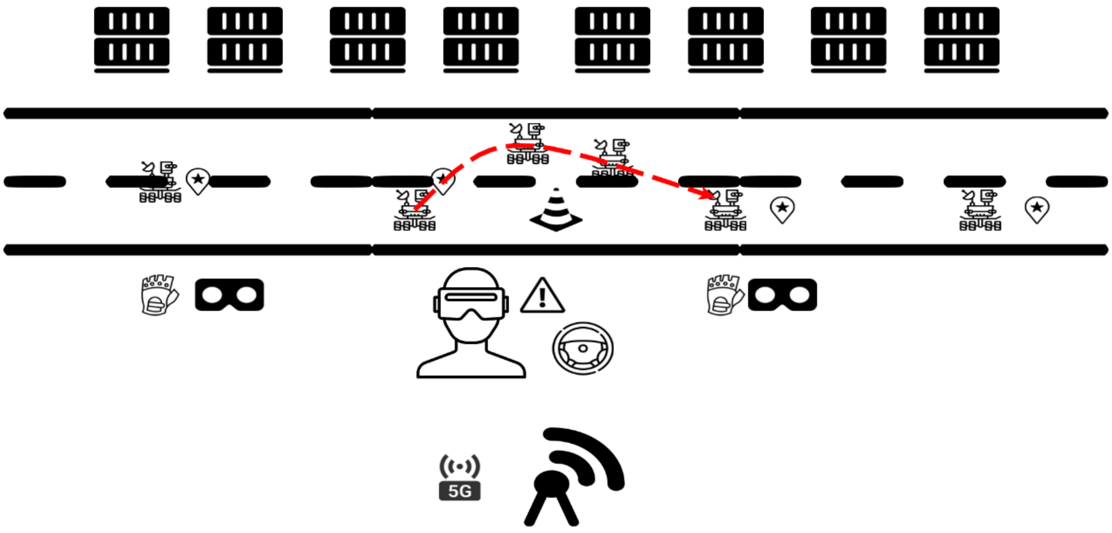


Figure 1. AGV Use Case in iNGENIOUS project

Considering the whole E2E architecture picture, the devices subject to this document and the architecture pieces related can be found below surrounded by red boxes. In this use case, the main actors are the Mixed Reality (MR) Cockpit and the Haptic Gloves devices which are in the right bottom corner, but they also have an explicit interaction with the rest of red boxes highlighted in the figure: Factory Robot, Automated Guided Vehicle, 5G Modem, 5G RAN, MEC, 5G Core and of course, the Smart & Tactile IoT Apps developed.

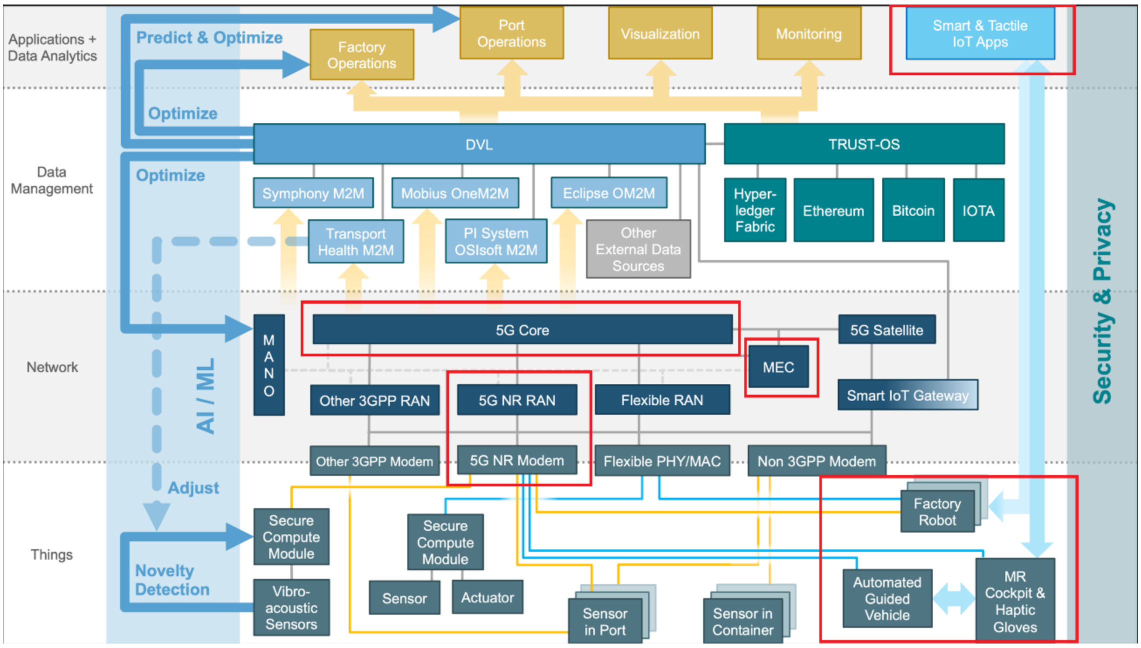


Figure 2. Components in iNGENIOUS

If we go in more detail and just paying attention to the before highlighted elements, we have a closer picture of how the components are connected.

From the perspective of the flows which are the responsibility of the HMD layer, the following picture depicts an overall view of the architecture.

Co-funded by the Horizon 2020 Framework Programme of the European Union

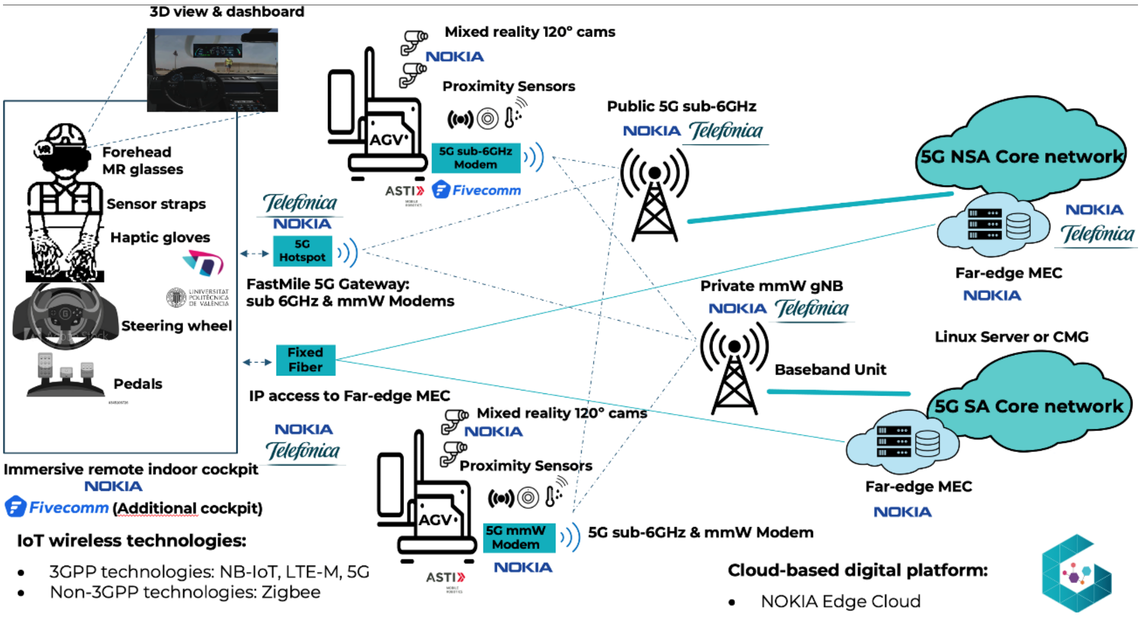


Figure 3. 5G ecosystem connectivity to HMD and Haptic Devices

There are two flows for the control commands which are the blue and red ones. They are used to transmit the movements to the AGV and to receive their acknowledgment message for everyone containing the telemetry information. A third flow shows the video which is transmitted in one direction. Finally, a security signal resulting from the detection of an obstacle by the depth camera is sent and results in vibration in the haptic glove.

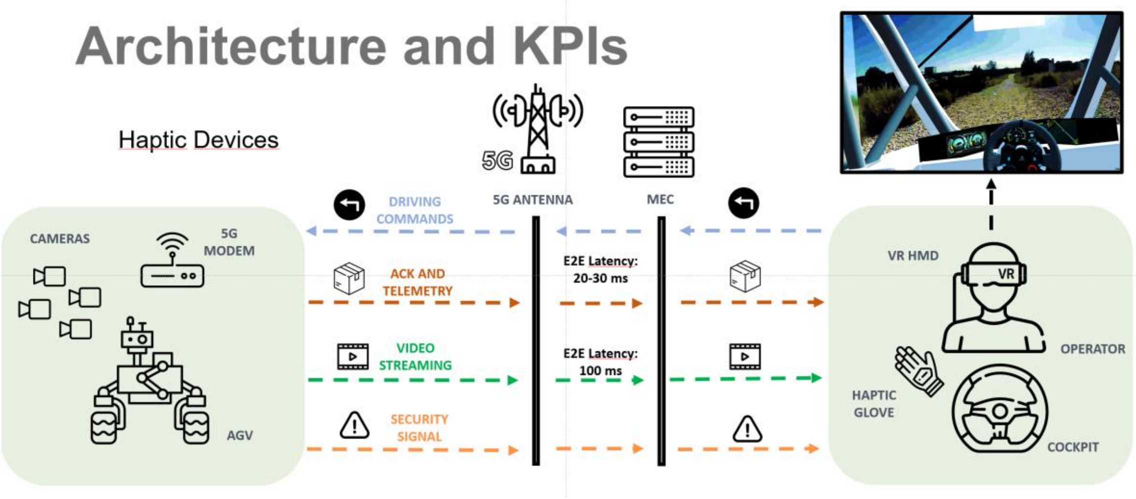


Figure 4. Data flows and KPIs in the global architecture

1.3.2 Scenarios

In this section we enumerate the possible scenarios and sub scenarios of the AGV use case explained from a User Story perspective.

Scenario 1: AGV moving autonomously, no video feed

In this scenario we will not need an HMD, however a Haptic device might be used to start/stop the movement of the AGV and/or to resume the movement to the last point in the route.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Scenario 2: AGV moving autonomously, video feed and telemetry info

In this scenario, the Cockpit is connected and sending remote driving commands which are ignored in the AGV. This means that, as the remote driver does not have the control, the commands are only sent to receive telemetry data and to be prepared to take the control at any moment. However, telemetry is sent back in the payload of the ACK of the commands. Video is received so that the person in the cockpit is a spectator. This mode can be considered as supervised mode.

Scenario 3: AGV encounters problem and asks for remote control, control granted and started

For this situation, a telemetry signal is received indicating that the human operator needs to take control to drive remotely the AGV. Once the control is taken, the control commands being sent from the cockpit will be executed by the AGV. The mode of driving changes to “Takeover” mode.

Scenario 4: AGV remotely controlled from HMDs and Logitech G29 wheel

Now, the commands are being sent from the cockpit part to the AGV with a period of 100ms. This period can be changed in the application acting as the smart component in terms of control. The commands are sent through the MEC to the AGV, that will acknowledge them by inserting the telemetry information into the cockpit. This telemetry information is used to calculate and render the KPIs in the Mixed Reality environment and to provide a realistic movement of the wheel in the car interior. UDP video continues to be normally transmitted during this phase.

Scenario 5: AGV bring control back to AGV

Once the obstacle has been passed, the remote driver can release the control and give it back to the AGV. Controls in the steering wheel are used to achieve this situation, for which the cockpit will send the information and wait until it has the same ACK UDP telemetry with this data to ensure that the remote driver does not need to retain control.

Co-funded by the Horizon 2020
Framework Programme of the European Union

2 Immersive HMD Devices

In this chapter we will explore in detail the couple of market devices we have selected to meet the use case requirements.

2.1 Overall architecture

The overall architecture Figure 5 can be seen below. Starting from the upper section of the schema the HMD is connected to the PC via OpenXR with an Oculus or Varjo layer depending on the manufacturer of the device. OpenXR was selected in order to be able to extend the application to the maximum number of different devices and having at the same time a unique application for all of them. Unity 2021 LTS is the version being used to ensure future maintenance of the application.

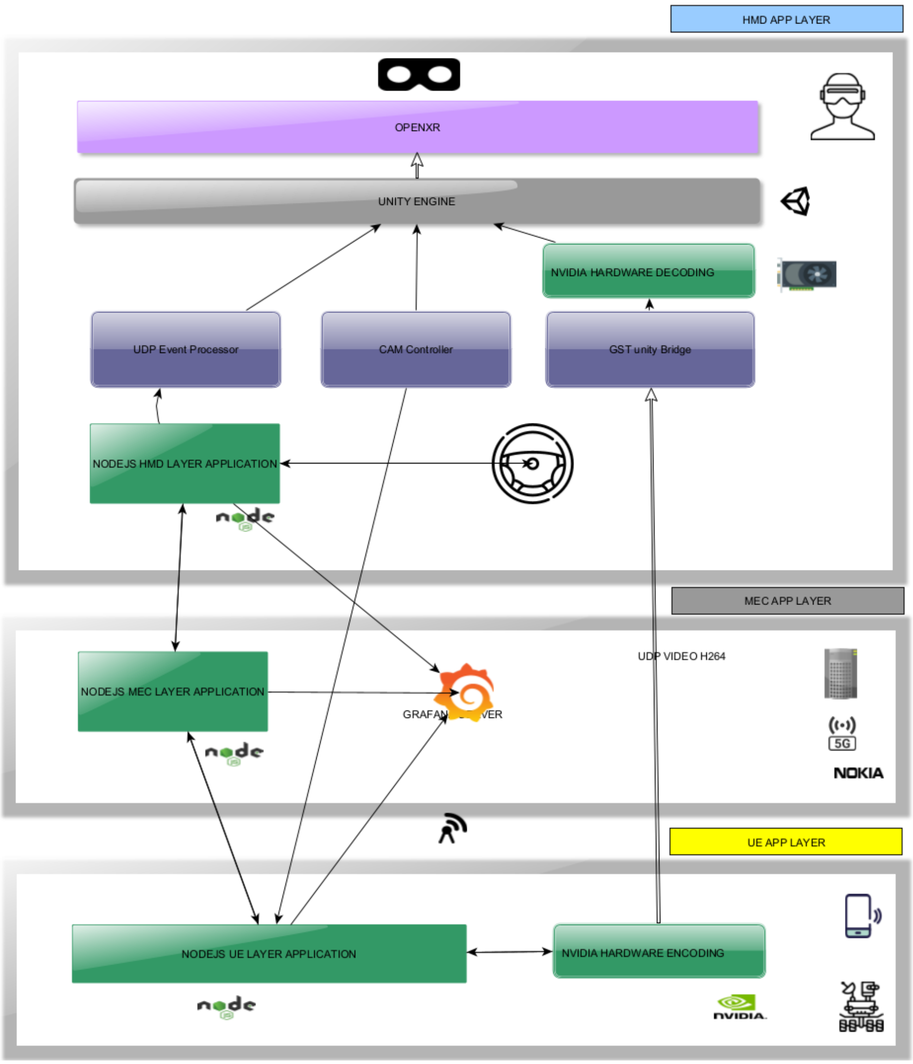


Figure 5. Three-layer overall architecture

The application runs in link mode, meaning that the HMD is always connected to a powerful PC via cable. The connection is different in the Varjo case compared to the Quest2 one. For VarjoXR3 two high-speed USB-C cables plus two DisplayPort cables are required to be connected to the PC and the graphic card respectively. The graphic card should support 2 simultaneous

Co-funded by the Horizon 2020 Framework Programme of the European Union

high-speed DisplayPort output. This only happens in NVIDIA for desktop models of RTX 2080, RTX3070 and RTX3080.

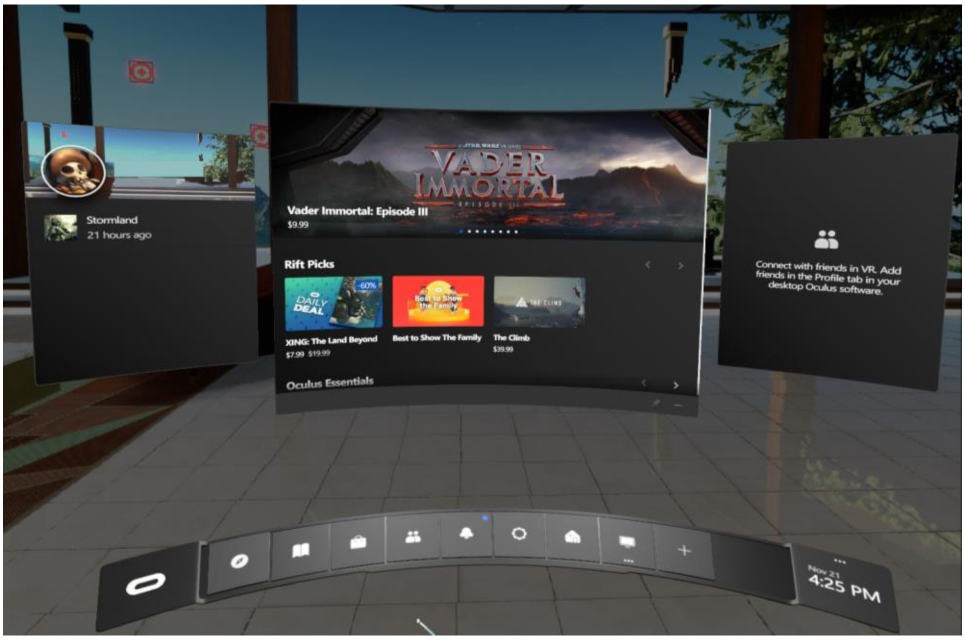
Once everything is connected, the speed is verified using the Varjo Base application. After that, the communication with the unity application can be done either via OpenXR or using SteamVR layer. For the Quests a single high-speed USB-c to USB3.0 cable is required.

2.2 Oculus Quest2

Oculus Quest2 is probably the most sold device in the market in terms of public market segment. At a reasonable price it offers both the possibility of working without cables using the Android layer behind it or directly linked to a powerful PC in which the latency and performance is quite better.

Figure 6. Oculus Quest 2

To make Oculus Link feature properly work in a PC, the Oculus application needs to be installed and opened with the Meta account which is associated to the device. Furthermore, it needs to be initiated with internet connectivity. It has been observed that sometimes there is some instability or strange behavior if the PC and the HMD are not connected to their Oculus Meta account. Once the cable is connected, we can see that the original menu from the Quest2 changes to another Oculus menu.



Connect with friends in VR. Add friends in the Profile tab in your desktop Oculus software.

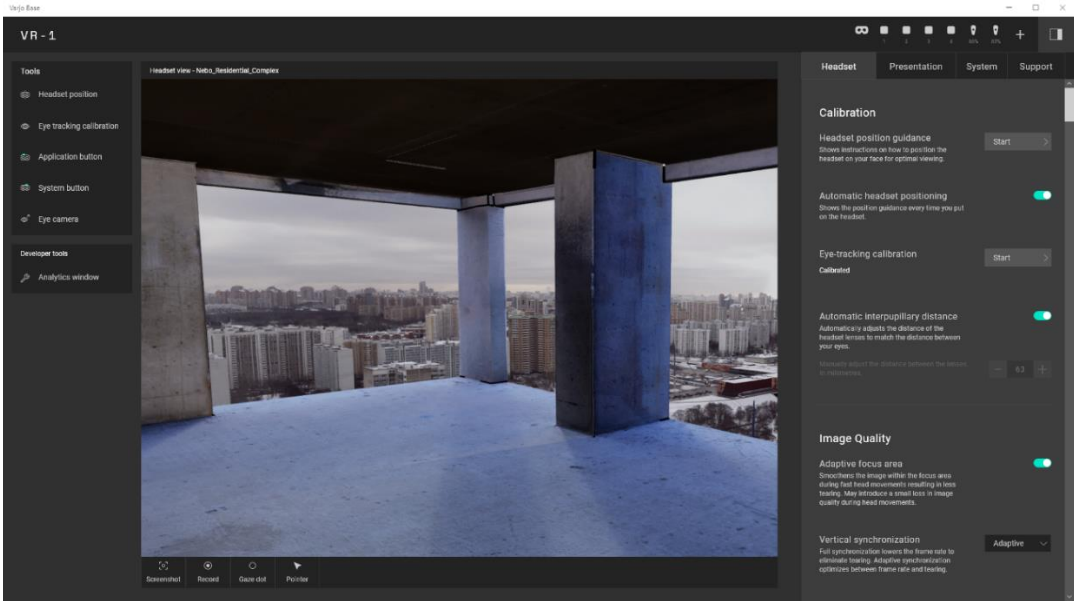
Figure 7. Oculus Link Virtual Environment

An additional flavor of the application was developed in the case of Quest2. The standalone version working in Android inside Quest2. The advantage of it is the fact that no cable is required. OpenVPN application is additionally required to be installed in this case. The disadvantage of this version is the high variability in terms of latency as it depends on the WI-FI connection. Another point is the speed in the hardware video decoding. If higher bitrates and four streams were required, some fluctuations were observed.

2.3 VarjoXR3

VarjoXR3 is the state of the art in Very High-Resolution HMDs. The idea behind it is that the area which is in the centre of the eyes is rendered with higher resolution than the rest. It requires two parallel high speed DisplayPort cables and USB-C high speed connection to work, because a larger amount of data is moved from the PC to the HMD.

To achieve this, high-end NVIDIA graphic cards are required. RTX3070 desktop model or better cards are needed. Two Display-Port connections will be used and most of the computational power will be used. Varjo Base software is the layer that will handle and allow the connectivity of this device to other VR platforms such as SteamVR or openXR.



Calibration

Headset position guidance
Shows instructions on how to position the headset on your face for optimal viewing.
Start

Automatic headset positioning
Shows the position guidance every time you put on the headset.

Eye-tracking calibration
Calibrated
Start

Automatic interpupillary distance
Automatically adjusts the distance of the headset lenses to match the distance between your eyes.
Manually adjust the distance between the lenses in millimetres.
63

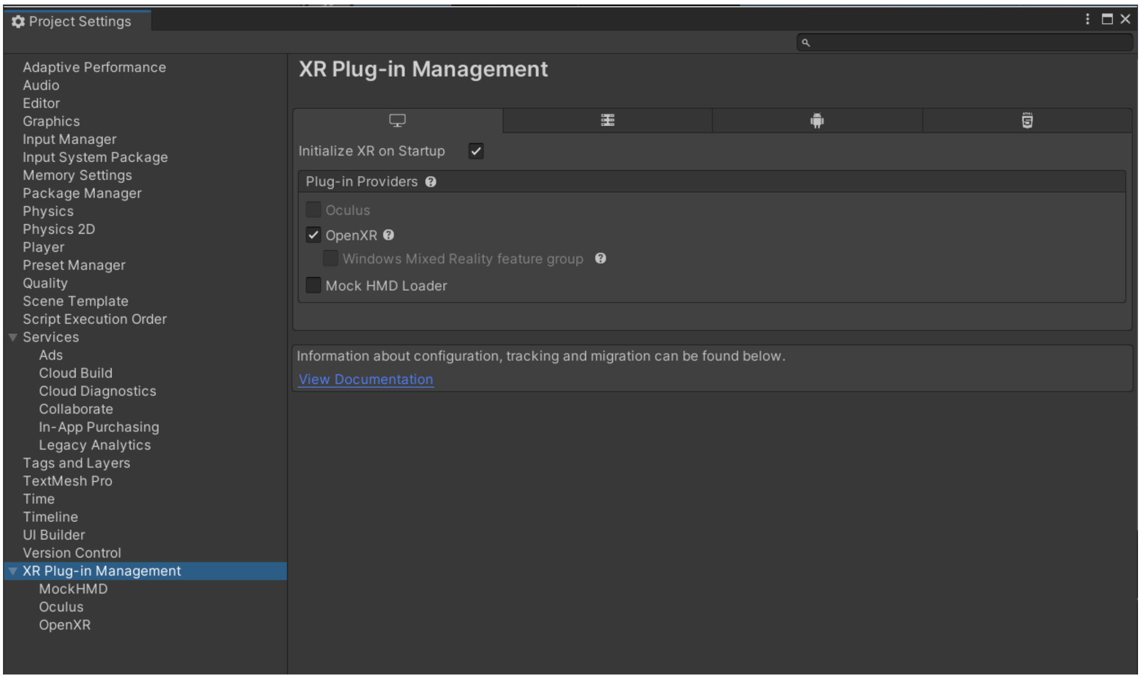
Image Quality

Adaptive focus area
Smoothens the image within the focus area during fast head movements resulting in less tearing. May introduce a small loss in image quality during head movements.

Vertical synchronization
Full synchronization lowers the frame rate to eliminate tearing. Adaptive synchronization optimizes between frame rate and tearing.
Adaptive

Figure 8. Varjo Base mediator software

The development of the application is identical to the case of Oculus as a single Unity application will be used. The difference will come in the Player configuration for the Project Setting in which we will not tick on the Oculus box in the XR Plug-in Management.


Project Settings

Adaptive Performance
Audio
Editor
Graphics
Input Manager
Input System Package
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
- Ads
- Cloud Build
- Cloud Diagnostics
- Collaborate
- In-App Purchasing
- Legacy Analytics

Tags and Layers
TextMesh Pro
Time
Timeline
UI Builder
Version Control
XR Plug-in Management
- MockHMD
- Oculus
- OpenXR

XR Plug-in Management

Initialize XR on Startup ☑

Plug-in Providers

□ Oculus
☑ OpenXR
□ Windows Mixed Reality feature group
□ Mock HMD Loader

Information about configuration, tracking and migration can be found below.
View Documentation

Figure 9. Unity configuration for XR Plug-in using OpenXR

2.4 Latency considerations

Latency is a critical aspect when considering remote driving. As explained in [4] it is recommended that the Round-Trip Time in experiencing the video feedback for a remote user driving a vehicle at low speeds should be under 125-150 ms for the AGV case. ID represents the impairment levels caused by end-to-end delay or responsiveness with 1.0 as no impairment. Please see Figure 10 showing the graph in which the effect is plotted for several scenarios.

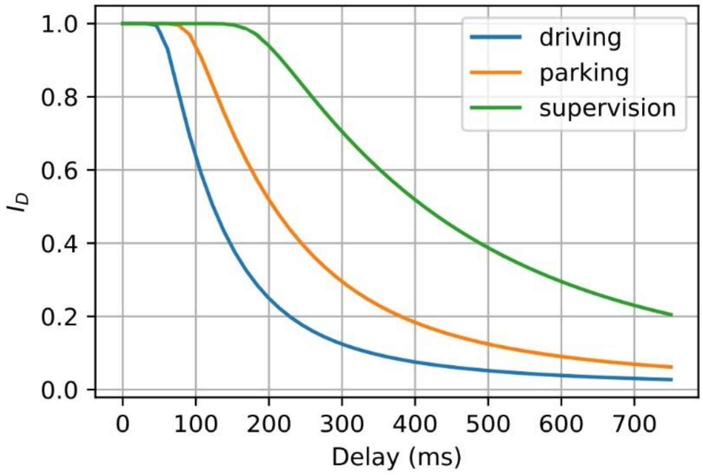


Figure 10. Maximum delay allowed based on speed and deviation curves

In the case of Oculus Quest 2 device there are two ways of running the MR application. First one consists of building an android application which will run in the HMD while the HMD connects either to the mobile phone or to the 5G MEC via WI-FI. The second alternative is connecting the HMD with a USB-C high speed cable to a PC which has a powerful GPU card inside it.

Co-funded by the Horizon 2020
Framework Programme of the European Union

If we use the first alternative, we have two aspects which will increase the End-To-End latency. First is the WI-FI communication compared to an ethernet cable communication direct or almost direct to the MEC. The second one is the time to decode the video streams as in the PC case we will use accelerated hardware decoding providing almost instantaneous decoding time (round 1ms). Because of this, the choice has been using the “link” approach for Oculus Quest 2. The unity application will be executed inside the PC a will offer superior performance.

2.5 Unity Application XR Description

In the case of Remote Driving application, the same unity application will be used for both HMDs. The only difference is the Varjo application layer that is required to run our software due to license requirements.

The XR application has the following components:

1. **Unity scene.** The unity scene is a combination of a car interior and exterior model which was initially modified using blender software and finally imported into unity.

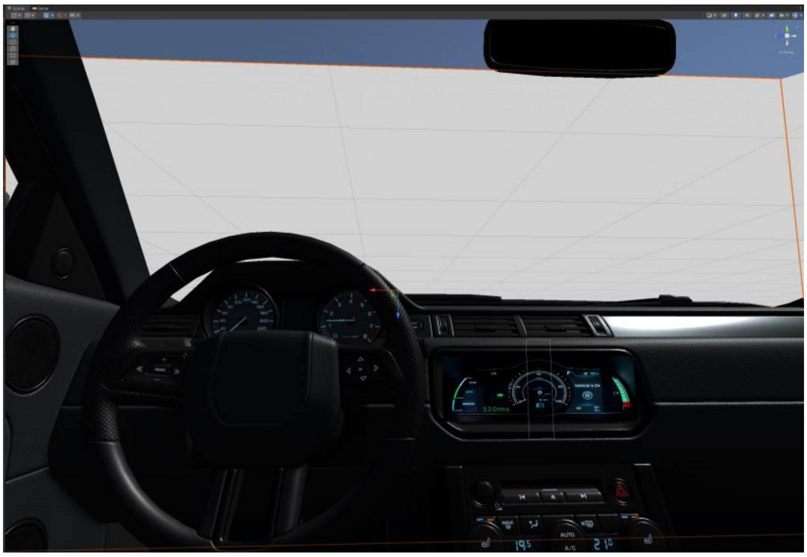


Figure 11. Unity scene including dashboard, model and video planes

2. **The gstreamer unity plugin.** This plugin will allow us to use the gstreamer software to receive the video stream using a generic pipeline to decode the H264 UDP flow we are receiving from the cameras. These video streams are projected into 4 rectangles that are in front and the sides of the car.
3. **The UDP C# scripts for receiving the telemetry information.** These scripts will receive UDP telemetry ACKs commands forwarded from the nodeJS cockpit application which is running in the same PC using port 8051. This information is rendered in the UI components in the car. The steering wheel is also moved according to the movement of the physical Logitech G29 steering wheel which is being used to remotely drive the AGV.

Co-funded by the Horizon 2020
Framework Programme of the European Union

From the point of view of target build platform, the application can be built on two OS: Android and Windows, but as explained before only Windows application will be compiled. Windows target is the application that will be executed using the high-speed USB-C connection cable in the Quest2 link setup. A standalone executable is generated that can be directly run from windows.

We selected OpenXR as final plugin for rendering in the player. In any case it is important that Oculus VR Application has been previously started with internet connectivity.

For the VarjoXR3 we will use SteamVR as the layer to communicate with the HMD apart from the Varjo software.

The application started being developed on Unity 2018 but has been migrated to Unity 2021.3.3.f1 LTS version to ensure compatibility and maintenance in the future.

2.5.1 Video Streaming Reception

2.5.1.1 Gstreamer plugin integration into unity and parameters

A gstreamer pipeline is used to receive the video streams from the AGV. This video is projected into three planes which are in front of the car model, and which represent the left camera view, the front camera view, and the right camera view. Then, an additional plane is in the rear mirror showing the rear view corresponding to the fourth camera. Video is horizontally flipped in reception, so it is the realistic view inside a physical vehicle.

This is the pipeline which is linked to the Unity GameObject for the front rectangle:

```
udpsrc port=3001 ! application/x-rtp,media=video,clock-
rate=90000,encoding-name=H264,payload=96 ! rtph264depay ! decodebin3 !
glupload ! glcolorconvert ! video/x-
raw(memory:GLMemory),format=RGBA,texture-target=2D ! fakesink sync=0
qos=1 name=sink
```

The rest of the pipelines are identical except for the UDP port in which the stream is received: 3002 for the left one, 3000 for the right one and 3003 for the rear one.

There are two flavors for gstreamer plugin. The one being used is the windows version which make use of 2 DLL:

- GstUnityBridge.dll
- DvbCssWc.dll.

The C# script offers several configuration parameters and links to gstreamer binaries. It is important to remark that one recent 64-bit version of gstreamer 1.0 needs to be installed in the PC in which the standalone Unity executable or Unity 2021 environment will run.

In addition, several system variables need to be set:

- `PATH C:\gstreamer\1.0\msvc_x86_64\bin`
- `GSTREAMER_1_0_ROOT_MSVC_X86_64 C:\gstreamer\1.0\msvc_x86_64\`

Co-funded by the Horizon 2020
Framework Programme of the European Union

- `GST_SDK_PATH C:\gstreamer\1.0\msvc_x86_64\`

2.5.1.2 Remote camera protocol – Remotely controlling camera streaming

The server part of the application is running in the Jetson Xavier NX Board located in the AGV. It is accessible through a REST API protocol that will initiate the UDP video flows towards the HMD.

This API can be invoked from the HMD perspective to control and initiate the UDP video streaming to be projected in the final scene.

The remote camera protocol node application allows to:

- List individual capabilities for a single camera in the jetson board

  ```
  http(s)://<JETSON_BOARD_IP>:3000/info/getCameraCapabilities?device=<VIDEO_DEVICE>
  ```

- Start streaming of a given camera

  ```
  http(s)://<JETSON_BOARD_IP>:3001/action/startCameraStream?device=<VIDEO_DEVICE>0&height=<HEIGHT>&width=<WIDTH>&framerate=<FRAMERATE>&bitrate=<BITRATE>&mtu=<MTU>&ip=<DESTINATION_IP>&port=<DESTINATION_PORT>
  ```

- Stop streaming of a given camera

  ```
  http(s)://<JETSON_BOARD_IP>:3001/action/stopCameraStream?device=<VIDEO_DEVICE>
  ```

- Check the current status of the cameras in the system

  ```
  http(s)://<JETSON_BOARD_IP>:3001/info/getActiveCameras
  ```

- Record the streaming from one camera into a file

  ```
  http(s)://<JETSON_BOARD_IP>:3001/action/startCameraRecording?device=<VIDEO_DEVICE>0&height=<HEIGHT>&width=<WIDTH>&framerate=<FRAMERATE>&bitrate=<BITRATE>&basedir=<BASEDIR>
  ```

- Stop recording video from one camera

  ```
  http(s)://<JETSON_BOARD_IP>:3001/action/stopCameraRecording?device=<VIDEO_DEVICE>
  ```

- Show the list of recordings

  ```
  http(s)://<JETSON_BOARD_IP>:3001/info/getRecordingList?basedir=<BASE_DIR>
  ```

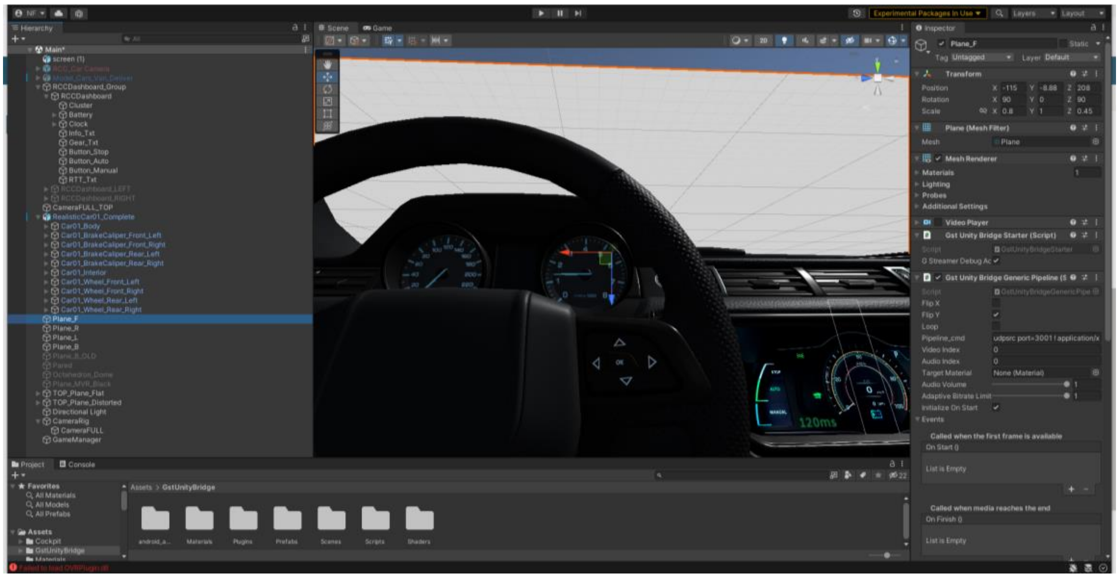
The table below describe the meaning of the parameters and shows some examples:

Parameter name	Description	Example
<JETSON_BOARD_IP>	IP Address in the VPN	10.8.0.3
<VIDEO_DEVICE>	Linux video device for the camera	/dev/video0
<HEIGHT>	Video Resolution (vertical)	1080
<WIDTH>	Video resolution (horizontal)	1920
<FRAMERATE>	FPS for video encoder	60
<BITRATE	Max bit rate for video (CBR)	4000000
MTU	Transport Unit Size for RTP	1278
DESTINATION_IP	IP of the HMD PC in VPN	10.8.1.4
DESTINATION_PORT	Reception port for UDP stream	3003
<BASEDIR>	Directory for recordings	/mnt/ramdisk/Capture1

Co-funded by the Horizon 2020 Framework Programme of the European Union

2.5.2 Unity Application Model

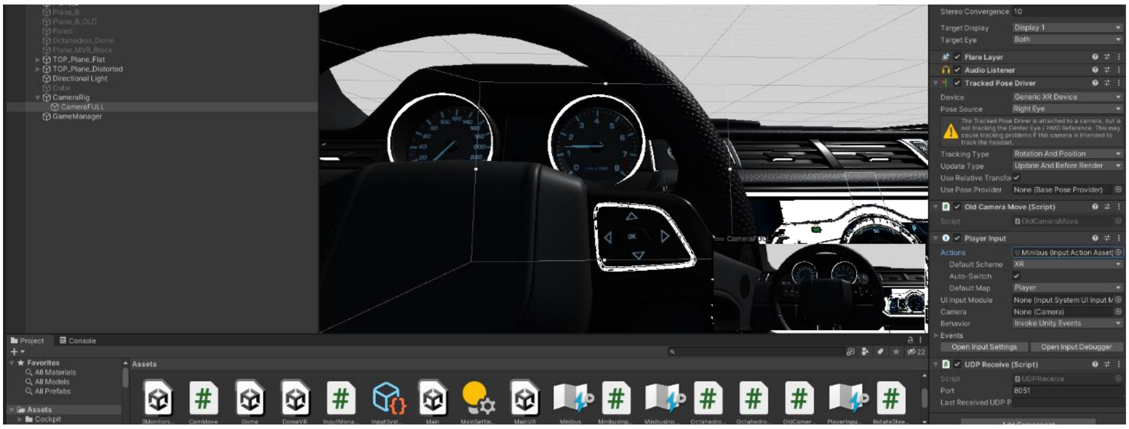
The Unity Application model is composed of a realistic car model, plus four additional planes in which the video is projected plus a dashboard for rendering some telemetry information.



```
Inspector
Plane_F                          Static
Tag Untagged     Layer Default
Transform
Position   X -115   Y -8.88   Z 208
Rotation   X 90     Y 0       Z 90
Scale      X 0.8    Y 1       Z 0.45
Plane (Mesh Filter)
Mesh                 Plane
Mesh Renderer
Materials            1
Lighting
Probes
Additional Settings
Video Player
Gst Unity Bridge Starter (Script)
Script               GstUnityBridgeStarter
G Streamer Debug Ac  ✓
Gst Unity Bridge Generic Pipeline (S
Script               GstUnityBridgeGenericPipe
Flip X
Flip Y               ✓
Loop
Pipeline_cmd         udpsrc port=3001 ! application/x
Video Index          0
Audio Index          0
Target Material      None (Material)
Audio Volume         1
Adaptive Bitrate Limit 1
Initialize On Start  ✓
Events
Called when the first frame is available
On Start ()
List is Empty
Called when media reaches the end
On Finish ()
List is Empty
```

Figure 12.Video plane with GSTUnityBridge script attached

The ability to adjust the position of the first user view of the camera is done via the joystick in the Quest2 controls. It is using the new Input system developed since Unity 2019 versions. This is attached to the main camera of the game so that the end user can place himself/herself into the most comfortable position inside the vehicle to initiate the driving process.



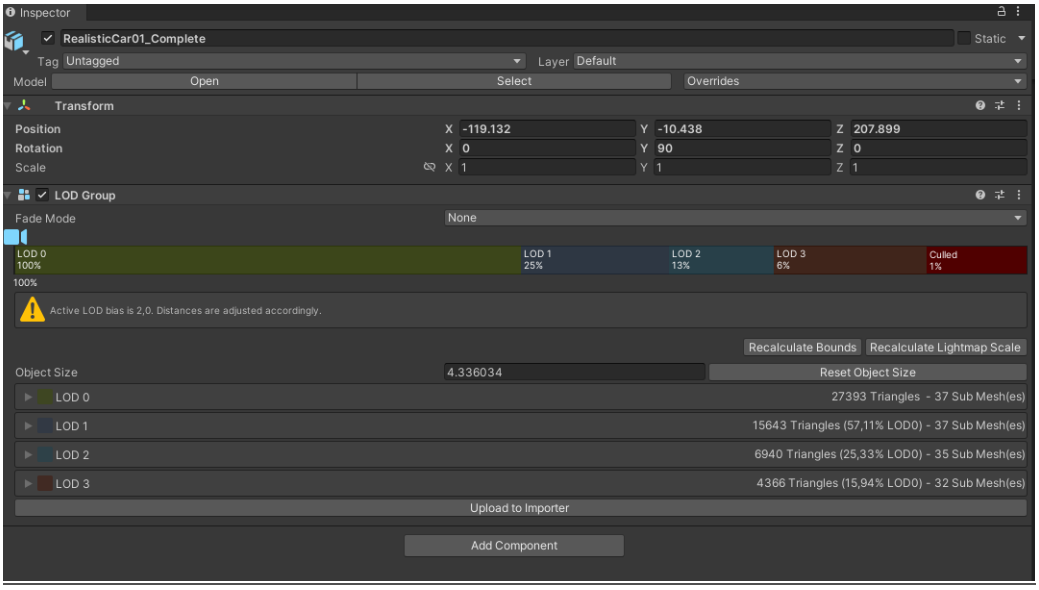
```
Stereo Convergence   10
Target Display       Display 1
Target Eye           Both
Flare Layer
Audio Listener
Tracked Pose Driver
Device               Generic XR Device
Pose Source          Right Eye
The Tracked Pose Driver is attached to a camera, but is not tracking the Center Eye / HMD Reference. This may cause tracking problems if this camera is intended to track the headset.
Tracking Type        Rotation And Position
Update Type          Update And Before Render
Use Relative Transfo ✓
Use Pose Provider    None (Base Pose Provider)
Old Camera Move (Script)
Script               OldCameraMove
Player Input
Actions              Minibus (Input Action Asset)
  Default Scheme     XR
  Auto-Switch        ✓
  Default Map        Player
UI Input Module      None (Input System UI Input M
Camera               None (Camera)
Behavior             Invoke Unity Events
Events
Open Input Settings   Open Input Debugger
UDP Receive (Script)
Script               UDPReceive
Port                 8051
Last Received UDP P
```

Figure 13. Main camera using new Input Device system in Unity

The model and the application are running into Unity 2021 LTS engine to guarantee and ensure the evolution of the model.

Several LOD (Level Of Details), this is, the number of triangles that are used to render in 3D the car model are available. The selected quality has been the highest one to ensure a realistic view of the car interior.

Co-funded by the Horizon 2020
Framework Programme of the European Union



☑ RealisticCar01_Complete

Tag Untagged

Layer Default

Position X -119.132 Y -10.438 Z 207.899

Rotation X 0 Y 90 Z 0

Scale X 1 Y 1 Z 1

☑ LOD Group

Fade Mode None

Active LOD bias is 2,0. Distances are adjusted accordingly.

Object Size 4.336034

LOD 0 27393 Triangles - 37 Sub Mesh(es)

LOD 1 15643 Triangles (57,11% LOD0) - 37 Sub Mesh(es)

LOD 2 6940 Triangles (25,33% LOD0) - 35 Sub Mesh(es)

LOD 3 4366 Triangles (15,94% LOD0) - 32 Sub Mesh(es)

Figure 14. Level Of Detail for Car Interior

2.5.3 Telemetry Information

The following telemetry information is retrieved from the AGV, and sent every 100ms.

- Longitude GPS Position : **GPS-LON**
- Latitude GPS Position : **GPS-LAT**
- RTT in milliseconds for the UDP commands: **RTT**
- Steering Angle of the wheels: **STEE**
- Speed in m/s: **SPEED**
- The protocol in the client side is also prepared to receive:
- Traffic lights status if sent: **TLIGHT**
- RPM of the motor: **RPM**
- Driving Mode: **MAN, AUTO, TAKEOVER**
- Energy of the battery: **BATT**
- Encountered object: **OBJECT**

This telemetry is forwarded to port 8051 in the HMD application and can be rendered as desired. It is also logged in the PC with lines like this:

```
{"TYPE":"CMD_ACK","CAR":"CARSIMU","SEQN":6619,"WTSU":1652454590139809,
"SPEED":2,"STEE":18,"BRAKE":0,"DEL1":-
1652454590139809,"MEC1":"127.0.0.1:5001","DEL2":0,"MEC_COCKPIT":"127.0
.0.1:5001","GEAR":1,"BUTTON_L2":0,"BUTTON_R2":0,"BUTTON_X":0,"SQN-
ACK":6619,"GPS-LAT":-7.977900000001554,"GPS-LON":42.035,"GPS-
HEI":484.9500000000021,"RPM":1199}
```

RTT latency is also continuously logged into the RTT.log CSV file.

2.6 Distributed Node Application

This application is the core part of the control system. It is distributed into three layers. The upper layer is the PC/HMD layer, the second layer runs in the

Co-funded by the Horizon 2020
Framework Programme of the European Union

Edge (MEC) and finally, the last layer runs in a Jetson Xavier NX board which is directly connected to the AGV and the cameras.

2.6.1 Distributed Node Application – Cockpit part

This part of the application runs in the PC which is directly connected to the HMD device via link cable. The connection between HMD and PC are performed via UDP using localhost schema. This part of the application is in charge of forwarding UPD events containing telemetry data so that they can be rendered in the VR view of the Unity application. This part of the application also reads the data from the Logitech G29 device and send it to the lower layer so that control commands flow towards the AGV. The real time position of the steering wheel is also read and directly mixed with the telemetry data received so that the end user perceives the steering wheel movement in the VR side synchronized with the physical one that she/he is touching.

2.6.2 Distributed Node Application – MEC Edge part

In this layer the VPN tunnel is created, all the logs are stored, and the scalability of the system is ensured. The edge is configured via several virtual machines using Microstack (Canonical Openstack flavour) as virtualization environment. The Grafana server is also running in this layer so that all the KPIs can be rendered and stored for the required purposes in the project.

2.6.3 Distributed Node Application – Jetson part

Finally, there is a layer of the application that runs on a Jetson Xavier board. This part is in charge of the translation of the command into the more specialized definition inside the AGV. It will also control and encode the video flows that are sent up to provide the consolidated rendered MR view in the HMD. Hardware encoders are required to guarantee the low latency requirements.

The video encoding parameters which have been tested in the field are so far:

```
/usr/bin/gst-launch-1.0 -v v4l2src device=$DEVICE do-timestamp=true !
image/jpeg,width=$WIDTH,height=$HEIGHT,framerate=$FRAME_RATE ! jpegdec
! nvvidconv ! nvv4l2h264enc  iframeinterval=30 bitrate=$BITRATE
insert-sps-pps=1 ! 'video/x-h264, stream-format=(string)byte-stream' !
rtph264pay mtu=$MTU_SIZE ! udpsink max-bitrate=$MAX_BIT_RATE
host=$DESTINATION_HOST auto-multicast=true port=$DESTINATION_PORT 2>&1
```

2.6.4 Edge and Radio

The Edge is directly interconnected with the cockpit place via fixed network high speed LAN routing. At the same time the MEC lives in the same physical network as the 5G CORE appliance so that the GTP tunnelling enjoys the fastest possible speed. The idea is that latencies from and to the AirScale radio server are under the 5ms range. The AGV is connected using a 1G ethernet cable to the board responsible for controlling and sending video upwards to the modem radio connection. The UE could be a mobile phone device, but the recommended setup is using a mmW modem with 1G or more ethernet connection. In this way, there is a complete optimization of the data path speed and latencies in the AGV level.

Co-funded by the Horizon 2020
Framework Programme of the European Union

NAT translation issues are solved by installing a VPN client in the board which is directly connected to the modem. The VPN server runs in the MEC and since a single a unique certificate is being installed in the board, the same IP address under the VPN subnetwork will be assigned all the time.

Regarding the penalty in terms of performance of the VPN it has been observed that is in the range of 3ms for the latency part and it does not have a relevant impact in the E2E latency under 50Mbps bandwidth conditions, which is what is being used for video transmission.

2.7 KPI Collection

It is important that the system has the capability of collecting KPIs that could potentially be analysed in another phase of the project as part of the evaluation of WP6 or even visualized in real time. The following ones have been selected as they are relevant for the use case:

- Frames received in the HMD
- Round Trip Time latency
- Bandwidth used
- GPS Latitude Position
- GPS Longitude Position

The first one is captured via gstreamer debug service which have been adjusted to almost the maximum level in the video decoding section.

The gstreamer plugin libraries will read the settings and will render a file called gub.log in the PC in which all the details are present. Additional nodeJS script has been developed to parse these statistics into a simpler CSV file.

Round Trip Time is measured from the application perspective meaning the time that an UDP control command requires to reach the AGV and get back to the HMD carrying telemetry information. Bandwidth is measured at the network interface level and expressed in Mbps units.

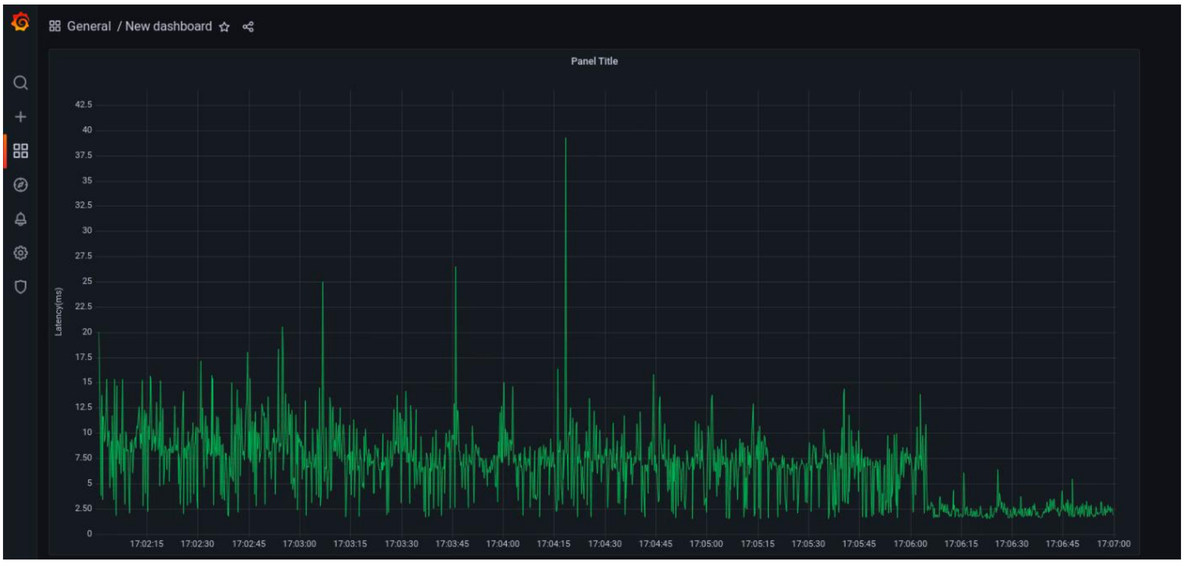


Figure 15. Round Trip Time of Remote Driving Command in Grafana

Co-funded by the Horizon 2020
Framework Programme of the European Union

Finally, the GPS position is being sent by the GPS RTK system inside the AGV and propagated back to the application inside the telemetry data.

To visualize and store all the data a Grafana based stack has been deployed in the MEC which receives asynchronous UDP messages from the nodeJS application in the HMD side and are absorbed by a Telegraf plugin that will inject them into Influx DB database.

Then, visualization dashboards can be set up to watch either the current KPIs in real time or older snapshots of the same type of measurements.

Received frames in the HMD

The gstreamer configuration allows to set a high level of debug for the video decoding activity. The traces that are rendered in the same directory that is being used to run the Unity application are like this:

```
1:22:03.873696000 0000018287D2E5C0 DBG videodecoder ../gst-
libs/gst/video/gstvideodecoder.c:3709:gst_video_decoder_clip_and_push_
buf:<avdec_h264-0> pushing buffer 00000182CA8745A0 of size 1419520,
PTS 1:22:03.733726300, dur 99:99:99.999999999
```

A nodeJS script has been developed to process these traces and evaluate how many frames have been successfully decoded. The output can be easily converted into CSV so that it can be plotted into a graphic.

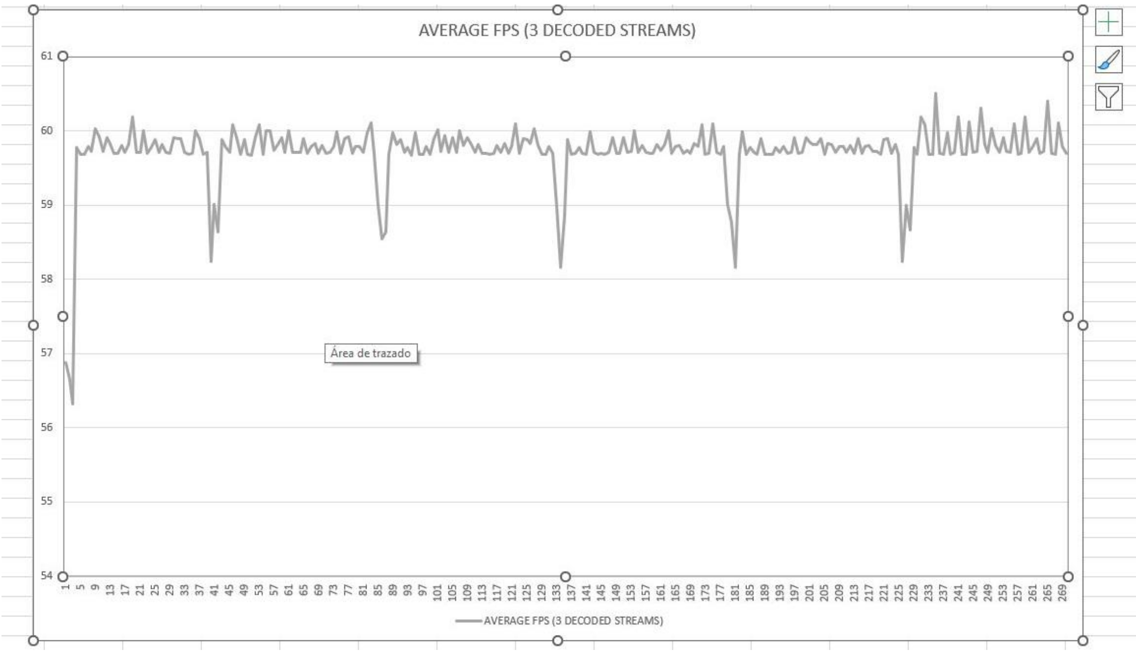


Figure 16. Frames Per Second Decoded at HMD Level

In a first attempt to characterize the KPIs of the Route Control application, the tool Wireshark was used to measure the local throughput handled by the Oculus Quest 2 goggles connected via USB-C, both in UL (device to PC) and DL (PC to device). Two profiles were tested for each configuration: Low profile (72 Hz, 3200x1632 pixels) and High profile (120 Hz, 5408x2736 pixels). The results show throughputs around 200 Mbps for DL and 100 Kbps for UL, with a slight increase when using the high profile:

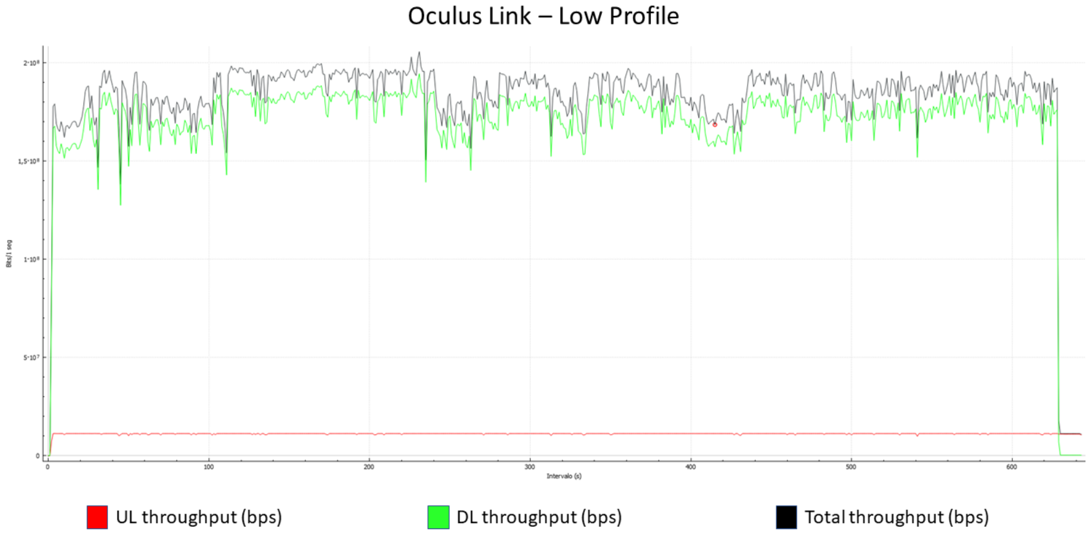


Figure 17. Oculus Link (wired) Low profile

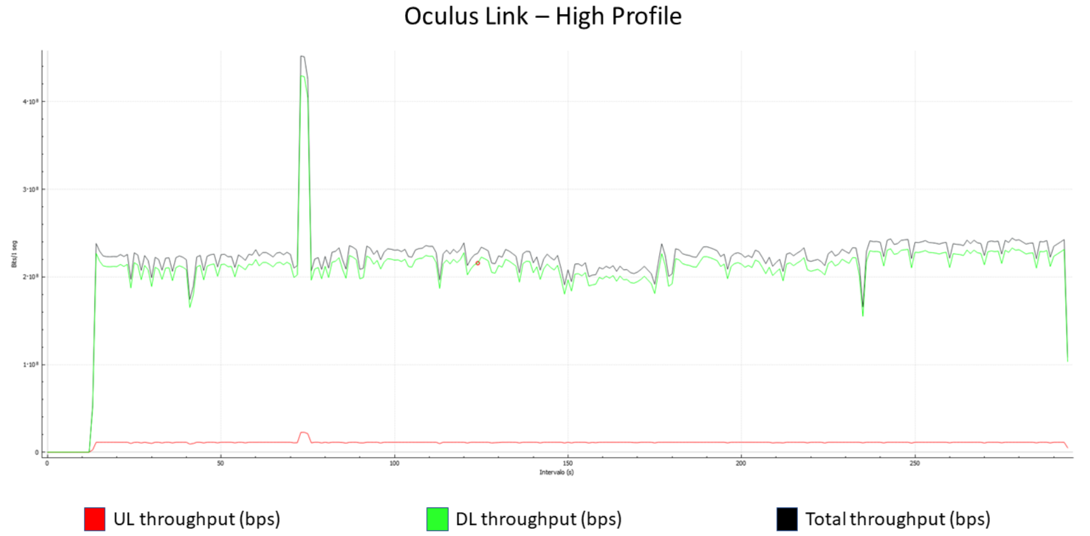


Figure 18. Oculus Link (wired) High profile

The same methodology was used for measuring the video stream throughput, which in this case is only for downlink. The gstreamer tool was used to configure different video resolutions, with the following results. The bitrate captured is 2,5 Mbps for 360p resolution, 8 Mbps for 720p resolution, and 16 Mbps for 1080p resolution.

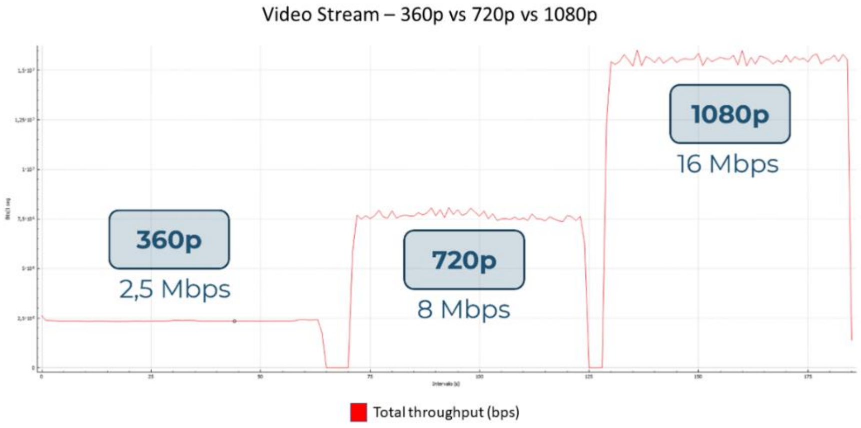


Figure 19. Bitrate for 480p and 720p video

The previous configurations were also used for testing the video latency. The methodology used was to display a millisecond-sensitive clock in the same screen that is running the Unity scene, and then point the robot camera to the same screen to compare the difference (see Figure 16). It should be noted that this has an error range of 50 ms, due to the camera frequency (30 fps, meaning an error of 33.33 ms) and the screen frequency (60 Hz, meaning an error of 16.66 ms), but several measures were obtained to minimize this effect.

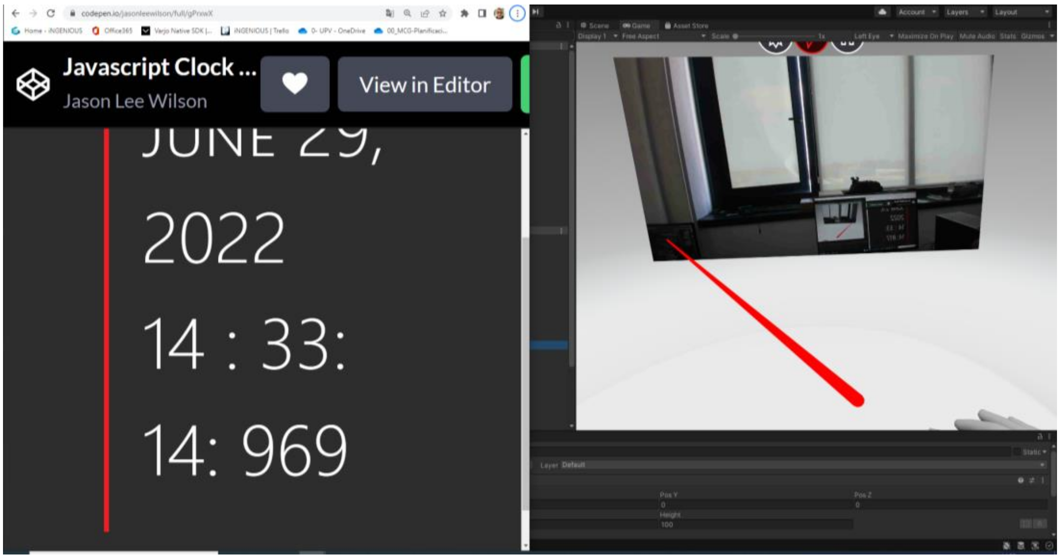


Figure 20. Methodology for measuring the video latency in the Route Control application

It can be observed that increasing the resolution of the video stream (the size of the frames) not only increases the throughput sent, but also the latency, because it takes more time to process and deliver the frames. For a standard video configuration of 720p, the latency measured was 156.4 ms. This fits in the requisites given by Figure 21, since the Route Control application is focused on supervision purposes and therefore is less latency-critical than the Remote Driving application:

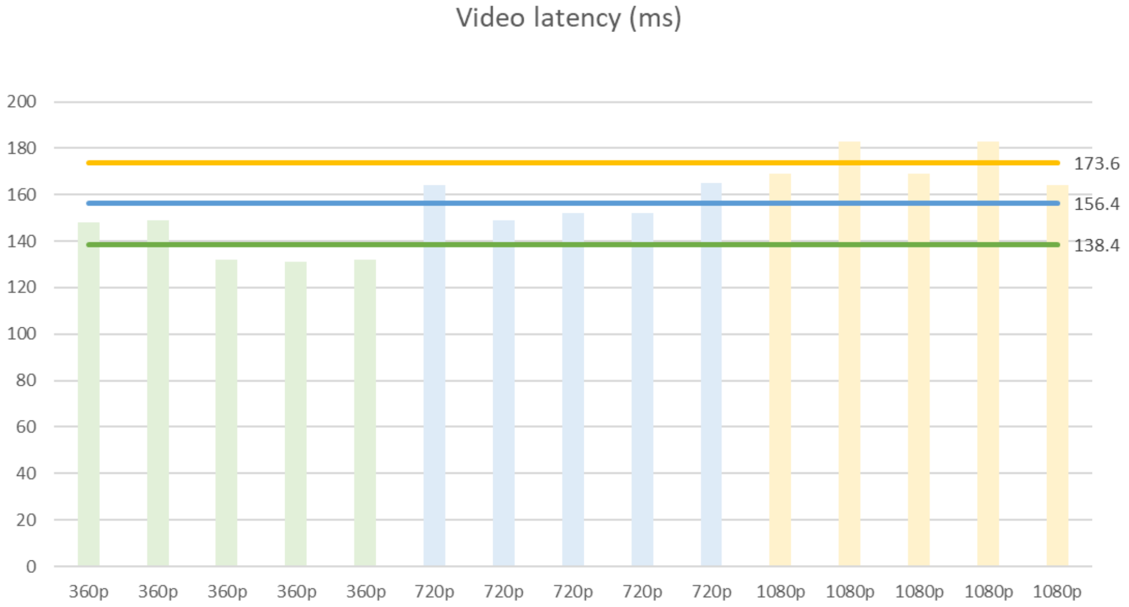


Figure 21. Results obtained for video latency in the Route Control application

Co-funded by the Horizon 2020
Framework Programme of the European Union

3 Haptic gloves and haptic sensors

Haptic gloves are a kind of wearable VR equipment that provides two kinds of main functionalities. On the one hand, as an input device, they provide a way to recognize the current hand pose so that this real set of positions can be translated to the virtual world if necessary. On the other hand, as an output device, they provide a way to apply feedback on the user by means of the haptic actuators positioned among the glove surface.

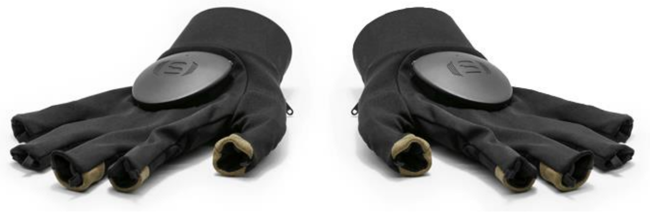


Figure 22. Sensorial XR haptic gloves.

3.1 Detailed description of Glove Capabilities

The Haptic Glove device used in this project is the Sensorial XR model created by Neuro Digital Technologies. This haptic glove has a total of 10 haptic points. Five of them located at each finger's top, and the other five are positioned among the palm, as show in Figure 23.

Each haptic actuator is a customized low-latency linear resonant actuator (LRA), that allows to define one among 1024 vibrations profiles. Their frequency is 205 Hz with a latency under 30 ms. They are used to simulate different kind of stimulus in the user during operation.

To perform the motion capture of the palm and each finger rotation Sensorial XR includes 7 IMUs (Inertial Measurement Unit), each of them a 9 axis one working over 225 Hz per second with a sample rate over 200 Hz. In order to compensate the IMUs drifting, the sensor fusion algorithm is run in the hardware.

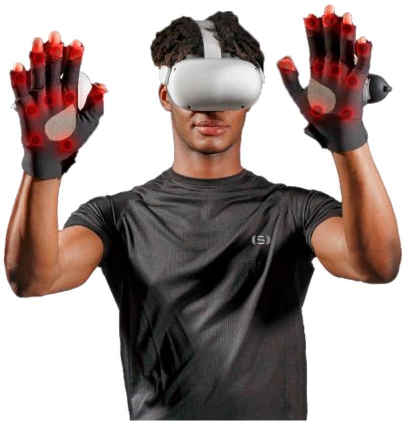


Figure 23. Haptic actuators position.

It's compatible with both kind of communications, wired and wireless. In the first case, the latency is negligible and a sample rate over 200 FPS is achieved. In the second case, there is 7.5 ms mean latency, with a sample rate around 120 FPS. The wireless connection was updated as part of the T3.3 to use the

Co-funded by the Horizon 2020
Framework Programme of the European Union

BLE (Bluetooth Low Energy) 5.0 protocol, and doesn’t require any kind of additional BT dongle, for an easy connection with any smart device.

Due to the difficulty to detect hand gestures with a 100% success probability, these haptic gloves include conductive fabric surfaces in the palm, and the top of thumb and the index and middle fingers, as show in Figure 24. Conductive fabric surfaces used by the smart controller action. When these surfaces contact with each other, a 100% accurate signal will be triggered. This way, even if the user is not holding any kind of controller, certain gestures where these conductive surfaces contact each other can be used as smart controllers/buttons to perform a certain input in the system.

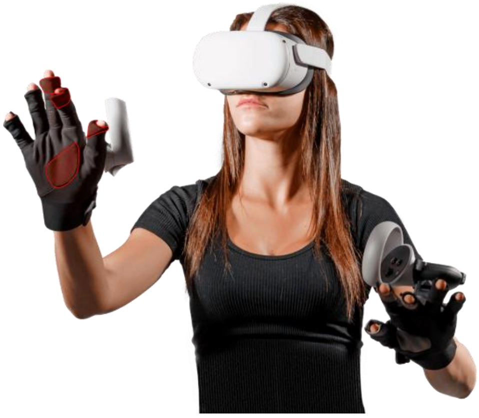


Figure 24. Conductive fabric surfaces used by the smart controller action

Finally, the battery included is a 600 mA Li-Po one that allows around 6-8 hours of continuous use, and 300 charging cycles.

3.2 Haptic Glove API and Versions

The haptic glove firmware communicates with the developers' applications by means of an API. At the start of the project, this API was developed in C# and therefore interoperability was restricted to certain systems, making it difficult to operate it from C and C++ applications. During the last months, a new C++ API was created from scratch so that the intercommunication can be extended to any kind of system, even if a simple wrapper or minor modifications must be defined in some cases, and in order to improve the performance.

This is especially important when the haptic glove is not going to be used to provide visual feedback of the hand position in a videogame engine such as Unity3D or Unreal Engine, as until now, the API was coupled with the SDK developed for these two platforms.

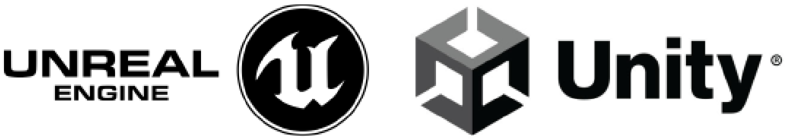


Figure 25. Unreal and Unity3D, the most used multipurpose 3D videogame engines

However, sometimes only the haptic feedback functionality is necessary, for example when the operator uses a physical simulator, and therefore a more basic development framework could be used, and these SDKs wouldn't be compatible or even necessary.

With this API update, we provide a solution for these cases and allow more traditional development framework's users to be able to handle the connection and management of the glove by means of the new API.

Also, the inclusion of the BLE 5.0 wireless communication was part of the new API development, as part of T3.2 in order to minimize the energy consumption. This way, communication with modern smart devices was possible without any additional BT dongle, but also it helps the battery to last longer. The main problem to implement BLE 5.0 was the amount of information to be communicated while using a full pair of gloves, so different strategies were applied in order to minimize the size of the data exchanged.

3.3 NeuroDigital Unity SDK

Together with the API, the Sensorial XR haptic gloves comes together with an SDK that provides additional functionality in both, Unity3D and Unreal Engine. Between them, the current project uses the Unity3D SDK.

There have been two main updates related to the SDK, and we will focus on the explanation of the Unity3D modification, as the Unreal Engines ones are equivalent:

1. The SDK was updated to include a new physical accurate hand object that interacts in a realistic way with the environment. This way, developers can achieve fingers slide on surfaces, forces simulation, inter-finger collisions and more advanced functionalities without the need to write a single code line by means of this pre-defined object. This allowed to improve the quality of the SDK solution and at the same time provide better tools for developers.
2. Second, the SDK was modified to use the new C++ API, so that it could benefit from the performance improvement obtained with the change.

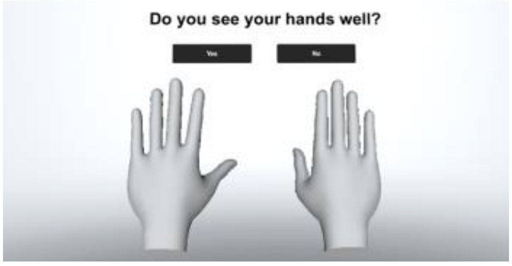


Figure 26. Sensorial XR SDK virtual hands in the Unity videogame engine

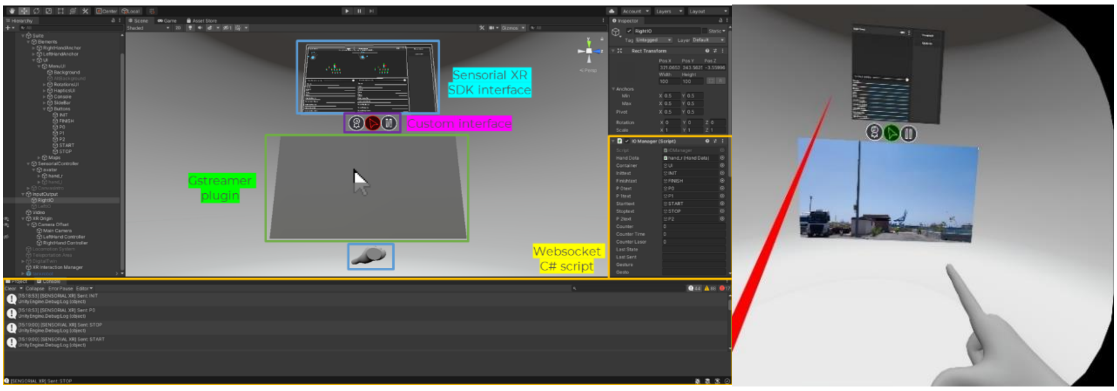
3.4 Mixed Reality Application

The Route Control application is a complementary approach to the Remote Driving application that consists of the supervision and command of autonomous AGVs, with the Sensorial XR haptic glove playing a pivotal role. The user is equipped with a pair of Oculus Quest 2 glasses and one Sensorial XR haptic glove, both attached to a PC that runs the Unity application and communicates the commands to the robot. This Unity application has the following components (see also Figure 27 below)

1. WebSocket C# script for handling the communication from and to the glove. This script creates a full-duplex connection between the PC and the robot, via TCP. It can send the gesture commands and receive haptic feedback asynchronously.
2. Gstreamer plugin for receiving the video feedback. The pipeline used consists of RTP and UDP protocols encapsulating JPEG frames, captured in different qualities.
3. Sensorial XR SDK interface for representing vibration levels, hand position and rotation, and gesture performed.
4. Custom interface for representing the current orders, such as the route number, the on/off flag, and the stop/resume flag.

The data gathered by the different sensors attached to the robot is used to create an immersive synthetic scenario with real-time video and haptic feedback, represented using the MR glasses and the haptic glove respectively. Unity3D software guarantees the interoperability of these devices by using the .NET environment and C# language, capable of running their dedicated SDKs and plug-ins as well as handling the connections. The Oculus Quest 2 goggles can connect wirelessly using the Wi-fi protocol (although USB-C solution was chosen for a better image quality and less artifacts), whereas the Sensorial XR glove can do so using BLE.

The HMD is connected via USB-C and OpenXR framework, in the same way that the Remote Driving application works. However, in this case the scene is dominated by the video stream, simulating the point of view of the robot.



```
[15:18:53] [SENSORIAL XR] Sent: INIT
UnityEngine.Debug:Log (object)
[15:18:53] [SENSORIAL XR] Sent: P0
UnityEngine.Debug:Log (object)
[15:19:00] [SENSORIAL XR] Sent: STOP
UnityEngine.Debug:Log (object)
[15:19:00] [SENSORIAL XR] Sent: START
UnityEngine.Debug:Log (object)
[SENSORIAL XR] Sent: STOP
```

Figure 27. Unity scene for the Route Control application

The protocols used for the communication are WebSocket (for sending gestures and receiving haptic feedback) and gstreamer (for receiving video streaming). WebSocket allows to establish a permanent bidirectional connection with added security and reliability, although its TCP-based scheme increases the latency with respect to the UDP-based solution. The different gestures that can be performed are the following: (i) go to a specific point in the route; (ii) stop the movement; (iii) resume the movement. Consequently, the user can avoid hazardous situations for the robot, for example when approaching to an obstacle in the route thanks to the reception of haptic feedback with an intensity dependant of the obstacle closeness (see Figure 33).

Co-funded by the Horizon 2020
Framework Programme of the European Union

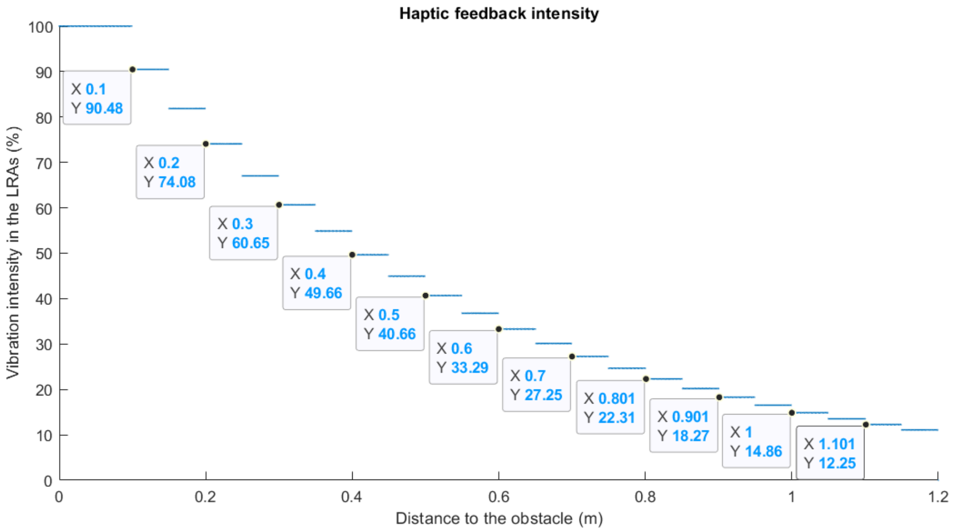


Figure 28. Haptic feedback intensity as a response to the obstacle closeness.

3.5 Objective KPIs collection

For the haptic gloves, some KPIs are gathered for future analysis in further phases of the project. The following KPIs have been considered important:

- Wired/wireless bitrate
- Indoor/outdoor wireless range
- E2E latency
- Network latency

The Haptic gloves can be both wired and wirelessly connected to the PC. Using Wireshark, the bitrate for both connectivity solutions have been captured. In Figure 29 the bitrate for Bluetooth connection is shown, being around 75 kbps, and in Figure 30 the bitrate for USB connection, which reaches the value of 375 kbps. These values correspond to UL data (device to PC), whereas the DL data (PC to device) is considerably lower and intermittent, since it is only sent for haptic feedback. The graphs also show the total throughput involved in the Bluetooth or USB connectivity, considering signalling as well as data delivery.

The functionalities of the haptic gloves are the same for both Bluetooth and USB options, being the differences of throughput due to: (1) Bluetooth messages refresh less frequent (120 FPS) than USB (200 FPS), and (2) Bluetooth uses compression of packets while USB does not.

Co-funded by the Horizon 2020
Framework Programme of the European Union

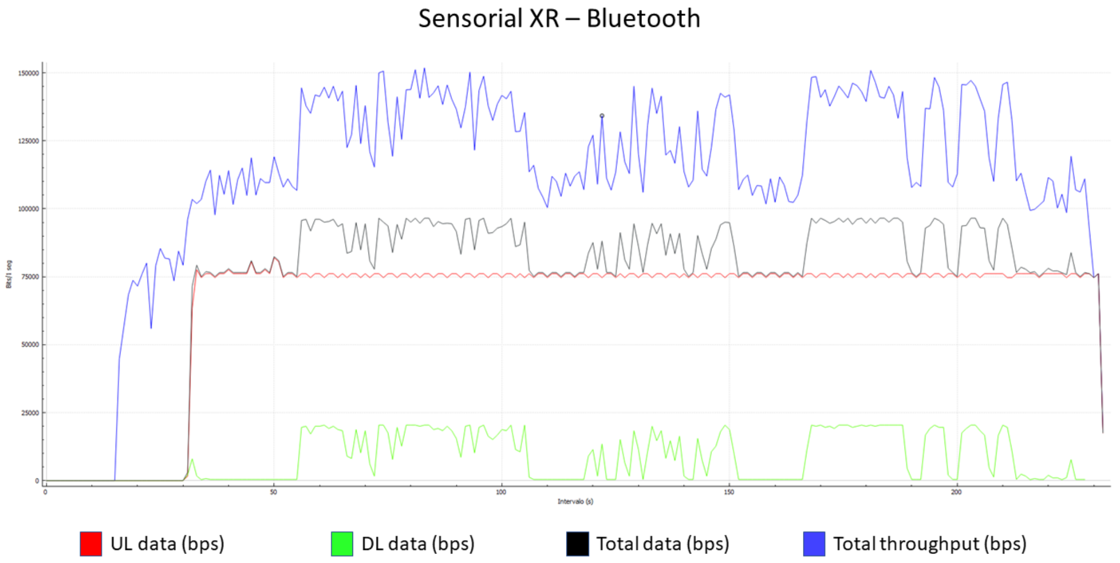


Figure 29. Bluetooth bitrate for haptic gloves communication

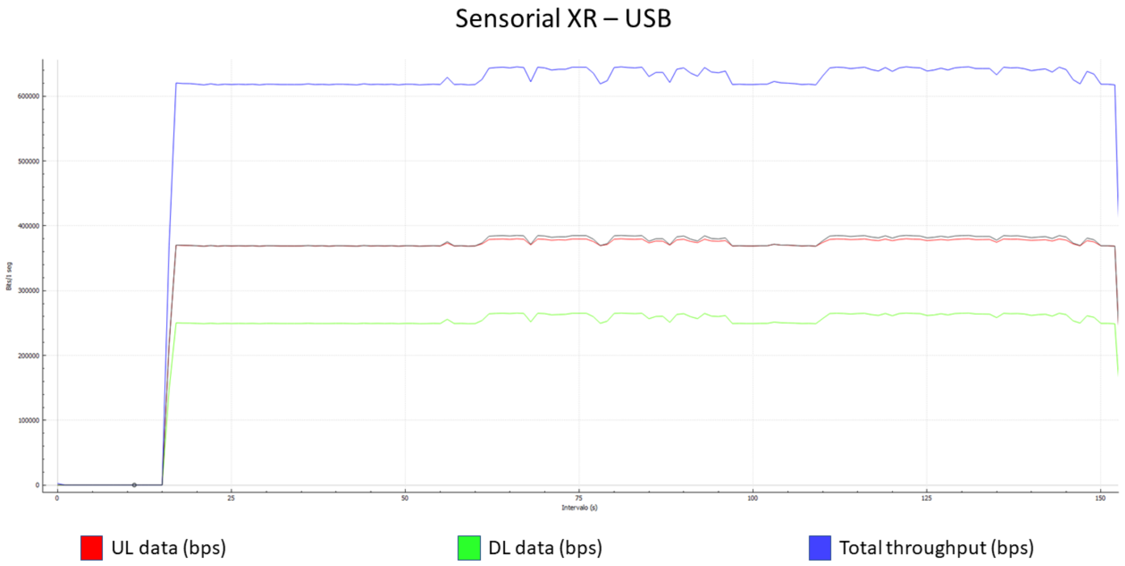


Figure 30. Wired bitrate for haptic gloves communication

The Haptic gloves have the option to work wirelessly, enabling the operator to move from the server. Nevertheless, the Bluetooth range is a KPI to have into consideration as it can be critical. If the operator moves out of these limits, the connection may fade and even stop. For evaluating this KPI, both indoor and outdoor test have been carried out in Fivecomm's office and the UPV campus respectively. The Bluetooth connections have been measured all around the maps detecting three different cases:

- Good performance (green)
- Connexion fades sometimes (yellow)
- Not working (red)

The indoor results obtained are shown in Figure 31. For the measurements, all the doors have kept open for a coherent test.

Co-funded by the Horizon 2020
Framework Programme of the European Union

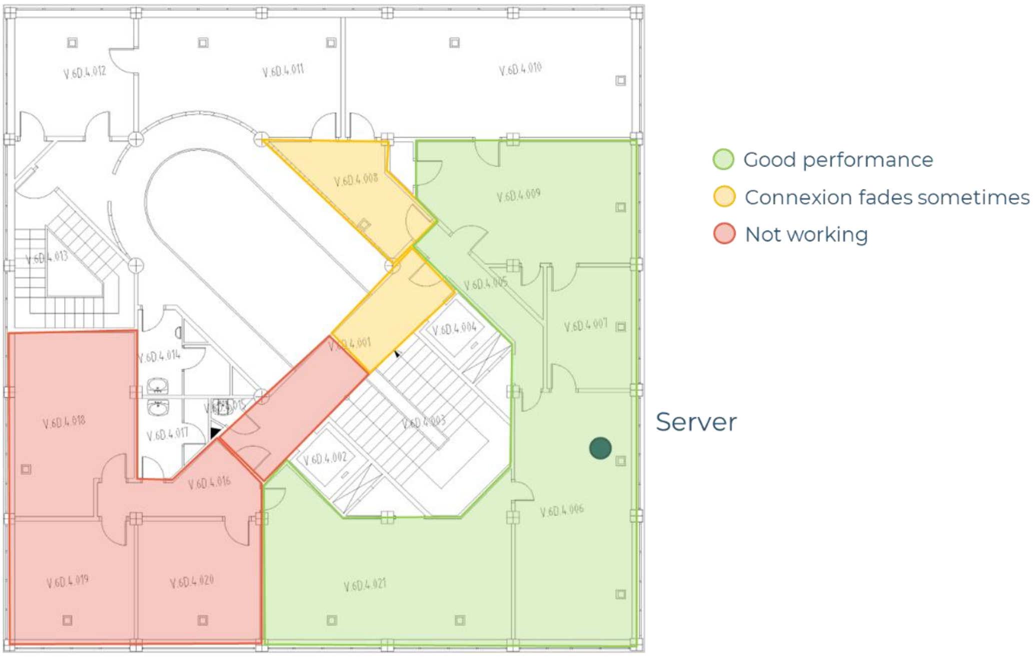


Figure 31. Indoor Bluetooth range for haptic gloves

On the other hand, the outdoor results obtained are shown in Figure 32.

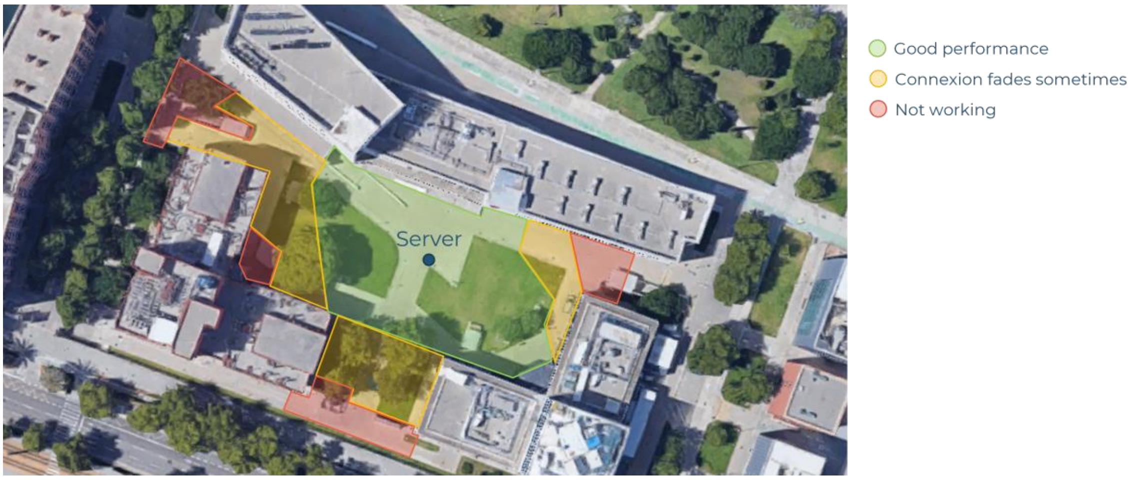


Figure 32. Outdoor Bluetooth range for haptic gloves

Regarding the glove's latency, two different methods were proposed: one automated method for measuring network delay, and other empirical method for measuring E2E latency. In the case of the network delay, the idea is to add a timestamp every time a gesture is sent (by the user) and received (by the robot), and therefore the delay is the difference between the two timestamps. However, for simplicity, this was implemented in a way that the robot sends back the messages to the user, meaning that the obtained value is the round-trip time (RTT). It excludes the latencies of the glove and the robot but gives an idea of how fast the 5G network is (29.14 ms in average).

Co-funded by the Horizon 2020
Framework Programme of the European Union

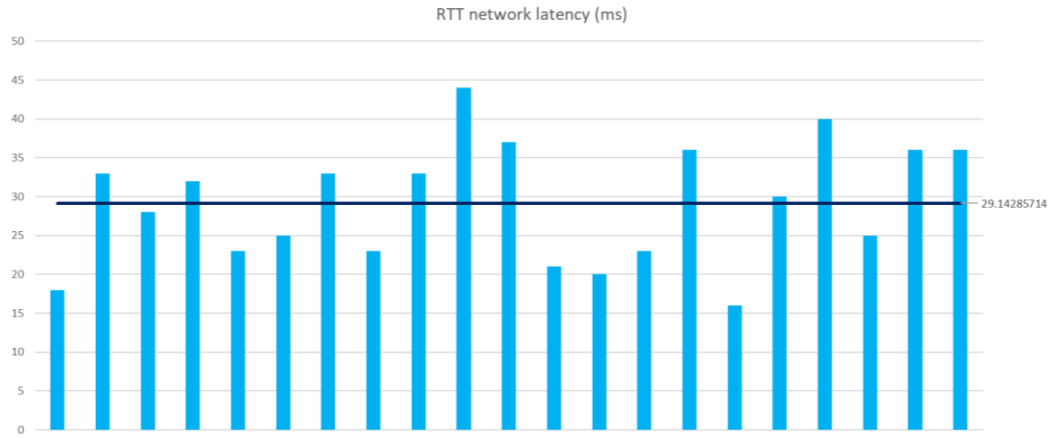


Figure 33. RTT network latency for haptic gloves messages

On the other hand, the E2E latency of the gestures includes several biases such as the behaviour of the glove and the robot. We defined it as the time it takes the robot to start moving after doing a gesture. This has been measured by capturing the on-site application with a slow-motion camera that provides 240 FPS, meaning an error rate of only 4.16 ms. Nevertheless, it should be noted that the main error rate in this case is the human perception because the values can vary depending on the frame that the user decides that the robot is actually moving.

In fact, the results displayed in Figure 34 show that the perceived latency is, in average, significantly higher when resuming a route or sending the AGV to a point (735 ms) than when stopping it (362 ms). This is due to mechanical reasons, that takes less time to stop the wheels with the brake than starting a movent. It may vary depending on the AGV employed. Apart from the human perception, we consider this gap to be due to the behaviour of the robot itself. It was observed that the robot performed considerably faster the brakes than the accelerations. For this reason, we consider these higher latencies acceptable, since the Route Control application is not as dependent of the robot responsiveness as the Remote Driving application.

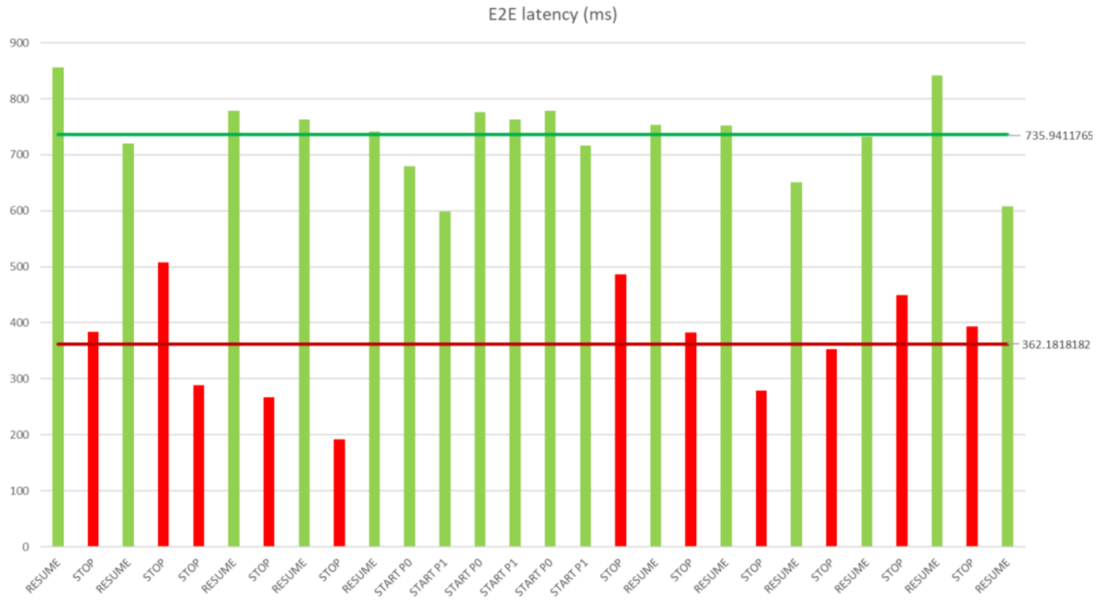


Figure 34. Perceived E2E gesture latency

Co-funded by the Horizon 2020
Framework Programme of the European Union

4 Integration of HMD and Haptic Glove in consolidated use case

The consolidated use case includes the initial movement of the AGV using predefined points in a route, the manual control from the glove as well as the teleoperated driving procedure when an obstacle Is encountered.

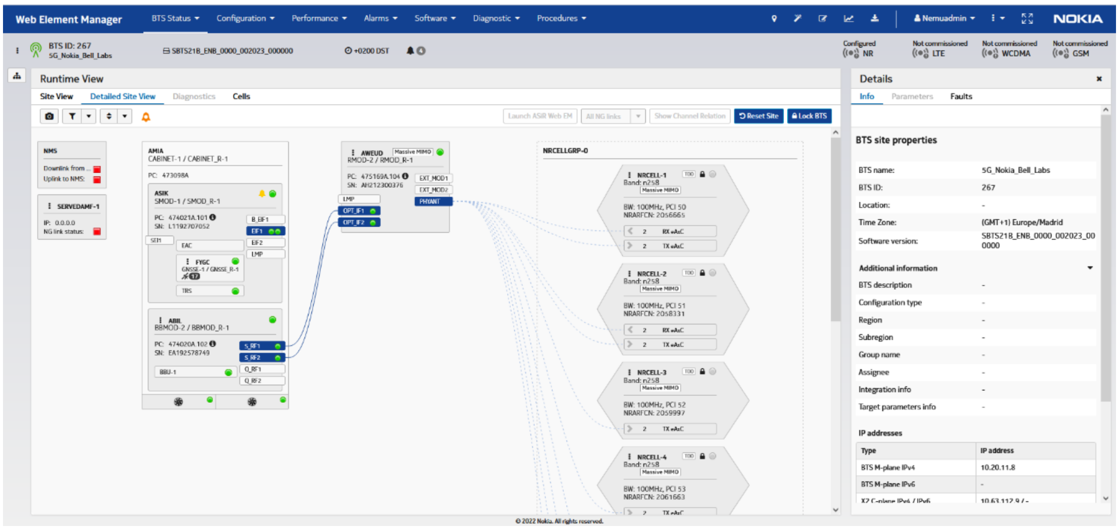
4.1 5G network integration

A more detailed view of the 5G Deployment and the possible link of UE devices to the AGVs is shown in this section. The MEC plays a fundamental role in this deployment as we need control commands to arrive to the UE as fast as possible while we need to monitor and register all the AGVs in some central place which offers better KPIs than a regular internet server in a data center far away. In addition, a VPN which will isolate NAT issues in the UE and will provide a secure channel for transmitting the video and control command is a must.

The 5G network offers low latency when connecting UEs and this make the use of using HMDs and Haptic Devices possible from a remote perspective. We need to remark the fact that no one of the HMDs or Haptic devices connects directly to 5G as they do not have a modem inside, so there are more than one way of getting connected to the 5G network.

1. Connect to 5G using an additional UE (Mobile or Modem) and then connect to this device using direct cable connection.
2. Connect to 5G using an additional UE (Mobile or Modem) and then connect to this device using WI-FI tethered connection.
3. Connect the HMD and the haptic device to a PC and this PC directly to a LAN which directly connects to MEC server.

The number of 5G cells being used for the downlink and uplink is also important to have better throughput for the video. There are up to 8 cells which can be configured for downlink and 2 for uplink in current 5G deployment. This is the default provisioning schema that most operators use. This can be seen in the detailed view of the radio configuration below.



BTS ID: 267
5G_Nokia_Bell_Labs
SBTS21B_ENB_0000_002023_000000
+0200 DST

Configured NR
Not commissioned LTE
Not commissioned WCDMA
Not commissioned GSM

IP: 0.0.0.0
NG link status:

AMIA
CABINET-1 / CABINET_R-1
PC: 473098A

ASIK
SMOD-1 / SMOD_R-1
PC: 474021A.101
SN: L1192707052

FYGC
GNSSE-1 / GNSSE_R-1

ABIL
BBMOD-2 / BBMOD_R-1
PC: 474020A.102
SN: EA192578749

AWEUD
RMOD-2 / RMOD_R-1
PC: 475169A.104
SN: AH212300376

NRCELL-1
Band: n258
BW: 100MHz, PCI 50
NRARFCN: 2056665
2 RX eAxC
2 TX eAxC

NRCELL-2
Band: n258
BW: 100MHz, PCI 51
NRARFCN: 2058331
2 RX eAxC
2 TX eAxC

NRCELL-3
Band: n258
BW: 100MHz, PCI 52
NRARFCN: 2059997
2 TX eAxC

NRCELL-4
Band: n258
BW: 100MHz, PCI 53
NRARFCN: 2061663
2 TX eAxC

BTS site properties

BTS name:	5G_Nokia_Bell_Labs
BTS ID:	267
Location:	-
Time Zone:	(GMT+1) Europe/Madrid
Software version:	SBTS21B_ENB_0000_002023_000000

Additional information

BTS description	-
Configuration type	-
Region	-
Subregion	-
Group name	-
Assignee	-
Integration info	-
Target parameters info	-

IP addresses

Type	IP address
BTS M-plane IPv4	10.20.11.8
BTS M-plane IPv6	-
X2 C-plane IPv4 / IPv6	10.63.112.9 / -

Figure 35. 5G mmW cells setup for HMDs use case

Co-funded by the Horizon 2020 Framework Programme of the European Union

The 5G Core is installed collocated with the microstack MEC deployment to make things easier and have faster connectivity. It is using open5GS and the SIMs can be provisioned either from the web interface or from convenient scripts in the edge appliance.

4.1.1 HMD and Haptic devices connection schema

The HMD and haptic devices are connected to 5G through a computer the server to whom they are connected as they do not have an integrated 5G modem. The connectivity options of the haptic gloves' devices are: (1) Bluetooth connection and (2) USB cable connection, while the options for the HMD are (1) Wi-Fi connection and (2) USB cable connection. Both devices have also the option to be directly connected to the MEC of the 5G network (3).

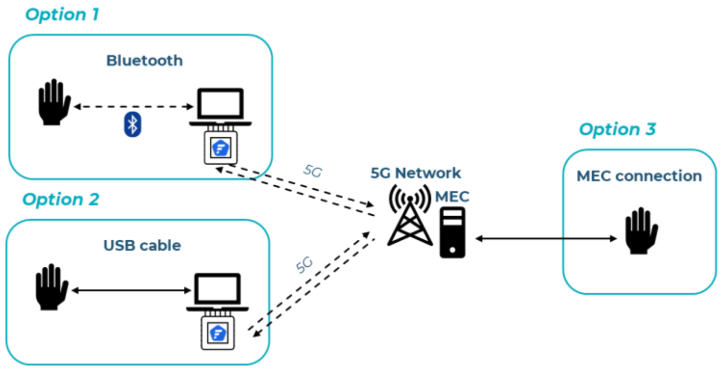


Figure 36. Connectivity options for the Haptic Gloves

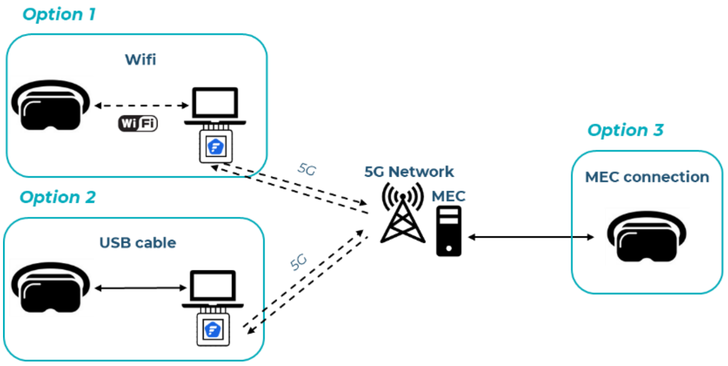


Figure 37. Connectivity options for HMD

4.1.2 mmW Modem

The lack of mmW modems and devices in the market make somehow difficult to choose between a big range of different devices. The availability of such UEs is also scarce so there only two of them that have been tested and integrated into the E2E solution.

The chosen device is a 5G modem which can operate in the N258 5G band and offers a 2.5Gbit ethernet link to the rest of the connected systems. A POE cable is required to get the DC input and potentially an AC/DC inverter is also needed to operate. The solution in the Robotnik AGV has been connecting the 48V DC output to an inverter, and then connecting the AC input of the

Co-funded by the Horizon 2020 Framework Programme of the European Union

modem to it. The POE adapter will carry the final DC input connection to the modem.

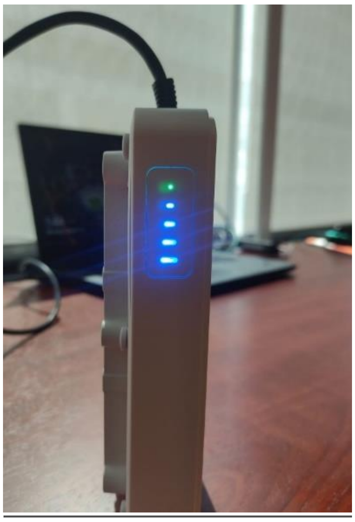
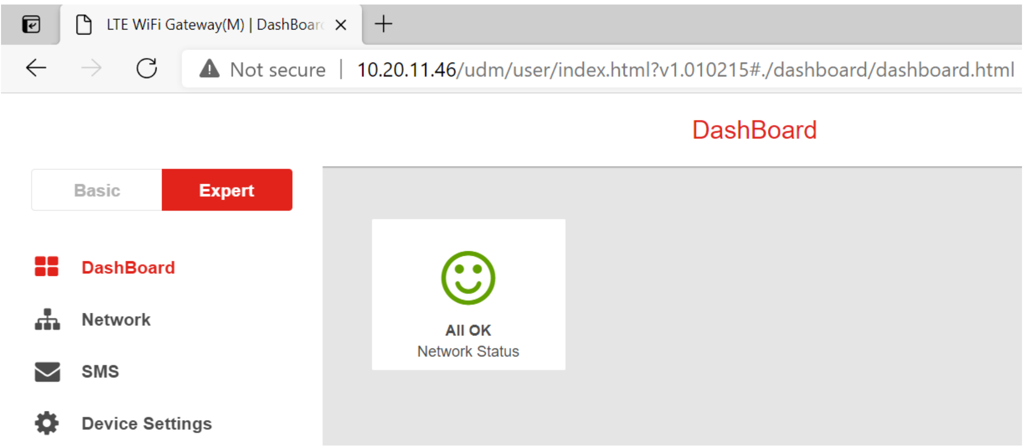


Figure 38. Askey mmW modem

This Askey mmW modem has been connected directly to the AGV controller board via ethernet cable. Askey modem also has a WEB UI which allows to be configured from web interface.



LTE WiFi Gateway(M) | DashBoar
Not secure | 10.20.11.46/udm/user/index.html?v1.010215#./dashboard/dashboard.html
DashBoard
Basic
Expert
DashBoard
Network
SMS
Device Settings
All OK
Network Status

Figure 39. Web Management Interface of Askey Modem

4.1.3 5G Modem in 3.5 GHz Band

Within the context of the project, a 5G modem working with Rel-15 in midband (3.5 GHz) will be employed. This 5G modem is technically described in [2]. Also, some tests have been carried out to evaluate the performance with other iNGENIOUS components. The results are described in [3].

Figure 40. 5G modem

4.2 E2E Integrated architecture step scenario and actors

It is important to understand that there are several active actors that play different roles. Some of them send and receive different information flows and are placed at the same points in space from the network deployment perspective.

The HMDs, Haptic gloves, PC and 5G modem are in what we could consider the cockpit zone, from which a human user is providing an extra level of safety to the autonomous operations of an AGV.

The 5G network is connecting everything and it is an infrastructure agent. It is connecting everything to the cockpit zone either by 5G modem or using a direct wired cable connection (maybe via some intermediate router) to the MEC. 5G network KPIs are relevant and need to be accessed in the cockpit area to take the appropriate decisions.

AGVs are connected using radio 5G due to their mobility requirements. They need to carry real-time capable video streaming devices or boards that are also connected both to the 5G network and to the robot internal network infrastructure.

4.2.1 SCENARIOS AND ACTORS

The actors involved in the use case are:

- **Operator:** The operator is the person that has the knowledge to control the AGV, and the user of the cockpit.
- **5G network:** Both commercial network and private deployment have been used, as well as mid-band and mmWave modems.
- **XR glasses:** They are used for visualizing the AGV surroundings in an immersive way. The sensation of the XR glasses is to feel that the operator is right there, so the control of the AGV is fluid and intuitive.
- **Haptic gloves:** This device contributes to the immersive experience, allowing the control of the AGV with different gestures, and the feedback using the vibration motors incorporated.

Co-funded by the Horizon 2020 Framework Programme of the European Union

- **Cockpit server:** It is a server that handles all processes involved in the use case, and is directly connected to the XR glasses, the haptic gloves and the 5G modem.
- **AGV:** Vehicle from ASTI, which is the subject to be controlled. It is integrated with a 5G modem enabling the 5G connectivity. It also includes a 360 camera for the video flow reproduced in the remote location with the XR glasses.

4.2.2 Integrated Architecture

There are two different architectures: on-site and remote control.

In the on-site architecture the operator can see the AGV in all moments and control it in real time while seeing how it is moving. The haptic gloves have to be wirelessly connected to the on-site cockpit by Bluetooth, so the operator has freedom to move. Then, the on-site cockpit processes the data sent from the gloves and sends to the AGV through the 5G network thanks to the 5G modems connected to both components. These modems use the Ethernet protocol to communicate with the AGV and the cockpit. Figure 41 visually represents the described architecture.

One flow sends the control commands from the gloves to the AGV, and other flow sends feedback information from the AGV to the cockpit. This feedback information includes the distance to the closest object, what produces in the haptic gloves a vibration to let the operator know if the AGV is getting closer to any object. This intensity can vary depending on the distance, being lower for long distances and higher for short ones.

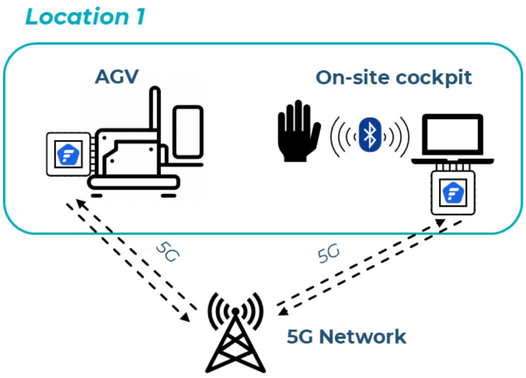


Figure 41. On-site control architecture

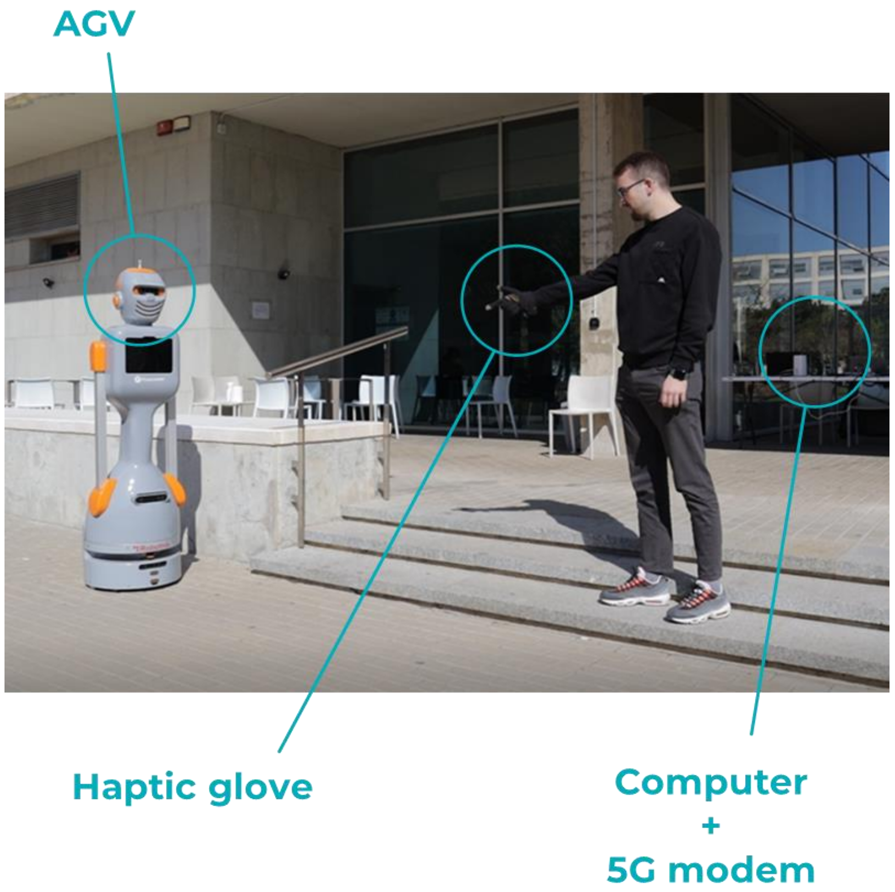


Figure 42. On-site control components

The remote architecture has two different locations: the AGV location and the cockpit location. The operator controls the AGV for a remote position. For this architecture, the haptic gloves are connected with a USB cable to the cockpit as well as the XR glasses. The cockpit is bidirectionally connected to the devices, and to the 5G network with a 5G modem via ethernet. This architecture is showed in Figure 43.

On the one hand, the haptic gloves send information to the AGV regarding the control orders. On the other hand, the AGV sends the same feedback information as in the on-site case, but also a camera flow. This is a key aspect, as the operator will not be in the same location, so there is no direct sight. This video flow is reproduced in the XR glasses for an immersive experience.

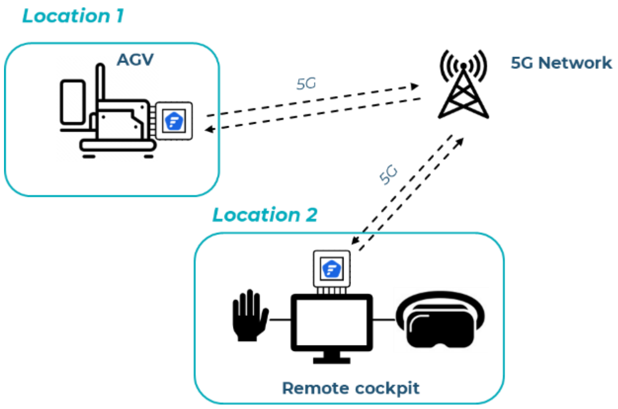


Figure 43. Remote control architecture

Co-funded by the Horizon 2020
Framework Programme of the European Union

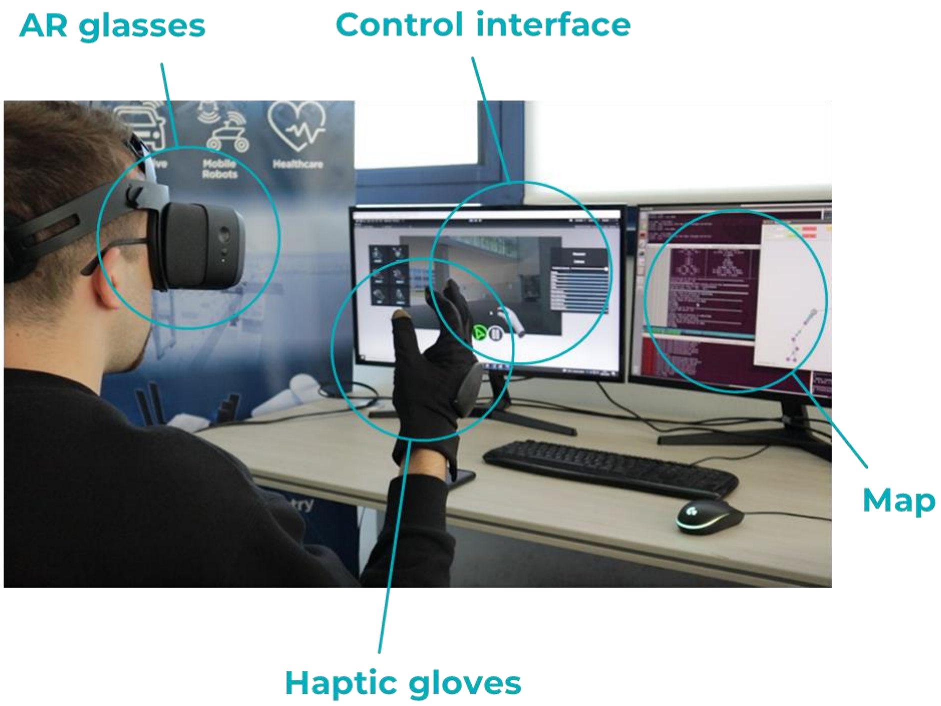


Figure 44. Remote control components

4.3 AGV Integration Architecture

Understanding the basic operation, architecture and capabilities of the AGVs the end user layer deals with will allow a smooth integration process from all the points of view. In a very simplistic view, the AGVs can receive simple control commands in order to move through predefined routes, allow the user to perform a free driving of the vehicle with the safety limitations imposed by their own firmware and communicate acknowledging and sending telemetry information. This integration points will be explored in the next subsections.

4.3.1 AGV predefined route and Haptic control

The autonomous control mode of the AGV consists of a predefined route with checkpoints. The AGV can autonomously move from one point to another following its own criteria and with the help of proximity lasers that can detect objects and avoid collisions. This navigation mode is commonly called Simultaneous Localization and Mapping (SLAM).

For controlling the AGV while working with SLAM, a solution based on haptic control has been implemented in iNGENIOUS. Concrete gestures are mapped to orders to the AGV, being possible the following actions:

- **Resume/Stop**: Stops or resumes the instant movement of the AGV, independently of the location.
- **Go to**: Send the AGV to a route point.

Co-funded by the Horizon 2020 Framework Programme of the European Union

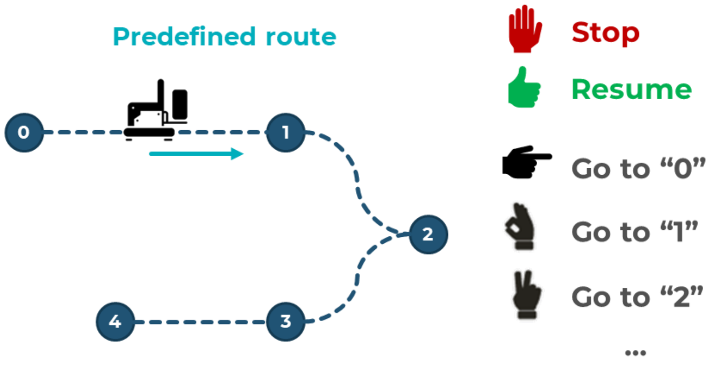


Figure 45. Predefined route and haptic control

4.3.2 Control Request from AGV

As explained in previous section, the way that the AGV should request control is by issuing an ACK response message in which TAKEOVER is set to 1. Once, this is received, the remote driver will see a popup flashing message indicating that a problem occurred, and he/she needs to take control. Then once it is accepted, a new control message will be sent in which MANUAL is set to 1 (TAKEOVER:0 and AUTO:0). From this time on, the cockpit oversees controlling and driving the AGV. At this point in time, some information will be rendered in the user interface using some warning or popup dialog message.

4.3.3 Communication Protocols

Several communication protocols are being used in the use case related to the haptic and VR devices that are subject of this document. We will not have a deep view on the ones which are more related to the 5G radio layers as they are not, strictly speaking, the subject of this deliverable. We will focus on the protocols that are being used to transmit data from the AGV to the HMD and haptic devices as well as the ones which are present in the communication between the PC which is connected to the devices and the 5G network.

In order to control the AGV a proprietary protocol has been developed by Nokia. This protocol uses UDP as its transport protocol. The idea of the protocol is that it is simple and fast.

This protocol has the following key aspects:

- Periodic messages over the 5G network.
- CMD message:
  - From remote control to AGV.
  - Contains: steering angle and target linear velocity + sequence number.
- ACK message:
  - From AGV to remote control.
  - Contains: actual steering angle, actual linear velocity, (others) + sequence number of CMD.
- Periodic message with T = 100 m (*):
  - If no CMD message is received for 3T = 300 ms (*), the AGV stops.

Co-funded by the Horizon 2020 Framework Programme of the European Union

- Message transport protocol: UDP.
- Message Payload format: JSON.
- Every JSON contains a field called SEQUENCEID. This field is returned in every ACK message so that it is possible to know:
    - If some messages have been lost.
    - Which is the Round-Trip Time of the message.

Regarding the video, the transport protocol is also UDP. To minimize latency, it is needed to rely on asynchronous protocols even though this implies that a potential loss of packets could take place.

As explained in other sections a VPN tunnel is also established between the UEs, the Edge and the Cockpit for the ToD scenario. OpenVPN software has been used to deploy this tunnel. Again, the choice between TCP and UDP for the transmission schema has been decided to be UDP.

Going back to the video, the transmission schema is RTP. This provides facilities for jitter compensation and detection of packet loss and out-of-order delivery, Inside the RTP stream, H264 encoded video is transmitted. The selection of H.264 versus H.265 was done as it has been demonstrated to be faster for encoding the frames even the video payload is bigger because of being less efficient. SPS and PPS fields are sent in every I-frame to ensure that the receptor of the video who needs to decode it can do that at any point in time (exactly every I-frame) and does not need to wait more. This is critical in our asynchronous video transmission schema and avoids the need of restarting the streaming from the cameras if for whatever reason the transmission was cut for some milliseconds.

Unity application is also receiving a forwarded version of the ACK response command containing the telemetry. This connection is also making use of almost plain UDP messages which have a payload which is a subset of the whole ACK message payload. They are triggering the visual rendering actions appropriately.

4.4 Haptic local operation

The nature of the haptic feedback triggered in the haptic gloves can be originated in lots of different scenarios, but we will focus on the meaning of haptics in local and remote operations. In the case of "local" operation, the haptic signal would be triggered due to a condition detected by the system the haptic gloves are directly connected to. It's indifferent if the connection is wired or wireless.

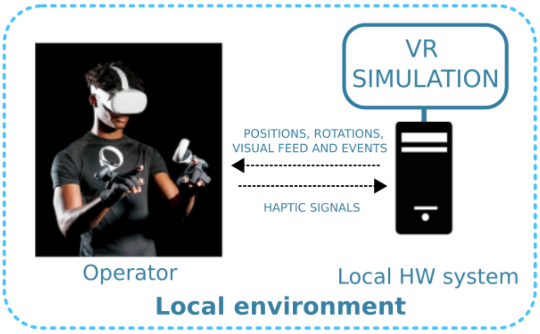


Figure 46. Local operation scheme where the haptic gloves are connected to a VR simulation

Co-funded by the Horizon 2020 Framework Programme of the European Union

An example would be a driving simulation, where the haptic gloves are connected to a computer with a videogame engine such as Unity, that using the SDK detects virtual incidences and trigger the appropriate signals.

In this case, all the inputs and outputs related to the haptic glove take place in the system they are connected to, and therefore, from the point of view of the gloves, these are local events. In this case, the delay between the incidences happening and the haptic response is negligible in most cases.

4.5 Haptic remote operation

In the case of "remote" operation, the haptic signal would be triggered due to a condition happening in a software or physical device (an AGV for example), not directly connected to the system the gloves are connected to. Let's call the system the haptic gloves are connected to "local environment", and the physical system where incidences take place "remote environment".

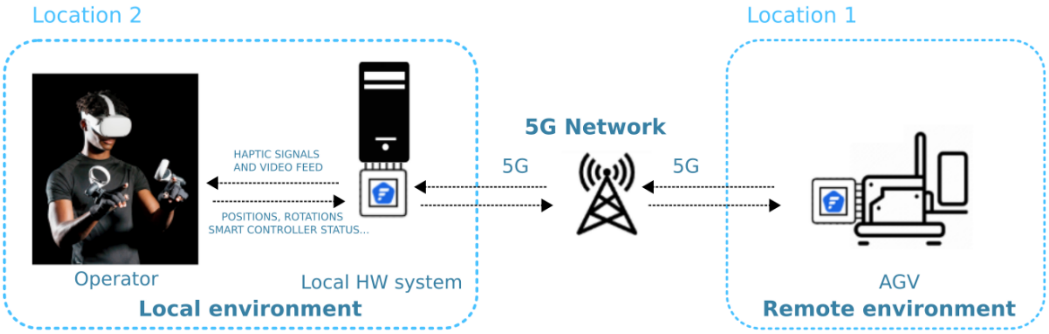


Figure 47. Remote operation where the haptic gloves are connected to a remote AGV

Due to the event taking place in a remote system, an additional delay will take place, as when the event is detected by the sensors in the remote environment, it must send the information to the local environment, and the local environment must then trigger the haptic feedback.

From the point of view of the haptic gloves, the main problem is not the data bandwidth, as the information necessary to decide if haptic feedback must be triggered or not is very compact, but the network lag that may cause the feedback to happen too late. If the haptic feedback takes place too late, it could cancel all the benefits of the haptic feedback approach, just the same way a video feed of a remote vehicle with a noticeable delay could result in problems while driving it.

The events that control the haptic feedback require minimal data transmission, and the use of a 5G Network allows us to guarantee the good performance of the haptic stimulus, just the same way it provides an adequate latency for a more complex signal, as the video feedback.

4.6 HMD Cockpit remote driving

The application in the HMD Cockpit needs to be aware of the point in time in which the AGV encounters an object and requests manual control to the physical end user behind the AGV. To do so an UDP message is sent back to the end user part of the application. This can be achieved with the telemetry information that the AGV is sending back to the upper layers of the system. The format of the message is quite simple. If inside the JSON payload a "TAKEOVER": 1 is found, a toast notification will be issued in the MR application and the remote driver will take control of the AGV.

It is important to note that before that takes place, the HMD application is receiving telemetry, video and the nodeJS application is sending remote driving commands normally, but the AGV should ignore them while TAKEOVER variable is 0.

4.7 Back to normal operation

Once the obstacle has been avoided, the remote driving procedure can be stopped and from the remote driving user all UDP commands will contain TAKEOVER variable set to 0. This change will be triggered by long press one of the Logitech G29 device which is attached to the cockpit setup.

4.8 Single Application Integration

Several HMDs, haptic devices and PC applications have been presented along this document. It should be desirable that all of them are either integrated or able to run together with a single narrative for the use case demonstration.

Haptic device can be easily integrated into a Unity project so that whenever some event takes place in the external world it is transmitted to it by means of vibration or any other way. The other way round will require to merge the existing unity project developed for using the glove as control device with the unity project used for ToD. In this fashion, a single application might be developed so that we have a uniform execution procedure.

Another possibility is just having two PCs connected to the MEC, one using the ToD application with Logitech G29 device and the other one with the haptic device connected. The two approaches are depicted in Figure 48.

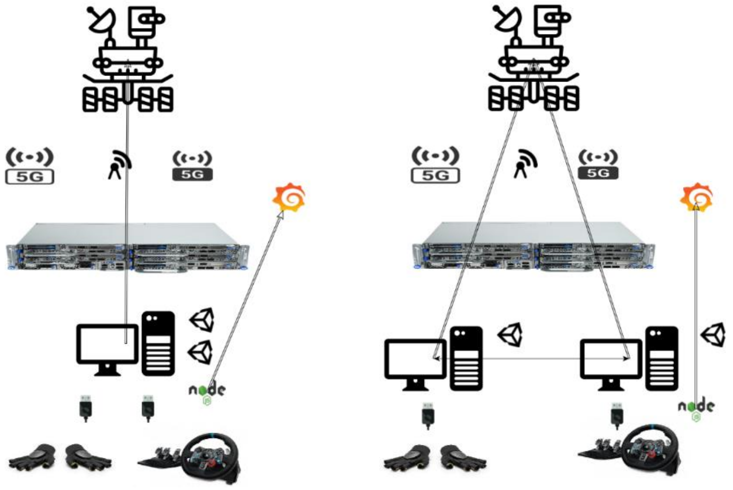


Figure 48. Application Integration Approaches

The conclusion is that the picture in the right will be chosen for simplicity reasons and taking into consideration that it is visually better for an external observer to understand the way that haptic and immersive devices can cooperate in the AGV use case scenario.

5 Exploitation, business potential and recommendations

5.1 Business Potential

The benefits for modelling future business cases related to the use of the combinations of these technologies can be described in a wide range of possible implementations.

One of them is **advanced surveillance in ports scenarios**: In this case we could have the fully remote tele operated driving scenario and a remote user is visualizing or supervising the video being streamed to the control centre. Moreover, an AI system running in the edge could advice of any anomalies and send a TAKEOVER signal to the cockpit altogether with a brief description and classification of the anomaly. From this moment on, the AGV could be manually driven to take a closer look at the detected anomaly and either take further action or discard it as a false alarm.

Surveillance in ports is typically based on fixed cameras but not on autonomous AGVs which are able not only to send video upstream to be analysed by humans or Ais but also interact with some other IoT elements. In this case, adding a pool of remote humans controlling these AGVs could provide a more comprehensive view of the status of the installations as well as offering a cheaper and safer service compared to having security personnel invigilating and monitoring the premises. Examples of some studies related to this can be found in [13]. However, most of the studies are focused on the water side rather than in the ground side as seen in [14] and [15].

Another topic is about **remotely handling small packets** or components to some areas of the port. This can be achieved by enhancing our current approach to mimic the haptic glove to some robotic arm attached to the AGV. In this way it should be possible to transport back and forth small things that otherwise would require an actual human to be performed. The benefit here has to do for the case in which 90% of the AGV duties are related to handling big packages or other maintenance tasks. Due to this, humans are required to work in a potentially dangerous environment. This extension in the AGV side could avoid the described situation and reduce some costs.

The **scalability and the increase in the capacity of the number of operations** that can be carried out **in maritime ports** is another benefit from using haptic and remote immersive devices due to the fact that it can automate and speed up some manual operations that currently are carried out by human persons.

Additionally, **people** who are hired for some of the jobs which are related to the AGV use case could **work remotely from their homes** by using the network infrastructure without the need to physically go to the port and avoiding some of the security risks associated with performing the tasks in site. **Training** for these jobs can be done by developing **simulations** of the real scenarios with a virtual environment combining immersive videos from real situations.

Co-funded by the Horizon 2020
Framework Programme of the European Union

5.2 Exploitation

Using Mixed Reality devices in several communication projects is crucial for companies such as Nokia who is aimed to understand the network requirements and experiment with industrial test cases such as remote driving with haptic feedback. The profile of the quality levels required in time and location will be different from typical immersive communications and this offers an opportunity to get a deeper knowledge of how a smart network might help.

The potential of the haptic glove in industrial environment combined with immersive applications in 5G is a field of exploration for NED, which could benefit from working and integrating with industrial and telecommunication partners to provide services which are still not massively deployed but just at investigation level.

A more detailed on this subject will be presented as part of deliverable D7.3 Final dissemination, standardisation and exploitation.

5.3 Recommendations

Some recommendations from the experiences in the iNGENIOUS project and the tests which have been performed so far are:

1. It is recommended if it is possible directly connect the PC controlling the HMD and/or the haptic device directly to the same subnetwork or to another one connected to the MEC/edge server. Cloud connectivity and Wi-Fi connections are not offering the best in terms of latency, reliability and low variability of KPIs for ensuring a safe and comfortable experience when dealing with remote vehicles operations.
2. E2E protocols should focus on asynchronous flavours such as plain UDP for messages or RTP on top of UDP for video streaming, ensuring that in case of a cut the service is immediately recovered
3. Tunnelling the communication is recommended so that the data is not intercepted, and NAT issues are no longer present when trying to redirect the data flows. This implies setting up a VPN connection either in the HMD or in the PC,
4. Every AGV should use their own way for receiving control commands, this means a sort of abstraction layer is required to run in a light board along the AGV so that the software which is running in the HMD part does not care about this.
5. Video and control flows should separate as much as possible. Ideally using different IP addresses. This will ensure that when the bandwidth is increased due to the video content, the control part is not affected so much.
6. HMD devices are OK for driving remotely, but probably not recommended for longer sessions. They are ideal for getting concentrated and focused for short periods of time.
7. Complementing the HMD visualization with haptic feedback either in steering wheels or haptic gloves could help remote driver to react faster to adversarial situations that just only using their eyes.

Co-funded by the Horizon 2020
Framework Programme of the European Union

8. Millimetre waves 5G works fine if the right antenna lobe is the one being used for transmitting and has a clear view path from the AGV. Latencies need to be constantly monitored in the HMD application so that the optimal path is followed if possible.
9. The complexity for developing and E2E application needs to be reduced as the HMD and Haptic devices developers should be able to have a simple API in the layers behind to configure the connectivity, control the AGVs, activate the video streams and parse the information being received from the real world including the AGV’s telemetry.

Co-funded by the Horizon 2020
Framework Programme of the European Union

6 Conclusions

This document has described how HMDs and haptic devices can really help to improve ToD scenarios for AGVs in industrial environments and which is the architecture in terms of software, network communications and design required to make everything work.

5G Networks, maybe in the form of private networks are critical to achieve this mission as they can cover a very controlled area such as a factory or port. Moreover, mmW deployments are more suitable for enabling very low latencies and scaling up with higher bandwidths of devices that need to stream in the uplink and are close to the 5G antenna.

Haptic devices help to produce a rich multisensorial input for the remote operator that helps him/her to react in times which are significantly lower.

There is a high complexity when developing complex E2E environments in which we include 5G, IoT, haptic and immersive devices that requires from knowledge of all these areas so that an efficient integration is possible.

Immersivity helps the end user feeling like in the real world, however, this could be potentially more tiring than using other non-immersive devices. So, it is important that there are some complementary traditional ways of interacting.

Subjective testing is also a key point to explore with real users how these technologies compare to the classical approaches to the problem.

Co-funded by the Horizon 2020
Framework Programme of the European Union

References

[1] iNGENIOUS Deliverable 3.1 available at: https://ingenious-iot.atlassian.net/wiki/download/attachments/2129921/iNGENIOUS-5G_D3.1_v1.0.docx?api=v2

[2] INGENIOUS Deliverable 3.2 available at: https://ingenious-iot.atlassian.net/wiki/download/attachments/1998858/iNGENIOUS-5G_D3.2_v0.5.pdf?api=v2

[3] INGENIOUS Deliverable 4.2 available at: https://ingenious-iot.atlassian.net/wiki/download/attachments/1245324/iNGENIOUS-5G_D4.2_v1.0.pdf?api=v2

[4] A parametric quality model to evaluate the performance of tele-operated driving services over 5G networks https://link.springer.com/article/10.1007/s11042-021-11251-x

[5] Toward the Distributed Implementation of Immersive Augmented Reality Architectures on 5G Networks https://ieeexplore.ieee.org/abstract/document/9722829

[6] ITU-T P.919 Subjective test methodologies for 360° video on head-mounted displays https://www.itu.int/rec/T-REC-P.919/en

[7] The Influence of Subjects and Environment on Audiovisual Subjective Tests: An International Study https://ieeexplore.ieee.org/document/6286980

[8] ITU-T Rec. P.910 Subjective video quality assessment methods for multimedia applications https://www.itu.int/rec/T-REC-P.910

[9] Teleoperated Driving, a Key Technology for Automated Driving? Comparison of Actual Test Drives with a Head Mounted Display and Conventional Monitors https://ieeexplore.ieee.org/document/8569408

[10] Enhancing Telepresence during the Teleoperation of Road Vehicles using HMD-based Mixed Reality https://www.researchgate.net/publication/301336871_Enhancing_Telepresence_during_the_Teleoperation_of_Road_Vehicles_using_HMD-based_Mixed_Reality

[11] ITU-T P.809 Subjective evaluation methods for gaming quality https://www.itu.int/rec/T-REC-P.809/en

[12]Teleoperation the Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters https://dl.acm.org/doi/10.1145/3342197.3344534

[13]An Overview of Maritime and Port Security https://up42.com/blog/tech/an-overview-of-maritime-and-port-security

[14] A Port waterside security systemic analysis https://www.researchgate.net/publication/251993442_A_Port_waterside_security_systemic_analysis

[15]A security and surveillance application of the innovative Monitoring Underwater Buoy Systems (MUnBuS) on the Protected Marine Area (AMP) of capo gallo (PA-IT)https://ieeexplore.ieee.org/document/7404560

Co-funded by the Horizon 2020
Framework Programme of the European Union

Appendix A: Subjective Testing Framework

This framework defines two technical experimental tests to separately evaluate video coding quality and end-to-end delay or latency. Both parameters are Key Performance Indicators (KPIs) used in a parametric quality model to evaluate the performance of tele-operated driving services over 5G networks.

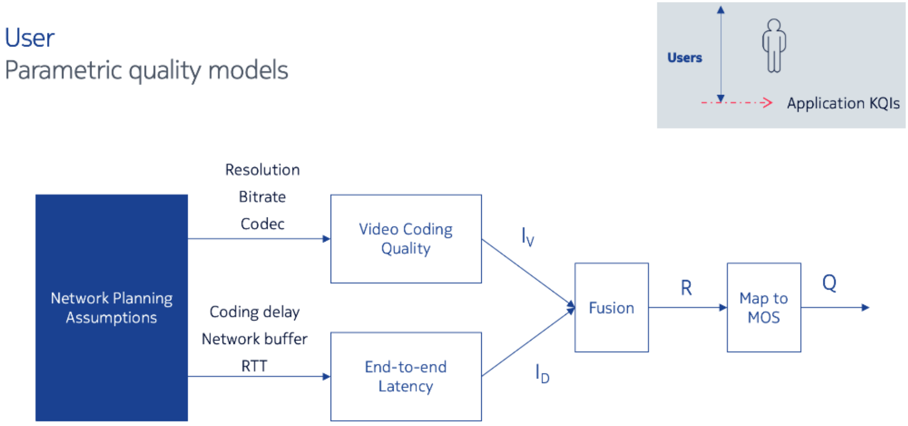


Figure 49. Parametric Quality Models for Subjective Testing

Video Coding Quality

There exists a trade-off between video quality and file size. High-quality videos result in large files and require high-speed bandwidth. On the other hand, low-quality videos are smaller files requiring low-speed bandwidth.

Video quality depends on the coding bit rate. The aim of the test is to determine which compression level ensures enough visual quality to support ToD.

This is a passive test in which the user will evaluate 360° pre-recorded videos using a headset. Each video will be evaluated through ACR. The method is based on a five-level rating scale.

Procedure for passive testing

1. Recording videos from a driving vehicle

Videos could be recorded in the port with a predefined circuit. The vehicle will be provided with a multi-camera system. One camera will be located at each side of the vehicle. Another camera will be placed in the front part and finally, a fourth camera is placed in the back.

Video characteristics:

- Time of each video: Following the recommendations of ITU-T P.919 [6] each video will last between 10 seconds and 30 seconds. As ToD requires time, the recommendation would be recording 30 seconds long videos.
- Preferably, an experiment should be designed so that each subject's participation is limited to 1.5 hours, of which no more than 50 minutes is spent rating stimuli, and no more than 25 minutes continuously.Video sequences: There are three different ToD use cases (driving, parking, supervision). Videos of these 3 situations are to be included.

- The number of times that each source stimulus is repeated also impacts subject fatigue. Among different possible test designs, preferably choose the one that minimizes the number of times a given source stimulus is shown. For this reason, the use cases of driving, parking and supervision should be displayed in a sequential order. For example, one user will watch driving, then parking, then supervision, and it would be followed by driving once more.
- Audio: This test is audio-visual. When a person is driving, it is forbidden to use headphones. When wearing headphones, the driver is most likely not aware of all the sounds around them. For that reason, to ensure ToD is safe, it may include real-time audio. This is the reason why the videos to be recorded could potentially include audio.
- Number of videos: We have 5 levels of rating, as we should include a video per level of quality there has to be a minimum of 15 videos (3 types of use cases times 5 different qualities – different bit rates).
- Speeds: Driving velocities should cover a range from 2 km/h to 20 km/h. Thus, it would be interesting to have videos every 4 km/h of difference (at 2, 6, 10, 14, 18)
- For parking it would be enough just to have videos at the same speed (usually 2 km/h) but at different qualities. This would mean having 25 videos of driving and 5 of parking.

2. Video compression at different levels

Spatial Information (SI) and Temporal Information (TI) are defined by ITU-T Rec. P.910 [8] as an approximate measure of video content complexity. They have been widely used in the field of quality assessment, for the selection of the video content to be used for the subjective tests, that should be representative of different complexity classes.

Each video will have a different setting. The parameters that may vary are:

- Codec
- Frame rate
- Frame size / Resolution
- Fields
- Bitrate

The test should show sequences that viewers would rate from bad to excellent.

3. User visualization

Each user will watch the 360° videos with a Head-Mounted-Display (HMD). They will visualize the videos at the Nokia offices in Madrid. The environment should be non-distractive and silent.

User's characteristics:

- Number of people Following the recommendations of ITU-T P.919 [6] at least 28 subjects must be used for experiments conducted in a controlled environment. I suggest performing the test on 30 people.

- Viewers with normal vision or corrected-to-normal vision.

Co-funded by the Horizon 2020
Framework Programme of the European Union

- Age, gender, cultural, social, and economic background A systematic study involving six laboratories in four countries found no statistically significant impact of subject's language and culture / country of origin on video quality ratings as explained in [7].

4. User evaluation through Absolute Category Rating (ACR)

User evaluation should take place at the same time the person is visualizing the videos in such a way that the person sees the video, rates it and automatically the next video appears to the individual.

5. Extras

ITU-T P.919 [6] recommends doing a training session with some example videos to familiarize with the evaluation method, the interface, and a reference of the range of available qualities. This will ensure the person has correctly adjusted the HMD (volume, camara focus, etc.).

They do not have to be related with driving sequences. Therefore, to simplify the procedure, I would recommend using videos which are already recorded.

6. End-to-end Latency

Latency is the delay in data transfer. It is a parameter for measuring network performance with subjective video quality. In this case, the context is a ToD situation.

Due to the physical limitation, the transmission of information of any kind can never be latency free, leading to one of the main problems with teleoperated systems: controlling a remote robot with delay.

This is an active test in which the user will drive an autonomous vehicle with a headset in an immersive environment whose scenario is a real-time situation.

Procedure for active testing

1. Circuit design: To avoid damaging the vehicle cube foam should be used. Example of a circuit from the paper [9].

1) **Narrow corner:** The operators must drive around a 90° corner.

2) **Close driving side:** Two foam cubes are positioned at the side of the street, reaching 25 cm into the lane.

3) **Stopping at a line:** The operators have stop as closely as possible at a line before an intersection.

4) **Narrow lane change:** Eight stacked cubes (4x2) are positioned as two gates.

5) **Obstacle avoidance:** Four cubes are stacked (2x2) and placed halfway on the street. The task for the operators is to drive as closely as possible straight forward onto the foam wall and then change the lane without hitting the cubes. This simulates waiting behind a parked car to let another car pass.

6) **Parking:** The operators must drive around the corner and then park the car between eight stacked cubes. This simulates parking the car into parking slot. It could be interesting to make the user complete this circuit twice.

Co-funded by the Horizon 2020
Framework Programme of the European Union

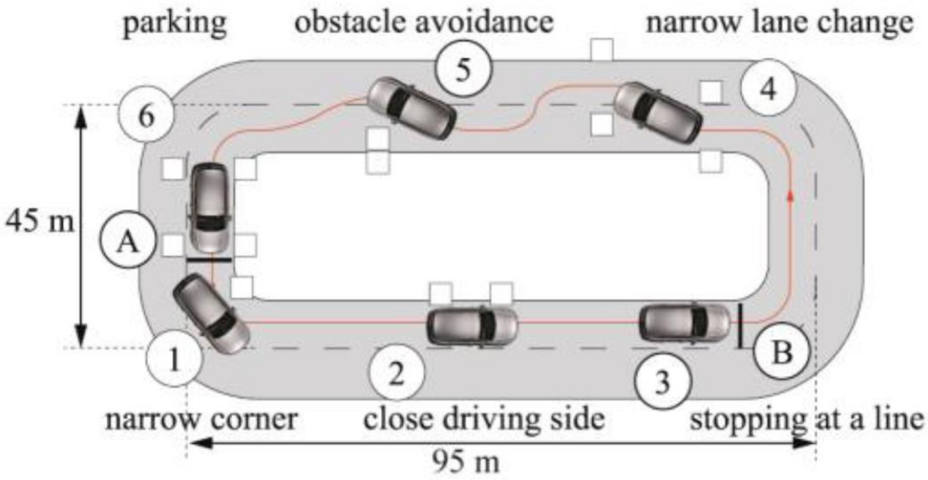


Figure 50. Typical Track Circuit for Tests

2. User experience testing

ITU-T P.809 [11] defines two different types of interactive tasks. Short interactive tasks (90-120 seconds) or long interactive tasks (10-15 minutes). Taking into consideration that in the real-case situation, if an autonomous device is facing a problem which requires the action of a person though tele-operated driving, probably the solution to the issue will not be short-lasting. As this is a preliminary study, I would recommend each experience to be 5 minutes long.

3. Analysis of results

Several parameters can be analyzed through this test:

- Time to complete the circuit
- Average driving speed
- Number of collisions with the cube foams
- Difference between distance perception and real distance
- Workload of the human operators during the test drives can be analyzed using the NASA Task Load Index (TLX).
- Subjective quality of the experience, ACR. This questionnaire should be performed on the subjects after they have completed the circuit.

4. Embodiment analysis

The second main problem with teleoperated systems is the limitation of information provided to the operator. Since the operator is not physically at the robot's location, he must build a mental representation of the robot's environments based on the feedback provided from the robot's sensors. These feedbacks are inherently imperfect and distorted leading to an incomplete description of the environment. This may lead to operator fatigue, weak situation awareness and a weak feeling of telepresence.

For this reason, embodiment questionnaires should be done to test whether the system allows for a seamless immersion in the ToD environment.

In [9] they propose the following embodiment questions to be rated from 1 (disagree completely) to 9 (fully agree):

Co-funded by the Horizon 2020
Framework Programme of the European Union

“My perception of the lateral distances was always good”
“My perception of the longitudinal distances was always good”
“I always trusted the system”
“The vehicle reacted immediately to my commands”
“My interaction with the environment was always natural “
“The interface feels realistic”
“I always had a good overview over the environment”
“I was able to handle the system from the beginning very well”
“The control method was intuitive, and I could always focus on the driving task”
“I was always in full control over the vehicle”

Extras

ITU-T P.919 [6] recommends doing a training session. In this case, the training session will consist of a test to learn how to drive the vehicle. This way, they can become familiar with the system, setup, and how is the field of view visualized so that embodiment is increased.

There is a paper for evaluating the degree of immersivity of the situation [10] in which they measure the steering wheel reversal rate (SRR) in which is said:

“Another important objective criterion to evaluate the lateral stability during teleoperated driving is the steering wheel reversal rate (SRR). Lower SRR values represent higher lateral control of the remote vehicle by the human operator. The total average of SRR values is increased using the proposed mixed reality HMI concept. The main reason for this performance degradation is that the human operator can see neither the steering wheel nor his hands during driving while using head-mounted display”.

By using segmentation, we could also include the hands of the user.

Paper [12] suggests the creation of dynamic scenarios with different levels of input/output and inter-frame latency by applying buffers.

HMD Considerations

Subjective testing is related to how a person lives the experience of using immersive HMD devices, in this case for driving a vehicle remotely.

To establish some basic rules, we can think of

- Which are the configuration parameters that we can change in the scenario? (1)
- During how long is the experience test conducted? (2)
- Which are the references for possible comparisons? (3)
- Which is the device under test: (e.g., Quest2 vs Varjo) (4)

Regarding 1, the parameters that we can tune are basically the bitrate of the UDP video streams being sent. We will work with a fixed resolution of 1920x1080p and 30 FPS in the camera side, but we can move the bitrate in the range of 3Mbps – 10Mbps for H264 encoding. We can also artificially increase the latency in the network… or just annotate the one under the test and repeat the tests several times under different 5G coverage conditions. Maybe this is another point to be taken into consideration in this kind of tests.

Co-funded by the Horizon 2020
Framework Programme of the European Union

For 2, it is important to have at least 4 different durations, starting in 5 minutes and being up to 45 minutes time. In this case another variable that matters is the fact that all the time the subject is remotely driving, only partially during the whole test time or she/he is just monitoring the AGV movement.

For 3, we need to think about external similar situation. One of them could be driving a real vehicle as we usually do in our daily life. Second one could be comparing to similar videogames with and without using VR. Third comparison maybe driving the vehicle in case of AGV using a typical video game remote controller.

Finally for 4 we could compare using Quest2 or VarjoXr3 devices.

A questionnaire can be written, and it is possible to register the answers to the combinations of 1-4 that we want to test. Levels ranging from 1-5 are recommended for this kind of questionnaires.

Haptic Devices Considerations

The subjective testing in the area of the Haptic devices should be focused in these areas:

- How is the reaction time improved when receiving haptic feedback instead or in addition to visual feedback? Possible answers are:
    - Much better reaction time.
    - Better reaction time.
    - Same reaction time.
    - Worse reaction time.
    - Much worse reaction time.
- When managing the movement and the commands toward the AGV, how is the experience in terms of naturality. Possible answers are:
    - I feel really close to the way I would do with my own hands
    - It is an additional degree of simplicity
    - I don't feel a real difference
    - It takes more time and effort to issue the commands
    - It is not possible for me to handle the AGV
- When combined with HMDs which are the characteristics of the immersivity that are augmented due to the use of the haptic device?
    - The sensation of being there
    - I can almost touch some items
    - Getting rid of the normal controller is good
    - The virtual world seems more realistic
    - Movements and gestures are easier, and I focus in the environment

This could be an initial questionnaire. Ideally a representative group including female and male persons and people who previously have used immersive devices and never or occasionally used them should be part of the study.

Co-funded by the Horizon 2020
Framework Programme of the European Union