



Virtual reality interface for the guidance of underwater robots

Final Degree Project in Videogame Design and Development

Marcos de la Cruz Soler

Supervisor: Dr. Pedro José Sanz Valero

Contents

1	TECHNICAL PROPOSAL	4
1.1	Introduction	4
1.2	Related subjects.....	5
1.3	Project goals.....	5
1.4	Expected results.....	5
1.5	Project Planning.....	6
1.6	Tools.....	7
1.6.1	Development	7
1.6.2	Audiovisual	8
1.6.3	Virtual reality.....	8
2	RELATED WORK	9
2.1	Background.....	9
2.1.1	Underwater Intervention Systems.....	9
2.1.2	User Interface Systems.....	12
2.1.3	Why Unity 3D and not UWSim?.....	15
2.2	HRI Design Specifications	16
2.2.1	Summary	16
2.2.2	Control Design	16
2.2.3	Underwater environment and items effects	19
3	MAIN DEVELOPMENT.....	24
3.1	Introduction	24
3.2	Arm integration in Unity	24
3.3	Vehicle integration and movement	27
3.4	Underwater Environment.	28
3.5	Canvas and interface information.	29
4	RESULTS.....	30
4.1	Performance Analysis.....	30
4.2	Usability Testing	31
5	CONCLUSIONS.....	35
5.1	Future Work	35
6	REFERENCES	36
7	ANNEXED DOCUMENTS.....	38

List of Figures

Figure 1. Project Gantt Diagram	7
Figure 2. A usual ROV control room.	10
Figure 3. G500(left) and its 3D model(right)	11
Figure 4. ARM5(left) and our 3D Model(right).	11
Figure 5. Immersion with “Oculus Rift” and LM.	12
Figure 6. Conventional teleoperation control(left) and Control using an HMD(right).	13
Figure 7. Multiple LM control, HMD and LM(left), with user in VR(middle) and with airflow-based haptic feedback(right).	13
Figure 8. HTC Vive starter Kit.	14
Figure 9. Twinbot in UWSim	15
Figure 10. Mode switch.	16
Figure 11. Vehicle controls.	17
Figure 12. Arm controls.	18
Figure 13. Engine Working.	19
Figure 14. Water Particles.	20
Figure 15. Seabed	21
Figure 16. Water Surface	21
Figure 17. No camera effects(Up), Blur and Fog effect(Down).	22
Figure 18. Black Box model (Left) Reachable Black Box(right).	23
Figure 19. Firs version of the arm in the simulation.	24
Figure 20. Slew Movement Script.	25
Figure 21. Order Boolean received by the VR controller.	25
Figure 22. Unity hierarchy of the Arm.	25
Figure 23. Script variables in Unity.	26
Figure 24. Jaw contact in Unity.	26
Figure 25. Order sent to the particle system of the object.	26
Figure 26. Addition of force up to the speed limit of the engine.	27
Figure 27. Original state of the simulation.	27
Figure 28. Engine particle systems force script.	28
Figure 29. Script of the shader in charge of the seabed and water surface noise .	28
Figure 30. Text display controller(Movement Section).	29
Figure 31. User visible canvas.	29
Figure 32. Charge test with different graphic cards.	30
Figure 33. Information of the pilot group in the usability test.	31
Figure 34. Average times and tries for every test.	31
Figure 35. First trial of the usability test.	32
Figure 36. Second trial of the usability test.	32
Figure 37. Third trial of the usability test.	33
Figure 38. Fourth trial of the usability test.	33
Figure 39. Likert result of the usability tests.	34

Motivation

The main motivation to establish this project was my interest in virtual reality, I am intrigued by the amount of possibilities it can offer and how it can evolve. I also wanted to make an interface that was useful once finished. Thanks to the professor P. J. Sanz, who was willing to guide a project of these characteristics and to his recommendations and help during all the development time we were able to make this project oriented to HRI in underwater interventions

1 TECHNICAL PROPOSAL

1.1 Introduction

The virtual reality is a branch of the technology which is gaining impact in the last years at a great speed, its main factor is that the immersion offered is superior if you compare it with other actual technologies.

With this in mind, we can say that, through the capacity of compress the information and show it in an accurate way, it is ideal for entertainment and simulators to learn how to pilot different vehicles.

One of the main topics in the evolution of robotics is the *human-robot interaction* (HRI) which groups all the human and machine relations. It is a multidisciplinary field with contributions from a lot of different fields of research such as human-computer interaction (HCI), artificial intelligence (AI), robotics, design, natural language understanding and social sciences. HRI is also very important in this project and we will guide the project to the Underwater Interventions.

Nowadays, if we talk about the underwater interventions with robots, the groups needs plenty of different equipment in order to receive the information to do the guidance of the robot, this generates a lot of stress in the robot guide and make them unable to do long sessions navigating the rob.

In this project, we are looking to make an interface that is not so aggressive with the user and can be used during a long period of time without generating stress on the guide. That is the reason why we have decided to make this project in a virtual reality environment, to reduce the number of equipment needed for the interventions and the stress of the guides.

1.2 Related subjects

- VJ1231 Artificial Intelligence
- VJ1227 Game engines
- VJ1224 Software Engineering
- VJ1208 Programming II
- VJ1203 Programming I

1.3 Project goals

- Facilitate the training of robot guides in underwater intervention missions.
- Shortening of the number of instruments necessary for the handling of said robots.
- Creation of a simple but reliable interface that is easy to manage and learn.

1.4 Expected results

We expect to make an interface that allows both, the training of robot guides and doing the interface a viable tool that can be use in real interventions which is possible to teleoperate the robs reducing the volume of equipment required.

The main goal is that the interface works in the simulation at the end of this project and in the future, we want to extend the functionality to make it usable with real robots.

1.5 Project Planning

Documentation

ID	Task	Period	Hours
TP	Technical Proposal	01/02/2019 to 03/02/2019	6
DA	Analysis and Design	22/02/2019 to 24/02/2019	7
PM	Project Memory	07/02/2019 to 03/06/2019	30
PDV	Project Defense Video	07/02/2019 to 05/06/2019	10
PDP	Project Defense Presentation	10/02/2019 to 05/06/2019	5
Total	60		

Research

ID	Task	Period	Hours
SC	Steam VR connection	25/02/2019 to 05/03/2019	5
VFX	Special Effects	06/03/2019 to 15/03/2019	5
RM	Ros modules	16/03/2019 to 01/04/2019	10
VRI	VR interfaces	02/04/2019 to 25/04/2019	20
Total	40		

Development

ID	Task	Period	Hours
UEI	Underwater Environment Implementation	25/02/2019 to 05/03/2019	30
VID	VR Interface Development	06/03/2019 to 15/03/2019	35
CI	Control integration	16/03/2019 to 01/04/2019	40
Total	105		

Design

ID	Task	Period	Hours
VC	Visual Effects Creation	25/02/2019 to 10/03/2019	15
TCV	Texture Creation for VFX	20/03/2019 to 25/03/2019	10
IG	Interface Graphics	06/03/2019 to 01/04/2019	30
MI	3D Model Integration	02/04/2019 to 25/04/2019	40
Total	95		

Testing

ID	Task	Period	Hours
DT	Development Testing	25-02-2019 to 05-03-2019	15
FIT	Final interface testing	10-05-2019 to 25-05-2019	15
Total	30		

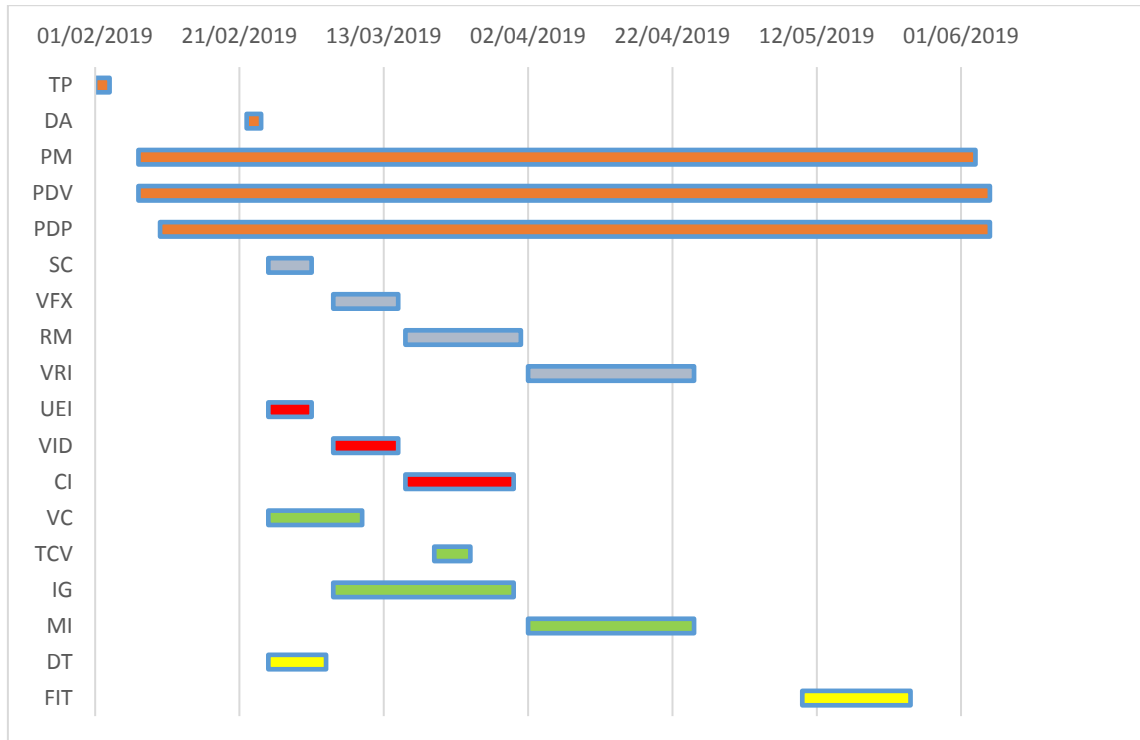


Figure 1. Project Gantt Diagram

1.6 Tools

1.6.1 Development

Unity 3D

Multipurpose game engine that supports 2D and 3D graphics and VR environments with the SteamVR plugin, Oculus and HTC Vive. It is the main environment where we will work during the project to set up the virtual environment of the interface.

Visual Studio

Microsoft Visual Studio is an IDE used to develop programs, as well as web sites, apps, services and mobile apps. It includes a code editor and has got an integrated debugger. We will use this IDE to make the scripts of the project and the shaders of the virtual environment.

SourceTree

An alternative to GitHub that was made by Atlassian Corporation. It is another desktop client for developers. It is used with a host service such as Bitbucket to make a version control system for our project.

Bitbucket

A web host service for projects that uses Mercurial and Git version control systems. We will use this with ST and Google Drive in the development of the project.

Google Drive

A file storage and synchronization service developed by Google that allows the user to store files and synchronize them across devices.

1.6.2 Audiovisual

Adobe Photoshop

Ps is a raster graphics editor developed by Adobe Systems. It can either create or edit raster images with multiple layers and supports masks, alpha and colors model like RGB, CIELAB, CMYK, duotone and spot color. We will use this tool to make all the graphics needed in the interface such as particle effects and the UI.

Vegas Pro

Vegas pro is the new name for Sony Vegas, a non-linear system editor of video created by Sony Creative Software and acquired by MAGIX.

OpenShot

Another video editor, but this one is open source and free, created by OpenShot Studios, LLC. We will use it with Vegas P. in order to make the videos related with the project.

1.6.3 Virtual reality

HTC Vive

A set of virtual reality glasses developed by HTC and Valve. The device is designed to use the space in a room and dive into a virtual simulated world where the user can walk and interact with multiple objects. This is the hardware that we will use in this project, it is a set with a cockpit, two controllers and two basestations.

SteamVR unity plug-in

A unity plug-in developed to create interactions between the VR systems and the virtual environment in Unity. We will use it in order to make easier the communication between the equipment and the videogame engine.

2 RELATED WORK

2.1 Background

2.1.1 Underwater Intervention Systems

During the first sections of this document we have talked about the project and its objectives. Now we will talk about the precedents of the project.

Following the control of robots in underwater interventions, the first teleoperation experiments took place at the end of the 19th century. (García, et al., 2015) in which the users did not press only one button to turn on the system, instead the user was given control of the system and they were able to make decisions based in the information that they had, which should not be perfect. Due to this, the human situation analysis is a very important factor, which has entailed the evolution of the systems to be designed focused on the user that will manage them and to avoid their fails and the system performance. This is the reason why the actual research is focused in reduce the stress generated in the user giving him tools that can improve their performance and reduce its fails.

Next and citing the reference (García, et al., 2015), among the most critical errors that can be committed by humans are the following, the user introduces a wrong command, the user use a command to late and the user don't use a command at all. The main causes of these errors are:

- Lack of training when acting on the information available, which may lead to the pilot not performing the actions correctly in an unforeseen situation.
- The time necessary to perform a correct analysis of the information. This may increase due to cognitive fatigue.
- Don't receive information from the sensors. The user should not see the information from a part of the interface or details when focused on one task.

It is important to have a plan B when this type of situation happens which in the worst case can lead to the loss of the vehicle, for example, The G500 has some instructions that make it emerge and activate a LED that blinks when the connection is lost. With this facts in mind we will explain how the simulation treat everyone of them.

First, the lack of knowledge of the user at the time of making an intervention the pilot must know how to use all the resources available and commit the minimum mistakes possible when it is managing the equipment and vehicles. The simulation is very important for this purpose like the fly simulators for the airplane pilots, the more time you do exercises in a simulated environment similar to the real intervention, the fewer the chances of error on it, because the user will have the competences to make decisions in different situations that can take place.

Second, in the our interface, one of the main objectives is to reduce the amount of equipment integrating the virtual reality to make easier the labor of the pilot, reducing the number of screens and sensors that the user needs to take care of.



Figure 2. A usual ROV control room.

That is why the simulation takes advantage of the canvas offered by the virtual environment to give information on different factors, such as the speed of the rove, and its rotation. All of this is linked to the third point, the user will always have in its field of vision the canvas with information despite of being performing a specific task, the pilot also has the possibility to observe all the relevant information without searching through different screens.

Foremost, in the last years the advances in virtual reality have revolutionized the world of simulations, granting a new perspective and a new flow of work to them. In order to make this project we also had in mind the problem that the robot guides experience during the sessions. That is why we have taken into account both virtual reality(VR) and underwater robotics(UR).

There are plenty of works and research about the UR but we have mainly paid attention to those accomplished at the university for example the Twinbot project^[1] and the Merbots project^[2]. The first is a cooperative project between UdG(Girona University), UIB (Balearic Islands University) and UJI(Jaume I University) consists of two robots that collaborate with each other to carry out

the interventions, more specifically, the IRSLab (Interactiva and Robotic System Lab) works in the communication section between the robots (COMOMUIS project), a branch of this project is the one of the user interfaces that is where the project that we describe would be cataloged. The second, is the previous project to the Twinbot, and between both, they have given rise to the G500 which is a vehicle developed by the UdG (Girona university) as a part of the project. We will use a 3D model of this robot in our simulation.

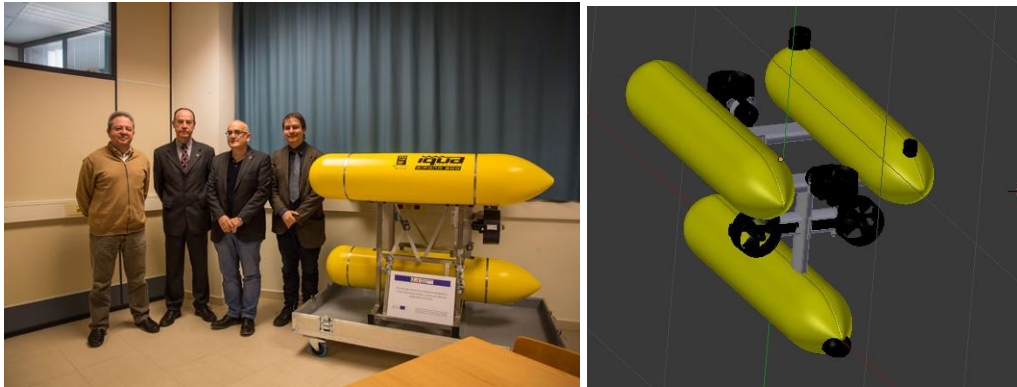


Figure 3. G500(left) and its 3D Model(right).

In our simulation, the robot has an arm that corresponds with a variation of the ARM 5 that was developed for the G500 [3]. It has several joints with 2 degrees of freedom, the slew has a range of 120° , the shoulder has a range of 85° , the elbow 130° and the wrist has a continuous range, the jaw can be open and close. The difference between our model and the ARM 5 is the jaw shape.

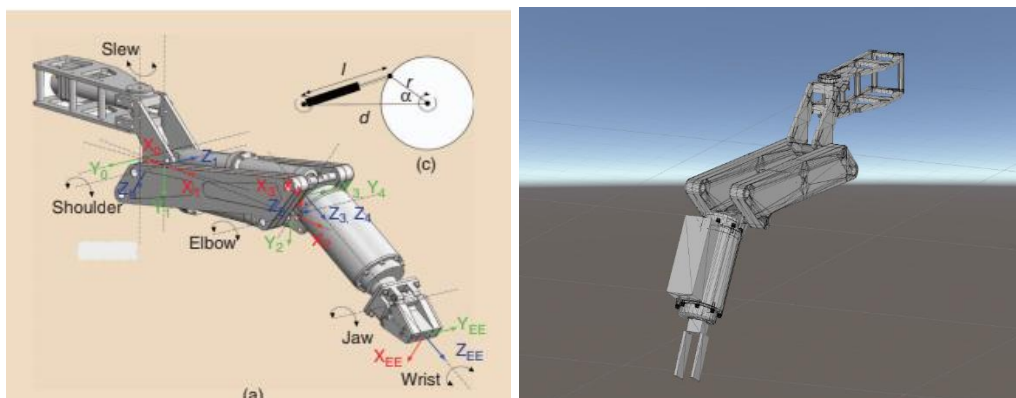


Figure 4. ARM5(left) and our 3D Model(right).

2.1.2 User Interface Systems

Regarding the references in virtual reality we have also taken into account precedents within the university but one of our main reference is the “*Vision-Based Control for an AUV in a Multi-robot Undersea Intervention Task.*” [4] which is one of the last works where the VR was use in UJI. It consists of a research work on the use of different technologies mixed with VR for the control of underwater robots.



Figure 5. Immersion with “Oculus Rift” and LM.

Based in the results obtained in this previous research and after looking at the advantages and disadvantages of every control combinations applied, we made the decision about which virtual reality system we should use. The first idea of use a joystick with different screens and devices that shows the information to the user, was discarded almost immediately since this violated one of our main points, to reduce the number of equipment needed to carry out the interventions in order to make the work easier to the pilot and improve its performance.

The second option that was considered was to use a virtual reality cockpit in combination with a joystick, this was one of the options closest to the one that has been used in the project but it had the problem that the number of options that one joystick could gave us was not enough to control fluidly the whole system, this can be solved with two joysticks, but this could make difficult to coordinate the different options and movements of the vehicle. That is why we decided to modify this a bit.



Figure 6. Conventional teleoperation control(left) and Control using an HMD(right).

The third option was to use the Oculus cockpit with a Leap Motion system that is able to recognize the position and gestures of the hands, due to the Rov control, establish a group of clearly differentiate gestures will suppose a problem for both, us and the final user because the development of this rules and its learning will be difficult due to the reduced precision of this system making a high possibility of error in a real intervention.

The two other versions that are covered in the referenced document are variations of this previous one, so the problems are similar. With all of these facts in mind, we decided that our best choice was to make use of controls that were already integrated in the virtual environment and gave us, at least the same number of options that two joystick can offer if not more, but in a simplified way. That's why we decided to change the cockpit of the Oculus on to the HTC Vive (HTC Corporation, n.d.) which has got two controllers and gave us a condensed dual joystick set up.



Figure 7. Multiple LM control, HMD and LM(left), with user in VR(middle) and with airflow-based haptic feedback(right).

The HTC vive controllers have integrated touchpads, which are a system that are able to recognize touch of the fingers, this reduce the problem of having to move excessively the hands that had the joysticks since no wrist movements are necessary to control the vehicle in the simulation and the user can access to all the options with only a finger move, this fact simplify its utilization. They also have lateral buttons that can be pressed clenching the hand , this gave us different options to use that buttons. Finally, they have got triggers that make us able to enable and disable options, an interesting fact of this triggers is that they have a build-in pressure controller which make us able to assign different options depending of the value of pressure if we want to.



Figure 8. HTC Vive starter Kit.

2.1.3 Why Unity 3D and not UWSim?

The IRSLab has its own software that simulates an underwater environment with robots called UWSim, then, why do we make a new one using Unity instead of use a simulator that we already have? The answer is simple, for the compatibility of each one with the VR equipment, while unity has a lot of plug-ins that make the use of this equipment an “easy” thing, UWSim has not this kind of options nowadays, that is why the usability test will be delayed and maybe, the hole project if we need to make the drivers and camera setup ourselves.

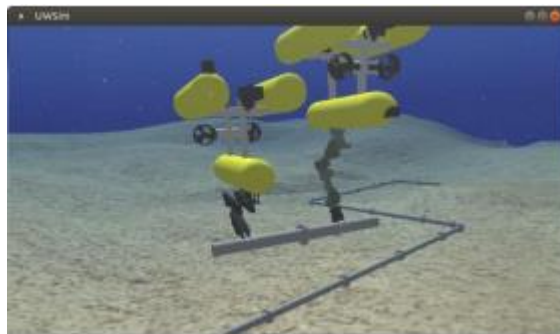


Figure 9. Twinbot in UWSim

The aforementioned, added to the fact that we had already used this tool before allowed us to complete a series of tests which we will talk about later.

2.2 HRI Design Specifications

2.2.1 Summary

The objective is a simulation as realistic as possible both visually and physically. That is why different factors have been taken into account, both in the robots and in the environment in which the simulation takes place.

The simulation should offer the user an easy interface to understand and learn, and to take the control of the vehicles.

As we have previously highlighted, we will use as vehicle the Girona 500 (*Figure 3*), which has three torpedoes connected and five engines that move the group of torpedoes, there are two engines for the frontward and the backward movement and for the turn, there are also two engines for the up and down movement and one for the lateral displacement.

The arm is a variation of the ARM 5 (*Figure 4*) which will have a different gripper but will have the same degrees of freedom in the joints which are two with a limited range of rotation for, almost, all the joints except the wrist which has a continuous range.

2.2.2 Control Design

The controls have been divided into two groups, those in charge of managing the vehicle and those in charge of managing the arm.

In the first instance, let's clarify that there are two independent modes among which is changed with the side button of the controller, this will also change the camera between the vehicle and the arm.

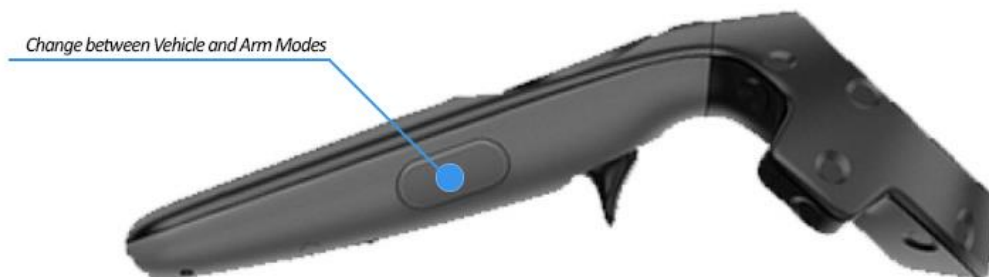


Figure 10. Mode switch.

Now, we will talk about the vehicle controls, the right controller is the one that manages the forward and backward movement and the rotation of the vehicle, the left one controls the up , down and lateral movement.



Figure 11. Vehicle controls.

In the end, we will talk about the arm mode, the right controller will manage the upper part of the arm(shoulder and slew) and the left one will control the lower part(Elbow and jaw) both of them has a trigger that will control the open and close of the gripper.

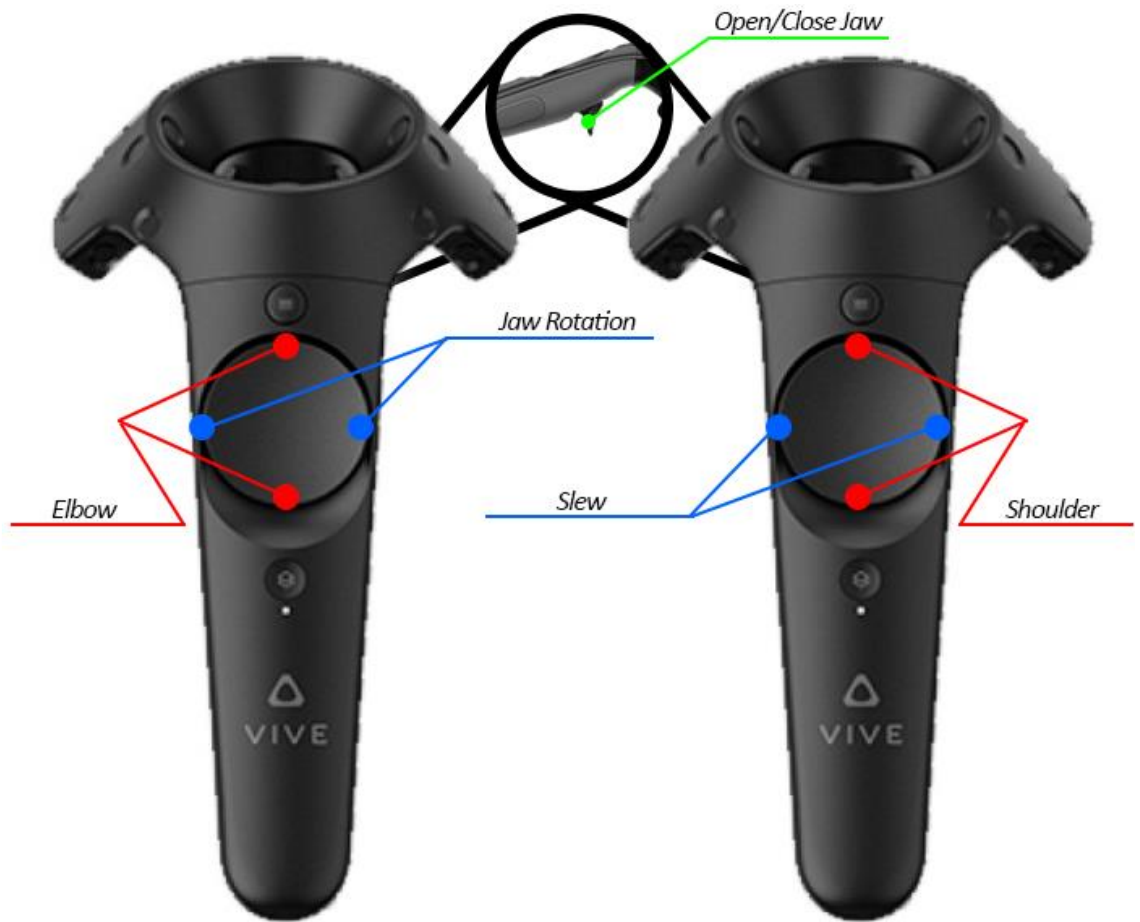


Figure 12. Arm controls.

2.2.3 Underwater environment and items effects

To fulfill the objective that the simulation is as close to reality as possible, we have created a series of effects that we will explain next:

Engine Particles

The different motors that the g500 has, when operating, generate a movement in the water represented as bubbles, which increases the amount of them based on the speed of the engines.

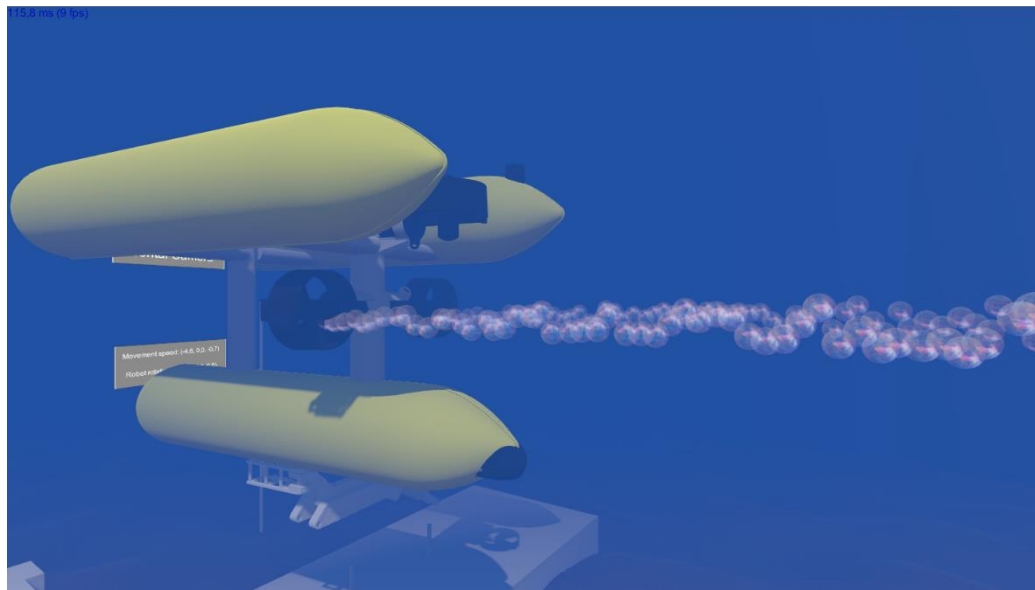


Figure 13. Engine Working.

Water Particles

They are represented by a simple particle system that shows the movement of particles that occurs naturally within underwater ecosystems, due to water streams.

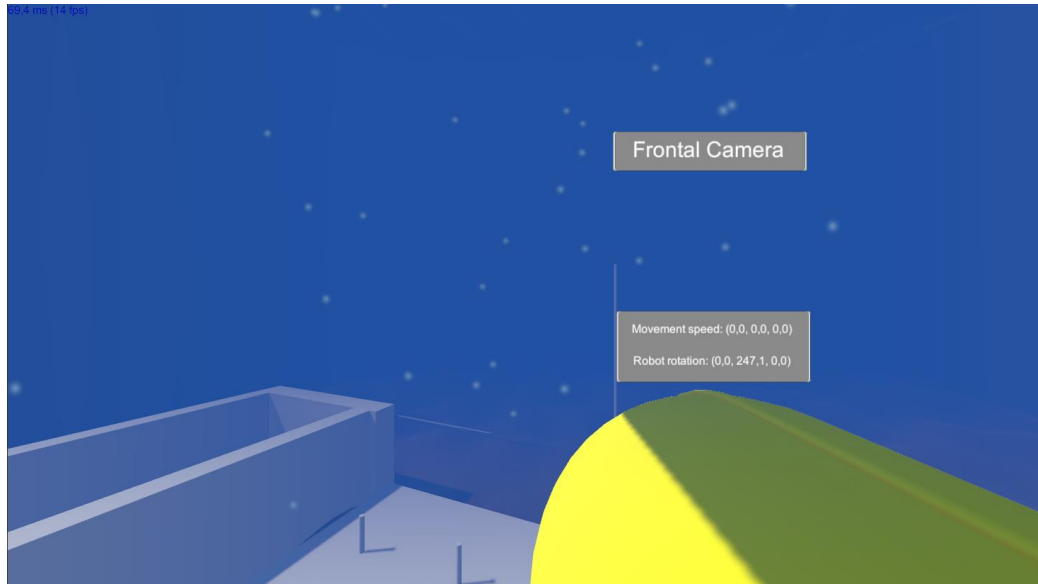


Figure 14. Water Particles.

Underwater Floor

We have used a series of materials and textures together with shaders that we apply to a plane in order to represent the seabed.

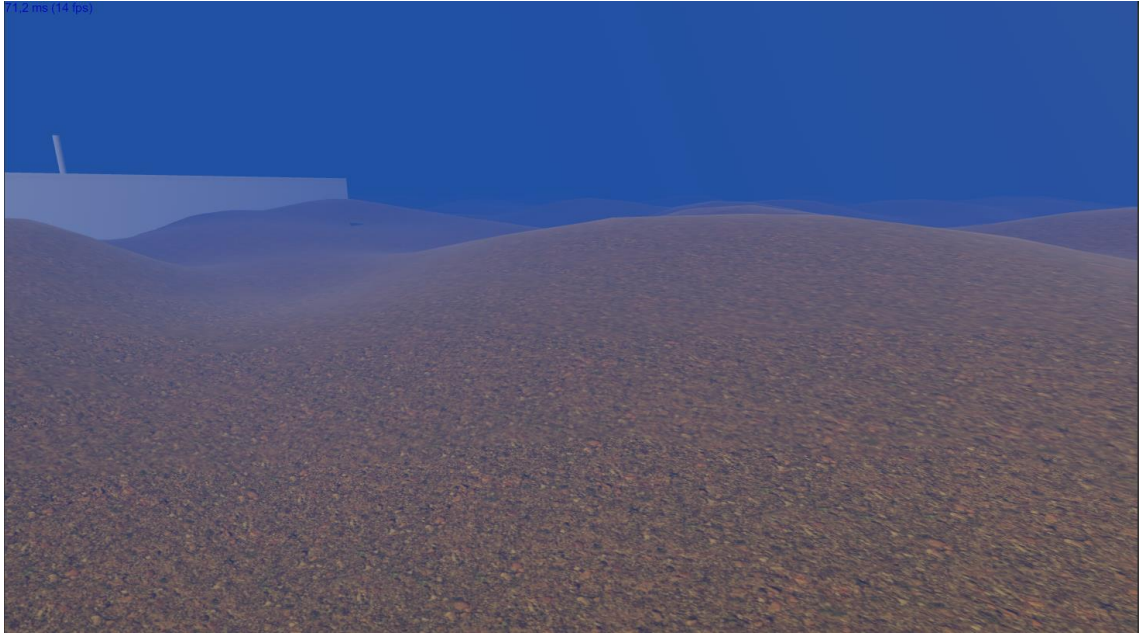


Figure 15. Seabed.

Water Surface

The same procedure applies to the surface of the water in which, through shaders, a movement has been added.

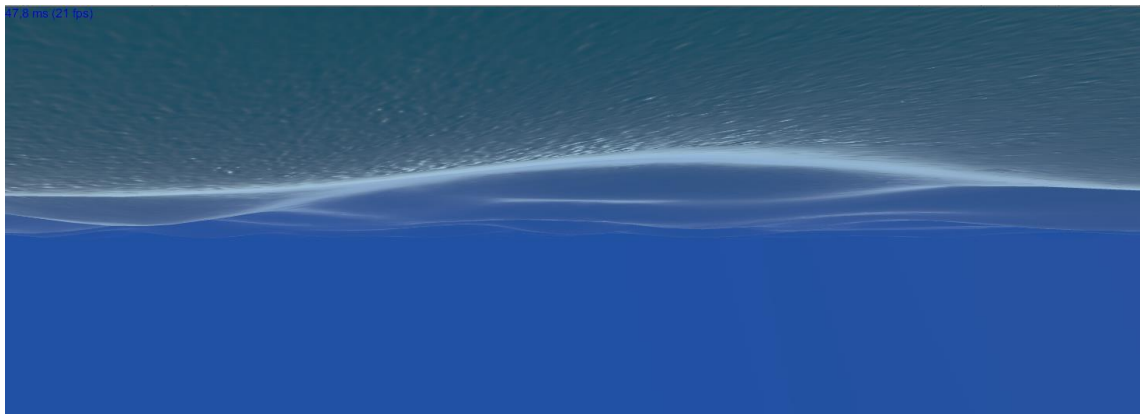


Figure 16. Water Surface.

Camera Effects

We have added, mainly, two effects to the camera based on the work flow of the AQUAS plug-in. The fog effect, which represents the turbidity of the water based on the distance and the blur effect, which represents the low visibility under water.

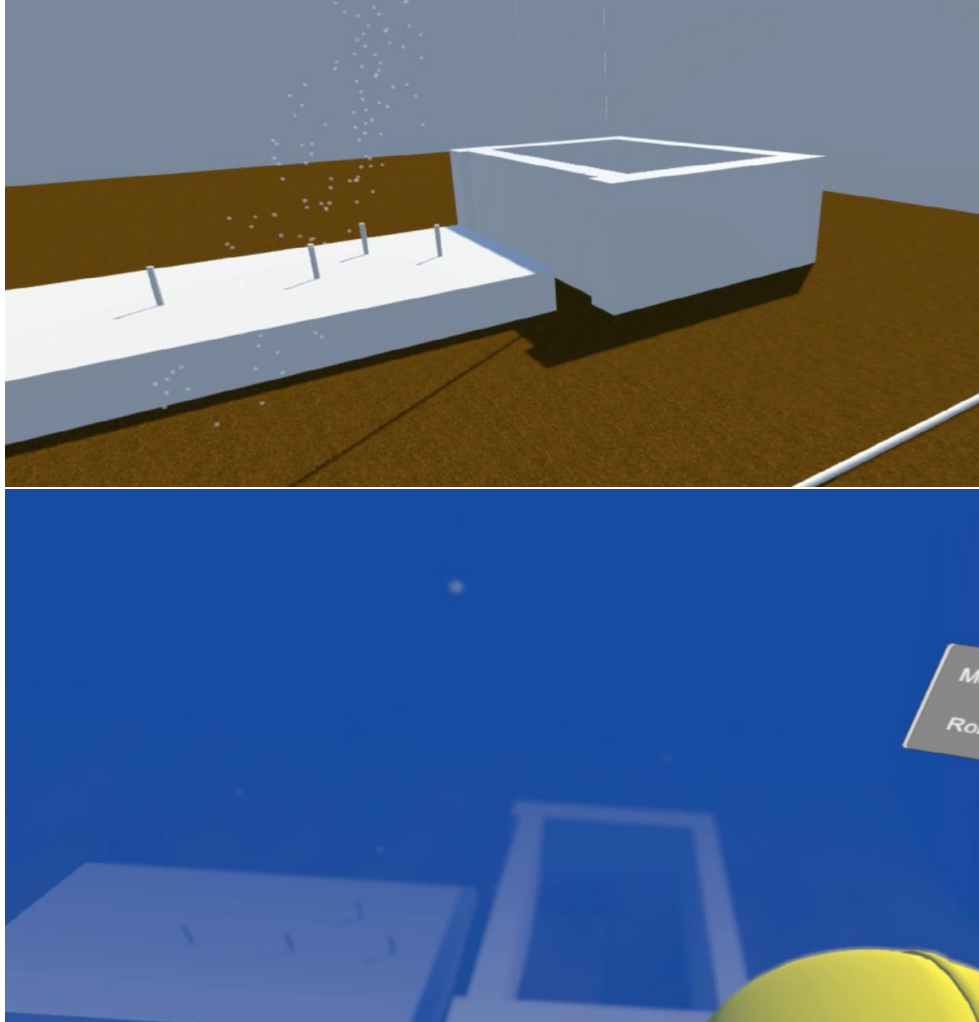


Figure 17. No camera effects(Up), Blur and Fog effect(Down).

Reachable Objects Effects

We have also decided to add a highlight when you can pick up an object by closing the gripper.



Figure 18. Black Box model (Left) Reachable Black Box(right).

3 MAIN DEVELOPMENT

3.1 Introduction

In this section we will present what has been the work routine that we have followed during the development of the project.

On the one hand, we integrated the arm in unity, together with objects that could be taken and its different movement.

After that, we prepared the vehicle movement and the integration of the G500 3D model in the simulation environment.

Lastly, we made the underwater environment of the simulation, and all the effects of the different elements.

3.2 Arm integration in Unity

As we have mentioned, the arm model is a variation of the ARM5 developed by in the UJI. The clearest variation is that of the clamp in which it has been replaced by a completely different one to simplify the user's grip.

The main script of the movement of the arm is a conditional statement where there are verified the orders received from the virtual reality equipment.

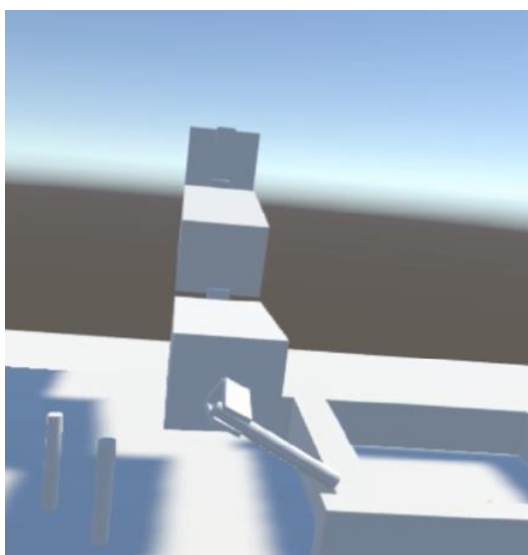


Figure 19. First version of the arm in the simulation.


```
//Slew Control

if (slewLeft)
{
    if (slewMinR < 60 ) {
        slew.transform.Rotate(0, 0, rotatS);
        slewMaxR -= rotatS;
        slewMinR += rotatS;
    }
}
else if (slewRight)
{
    if(slewMaxR < 60)
    {
        slew.transform.Rotate(0, 0, -rotatS);
        slewMinR -= rotatS;
        slewMaxR += rotatS;
    }
}
}
```

Figure 20. Slew Movement Script.

```
SteamVR_Actions.default_TouchTouchpad.stateUp
```

Figure 21. Order Boolean received by the VR controller.

Within the unity hierarchy the script is applied to the arm element that contains all the joint of the same. This allows the script to know which joint is the one that should move in every situation.

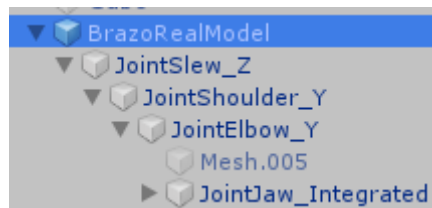


Figure 22. Unity hierarchy of the Arm.

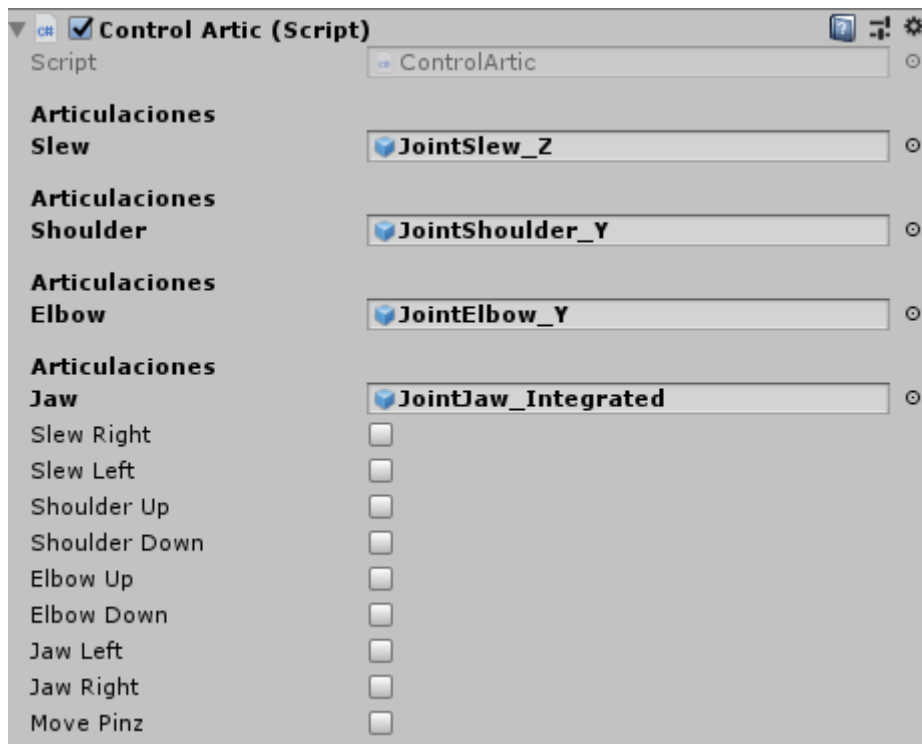


Figure 23. Script variables in Unity.

To detect if you can grab an object we have added to the gripper a collider that detects when there is an object between the clamps, apart from this, the collider itself throws an order to the object's script that activates the highlighting through a system of particles.

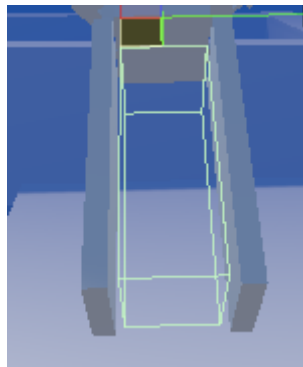


Figure 24. Jaw contact in Unity.

```

if (Inter(other))
{
    objetoAgarrable = other.gameObject;
    triggerObj = objetoAgarrable.GetComponent<ParticleSystem>().emission;
    triggerObj.enabled = true;
}

```

Figure 25. Order sent to the particle system of the object.

3.3 Vehicle integration and movement

The vehicle is a g500 as we have previously reported, in this section, contrary to the previous one, it has been decided to implement the model first and then the programming.

The way to receive orders is very similar to the arm because the controls are a variation of the same. The difference lies in the behavior of the robot, the engines are gaining strength as one direction is held down to a maximum value, in order to simulate the forces of the same.

```
if (rb.velocity.magnitude < 10)
{
    AddForceForward(speedMove);
    ForwardBackwardEngineEnable(true);
}
```

Figure 26. Addition of force up to the speed limit of the engine.

The collision system is quite meaningful because of how the simulation reacts to them, when a collision between the vehicle and the seabed or the water surface occurs the simulation stop and restarts the original state.

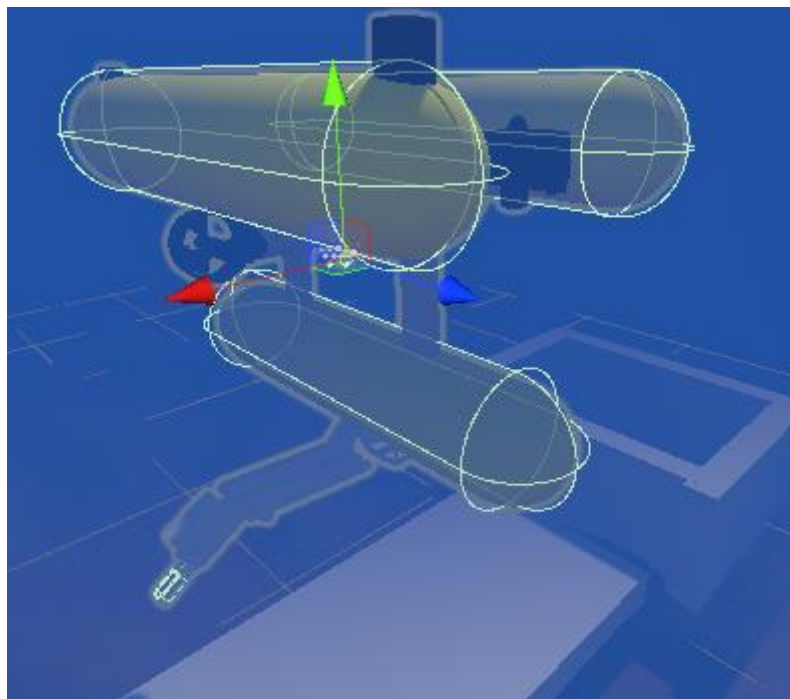


Figure 27. Original state of the simulation.

3.4 Underwater Environment.

There are many things to talk about in this section, we will start with the representation of particles.

Starting with the particles of the engines, these begin to launch when the engines start working and gain strength in the speed of the G500.

```
private void RateControl(float speedMagnitude,ParticleSystem.EmissionModule bubblePart)
{
    bubblePart.rateOverTime = speedMagnitude/2;
}
```

Figure 28. Engine particle systems force script.

To represent the particles of the underwater system, both the position of the particles and the rotation of the entire system are transformed with time.

Taking about the seabed and the water surface they are essentially the same, the only difference is that the water surface shader change with the time to simulate the surge.

```
float3 v0 = v.vertex.xyz;
float3 bitangent = cross(v.normal, v.tangent.xyz);
float3 v1 = v0 + (v.tangent.xyz * 0.01);
float3 v2 = v0 + (bitangent * 0.01);

float ns0 = _NoiseScale * snoise(float3(v0.x + _NoiseOffset.x, v0.y + _NoiseOffset.y, v0.z + _NoiseOffset.z));
v0.xyz += ((ns0 + 1) / 2) * v.normal;

float ns1 = _NoiseScale * snoise(float3(v1.x + _NoiseOffset.x, v1.y + _NoiseOffset.y, v1.z + _NoiseOffset.z));
v1.xyz += ((ns1 + 1) / 2) * v.normal;

float ns2 = _NoiseScale * snoise(float3(v2.x + _NoiseOffset.x, v2.y + _NoiseOffset.y, v2.z + _NoiseOffset.z));
v2.xyz += ((ns2 + 1) / 2) * v.normal;

float3 vn = cross(v2 - v0, v1 - v0);

v.normal = normalize(-vn);
v.vertex.xyz = v0;
```

Figure 29. Script of the shader in charge of the seabed and water surface noise .

We have been working with shaders applied to materials in the previous stage, but, for the fog we will make a global one that affects all the scene and simulates the underwater vision, for this we will use, also, a blur material that we will apply in front of the camera and will gain strength with the distance.

3.5 Canvas and interface information.

Here, we will talk about the information which is displayed in the screen every moment, we have 3 sections, the FPS and input lag section, the camera name section and the movement one.

In the FPS and Lag section will show the information of images per second and delay between time of sending and processing of the signal this information is not showed to the user but is shown on the computer monitor.

The second one, the camera name section, shows the name of the actual active camera and mode.

The third one, shows the information about the vehicle movement speed and rotation.

```
private void UpdateTexts()
{
    for(int i=0; i < velocities.Length; i++)
    {
        if (i % 2 == 0)
        {
            velocities[i].text = "Movement speed: " + rb.velocity;
        }
        else if (i % 2 == 1) {
            velocities[i].text = "Robot rotation: " + rb.rotation.eulerAngles;
        }
    }
}
```

Figure 30. Text display controller(Movement Section).

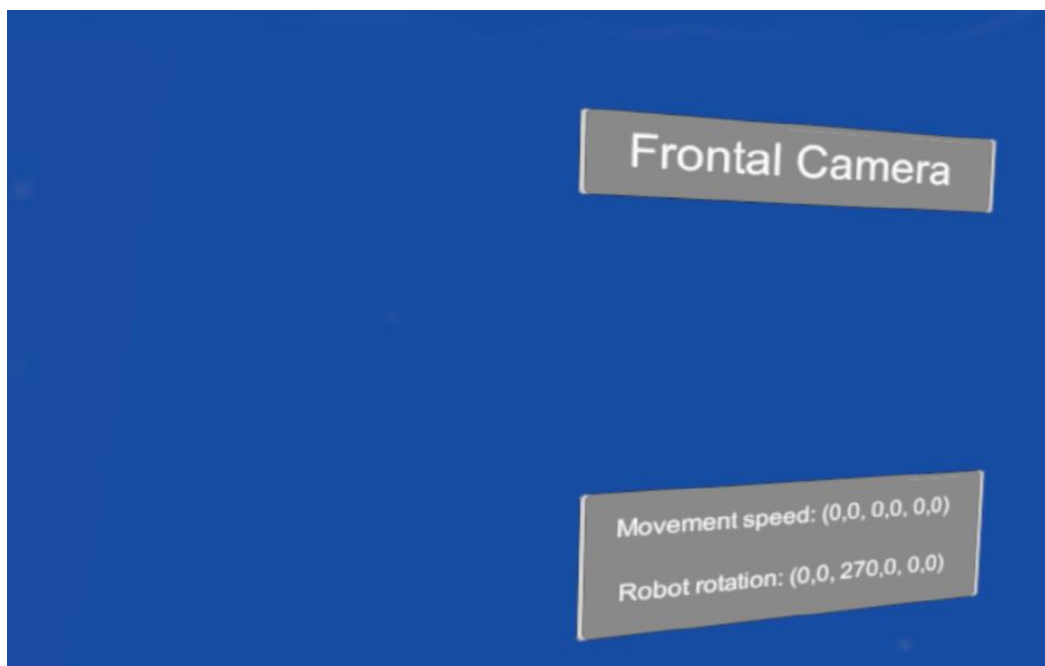


Figure 31. User visible canvas.

4 RESULTS

The results obtained are much higher than expected, besides completing the pre-established objectives, we have been able to perform different tests that have allowed the improvement of the project, from performance tests in different computers and situations, which we will discuss below, to a usability test that made us able to receive feedback from different people.

4.1 Performance Analysis

The tests on the equipment consisted of measuring the latency according to different situations of stress for them.

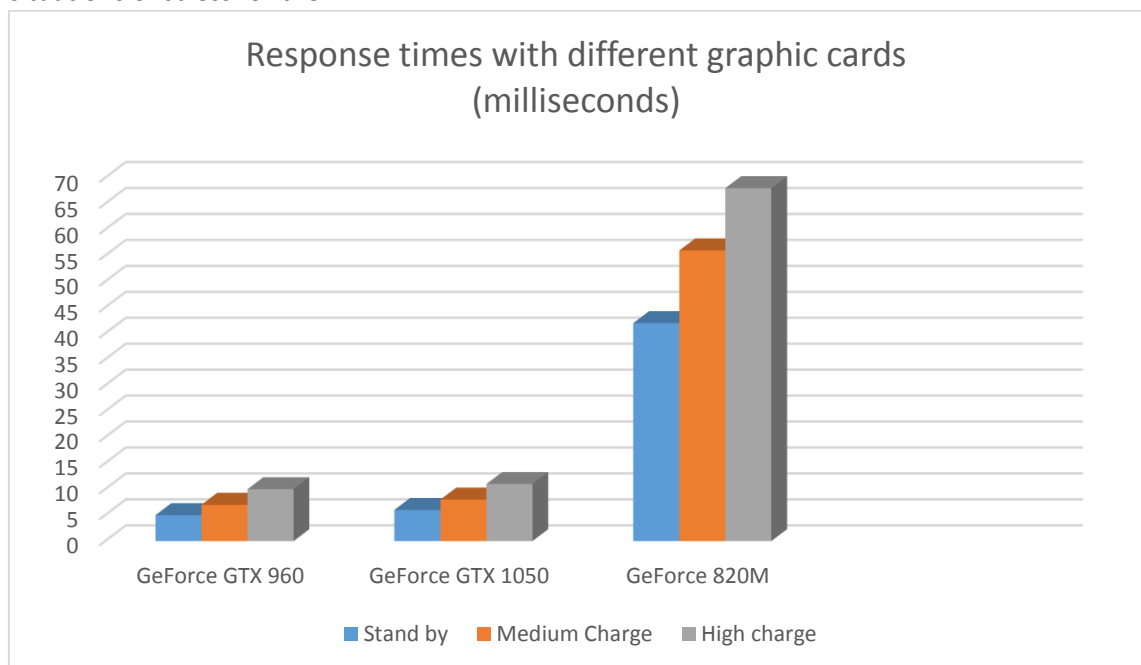


Figure 32. Charge test with different graphic cards.

Surprisingly it looks like the table can be seen that a previous card(GTX 960) has achieved better results than a more modern one(GTX 1050), the previous one with a delay of 8ms while in high charge, when the modern one has 10ms of delay, this is due two factors, first the GTX 960 has better performance despite of being older, and the processor of the computers . The last one, as we can see has the worst results in the test but can handle the VR even though theoretically it should not be able to do it, but it has more than 65ms of delay in high charge.

A paper including this tests, among other results, will be presented in the Spanish Robotics Workshop(Alicante, June 2019). See Annex 1.

4.2 Usability Testing

We have made some usability test to see if the simulation accomplish our previous requirements of being easy to learn, realistic and not fatiguing.

First of all, we will talk about our test group, they have different skills and expertise in simulators and videogames, as we can see in the table, there is a component called “affinity” this is a value established by us taking into account the background of the user, for example, if the user plays videogames every day, he/she will have a higher affinity than someone that don’t play videogames or uses simulators.

Age	Gender	Test 1	Tries	Test 2	Tries	Test 3	Tries	Test 4	Tries	Affinity	Pleasantness
23	Male	352	1	199	1	1182	7	183	1	0,5	7
43	Male	265	1	385	1	395	2	296	2	0,2	8
27	Male	430	1	650	3	201	1	318	1	0,7	9
22	Male	543	3	138	1	156	1	236	1	1	9
28	Male	463	1	286	1	274	1	421	1	0,4	8,5
45	Male	647	3	360	1	558	2	352	1	0,1	7,5
										0,483333333	8,166666667

Figure 33. Information of the pilot group in the usability test.

On the other hand, in the usability tests, a series of data was collected to discover if the interface had been made easy to use, and to receive the feedback from real users and not just the developer. The test consisted of four scenarios with different difficulties to see the time and the number of attempts that it cost to the user to complete each difficulty.

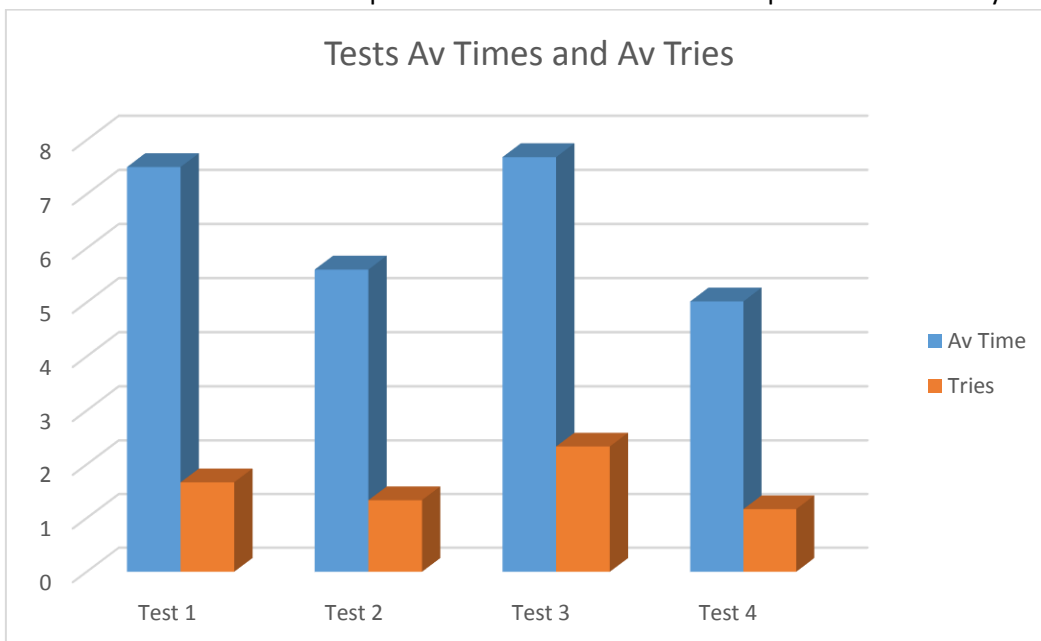


Figure 34. Average times and tries for every test.

In the first difficulty the cameras are in third person and the user needs to take the black box and bring it to the white container. The user only knew about the camera change button and that the elements moved with the touchpad. The average time of this trial is

450 seconds (7 minutes and 30 seconds) and 1.6 tries per person the best time in the last try is 120 seconds (2 minutes) which is only 20 seconds behind the developers time.

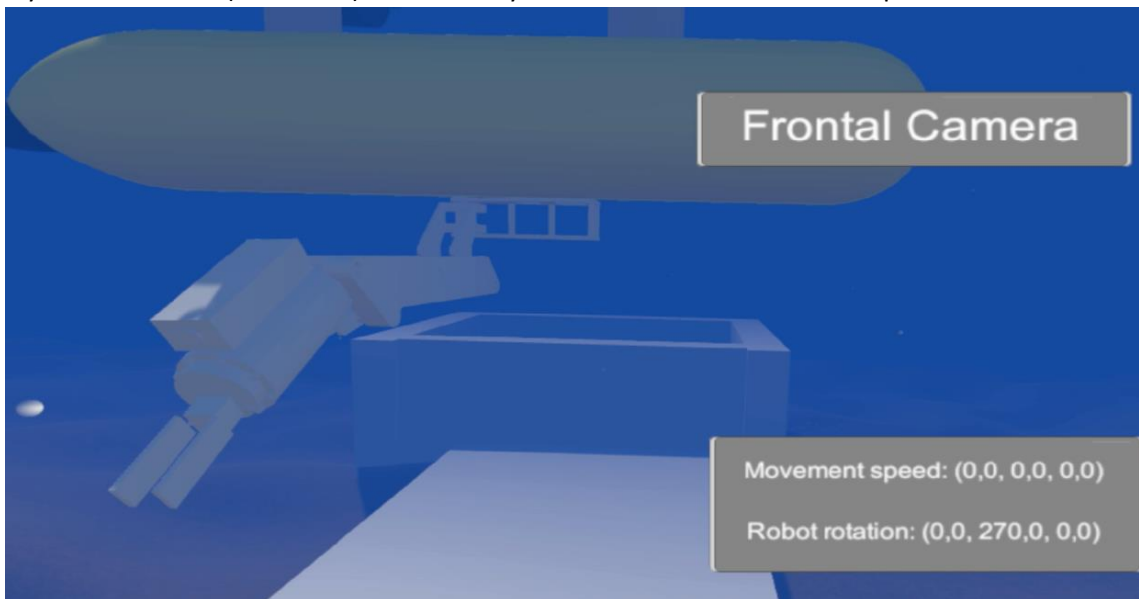


Figure 35. First trial of the usability test.

In the second proof the arm camera is set on its place, the times are a better, around 4 minutes and 40 seconds, and the opinions are also better than the first one, the user thinks that is easier to take the object when you have the arm camera in its place.



Figure 36. Second trial of the usability test.

In the third one the vehicle camera is also set on its place, the times keep improving, 4 minutes, and the opinions are the same of the second one, but related to the vehicle camera.



Figure 37. Third trial of the usability test.

In the last one a random obstacle is set between the G500 and the black box and the user need to avoid it, as expected the times increase due to the obstacle, the average one is around 6 minutes.



Figure 38. Fourth trial of the usability test.

At the end of the test we asked some questions to the users about their feelings in relation with the simulation, they all gave different opinions, but everyone agreed that a tutorial would simplify the learning.

One of the main questions leaving aside the tutorial was a satisfaction survey from which the following table arises.

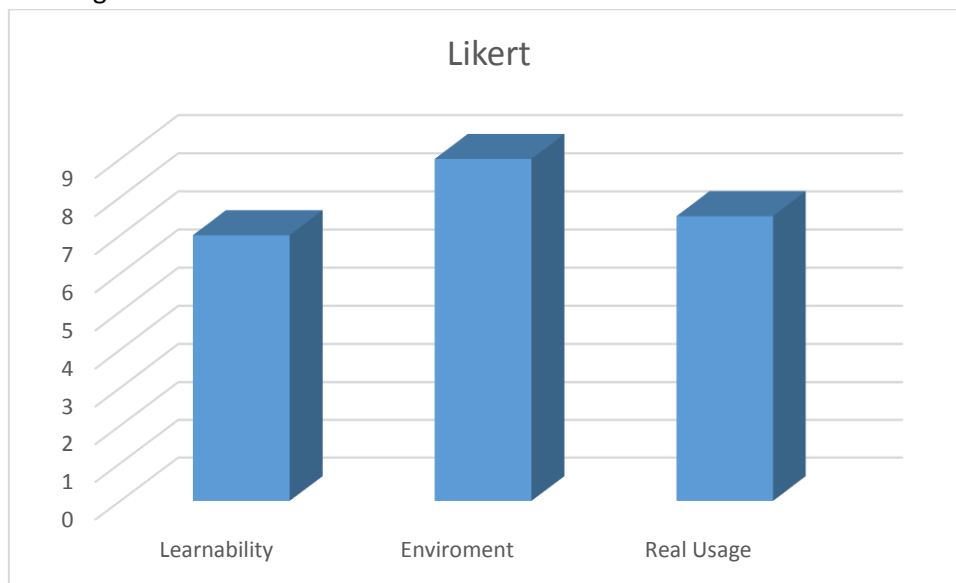


Figure 39. Likert result of the usability tests.

The table shows how easy the users think that they were able to learn all the controls in the simulation and make a good use of them, how realistic the environment was for them and if the simulation can be useful in a real intervention in their opinions.

For the learning time a score of seven out of ten was obtained which, taking into account that the basis of the test was, that the users had no information and that they should discover the controls beyond they use the simulation, this supposes a higher note than expected in the first instance.

As for the environment the note was a nine out of ten, the most highlighted comment is that we could try to simulate water streams with enough force to hinder the handling of the robot, which would be interesting in a future extension of the project, but in general the received note is excellent for our expectations.

Finally, the user was asked about the real usefulness that they believed the interface would have in a real intervention and they gave a score of 7.5 out of ten was obtained, giving comments like, "if used as a training tool, it might be a good idea to try to adjust the 1Hz refresh rate to simulate a wireless intervention, and that could be added to the information shown in the glasses some aspects such as depth", this would also be interesting in the future.

5 CONCLUSIONS

This document has presented a simulation of underwater robots in an immersive VR environment, oriented to the training of robot guides. The simulation has gone through all the phases and they had been documented.

VR equipment setup and implementation has helped to understand how the information is treated and sent to the different equipment involved in the process and how they represent it in the virtual ambient.

With this project we have been able to make an approach between two branches of technology, underwater robotics and VR, and we have seen that they go very well together and in the future the implementation of VR can make interventions much easier.

Also, some users with whom we have tested claim to have dizziness using VR before, which has not happened to them now. If we should mention a problem is that users who have vision problems need to use contact lenses to see the canvas information.

5.1 Future Work

It would be interesting to increase the dimension of the project by adding artificial intelligence to reduce user intervention.

Also, it would be a good idea to represent the problems of underwater wireless communication such as low image refresh and quality of the real systems, and adapt the interface to be usable in real interventions.

However, the most inviting thing is to make a translation between the controller instructions into Ros[7] instructions to make the interface available to be used in UJI experiments and apply it to the Twinbot project.

Thanks to the usability test, some ideas have been taken into consideration that could be very interesting to apply in the future, such as the inclusion of some extra data in the information that is shown to the user in the glasses, information like the actual vehicle depth, the pressure and the forces applied to the Rov. Another interesting proposal was to change the refresh rate of the simulation to 1Hz in order to represent a real wireless connection with the robot through which you are not able to send full hd pictures with a 60Hz rate like the actual wireless interventions has to train the pilot in that situations.

6 REFERENCES

- [1] IRS Lab UJI. (2018). *Twin Robot Project*. Retrieved from <http://www.irs.uji.es/project/twinbot>
- [2] IRS Lab UJI. (2015). *Merbots Project*. Retrieved from <http://www.irs.uji.es/project/merbots>
- [3] Universidad de Girona. (n.d.). *Girona 500 AUV*. Retrieved from <https://cirs.udg.edu/auvs-technology/auvs/girona-500-auv/>
- [4] García, J. C., Patrão, B., Almeida, L., Pérez, J., Menezes, P., Dias, J., & Sanz, P. J. (2015). A Natural Interface for Remote Operation of Underwater Robots. *IEEE Computer Graphics And Applications*, 34-43.
- [5] HTC Corporation. (n.d.). *HTC Virtual Reality*. Retrieved from <https://www.htc.com/es/virtual-reality/>
- [6] Sanz, P. J., de la Cruz, M., Lunghi, G., Veiga, C., Marín, R. & Di Castro, M (2019). The Role of HRI within COMOMUIS Research Project.(Annex 1)
- [7] Dogmatic Games. (n.d.). *Aquas Unity Plugin*. Retrieved from <https://assetstore.unity.com/packages/tools/particles-effects/aquas-water-river-set-52103>
- [8] Fernández, J. J., Prats, M., Sanz, P. J., García, J. C., Marín, R., Robinson, M., . . . Ridao, P. (2013). *Grasping for the Seabed*.
- [9] Lunghi, G., Marín Prades, R., Di Castro, M., Masi, A., & Sanz, P. J. (2018). *Multimodal Human-Robot Interface for Supervision and Programming of Cooperative*. Castellón de la Plana.
- [10] Peshkova, E., Hitz, M., & Kaufmann, B. (2017). *Survey on Natural Interaction Techniques for an Unmanned Aerial Vehicle System*.
- [11] Bruno, F., Barbieri, L., Lagudi, A., Cozza, M., Cozza, A., Peluso, R., & Muzzupappa, M. (2017). Virtual dives into the underwater archaeological treasures.
- [12] Marsh, T. (1999). *Evaluation of Virtual Reality Systems For Usability*
- [13] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T.(1994). *Human-Computer Interaction*. Addison-Wesley Longman Ltd., Essex, UK, UK. ISBN:0201627698

7 ANNEXED DOCUMENTS

The Role of HRI within COMOMUIS Research Project [◇]

Pedro J Sanz ^{a,1,2*}, Marcos de la Cruz^a, Giacomo Lunghi^b, Carlos Veiga^b, Raúl Marín^{a,b}, Mario Di Castro^b

^a IRS-Lab, Computer Science and Engineering Department, Jaume I University, Avd. Sos Baynat s/n (12071 Castellón de la Plana, Spain).

^b CERN, EN-SMM Survey, Measurement and Mechatronics group, Geneva (Switzerland).

Abstract

The three-year National Coordinated TWINBOT Project (TWIN ROBOTS FOR COOPERATIVE UNDERWATER INTERVENTION MISSIONS, 2018-2020), has been broken down into three sub-projects, each one of them under the responsibility of a partner in correspondence to their main area of expertise. In particular, the UJI subproject, COMOMUIS (Cooperative Mobile Manipulation for Underwater Intervention Systems), is responsible of the mechatronics and planning aspects dealing with the required manipulation skills, wireless underwater communications, and Human-Robot Interaction (HRI) issues. In particular, the work shown in this paper discuss new progress, under UJI's development, mainly focused on the HRI context, including immersive Virtual Reality (VR) capabilities for assisting telemanipulation tasks. *Copyright* © CEA.

Keywords:

HRI, Virtual Reality, Marine Robotics, Robot Cooperation, Underwater Intervention.

Project Info:

Title of the Project: Cooperative Mobile Manipulation for Underwater Intervention Systems - COMOMUIS

Reference: DPI2017-86372-C3-1-R

Coordinators: Pedro J Sanz, Raúl Marín

Type of project: National

Funding Institution: Ministerio de Ciencia, Innovación y Universidades

Starting date/ending: 1/01/2018 to 31/12/2020

Introduction

The TWINBOT coordinated research project aims at the use of two I-AUVs, cooperating for enabling complex underwater intervention tasks with high autonomy never demonstrated before. The project objectives include the design of multimodal user interfaces, advanced control strategies, online 3D reconstruction, target detection and tracking, underwater wireless communication and cooperative mobile manipulation of 2 I-AUVs. The complexity of the proposed system requires a multidisciplinary team including mechanical, electronics and computer engineers as well as experts on computer science, mathematics and telecommunications. To tackle this complexity, the project has been split in three sub-projects according to the expertise of each partner of the consortium and to the facilities they have

available. These partners are UdG, UIB and UJI as Coordinator. This is a very challenging project, with a lot of different underlying building blocks (hardware and software) to implement and integrate, effectively, in a great variety of technology domains. Thus, this paper will be specifically focused within HRI problem, under UJI's responsibility (i.e. Subproject COMOMUIS) letting other contributions out of scope of this work.

1.1. Background

RAUVI (DPI2008-06548) was the first coordinated project in which UJI, UdG and UIB started their cooperation. The project was focused on the design and implementation of a Reconfigurable AUV for Intervention Missions. The major outcome of RAUVI was the GIRONA 500 I-AUV (Ribas et

[◇] HRI = Human Robot Interaction.

¹ Head of IRS-Lab (UJI, Castellón).

* Corresponding author.

Emails: sanzp@uji.es (1st Author),
al343449@uji.es (2nd Author), giacomo.lunghi@cern.ch (3rd Author), carlos.veiga.almagro@cern.ch (4th Author) rmarin@uji.es (5th Author),
mario.di.castro@cern.ch (6th Author)
<http://www.irs.uji.es/sanzp/> (1st Author)

al., 2012), equipped with an electrically driven arm with 4 DOFs and a stereo camera (Fernandez et al., 2013) which was used to demonstrate autonomous object recovery, firstly in a water tank environment (Prats et al., 2012a) and later in a harbor (Prats et al., 2012b). This concept was later on extended in the TRIDENT FP7 EU project (Sanz et al., 2013), proposing a new methodology for multi-purpose underwater intervention missions. A team of two cooperative heterogeneous robots with complementary skills, an ASC (Autonomous Surface Craft) and an I-AUV endowed with a dexterous 7 DOF manipulator and a 3-fingered hand was used to perform underwater recovery tasks. Our next step was the Spanish project TRITON (DPI2011-27977), where different capabilities were proved, such as autonomous docking onto a custom subsea station, fixed-based manipulation for valve turning and hot stab connection, and free floating manipulation for valve turning (Palomeras et al., 2016). As a continuation of the TRIDENT concept, TRITON used a second AUV as a surface vehicle to support the I-AUV through range-only, active, single, mobile-beacon navigation. Finally in our last consortium project, called MERBOTS (DPI2014-57746) (Sanz et al., 17), we explored collaborative strategies between an intervention HROV and an AUV operating as a flying camera (Garcia-Fidalgo et al., 2017). Both systems were interconnected to an operator using RF, VLC (Visible Light Communication) and acoustic communications through an ASC working as a gateway.

1.2. The UJI (IRS Lab) expertise

IRS Lab has been demonstrating a well-known experience working in robotic manipulation, from a very spread point of view, since nineties. This research includes multi-sensory based control, wireless communications, multipurpose autonomous manipulation, and multimodal interfaces for telemanipulation techniques, among others. Some previous national (DPI2004-01920; DPI2001-3801) and international projects (GUARDIANS-IST-045269, GRASP-ICT-215821) guarantee the potential research in these areas of expertise. Concerning the context of TWINBOT project about underwater manipulation and HRI, it is noticeable that the UJI team has coordinated four previous projects since 2009, three national ones (RAUVI, TRITON and MERBOTS) and another one supported by the EC (TRIDENT-ICT-248497). Moreover, during the previous project MERBOTS, a new research line was started dealing with the underwater wireless communication problem. It is worth mentioning that the other two groups, UdG and UIB, have also been partners in all the projects concerning underwater robotics, since 2009, above mentioned, thus guaranteeing an excellent rapport between the three teams. On the other hand, the UJI team has also current projects cooperating with industrial companies. Another important cooperation agreement is running with the CERN Robotics group for bi-manual telerobotics interventions in hazardous environments. Other cooperation activities include educational robotics with some institutions. In summary, different issues related to the physical interaction capabilities, including cooperative mobile manipulation aspects, and dealing with intervention tasks, HRI and wireless communication aspects, addressed in TWINBOT project, are under responsibility of UJI group. Further technical details can be found elsewhere <http://www.irs.uji.es/>.

1.3. Proof of concept and goals

Let us briefly consider a representative mission scenario, corresponding to an autonomous object transportation task. A team of two single-arm I-AUVs (Fig. 1) is deployed in an area to cooperatively search for a set of objects located at unknown positions within a bounded region.

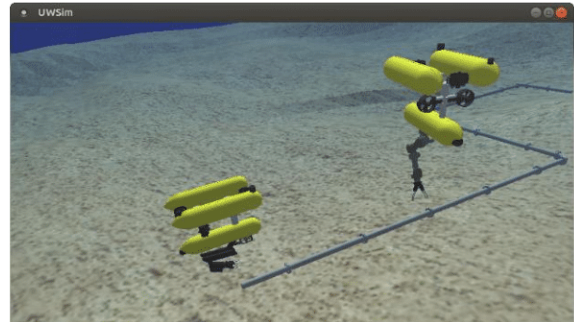


Figure 1: Cooperative Survey.

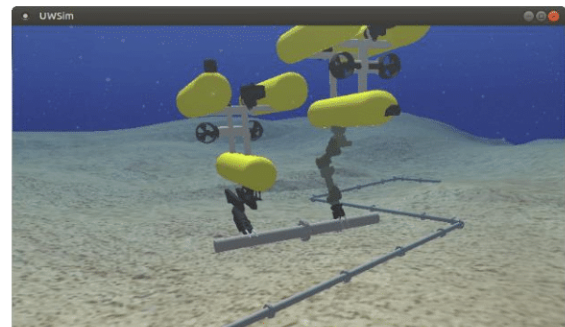


Figure 2: Cooperative Transportation.

The objects are assumed to be known a priori, since they are the parts to be assembled to build a complex object and hence their 3D models are available a priori. The vehicles divide the search area to map it exhaustively. Task-driven communication methods will be used to keep limited the amount of information interchanged, through a multi-modal (acoustic/RF/VLC/WiFi) communication architecture, and focused on the undertaken task. Whenever an object appears within the sensor field of view, object detection algorithms will be used to detect it. Next, object identification methods will try to recognize it and determine its pose in order to launch active perception strategies to guide the I-AUV for mapping the object in a smart way. Cooperative SLAM techniques will be used for a consistent real-time navigation of both I-AUVs, making each vehicle aware of the pose of its twin. To this aim, each robot will perform a visual metric SLAM for accurate navigation while using the recognized objects as semantic features whose observations will be shared with its twin I-AUV through the multimodal communications. This will allow to keep a consistent map although distributed between the vehicles. After the survey, the vehicles will be surface to achieve sufficient communication bandwidth to share their map copies and use map fusion techniques to join the distributed maps into a single consistent one, copied in both robots as well as in the HRI where the assembly mission will be planned by the user. Next, the robot will dive to

perform the plan to bring all the objects together to assemble them to build a complex object. Component objects will be grasped and transported using cooperative I-AUVs configuration (Fig. 2). A final goal, dealing with the cooperative assembly, will be tackled only if we succeed with the aforementioned stage but currently it is out of the scope of this paper.

After this review to the general envisioned concept of TWINBOT it is needed to clarify now the specific goals under UJI responsibility, through COMOMUIS subproject, which are the following:

- Design & Implement a 6 DOF electric arm and a multisensory grasping device
- Cooperative mobile manipulation
- Human Robot Interaction (HRI)
- Wireless communications (RF)

As stated before, due to practical considerations, the next part of the paper is devoted to the HRI goal. In summary, the UJI team will be responsible of all the aspects related to the mission specification by the user and simulation, integrated both through the HRI module.

2. HRI in progress

From a general perspective, in order to implement the TWINBOT's Work Plan and so, of the UJI subproject, COMOMUIS, there exists a specific task in charge of developing the HRI needed to assist the user during all the mission phases. Note that, despite the whole system will work autonomously, the user will monitor the intervention launching and stopping the tasks, when needed and also will be crucial the capability of supervised control letting available, when needed, the teleoperation mode. The interface developed by UJI in previous projects will serve as starting point and extended to deal with the new proposal challenges (García et al, 2017). The survey specification will start by loading a raw geo-referenced map of the work area (available a priori) and indicating a set of waypoints. Each robot will follow its assigned waypoints, surveying the area and refining the map geometry with the online 3D mapping module. These precise maps, will be later on fused into a single consistent one which will be sent to the user immersive VR device whenever the communication bandwidth makes it possible (this is bringing the vehicles to the surface after the survey to achieve Wi-Fi). Using the I-AUV position broadcasted through the acoustics, the HRI will provide the user with a 3D representation of the twin I-AUVs during the mission stages, and so the user can specify and monitor both the survey and the intervention to be done.

In particular, the main objective of this part of the project is the integration of the different guidance controls that exist for the intervention robots into a single VR interface, including immersion capabilities (Gandhi et al., 2018). This new interface should be enough intuitive and friendly to simplify, as much as possible, the pilot's work, reducing expended time and inherent complexity of this kind of systems for operating and running the intervention mission in a suitable manner.

It is also intended to observe through the simulation tests which of the different virtual reality equipment, available on the market, are the most optimal in terms of accuracy and

response times due to the need of both factors in the different underwater intervention missions.

The system also includes a simulation of the TWINBOT project scenario (see figs. 1 and 2) on the server side, in preparation for the next integration tests for sea trials. The server is designed to provide the user interface with images close to reality, so that vision algorithms can be tested, as well as the effect of the contact of the robots' claws on the object to be manipulated. The server provides access to sensors and controls of robots in a multimodal manner, through different communication protocols (i.e. TCP, JSON, HTTP), easily adaptable to the control system of real robots.

3. Experimental setup and VR developments

3.1. "HTC Vive" vs "Oculus Rift"

The current experimental setup can be observed at fig. 3, where the "HTC Vive" system is running for guarantee a realistic 3D immersion in a friendly manner. Its main specifications are: 110° field of view, 90 frames per second, 2160x1200 resolution, 32 sensors in the glasses for spatial localization and 24 in each controller. It is recommended to use a graphics card higher than the 1060 range of NVidia. In our case, it has been used an NVidia 960GTX, with 8GB of DDR3 RAM, and an Intel Core I7-4790 3.60 GHz.



Figure 3: 3D immersion with HTC Vive system.

A question arise here: Why we are using the "HTC Vive" system in place of previous one (García et al, 2017) that included "Oculus Rift" and Leap Motion (LM, see fig.4)? The main reason to discard a LM device is the precision of the movements, which would be less than that of a joystick for example, the second is that these devices are designed for the detection of hands and gestures which could make learning a little difficult of the control due to the amount of possible actions in the project which would have to be differentiated in a clear way to avoid fatal errors when controlling the robots. For a similar reason to the latter we have left aside the joystick, one could only be insufficient while two could require a lot of learning time for the user. Regarding the glasses, we decided to use the HTC because it includes its own controls (instead of joysticks for instance) that are perfect because of the different buttons and pads they have, in addition to the fact that the relationship between the controls and the glasses is perfect for having been built, literally, for each other, thus simplifying the communication between both elements.



Figure 4: 3D immersion with "Oculus Rift" and LM.

3.2. Unity (game engine) vs UWSim

Till present, IRS-Lab developments dealing with HRI used UWSim (Prats et al., 2012) Hardware in The Loop (HIL) simulator developed during the TRIDENT project, adapting hardware and software to the intended mission scenario. But now the new challenge is to improve the immersive capacity (VR) 3D of the system. So new, better-positioned possibilities open up than UWSim, like the Unity gaming engine, by Unity Technologies, (Unity User Manual, 2018). In fact, the main reason to use Unity in place of UWSim, is that the latter does not have integrated VR while Unity if it has integrated (as a plug-in) several drivers for different VR equipment such as "Oculus Rift" or "HTC Vive" while to perform the implementation, from scratch, in UWSim would take a long time which would delay the usability tests and so on.

It is noticeable that UWSIM is demonstrating to be a great tool for education, simplifying the interaction with the real G500 ROS-based robot control, as well as incorporating the possibility to configure the currents and water turbidity in the scene itself. Also, it includes a benchmarking module, which enables to register and compare the results from different robot control algorithms, in a web-based manner. Other critical point, under consideration, is the multibody interaction problem. In fact, the UWSim physics for modeling the simulation of multibody interaction, such as in the case of TWINBOT project, is not perfectly solved yet. Some solutions have been designed in UWSim for this (e.g. ObjectPicker), which are very interesting for fast prototyping, while the Unity physics model is giving a more realistic response to this problem. On the other hand, UWSim presents a much better integration with ROS than Unity, which is very interesting for current research and potential applications, like education and so on.

In summary, Unity and UWSim represents two different tools for two distinct purposes. Moreover, Unity platform enables, as will be seen in the following sections, the design of simulation servers that provide more realistic sensor inputs, especially in the case of water scenarios. So, the suggestion would be just to select the best tool for each specific necessity.

4. Preliminary achievements

4.1. Suitability of HTC glasses for immersive VR 3D

Currently, it has been got to perform a functional simulation from which it can be starting to extract performance and accuracy data from the virtual reality (VR) interface developed.

In fact, some tests have begun to identify the performance of the graphics cards used. Keep in mind that the response time of these cards will be key to properly process the orders sent through the controls of the glasses used (HTC Vive). It is noticeable that the possibility of applying VR environments to the control of underwater robots is dependent of a suitable graphics performance, among other factors. Thus, some specific tests were implemented showing how the performance looks (i.e. response time) of the graphic cards, regarding the orders processed through the controls of the HTC glasses (see fig.5).

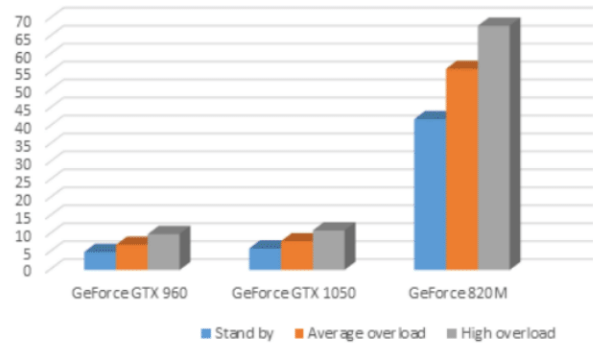


Figure 5: Graphics cards performance (milliseconds).

In fig. 5 is shown the results obtained in three different computers, in three different computational overload tests. The best results have been obtained with the GeForce GTX 960 graphics card which correspond to 4 milliseconds when working in "Stand by" that corresponds to an almost zero workload in which within the virtual environment the robot is standing, facing simply moving the robot that corresponds to an average overload, have achieved results of 5 milliseconds and finally moving the arm and picking and dropping objects constantly, which we have denominated as high overload, we have obtained results of 10 milliseconds of response time. The worst results correspond to those of the GeForce 820M that is part of the range of graphic cards for laptops.

4.2. VR functionalities under development

With the aim to improve the assistance of the user through the teleoperation process, within an intervention mission, new VR functionalities have been implemented. As a proof of concept a realistic scenario has been implemented dealing with recovering an aircraft's black box on the bottom sea. Thus, a couple of images are displayed, in figs 6 and 7, where the target (i.e. black box) intervention area is shown, before and after to be approached by the robotic gripper in charge of the recovering actions.

Shortly, fig. 6 shows the initial state of the intervention target, with the black box model included inside the simulation as a graspable object. These kind of objects have inside the "Unity engine" a tag called "Takeable" in all the objects able to be grasped by the robotic gripper. In fact, this functionality is supported by a communication process between several classes activates and deactivates the highlighters of the objects that gives a special color to the objects when they can be picked up by closing the gripper (see fig. 7).



Figure 6: black box before to be approached by the robotic gripper.

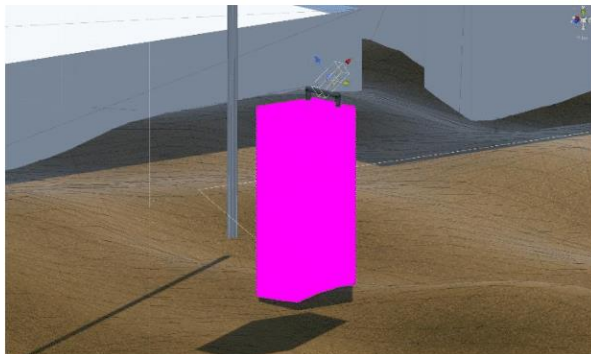


Figure 7: black box after to be approached by the robotic gripper.

So, in this manner a new functionality is now helping the user teleoperating a target, through available VR, guaranteeing to see if he has placed the arm and the robot correctly at any time.

4.3. The training and integration server

Finally, as can be observed in fig. 8, a training and integration server is under development also that is integrating all the hardware and software for the complete TWINBOT scenario. This complex VR 3D scenario includes two I-AUV's (Girona-500), with the arms, cameras and all kind of sensors within a harbour context or similar. Implementation details are out of the scope of this paper because are linked with the global TWINBOT project.



Figure 8: Unity development for the TWINBOT concept.

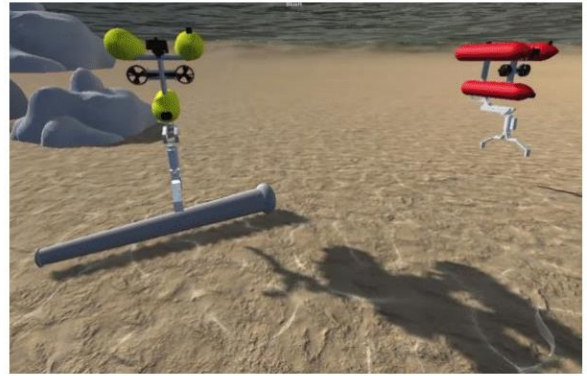


Figure 9: TWINBOT Simulation Server: Approaching Robots to the pipe.

The simulation server accepts network commands in several formats, such as TCP/IP and HTTP, providing the sensor information (e.g. robot positions and camera images) in http/JPEG and JSON packets respectively, while a ROS interface is under development.

Examples of the TWINBOT Simulation Server (TSS) state during an interaction with the GUI are shown in Figures 10-13.

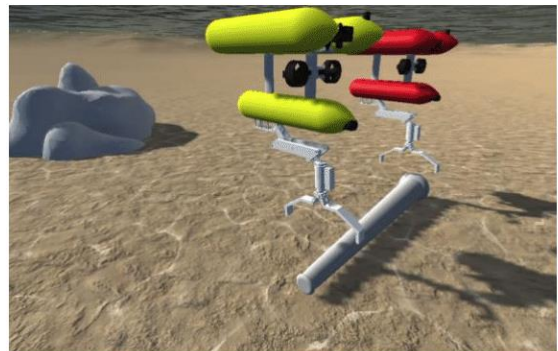


Figure 10: TSS: Approaching the grasping points.

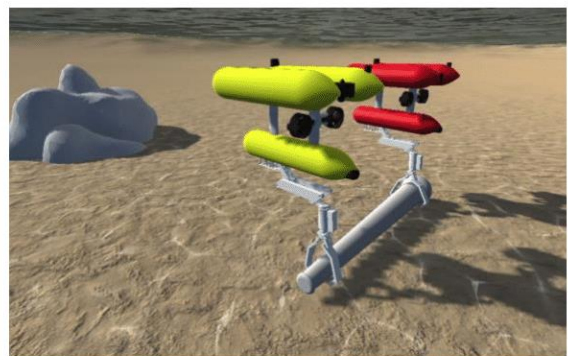


Figure 11: TSS: Closing simultaneously the grippers.

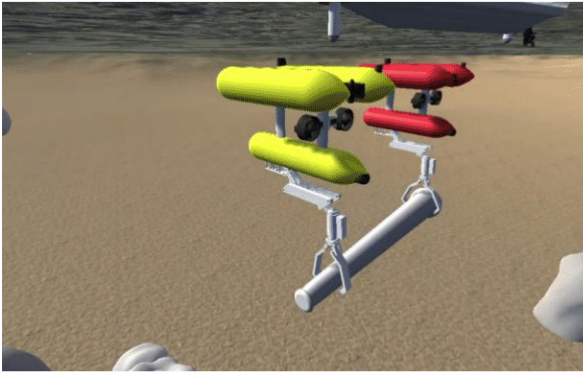


Figure 12: TSS: Transporting the pipe in a cooperative manner (underwater).

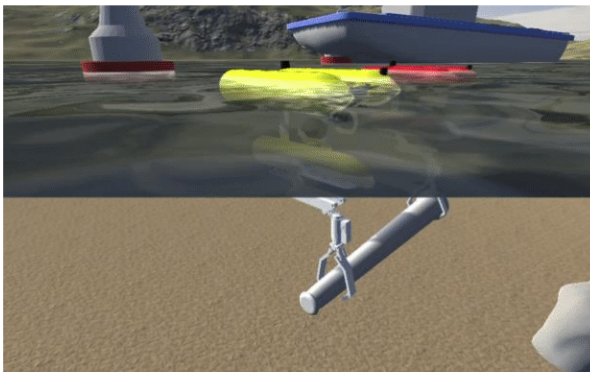


Figure 13: TSS: Transporting the pipe in a cooperative manner (approaching the surface).

Also, in collaboration with CERN Mechatronics and Robotics Section, the Unity simulation Server has incorporated the interface of the CERN Robotic Framework, which enables the interaction of the server in the same way the robots pool at CERN work, enabling the use of the Multimodal Unified User interface

In fact, the use of the CERN Robotic Framework User interface facilitates enormously the interaction of a single operator with the robotic team, especially by using the cooperative scripts edition tool, which permits the simple design, activation and deactivation of semi-autonomous behaviors for robotic vision and synchronized leader-following tasks. For example, one the vision-based behaviors that are being developed is the 2D/3D grasping determination algorithm applied to cooperative manipulation.

5. The system in action

With the aim to better understand the kind of facilities available through the VR under development (see fig. 14) a preliminary video sequence is available here: https://drive.google.com/file/d/1o4N4_oH1KyKjhEPFMizaCe5i1C6NeXbV/view?usp=sharing.

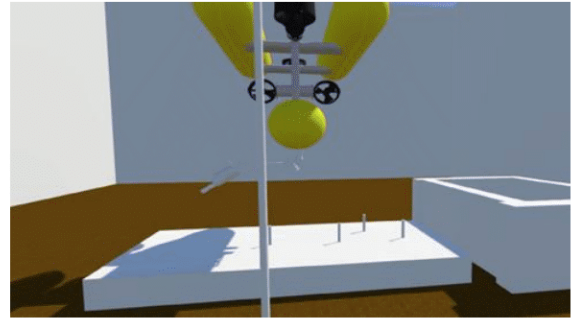


Figure 14: Snapshot extracted from the referenced video.

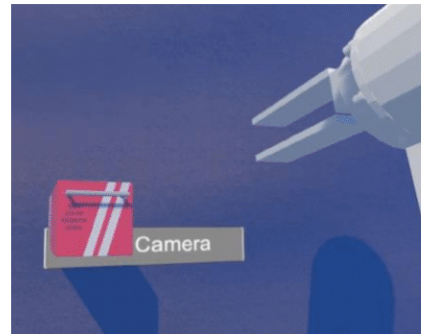


Figure 15: black box before to be approached by the robotic gripper.

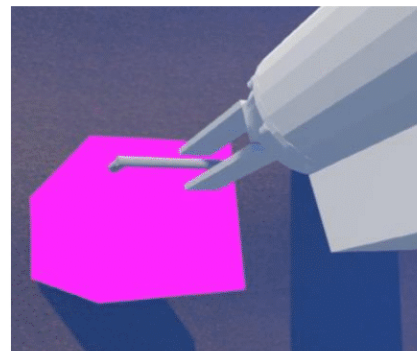


Figure 16: black box after to be approached by the robotic gripper.

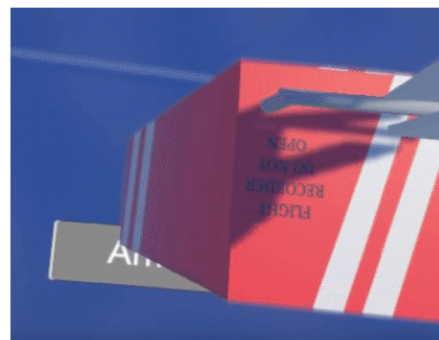


Figure 17: black box after to be grasped and recovered by the robotic gripper.

The complete sequence of actions (i.e. approached, grasp and recovery of a black box) can be observed through this available video at the link:

<https://drive.google.com/file/d/1HEWAS3iRfuBrlSckTAr6heunLGHObPLq/view?usp=sharing>

6. Concluding remarks

A short description concerning the UJI subproject (as part of Coordinated TWINBOT project) has been done. Letting apart some generalities of this subproject some specific details were shown concerning the HRI block. In particular those aspects dealing with immersive VR capabilities, within the underwater intervention context, has been discussed. Moreover, the named "Unity simulation Server" has incorporated the interface of the CERN Robotic Framework, in cooperation with our team, enabling the possibility for simulating cooperative robotic actions like those here presented. Finally, some videos are linked, in the paper, with the aim to better understand the system in action.

Acknowledgements

This work has been partly supported by Spanish Ministry of Economy, Industry and Competitiveness under grant DPI2017-86372-C3-1-R (COMOMUIS Project), by Valencian Government under GVA-PROMETEO/2016/066 grant, and by UJI under UJI-B2018-34 grant (NEPTUNO Project). This work has been partly supported by the CERN Engineering Department, under CERNTAURO Project.

References

- Fernandez, J. J., Prats, M., Sanz, P.J., García, J.C., Marin, R., Robinson, M., Ribas, D., Ridao, P., 2013. "Grasping for the Seabed: Developing a New Underwater Robot Arm for Shallow-Water Intervention," in IEEE Robotics & Automation Magazine, vol. 20, no. 4, pp. 121-130.
- Gandhi, R.D., Patel, D.S., 2018. Virtual Reality – Opportunities and Challenges. International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 01. www.irjet.net
- García, J.C., Patrao, B., Almeida, L., Perez, J., Menezes, P. Dias, J., Sanz, P.J., 2017. A Natural Interface for Remote Operation of Underwater Robots. IEEE Computer Graphics and Applications. 37(1), pp. 34 - 43.
- García-Fidalgo, E., Ortiz, A., Massot-Campos, M., 2017. Vision-Based Control for an AUV in a Multi-robot Undersea Intervention Task. Springer International Publishing. ROBOT 2017: Third Iberian Robotics Conference.
- Palomeras, N., Peñalver, A., Massot-Campos, M., Negre, P.L., Fernández, J.J., Ridao, P., Sanz, P.J., Oliver-Codina, G., 2016. I-AUV Docking and Panel Intervention at Sea. Sensors. 16 – 10.
- Prats, M. et al., 2012b "Multipurpose autonomous underwater intervention: A systems integration perspective," 20th Mediterranean Conference on Control & Automation (MED), Barcelona, pp. 1379-1384.
- Prats, M., Pérez, J., Fernández, J.J., Sanz, P.J., 2012. "An Open Source Tool for Simulation and Supervision of Underwater Intervention Missions Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2577-2582. Vilamoura, Algarve, Portugal.
- Prats, M., Ribas, D., Palomeras, N. et al., 2012a. Intel Serv Robotics, 5: 19.
- Ribas, D., Palomeras, N., Ridao, P., Carreras, M. and Mallios, A., 2012. "Girona 500 AUV: From Survey to Intervention," in IEEE/ASME Transactions on Mechatronics, vol. 17, no. 1, pp. 46-53.
- Sanz, P.J., Marín, R. 2017. An Underwater Robotic System for Cooperative Archaeological Intervention. Proc. of Jornadas Nacionales de Robótica, UPV, Valencia, Spain. <http://jnr2017.ai2.upv.es>
- Sanz, P.J., Ridao, P., Oliver, G., Casalino, G., Petillot, Y., et al., 2013. "TRIDENT: An European project targeted to increase the autonomy levels for underwater intervention missions," OCEANS - San Diego, CA, pp. 1-10.
- Unity User Manual, 2018. <https://docs.unity3d.com/Manual/index.html>