School of Information Technologies

Stanislav Jeršov 155251IAPB

DEVELOPMENT OF DIGITAL TWIN IN EXTENDED REALITY WITH UNREAL ENGINE 4

Bachelor's Thesis

Supervisor
Aleksei Tepljakov, Ph.D.
Research Scientist

TALLINN 2020

Infotehnoloogia teaduskond

Stanislav Jeršov 155251IAPB

DIGITAALSE KAKSIKU LOOMINE LAIENDATUD REAALSUSES UNREAL ENGINE 4 ABIL

Bakalaurusetöö

Juhendaja
Aleksei Tepljakov, Ph.D.
Teadur

TALLINN 2020

Declaration of Originality

I hereby declare that this thesis, my original investigation and achievement, submitted for the Bachelor's degree at Tallinn University of Technology, has not been submitted for any degree or examination. All the works of other authors used in this thesis have been referenced.

Date: May 26, 2020

Signature: ....Stanislav Jeršov

Abstract

The present thesis is dedicated to investigating potential applications of Digital Twins in extended reality to control systems. The pursued goal is to develop a pilot prototype, which, after being tested to improve the quality of training on real examples at the university, can also be transferred to the industrial context and used for training engineers of automatic control systems, thus facilitating industrial use of XR and DT.

The achievement of goal stated above requires the following problems to be solved:

- Performing a comprehensive literature overview, since any technical thesis should draw from previous R&D results.
- Creation of 3D digital representation of the real object.
- Implementation of real-time system control using PI controller.
- Implementation of real-time change of control parameters and subsequent change of system behavior.
- Comparison of theoretical predictions and practical results to verify the validity of system control.
- Implementation of two-way communication between the real object and the DT.
- Implementation of the ability to influence the real object through interactions with the DT (setting custom set points).
- Data visualization in an intuitive way.

The thesis is in English and contains 75 pages of text, 9 chapters, 43 figures, 1 table.

Annotatsioon

Digitaalse kaksiku loomine laiendatud reaalsuses unreal engine 4 abil

Käesolev lõputöö on pühendatud digitaalsete kaksikute võimalike rakenduste uurimisele laiendatud reaalsuses süsteemide juhtimiseks. Eesmärgiks on arendada piloot-prototüüp, mida pärast katsetamist ülikoolis saab üle viia ka tööstuskonteksti ja kasutada automaatsete juhtimissüsteemide inseneride koolitamiseks, lihtsustades sellel viisil tööstuslikku XR ja DT kasutamist.

Ülaltoodud eesmärgi saavutamiseks on vaja lahendada järgmised probleemid:

- Põhjaliku kirjanduse ülevaate koostamine, kuna iga tehniline lõputöö peaks põhinema teadus- ja arendustegevuse varasematel tulemustel.
- Pärisobjekti digitaalse esituse loomine.
- PI kontrolleri abil reaalajas töötava süsteemi juhtimise loomine.
- Juhtimisparameetrite reaalajas muutmise ja süsteemi käitumise muutmise implementeerimine.
- Teoreetiliste ennustuste ja praktiliste tulemuste võrdlus süsteemi juhtimise õigsuse kontrollimiseks.
- Kahesuunalise kommunikatsiooni implementeerimine reaalse objekti ja DT vahel.
- Reaalse objekti mõjutamise implementeerimine DT-ga interaktsioonide abil (seadesuuruste seadmise kaudu).
- Andmete intuitiivne visualiseerimine

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 75 leheküljel, 9 peatükki, 43 joonist, 1 tabelit.

Nomenclature

DT	Digital Twin
VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
XR	Extended Reality
SCADA	Supervisory Control And Data Acquisition
HMD	Head-Mounted Device
HIL	Hardware-In-the-Loop
PID	Proportional-Integral-Derivative (controller)
API	Application Program Interface
ITAE	Integral Time Absolute Error
SP	Set Point
SVG	Scalable Vector Graphics
FBX	Filmbox
UMG	Unreal Motion Graphics
I/O	Input/Output
CAN	Controller Area Network (message)
UDP	User Datagram Protocol
SDK	Source Development Kit
PNG	Portable Network Graphics

Contents

List of Figures

List of Tables

1. Introduction

Over the last years there have been substantial advances in computer science and technology and, consequently, many areas of applications have benefited from their development and progress. Production design, manufacturing, education and engineering as a whole are capable of raising the efficiency and productivity to a new level due to technological progress in both hardware and software directions. An exponential growth in computational power gave rise to new methods, algorithms and theories. Unsolvable problems, or more precisely, formerly lengthy problems, became solvable in real-time. Thereby, a technology of creating copies of virtual systems and environments emerged, namely, Virtual Reality (VR) [1].

Contrary to popular belief, VR is not being primarily used in the field of entertainment. Owing to its exceptional visualization capability and provision of immersive experiences and advanced interaction mechanics, VR has found applications in many forms. Among notable examples are virtual reality control systems, virtual laboratories and simulation/visualization of processes and products. Given the fact that a lot of information about the surrounding world is received through vision, visualization of processes can be very useful and cognitive. These development directions will be described in subsequent sections [2].

The concept of VR, together with the concept of augmented reality (AR), which allows digital objects to be superimposed over a real-life environment, gave rise to the concept of Mixed Reality (MR) merging both real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time [2]. Onward, referring to all real-and-virtual combined environments, a term Extended reality (XR) was introduced.

On the other hand, digital realization of objects as both mathematical and three-dimensional models has given rise to the concept of Digital Twins (DT) which also have obvious industrial applications in many areas [3]. One such area is control systems with many open problems [4].

The present thesis is dedicated to investigating potential applications of Digital Twins in extended reality to control systems. Specifically, facilitating industrial use of XR and DT by initially implementing a training application that can be used in the classroom is of the primary interest [2].

1.1. XR and its representative forms

In this section a more detailed definition of Extended Reality, its representative forms - VR, AR, MR along with related concepts will be provided.

Reality - virtuality continuum Being first introduced by Paul Milgram, the reality-virtuality continuum is a concept that encompasses all possible forms and compositions of real and virtual objects, spanning from completely real to completely virtual environments [5].

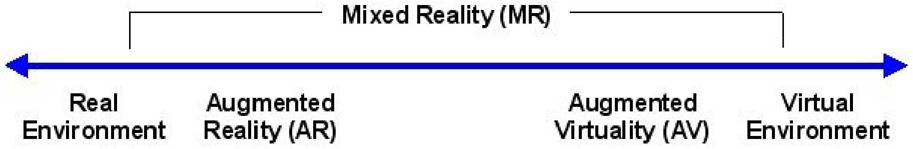


Figure 1. Reality-virtuality continuum

Taking into consideration the fact that this concept ties together different kinds of experiences, understanding of it provides the required basis necessary for further development of the theme of present thesis.

Virtual Reality Virtual Reality is a simulated experience created using technical means, that can be similar to or completely different from the real world. Currently standard virtual reality systems use either virtual reality headsets or multi-projected environments to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual environment. A person using virtual reality equipment is able to look around the artificial world, move around in it, and interact with virtual features or items providing a strong sense of immersion [6].

The effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes, but can also be created through specially designed rooms with multiple large screens. Virtual reality typically incorporates auditory and video feedback, but may also allow other types of sensory and force feedback through haptic technology [6].

Taking into consideration the fact that people are accustomed to receiving an instant feedback to their actions from the real world, it is essential for the virtual environment to be simulated in real-time in order to maintain the sense of immersion [6, 7].

Augmented Reality Whereas it is most commonly believed that virtual reality completely replaces the user's real-world environment with a simulated one, augmented reality tends to alter one's ongoing perception of a real-world environment. It can be defined as a system that fulfills three basic features: a combination of real and virtual worlds, real-time interaction and accurate 3D registration of virtual and real objects. An essential part of augmented reality experience is the manner in which components of the digital world blend into a person's perception of the real world. Merely displaying portions of additional data is not sufficient for provision of remarkable augmented reality representation, instead the approach should include the integration of immersive sensations, which are perceived as natural parts of an environment. Real environments can be enhanced across multiple sensory modalities, such as visual, auditory, haptic etc. In this way, the experience is seamlessly interwoven with the physical world [7, 8].

The overlaid sensory information can be of two types: *constructive*, which means additive to the natural environment or *destructive*, i.e. masking of the natural environment [8].

Mixed Reality Mixed reality (MR) is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real-time. Mixed reality does not exclusively take place in either the physical or virtual world, but is a hybrid of reality and virtual reality, encompassing both augmented reality and augmented virtuality via immersive technology.

Engaging the concept of reality-virtuality continuum previously described in earlier section, MR takes place in the area between the two extremes, where both the real and the virtual are mixed [7, 9].

This is the point where confusion may arise: the similarities between definitions of AR and MR are clearly seen. One may wonder that Mixed Reality is just a "Reality" that has been "Augmented". Technically, this is correct. However, there are certain differences in user experience and developmental complexity, that do not afford these terms to be interchangeable [7].

Extended Reality Term Extended Reality refers to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables, where the 'X' represents a variable for any current or future spatial computing technologies. It is the ultimate umbrella category that covers all the possible forms of computer-altered reality, including: VR, AR, MR and the areas interpolated among them [7]. The levels of virtuality range from partially sensory inputs to immersive virtuality, also called VR.

As it was already defined, XR is a superset which includes the entire spectrum from "the complete real" to "the complete virtual" in the concept of reality-virtuality continuum. Still, its connotation lies in the extension of human experiences especially relating to the senses of existence (represented by VR) and the acquisition of cognition (represented by AR). With the continuous development in human-computer interactions, this connotation is still evolving [10].

1.2. DT definition and overview

The combination of immersive visuals and wide functionality led to growing interest and development of Digital Twins. This concept originates from the Industry 4.0 developments and describes a one-to-one representation of a real world artefact (such as machine, component or part of the environment) comprising its visual representation, its behavior and its communication facilities [11].

The paper [12] defines DTs as software implementations of their physical counterparts that act through Application Program Interfaces (API) to the physical devices they monitor, engage with and possibly control. In paper [13] DTs are defined as integrated multi-physics, multi-scale, multi-disciplinary attributes with real-time synchronization, faithful mapping, high fidelity, and the ability to implement the technical means of interaction and integration between the real world and the information world. Combining this with appropriate simulation algorithms enables to execute the model, making the Digital Twin “experimentable”. Therefore, it becomes possible to test and evaluate the performance of newly developed overall-systems under different boundary conditions, interactions between components and subsystems in various operational scenarios and intended modifications without making any changes to the actual system [14, 15].

DTs can be controlled and programmed in real-time from the computer and are, if necessary, capable of making operational changes in the real object [16, 17] . As a product of two-way communication between the digital and the real object, the physical twin can use knowledge from digital twins to improve its characteristics [17].

1.3. Problem statement

The present thesis is dedicated to investigating potential applications of Digital Twins in extended reality to control systems. The pursued goal is to develop a pilot prototype, which, after being tested to improve the quality of training on real examples at the university, can also be transferred to the industrial context and used for training engineers of automatic control systems, thus facilitating industrial use of XR and DT.

When engaging extended reality solutions in scope of control systems, industrial partners are primarily interested in three applications [2]:

1) Training;

2) Process monitoring and analysis;

3) Controller design.

When dealing with industrial applications, the digital twin can contribute to understanding process dynamics. A mathematical model can represent the latter, and if this model is sufficiently accurate, then seamless switching between the real object and the model with the digital twin should be possible inside an appropriately designed Supervisory Control And Data Acquisition (SCADA) application. Therefore, the items listed above should be successfully covered by a suitable application of the digital twin for any given process [2].

In the present thesis a real-life laboratory model of a multi-tank hydraulic system will be used as a use case. The work will be focused on the proposed method and the technical implementation. As the end result, a Digital Twin in extended reality with necessary interaction mechanics will be developed and connected to the real plant for performing real-time Hardware-In-the-Loop (HIL) experiments. Unreal Engine 4 will be used as the basis for visualization and functional implementation of the digital twin, whereas MATLAB and Simulink environment will be used for the implementation of control logic and an interface to the actual real-life system. In terms of system control, PI control will be focused on, since it is considered to be one of the most popular industrial control methods [18]. However, the proposed methodology can be applied to an arbitrary control technique. Also, in order to observe and manipulate the perceived environment, VR HMD (Head Mounted Device) and motion controllers will be used.

The achievement of goal stated above requires the following problems to be solved:

- Performing a comprehensive literature overview, since any technical thesis should draw from previous R&D results.
- Creation of 3D digital representation of the real object.
- Implementation of real-time system control using PI controller.
- Implementation of real-time change of control parameters and subsequent change of system behavior.
- Comparison of theoretical predictions and practical results to verify the validity of system control.
- Implementation of two-way communication between the real object and the Digital Twin.
- Implementation of the ability to influence the real object through interactions with the DT (setting custom set points).
- Data visualization in an intuitive way.

1.4. Methodology

This thesis is based on practical approach and the proposed method. In scope of the present work much attention is paid to development of a working application. In addition, a certain amount of analysis of underlying concept is included. Moreover, several goals and criteria are established in order for technical implementation to be oriented towards them.

1.5. Outline of thesis

The present thesis is structured in a logical way so that each subsequent part provides a description of a new aspect of development, while taking into consideration previous parts and referring to them, thus making it possible to track a logical and consistent development process.

Chapter 1

The first chapter provides an introduction to the present thesis and specifies the pursued objectives along with problems that need to be solved in order to successfully achieve the stated goal.

Chapter 2

In the second chapter a comprehensive State of the Art is provided, thus forming the necessary base that enables further research and development.

Chapter 3

This chapter gives a general overview of the developed experimental system, describing main components, their interconnection and purpose as well as defining a detailed workflow.

Chapter 4

The fourth chapter depicts the iterative process of creating a 3D model of a multi-tank hydraulic system that is going to be used further throughout the whole development process.

Chapter 5

This chapter is devoted entirely to clarifying UE4 solution and the functionality of the digital twin in accurate detail. Not only the implementation of main mechanics of the digital twin, but also core functionality these mechanics rely on are described.

Chapter 6

The main purpose of the sixth chapter is to provide a description of the way the data transmission between the digital twin and the real object is achieved.

Chapter 7

Being an essential part of convenient and useful user experience, data visualization has an important role in the given system and is described in the present chapter.

Chapter 8

This chapter provides a description of the mathematical model used to define the behavior of the real plant, system control with a PI controller as well as a conducted real-time experiment, the results of which are compared to theoretical results, thereby proving the effectiveness of the developed solution.

Chapter 9

In the last chapter conclusions are drawn, summarizing everything that has been achieved throughout the development.

2. State of the Art

In this section, research related to XR, digital twins, control systems, and virtual laboratories is reviewed. This knowledge must guide further research and development efforts where the ultimate goal is to not only provide advanced XR training facilities, but also implement these techniques for developing novel SCADA systems.

2.1. Virtual laboratories

Over the last years virtual laboratories have made a significant contribution to education, because computer simulations that are properly designed are useful for a variety of contexts that can promote student learning [19]. In its most common instructional design, the virtual laboratory provides a simulation that mirrors a physical laboratory at the university [20]. In addition, it allows students to experiment with the physical phenomena in a safe environment and may provide students with information not available as well as inclusion of visual cues or alternative representations not possible to observe in physical laboratories (e.g. visualization of emergent molecular interactions) [20, 21].

Whereas many virtual laboratories are only accessible within the university, there are also remote experimentation laboratories (although not always supporting VR devices like Oculus or HTC Vive—a noteworthy development nevertheless) that are accessible as part of a web-based learning approach. Expedited by advances in web applications and related technologies, remote experimentation laboratories have evolved in recent years, setting the accurate recreation of on-campus laboratory-based student experience with an equivalent level of user access, functionality and flexibility as the end objective [22]. Such remote experimentation facilities afford several critical benefits, augmenting and complementing web-based learning material by facilitating distant access to based in campus physical resources [23]. This is consistent with the statement given in the paper [24], which is: “Simulation based labs cannot provide a ‘feel’ for real things. Students need to use real devices and execute commands on real tools to gain necessary practical skills.”. For engineering related distance education courses the use of a web-based delivery mechanism is the only realistic method of providing practical hands-on experience, allowing remotely located students to complete laboratory assignments, unconstrained by time or geographical considerations while developing skills in the use of real systems and instrumentation [25, 26]. Presumably, the support for VR head mounted devices and controllers will be introduced in most remote experimentation laboratories in foreseeable future.

The paper [27] presents the description of virtual laboratory experiments in different conditions: physical only instruments, virtual only instruments and combination of both. Observation results show that in the combination condition students profited from comparing two potentially different representations of the same phenomena and using abstract reasoning to analyze the differences.

The paper [28] presents the virtual laboratory, where students can, among other experiments, participate in control system experiment which involves a Magnetic Levitator. The main concepts illustrated by this experiment are: PID (Proportional-Integral-Derivative) controllers, root locus controller design and ITAE (Integral Time Absolute Error) optimal controller design.

2.2. Virtual reality control systems

Virtual reality control systems are broadly used in system prototyping, design and simulation providing such benefits as reduced costs of development, mitigation of nonconformity and failure risks and greatly reduced prototyping and design time [1]. With respect to control systems, it is beneficial to display the virtual world and the real world synchronously. Comparing the real system with its virtual representation allows to get a profound understanding of control effectiveness of the whole system [29].

The paper [30], with a view to analysis of the dynamic behavior of the water jet cutting machine, presents a virtual machine, which consists of a real machine control bi-directional connected to a machine simulation, which is bi-directional connected to 3D visualization of the machine. This setup enables realistic system simulations, since the simulation input comes from a real control. Following this approach, it is possible to test large numbers of variants and optimize the product with the aid of a minimum of physical prototypes. The paper [31] brings the virtual machine approach further. The concept facilitates total systems simulation, including the dynamics of the control and the electrical and mechanical parts of the machine. This makes it possible to assess systems design suggestions at an early stage of development and to perform systems optimization. Thereby, the virtual machine concept is moving towards becoming a useful tool for supporting machine tool systems development.

2.3. Digital twins

Digital twins are being used in various industrial sectors to optimize the operation and maintenance of physical assets, systems and manufacturing processes [32].

For instance, General Electric (GE), builds cloud-hosted digital twins of its machines that process information collected from sensors using artificial intelligence, physics-based models, and data analytics to better manage those machines [33, 34].

The paper [16] presents the development of the DT of industrial robot, which is made as a precise digital copy of the real object. Aside from the functionality of the real object (e.g. movement of moving parts) such valuable concept as collision detection is described. Given that the DT is an exact digital copy of the real robot, as an alternative to providing the real object with proximity sensors it is possible to use the DT's geometry to monitor the position of the robot in its workspace and prevent it from accidental collisions with the environment. This collision detection approach provides almost the same level of reliability but does not require any extra physical equipment or significant setup time.

To give an example of work previously performed at Centre for Intelligent Systems, TalTech, and, in part also to control systems in XR, the paper [35] presents the description of the project aimed at providing means for including voluntary synesthetic experiences (synesthesia is the act of experiencing one sense modality as another) through the VR environment. The application of an acoustic localization method to the problem of locating the sound source in a room is investigated. Afterwards, sounds are processed and turned into visual signals, in other words visualized. In this project, the Kalman filter (an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements) is applied in order to reduce noise generated by the uncertainty of sound source location prediction. This is an important addition, since The Kalman filter has numerous applications in technology, including guidance, navigation, control of vehicles, signal processing, system control and many others. While this project was not directly related to control systems, its development provided necessary insights into immersive aspects of XR as well as interaction mechanics. In the present effort, this knowledge is fully leveraged to develop a digital twin of a control object, make it interactive, and provide facilities for the control system aspect, namely, process monitoring, control, and management. This is detailed in the following sections.

3. Overview of developed experimental system

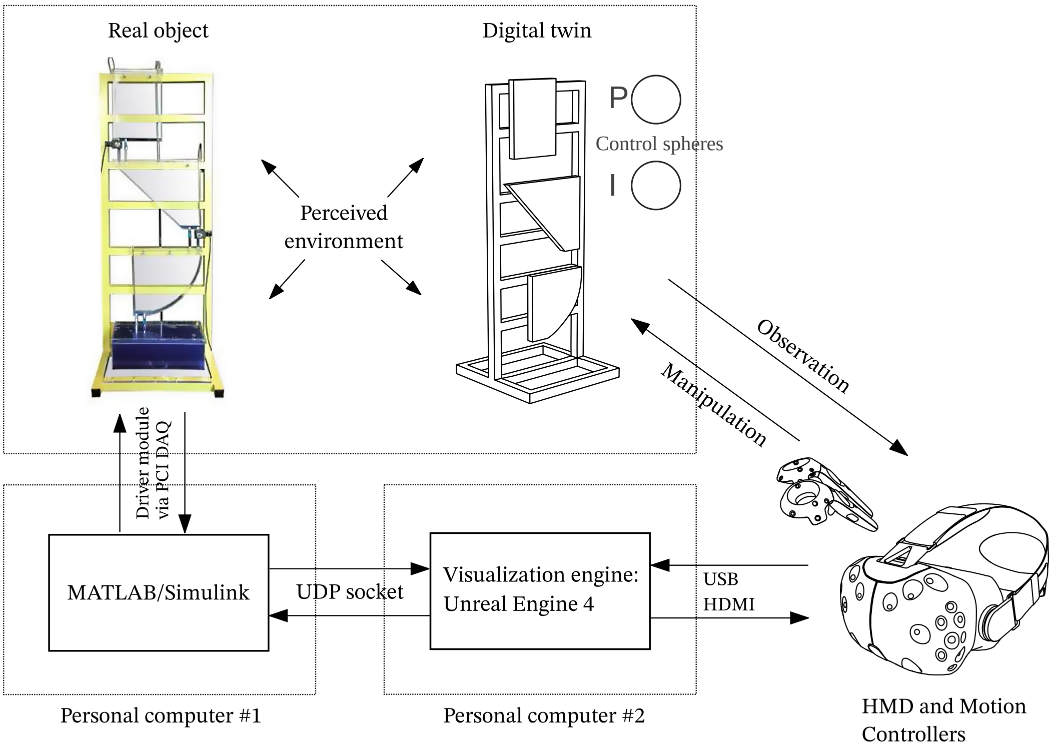


Figure 2. Architecture of the developed experimental system including the real object, its digital twin and necessary hardware components.

The main components of the developed experimental system are:

- Real object;
- Digital twin;
- MATLAB/Simulink solution;
- Unreal Engine 4 solution;
- Two-way UDP communication;
- HMD and motion controllers;
- Control spheres.

In general sense, the system consists of the multi-tank hydraulic system (the “real object”) that is connected to a PC with a PCI DAQ board and its digital twin that is rendered in Unreal Engine 4 on another PC. The real object represents three tanks attached to one base and interconnected via valves and tubes. With the help of valves and water pump the system raises the level of water in accordance with given set points (SP). This process is controlled by a controller implemented in MATLAB/Simulink environment. The two-way communication between the real object and the digital twin is established through UDP (User Datagram Protocol) socket. Then, VR HMD and motion controllers are used in order to observe and manipulate the perceived environment [2].

As it was mentioned previously, the PI controller is used in order to control the given plant. Being an essential element of efficient control performance, the correct adjustment of proportional gain K_p and the integral gain K_i parameters is a matter of high importance. With a view to providing the user with the opportunity to control K_p and K_i values and see the result of these changes in real-time, the "Control Spheres" are introduced. They represent an intuitive way to scale K_p and K_i values in the desired range by "stretching" movable corresponding spheres with both motion controllers. Values are then sent to MATLAB via UDP socket and processed for subsequent change in system behavior in accordance with the obtained parameter changes [2].

The workflow of the given system can be represented as follows [2]:

1. Initialization of MATLAB/Simulink model for the real object and real-time communication;
2. MATLAB model listens on UDP port, expecting the values for set points and PI controller gains;
3. Initialization of Digital Twin solution in UE4;
4. Initialization of communication of the Digital Twin solution in UE4;
5. Values of Set Points and controller gains for real tanks are sent to MATLAB via UDP socket;
6. MATLAB receives values of Set Points and controller gains from UE4 and processes them;

7. The system raises the level of water in real tanks in accordance with SPs using the controller updated with the received K_p and K_i gains;

8. Every time step (defined in the model) the water level values are sent to UE4 via UDP socket;

9. UE4 gets real water level values from MATLAB and visualizes the water in digital tanks at given level;

10. The user sets custom SPs in digital tanks by pointing at them with motion controllers and the gains of the controller by resizing two spheres ("P" and "I", respectively);

11. Custom SP values and controller gains are sent to MATLAB via UDP socket;

12. Back to step (6).

The detailed description of all engaged components of the system along with explanatory figures will be provided in the following sections.

4. Visualization of DT

Taking into consideration the fact that visualization of Digital Twin plays an important role in providing immersive experience and perception of the system, creation of 3D model that corresponds to a real object was essential.

4.1. Creation of 3D model and import to UE4

The first step towards creating a 3D model of a multi-tank hydraulic system was taking a picture of the actual object located at the laboratory. Next, this picture was imported to Inkscape in order to circle the shape and thus get flat representation of the object. The result then was exported as an SVG (Scalable Vector Graphics) file. Afterwards, the acquired file was imported to Blender in order to add volume and configure preliminary collision and the position of pivot point. As a result, the first version of the 3D model was ready to be imported to Unreal Engine 4 as FBX (Filmbox) file to create a Static Mesh component and be used in further development.

The ruler used on all three tanks was created in Inkscape and exported as a PNG (Portable Network Graphics) file. Afterwards, the file was imported to UE4 as a texture to be used to create a Material and apply it to a plane. Then, this plane was attached to each of three tanks to show the level of liquid in the future.

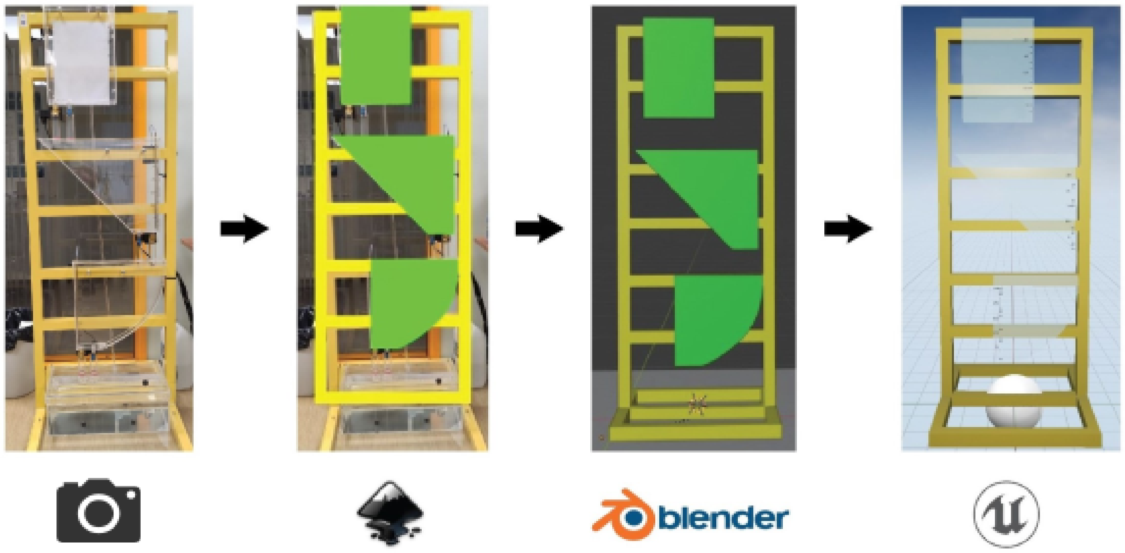


Figure 3. The first iteration of 3D model.

The second iteration of 3D model represents a highly detailed multi-tank system model in more accurate scale which will be used throughout further development.

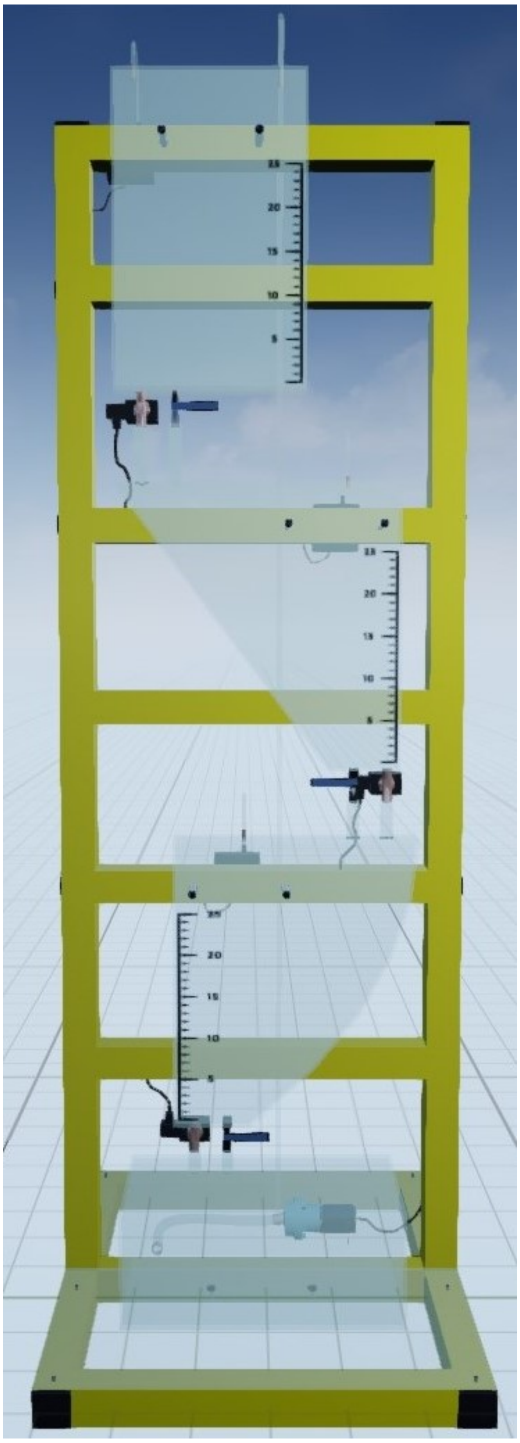


Figure 4. Detailed 3D model.

5. Unreal Engine 4 solution

Once the 3D model is ready it is possible to use it in further development of the Digital Twin.

The next stage of development includes adding necessary components, implementation of liquid level and interactions.

Most of functionality is developed using Blueprints—a flexible visual scripting system inside UE4, however there are certain parts of the project that are developed in C++ and exposed to Blueprints. A more detailed description of C++/Blueprint mixed development will be provided in the following sections.

5.1. Adding sockets

In order to effectively move further to liquid level implementation certain preparatory work has to be done. More precisely, so-called Sockets are placed at lowest points of each tank. It allows many manipulations to be performed, in case of the present thesis they are going to be used in order to easily get the location of the lowest point of each tank at any moment.

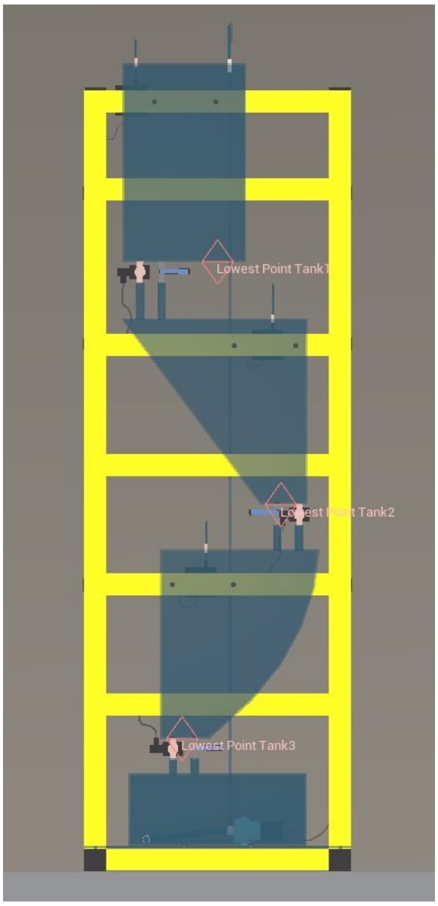
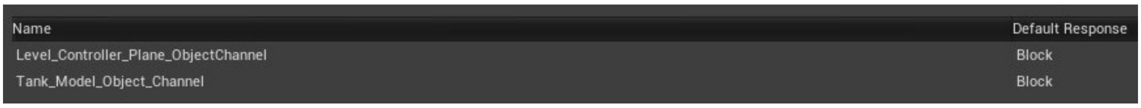


Figure 5. Sockets at lowest points of tanks.

5.2. Configuring collision

Prior to implementing interaction mechanics it is necessary to configure collision settings. More precisely, custom object channels are being used in the present solution. There are several collision presets initially available “out of the box” in UE4, each with its own specifics and suitable for various applications, but defining custom channels and the way an object is supposed to react to them gives developer more freedom and extends boundaries of the default functionality. It allows, for example, to make objects only of certain type to collide with objects only of another certain type.



Name	Default Response
Level_Controller_Plane_ObjectChannel	Block
Tank_Model_Object_Channel	Block

Figure 6. Custom object channels.

The figure above shows object channels that were defined in the present solution.

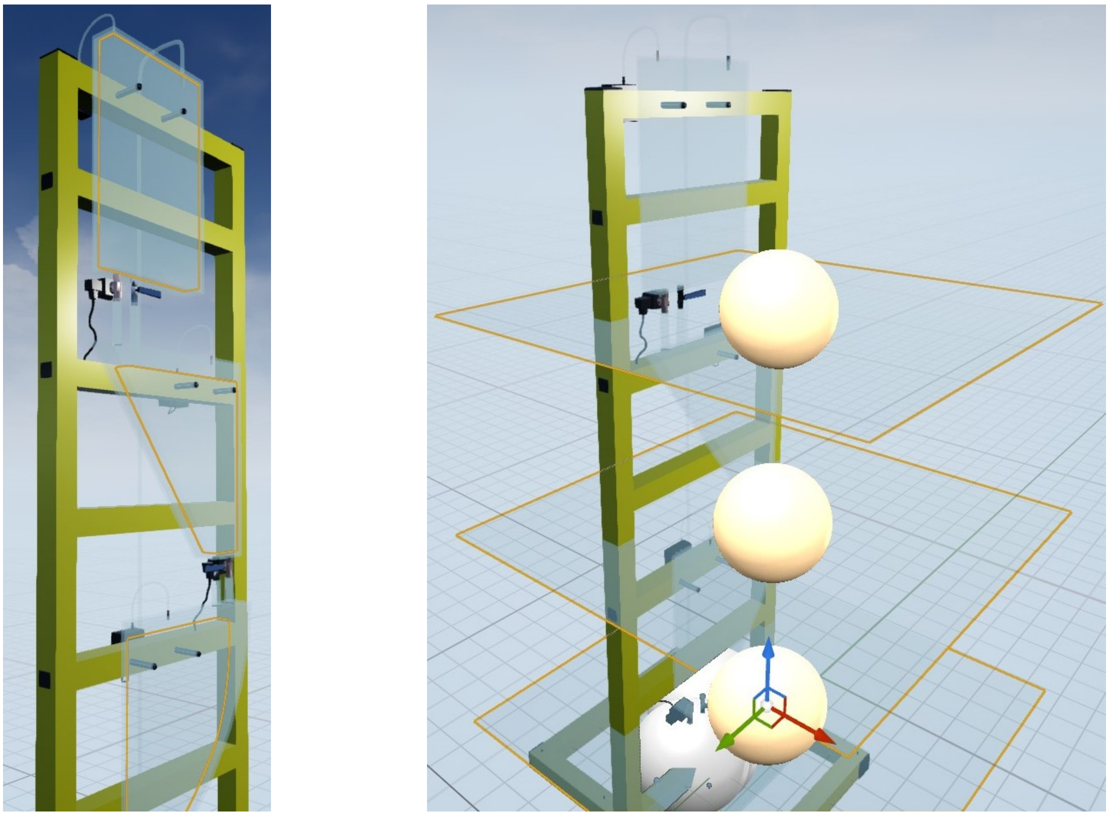
- “Level_Controller_Plane_ObjectChannel” will be used for setting custom Set Points.
- “Tank_Model_Object_Channel” will be used mostly for moving the Digital Twin around.

A more detailed description of using aforementioned channels will be provided in future sections.

5.3. Liquid level

One of the most important aspects of the DT presented in this work is the ability to display the level of liquid corresponding to the real object. In this section a detailed description of how it is achieved will be provided.

To begin with, three additional objects (henceforth “TankLiquids”) are added to the model. The shapes of TankLiquids correspond to shapes of visible tanks, but the size is slightly smaller in order to imitate the thickness of tank’s walls. Also, a particular material called Liquid_Tank_Material is applied to each TankLiquid. Then, there is a special plane called BPHider attached to each TankLiquid. By default, BPHider is located at the lowest point of each tank. This is easily achieved by acquiring the position of sockets (described in Section 5.1) and attaching BPHiders to them. Once the BPHider starts to move up/down (along Z axis), the TankLiquid is being “cut-off” depending on BPHider’s position.



(a) TankLiquid objects selected.

(b) BPHider objects selected.

Figure 7. TankLiqid and BPHider objects selected.

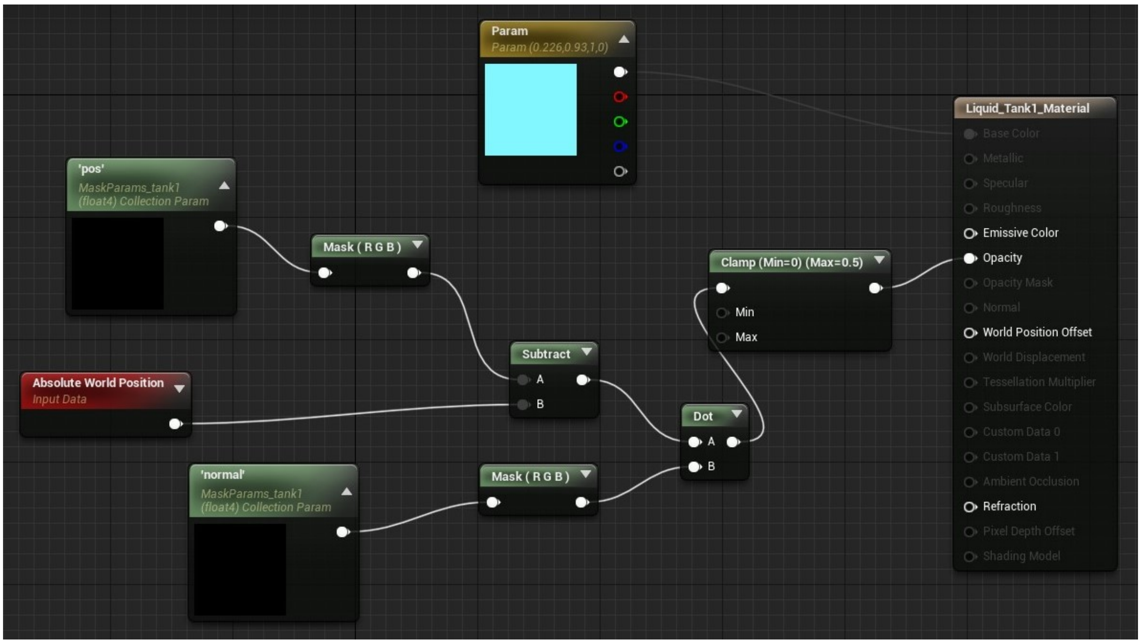


Figure 8. Liquid_Tank_Material graph.

Liquid_Tank_Material is created using the following parameters passed to the result node of the material:

- Vector Parameter—an expression that outputs a four-channel vector value, in other words, four constants numbers and can be modified in instances of the material and through code. For example, an RGBA color can be represented by it, where each channel is assigned to a color: red, green, blue, alpha [36]. In the present solution the value is set to (0.226, 0.93, 1, 0) which corresponds to a light sky-blue color.
- Opacity—an expression that controls the translucency of the material.

Opacity calculation relies on the use of the following parameters and data:

- Material Parameter Collection - an asset that can store an arbitrary set of scalar and vector parameters which can be referenced in any material [36]. In the present solution it contains 2 vector parameters: “pos” and “normal”.
- Absolute World Position—an expression that outputs the position of the current pixel in world space [36].

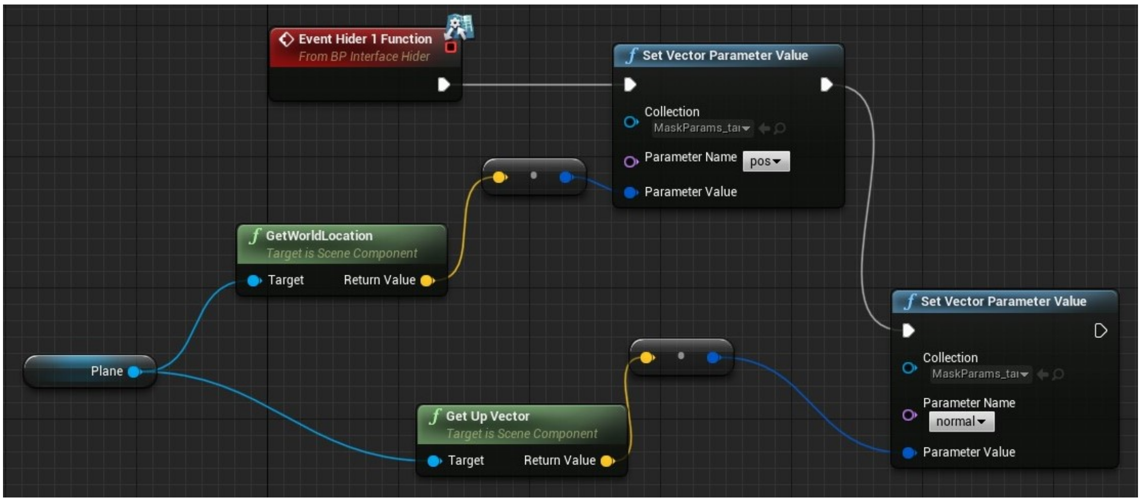


Figure 9. BPHider function.

Once the application is run, every frame “pos” and “normal” are assigned values of BPHider’s world location (the position in world space) and up vector (Z unit direction vector in world space) respectively.

The actual Opacity calculation is performed with the use of Math Expressions (expressions that perform mathematical operations on one or more inputs [36]). The dot product of two vectors is calculated: the normal vector of the BPHider represented by "normal" and the location vector between BPHider and the position of current pixel in world space. The acquired value is either positive or negative. Then, the value is clamped to constrain it to a specified range from 0.0 to 0.5.

Onward, Vector Parameter is passed to the result node of the material as Base Color and Opacity is set to the result of aforementioned calculation.

Since by default BPHider is located at the lowest point of tank, the "liquid", i.e. TankLiquid is completely "cut off" and therefore not seen - the tank is "empty".

The figure below depicts a situation where the BPHider is raised to approximately the middle of the tank.

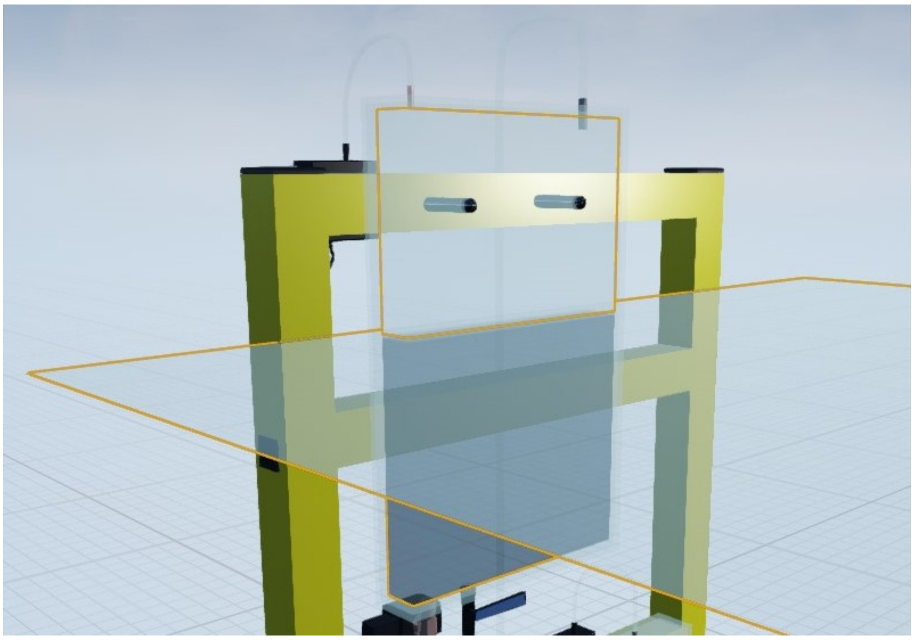


Figure 10. BPHider raised to the middle of the tank.

In this way, the functional base for visualization of liquid inside of the tank is complete, the only thing left is to programmatically set the position of the BPHider along the Z axis to a corresponding level of liquid in the real object. This will be covered in further sections.

5.4. Interactions

In the present thesis interaction mechanics are being emphasized on, since it is one of essential aspects of the developed solution. This section is devoted to providing a detailed description of developed fundamental methods on which the system of interaction in the given project relies.

5.4.1. MotionControllers

By default an "out of the box" MotionController class is used in the project for such basic functionality as tracking of controllers, setting controller visibility etc. However, implementation of interaction mechanics requires this limited functionality to be extended.

5.4.2. LineTrace

LineTrace is a convenient tool that is commonly used in UE4. It does a collision trace along the given line and returns the first hit encountered. This can be applied to various kinds of mechanics, for instance LineTraces can be used in order to operate and navigate in a VR menu or when creating a character that moves and behaves in accordance with the surface on which it walks by casting traces from the bottom of the feet.

In the present solution a LineTrace of a particular type, a LineTraceForObjects, which only returns the hit from objects that are of a type specified by object channels, is involved and will be used for two purposes:

- To set custom Set Points by pointing at the desired level in digital tanks.
- To move the DT around.

5.4.3. Moving the DT

The multi-tank system model is set to be of type Tank_Model_Object_Channel (described in Section 5.2) and the LineTrace is set to react only to this type of objects, disabling any interactions with objects of different type.

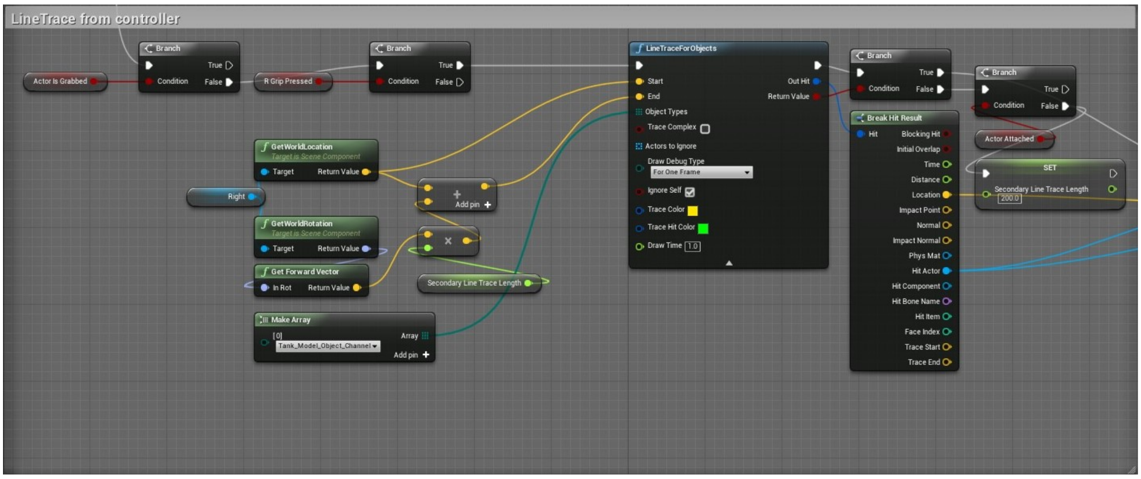


Figure 11. Casting a LineTrace from controller.

LineTrace is cast from the right controller once the Grip button is pressed. It is possible to configure the desired length of the trace, which is 200 cm by default. Then, the condition whether there is a hit with an object of type Tank_Model_Object_Channel is checked.

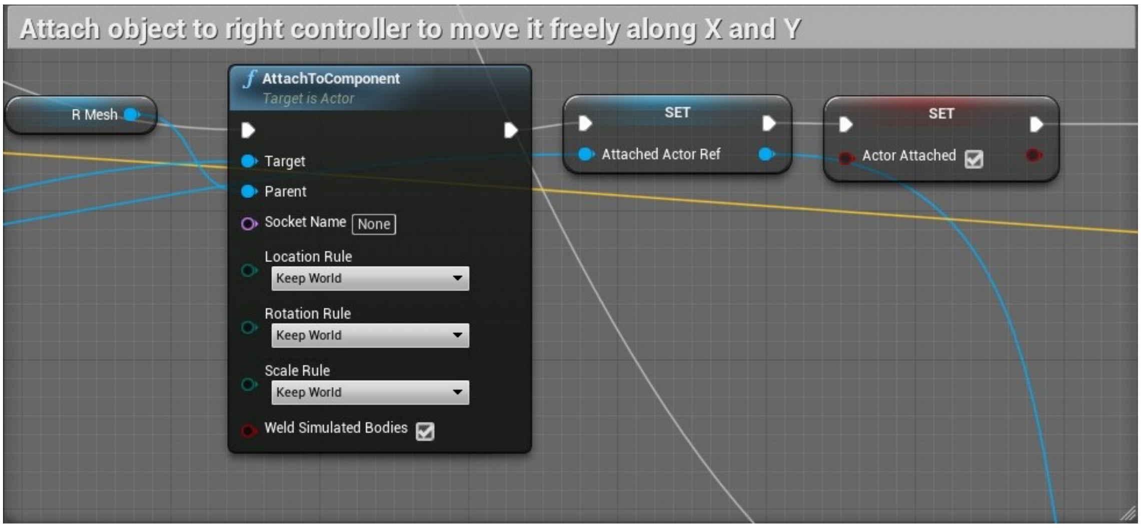


Figure 12. Attaching the object to controller if there was a hit.

If there is a hit, the object is attached to the controller, preserving its location, rotation and scale. In this way, the object is not "teleported" to the location of the controller, but is kept “on a leash” allowing the user to move the object freely around the workspace along X (forward) and Y (right) axis. Moving the object up/down is going to be covered further.

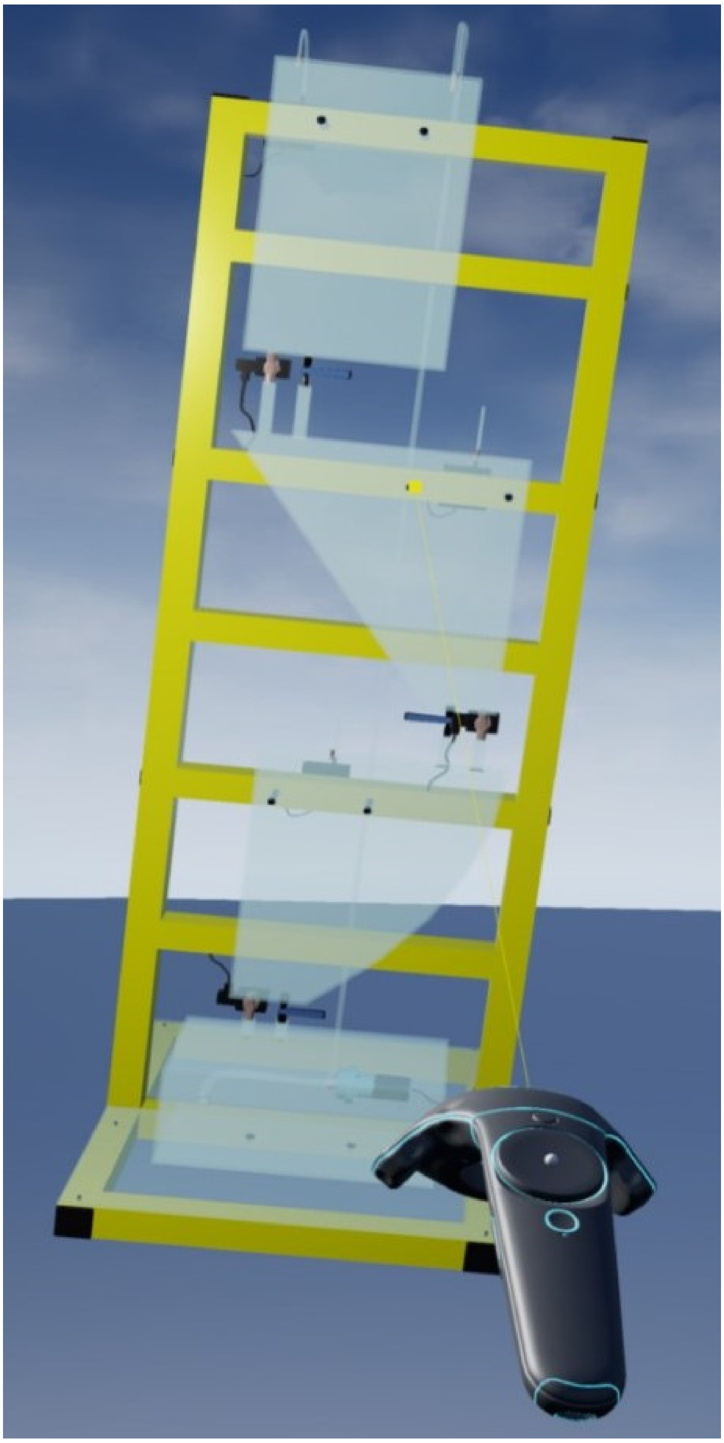


Figure 13. Attached object.

To prevent the LineTrace from “cutting through” the object in case if the distance between the object and the controller is shorter than the LineTrace length, the trace is “truncated” at the point of hit.

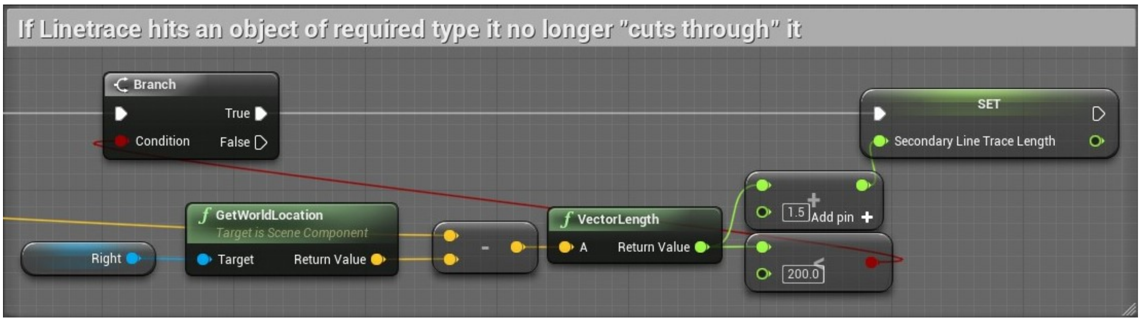

If Linetrace hits an object of required type it no longer "cuts through" it

Figure 14. Truncating the LineTrace.

Aside from simply moving the object along X and Y it is necessary to provide the user with the ability to adjust its rotation and position along Z (up) axis. When the object is already being held, every frame its location vector is acquired, split into component values and the Z value is incremented/decremented depending on the pressed button on the trackpad.

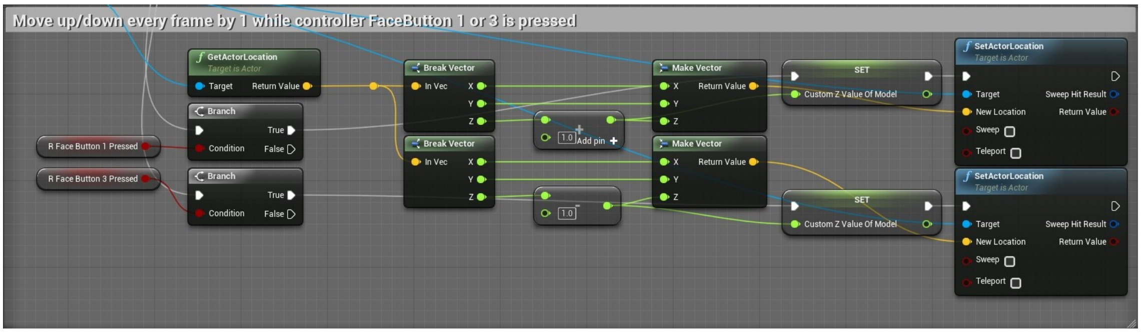

Move up/down every frame by 1 while controller FaceButton 1 or 3 is pressed

Figure 15. Moving the object up/down.

Object rotation around the Z axis is implemented in a similar way.

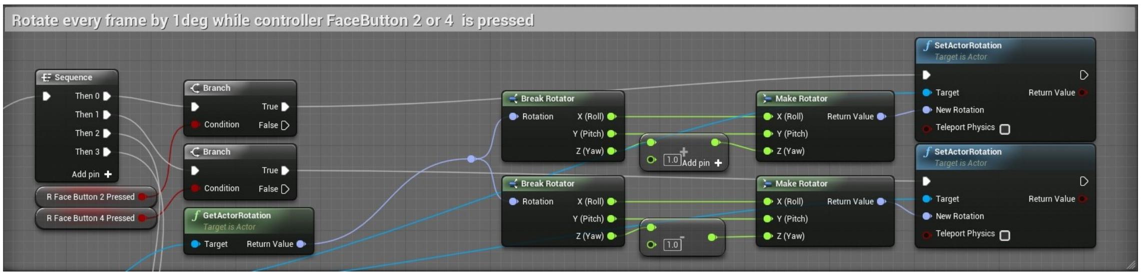

Rotate every frame by 1deg while controller FaceButton 2 or 4 is pressed

Figure 16. Rotating the object.

With an intention of making the use of aforementioned mechanics simpler and more intuitive an indicator of user’s current thumb position on the trackpad is implemented. Once the user touches the trackpad a little sphere moves to the location of the touch.

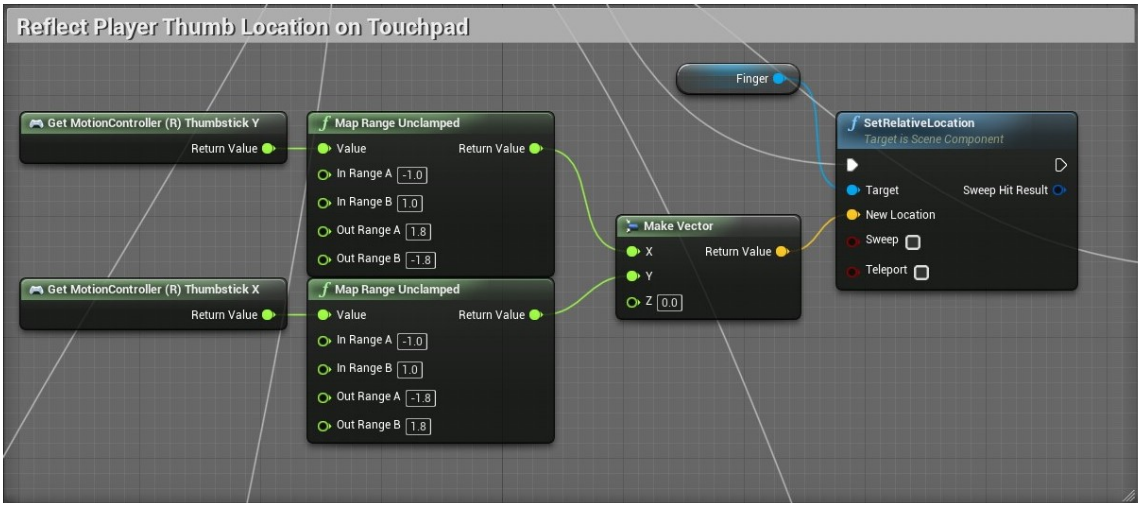


Figure 17. Thumb indicator graph.

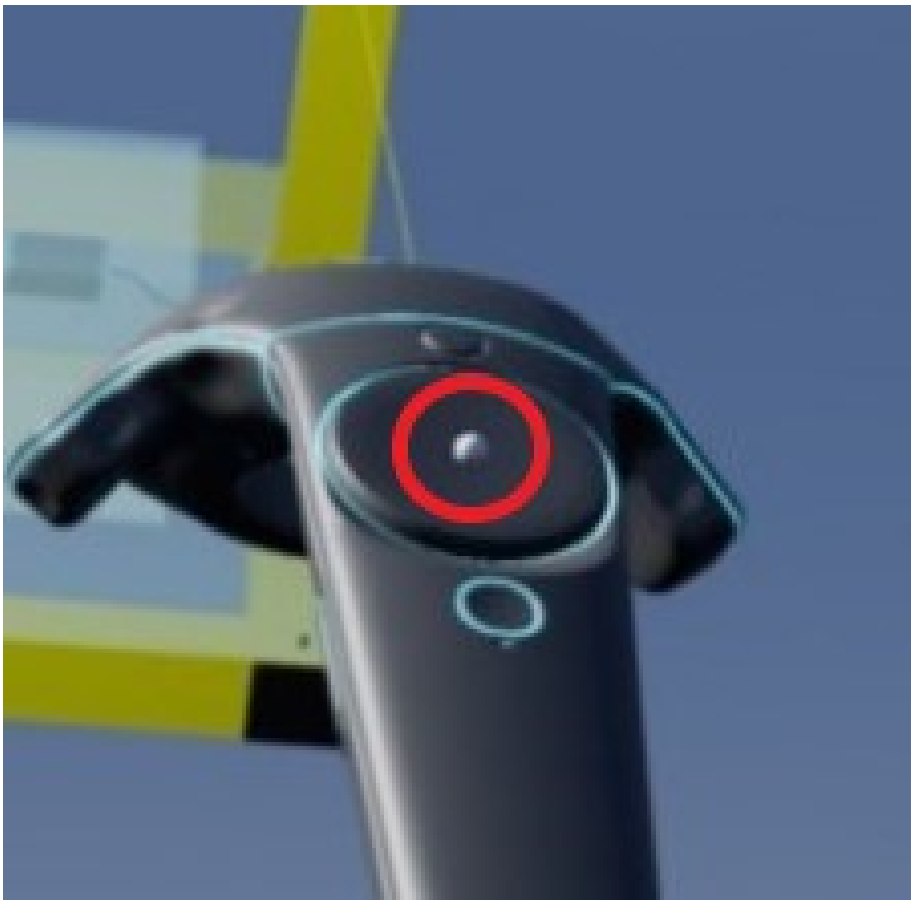


Figure 18. Thumb indicator.

5.4.4. Setting custom Set Points

The implementation of setting custom Set Points is partially similar to moving the object. However, certain modifications need to be made.

First of all, additional objects (henceforth "LevelControllerPlanes") are added to the model. Each LevelControllerPlane has a parameter Tank ID assigned to it (1, 2 and 3 for each tank respectively) and is set to be of type Level_Controller_Plane_ObjectChannel (described in Section 5.2). The LineTrace is set to react only to this type of objects, disabling any interactions with objects of different type.

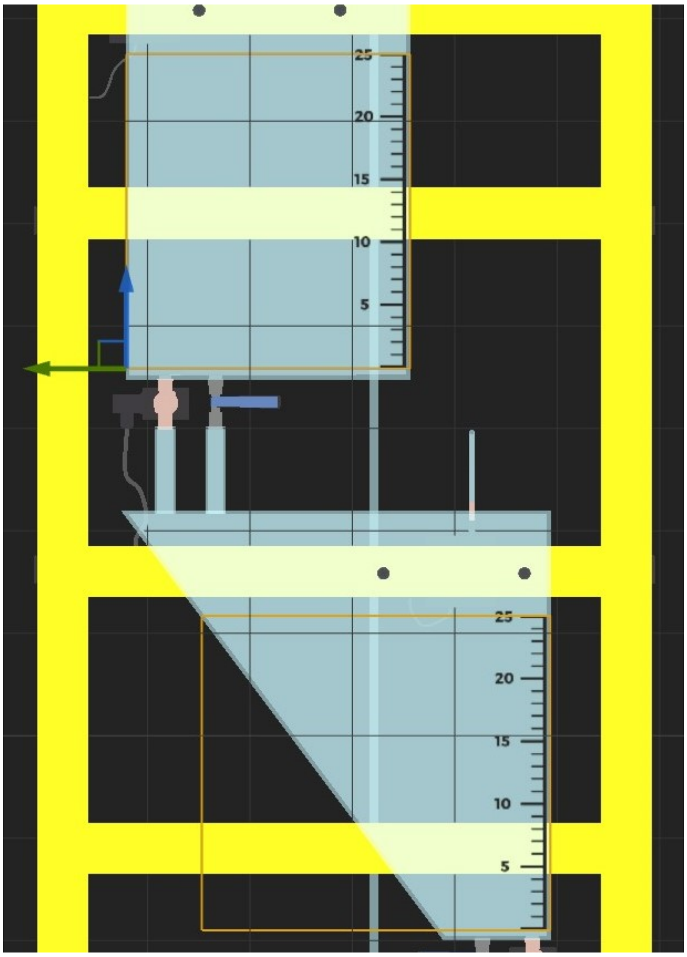


Figure 19. LevelControllerPlane objects selected.

If the LevelControllerPlane is hit the location vector of the hit in world space is acquired. Then, InverseTransformLocation is used in order to transform the position of the hit from world space to local space. Since the level of liquid is measured along vertical axis, in this case X, the location vector is split into component values and the X value is acquired. Now, the only step left is to interpolate the acquired value so that it can be sent to MATLAB.

In MATLAB the level of liquid is represented by a float value in range 0.0–0.25, which means the value acquired in the previous step should also fit into the given range.

To accomplish this, a linear interpolation formula is used:

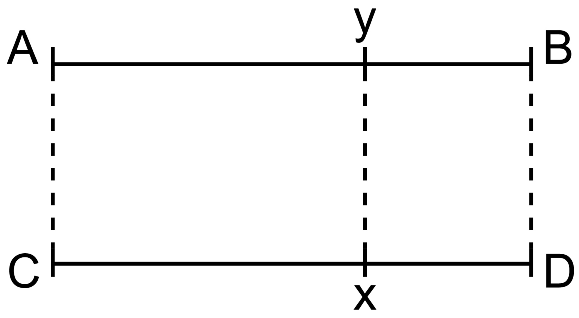
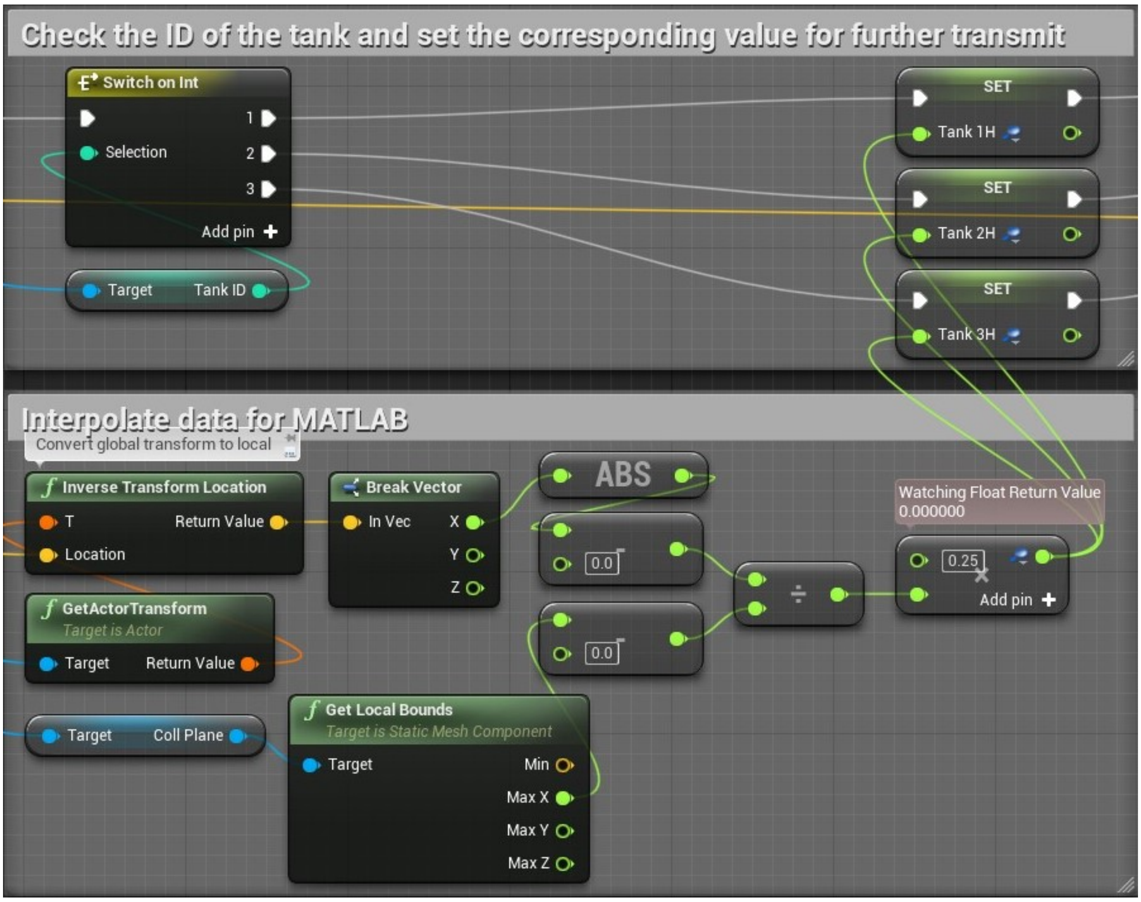


Figure 20. Linear interpolation scheme.

$$x = (D - C)\left(\frac{y - A}{B - A}\right) + C. \tag{1}$$

Then, depending on Tank ID of the given LevelControllerPlane a corresponding variable (Tank1H, Tank2H or Tank3H) is assigned the interpolated value. As a result, the custom Set Point is set and ready to be sent to MATLAB.



Check the ID of the tank and set the corresponding value for further transmit

Interpolate data for MATLAB

Convert global transform to local

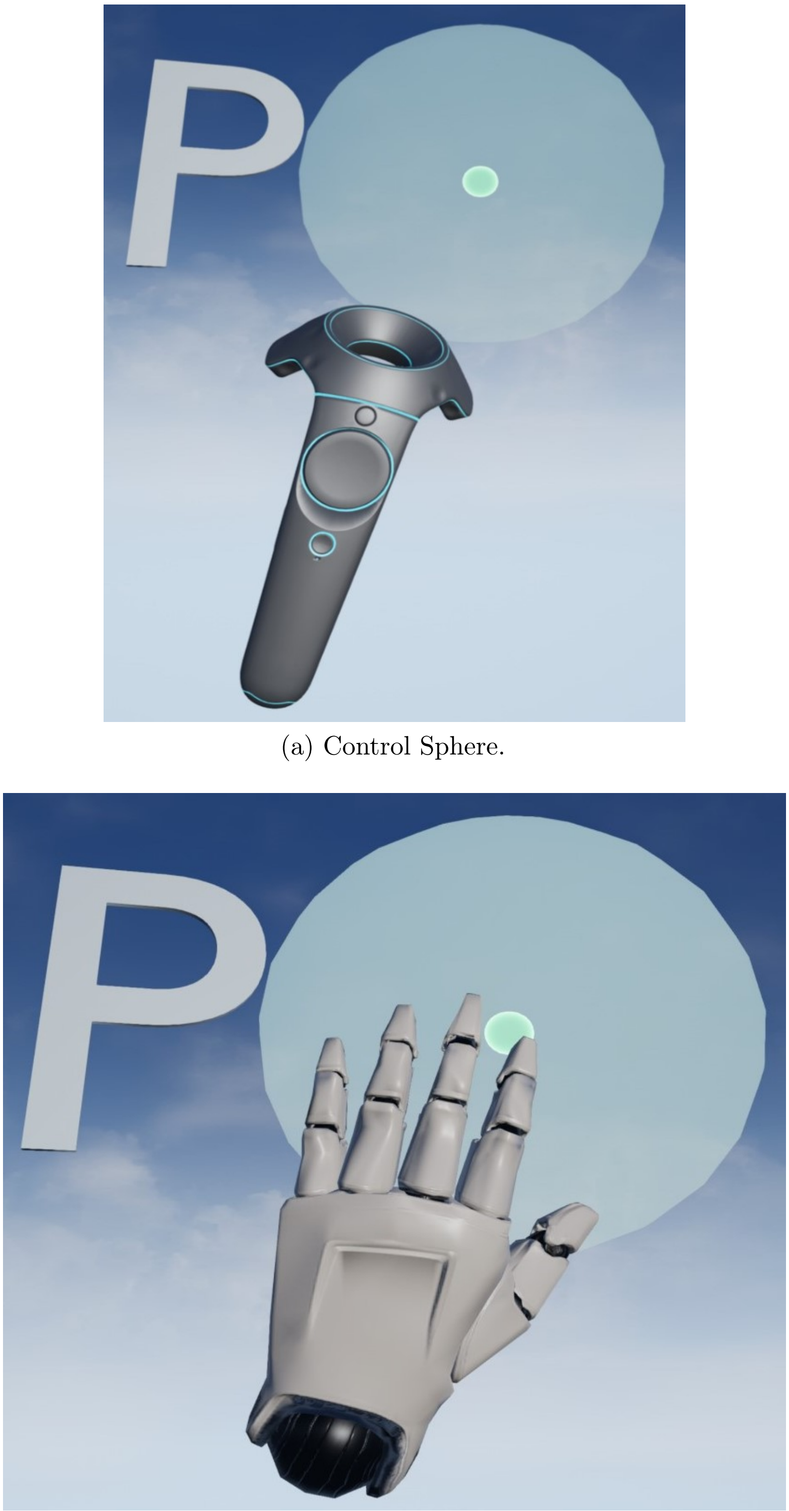
Watching Float Return Value
0.000000

Figure 21. Set Point calculation.

5.4.5. Control spheres

Prior to moving further to MATLAB and Simulink solution, the functional base of Control Spheres needs to be described. Although they are directly related to system control, which is going to be covered later on, it is more reasonable to clarify the functionality of Control Spheres in the present section.

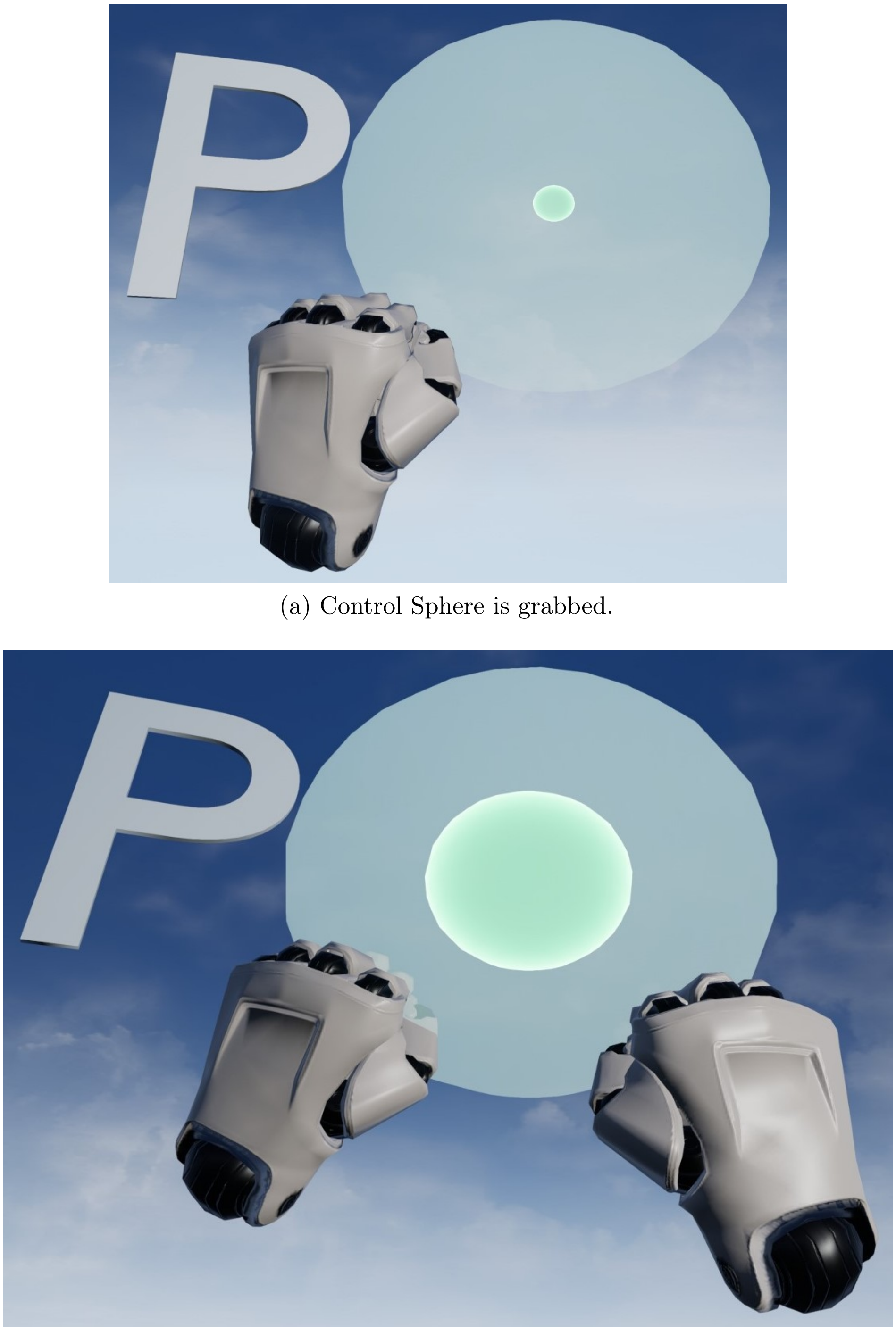
As it was mentioned previously, Control Spheres represent an intuitive way to adjust control parameters of the system, namely the proportional gain K_p and the integral gain K_i, by "stretching" movable spheres with both motion controllers. To achieve this a CollisionSphere is attached to the left controller. Every frame a condition whether CollisionSphere overlaps the Control Sphere is checked. If the condition is true, the controller model changes to "palm" and the Control Sphere can be grabbed and resized.



(a) Control Sphere.

(b) CollisionSphere on controller overlaps the Control Sphere, controller model changes to “palm” and the sphere can be grabbed.

Figure 22. Overlapping the Control Sphere with a controller.



(a) Control Sphere is grabbed.

(b) Control Sphere is resized.

Figure 23. Control Sphere grabbing and resizing.

In general sense, Control Spheres consists of two separate spheres centered at one point: outer sphere and inner sphere. Only the inner sphere is being resized while outer sphere remains the same. This is done intentionally to make the use of Control Spheres more intuitive and visualize the change of the parameter value and its limits in a convenient manner. The minimum parameter value is represented by the inner sphere reduced in size to the minimum limit while the maximum parameter value on the contrary is represented by the inner sphere increased in size to the maximum limit.

More precisely, each Control Sphere has an ID assigned to it (0 or 1) that corresponds to parameter K_p or K_i respectively. During the initialization the scale of the outer sphere is set to the value defined by the user, at the same time this value represents the max limit for the inner sphere since it is nested inside. Then, it is necessary to verify if user-defined value for minimum inner sphere scale value is less or equal to maximum inner scale value to prevent any incorrect behavior. In case the value is invalid, it is set by default to be equal to 90% of outer sphere scale. Once the validation is complete, both spheres are scaled according to initial values and the scale of the inner sphere is interpolated for the first time to prevent an undefined parameter from being sent to MATLAB. Thus, the initialization of the Control Sphere is complete.

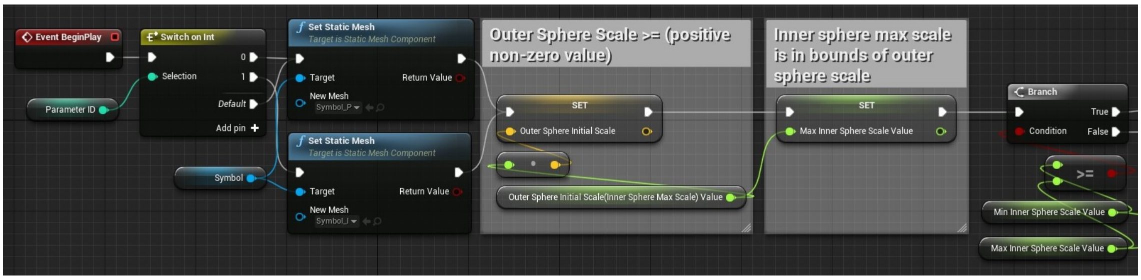

Event BeginPlay
Switch on Int
Selection
0
1
Default
Add pin
Parameter ID
Symbol
Set Static Mesh
Target is Static Mesh Component
Target
Return Value
New Mesh
Symbol_P
Symbol_I
Outer Sphere Scale >= (positive non-zero value)
SET
Outer Sphere Initial Scale
Outer Sphere Initial Scale(Inner Sphere Max Scale) Value
Inner sphere max scale is in bounds of outer sphere scale
Max Inner Sphere Scale Value
Branch
Condition
True
False
>=
Min Inner Sphere Scale Value
Max Inner Sphere Scale Value

Figure 24. Control Sphere initialization 1.

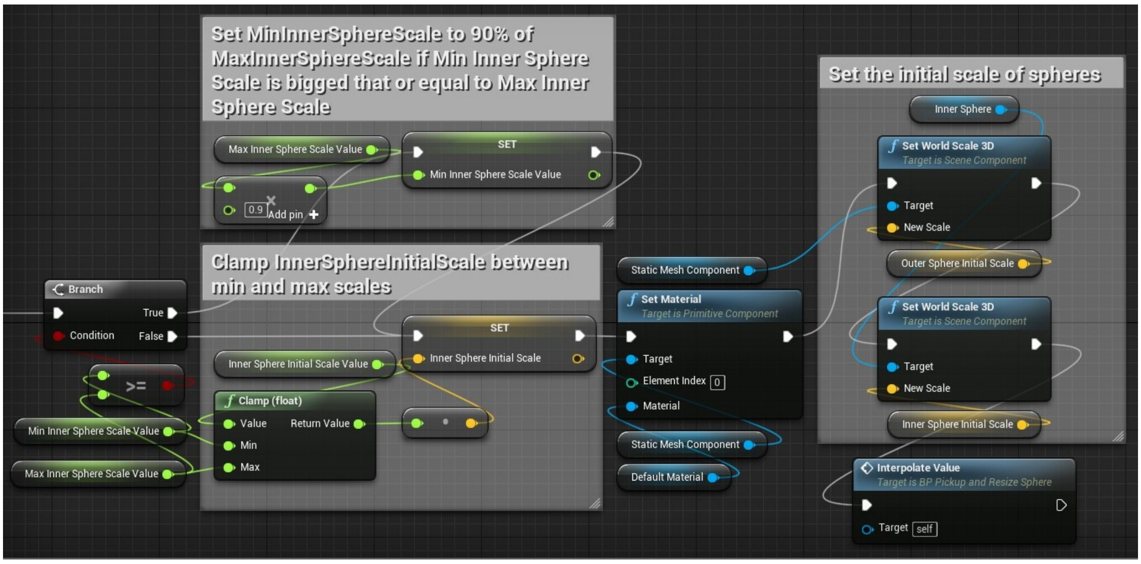

Set MinInnerSphereScale to 90% of MaxInnerSphereScale if Min Inner Sphere Scale is bigged that or equal to Max Inner Sphere Scale
Max Inner Sphere Scale Value
SET
Min Inner Sphere Scale Value
0.9
Add pin
Clamp InnerSphereInitialScale between min and max scales
Inner Sphere Initial Scale Value
Inner Sphere Initial Scale
Clamp (float)
Value
Return Value
Min
Max
Branch
Condition
True
False
>=
Min Inner Sphere Scale Value
Max Inner Sphere Scale Value
Static Mesh Component
Set Material
Target is Primitive Component
Target
Element Index
0
Material
Default Material
Set the initial scale of spheres
Inner Sphere
Set World Scale 3D
Target is Scene Component
Target
New Scale
Outer Sphere Initial Scale
Inner Sphere Initial Scale
Interpolate Value
Target is BP Pickup and Resize Sphere
Target
self

Figure 25. Control Sphere initialization 2.

Onward, once the Control Sphere starts being resized, sphere scale and the distance between the left and the right controller is saved.

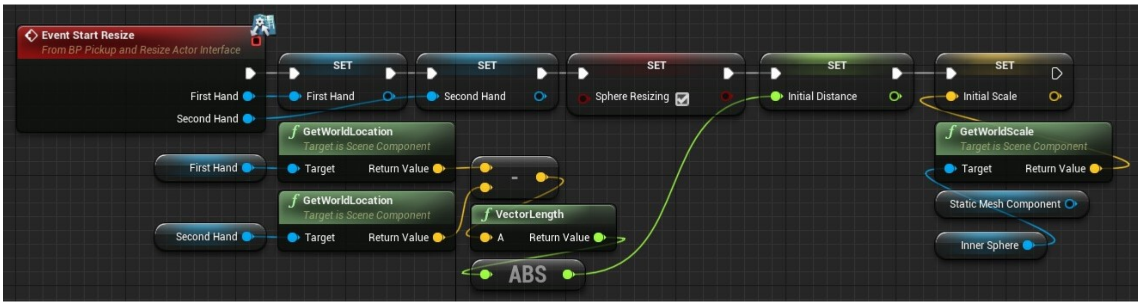


Figure 26. Control Sphere is grabbed and starts being resized.

Every frame new distance between controllers is calculated and divided by the distance acquired in the previous step, thus producing a proportion. It is greater than 1 if controllers are spread and lesser than 1 if moved towards each other. At this point new sphere scale is ready to be calculated. Old scale is multiplied by the proportion and as a result of these calculations a scale vector is acquired which is then split into component values. Each value is then clamped between minimum and maximum inner sphere scale values to keep it within permitted range. Once the values have been clamped, a new scale vector that is composed of these values is made and the inner sphere is resized. At this point the only step left is to interpolate the scale value once again. The interpolation is performed using the same formula as in the Section 5.4.4 and the interpolation range is also user-defined. As a result, depending on the Control Sphere ID, a corresponding variable (SphereScaleToBeSentSphere1 or SphereScaleToBeSentSphere2) is assigned the interpolated value and is ready to be sent to MATLAB.

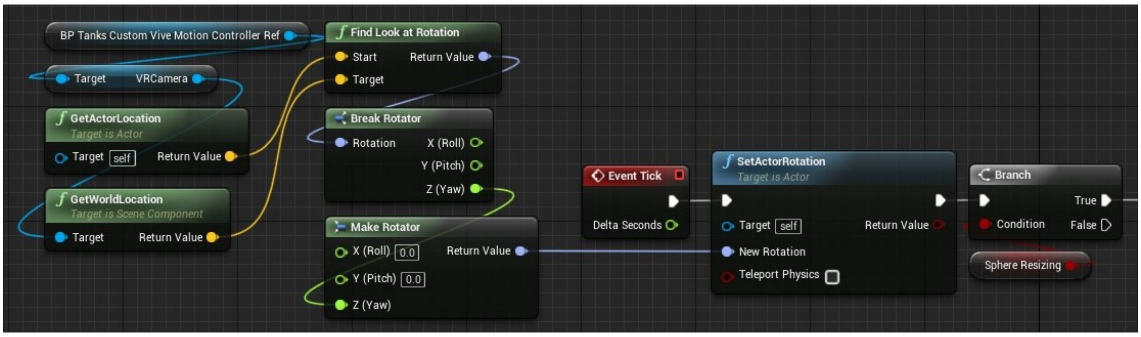


Figure 27. Control Sphere Tick function 1.

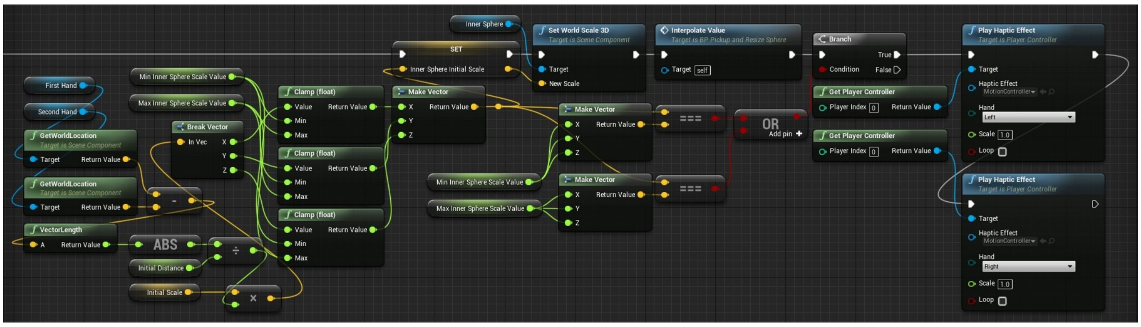


Figure 28. Control Sphere Tick function 2.

In addition, to make user-experience more flexible and intuitive some additional functionality is implemented. More precisely, it is possible to customize the working environment to a certain extent by configuring colors and size of the spheres. Also, the letters “P” and “I” that are located next to the Control Sphere, are set to always face the user regardless of location and rotation.

5.5. HMD

The VR HMD used in this project is HTC Vive Pro. Being equipped with front-facing cameras, it allows to use a see-through module, enabling the user to simultaneously see both the real environment and the digital twin. This is achieved with the use of SRWorks which is an SDK (Source Development Kit) developed by VIVE Software team. The SDK includes support for native development with plugin for Unreal Engine 4, making it easy to install and use [37].

Thus, a more immersive experience is created providing the user with the ability to seamlessly affect the real object through the digital twin and instantly see the result of these interactions.

6. Data transmission between UE4 and Simulink

Information exchange between the digital twin and the real object is an essential component of any digital twin solution. Without it any real-time communication or interactions such as system control, sensor tracking, remote control etc. become impossible.

In the present thesis, the communication between the DT and the real object is established via UDP (User Datagram Protocol). Both UE4 solution and Simulink model send and receive data on specified ports and process this data on their side afterwards.

6.1. UDP protocol, data transfer between UE4 and MATLAB

Development in Unreal Engine 4 supports implementation of functionality through C++ code, Blueprints or a combination of both. The latter is called C++/Blueprint mixed development and is a flexible tool that allows, for example, classes, variables and functions implemented in C++ to be exposed to Blueprints. First of all, corresponding macros, for example, UFUNCTION and UPROPERTY are added to the line above the function/property declaration in the header file to make them recognizable by the UE4 reflection system. Then, the macro will support Specifiers that can be used to expose UFunction and UProperty to Blueprint graphs, which provide a way for developers to call or extend UFunctions and Uproperties from Blueprint Assets without having to alter C++ code [36].

There are two C++ classes: UDPSender and UDPReceiver that handle sending and receiving data respectively, both of which use a UCLASS macro to indicate that they are reflected. In UE4 Editor, two blueprint classes that inherit from UDPSender and UDPReceiver (with similar names) are created. They are going to act as utility classes that other elements of UE4 solution will refer to in order to send already processed data to MATLAB or receive data and process it afterwards to affect the behavior of the digital twin. Both classes have functions and properties that are exposed to blueprints. For instance, functions GetData() in UDPReceiver and UDPSendArray() in UDPSender are exposed to blueprints using the specifier BlueprintCallable which means that these functions can be executed in a Blueprint or Level Blueprint graph.

```
UFUNCTION(BlueprintCallable, Category = "UDPCommunication")
FUDPData GetData();
```

Figure 29. C++ GetData() function.

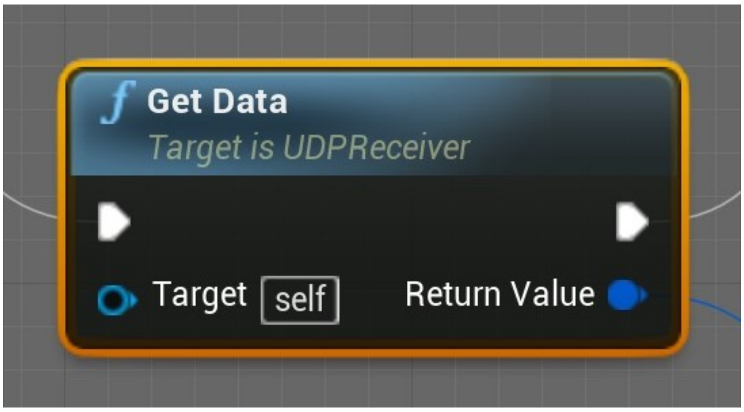


Figure 30. GetData blueprint function.

```
UFUNCTION(BlueprintCallable, Category = "UDPCommunication")
bool UDPSendArray(FUDPData data);
```

Figure 31. C++ UDPSendArray() function.

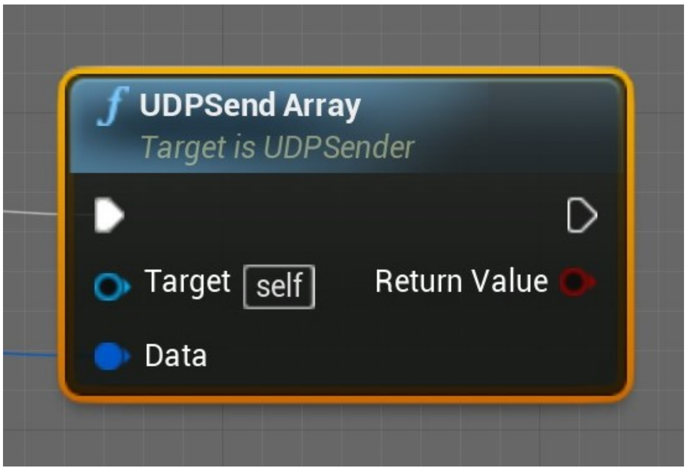


Figure 32. UDPSendArray blueprint function.

6.1.1. UDPReceiver

Once the UDPReceiver is initialized, it is ready to receive the values of liquid level in real tanks every frame. First of all, locations of lowest points of each tank are obtained. They are going to be used as starting points, relative to which the BPHider object is going to move up/down to change the level of liquid in the digital tank. Then, the data from MATLAB is acquired using UDP C++ functions that are exposed to the blueprint. The data received has the form of float values that correspond to level of liquid in real tanks. Each of these values is passed to the Tank Level Change function as an argument for further processing.

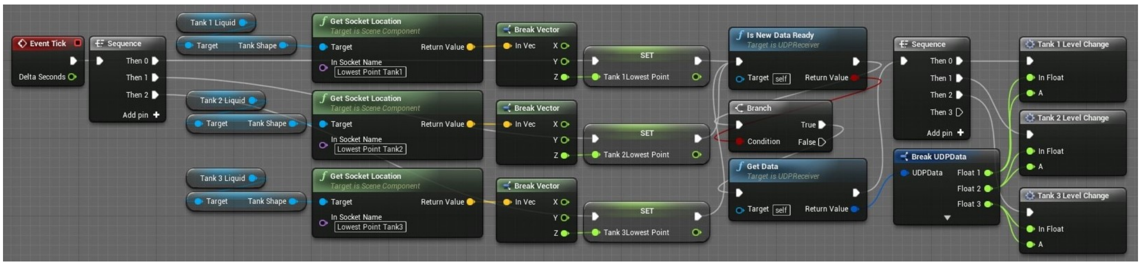


Figure 33. UDPReceiver.

In Tank Level Change function a new location for BPHider is calculated depending on the received value. This value first has to be multiplied by a Value Multiplier Const in order to convert it to centimeters (which is a default unit in UE4). Thus, the value that indicates how much the BPHider has to be raised relative to the lowest point of tank is calculated. The BPHider is then moved up-down depending on the calculated value. Thus, the level of liquid in the digital tank is changed.

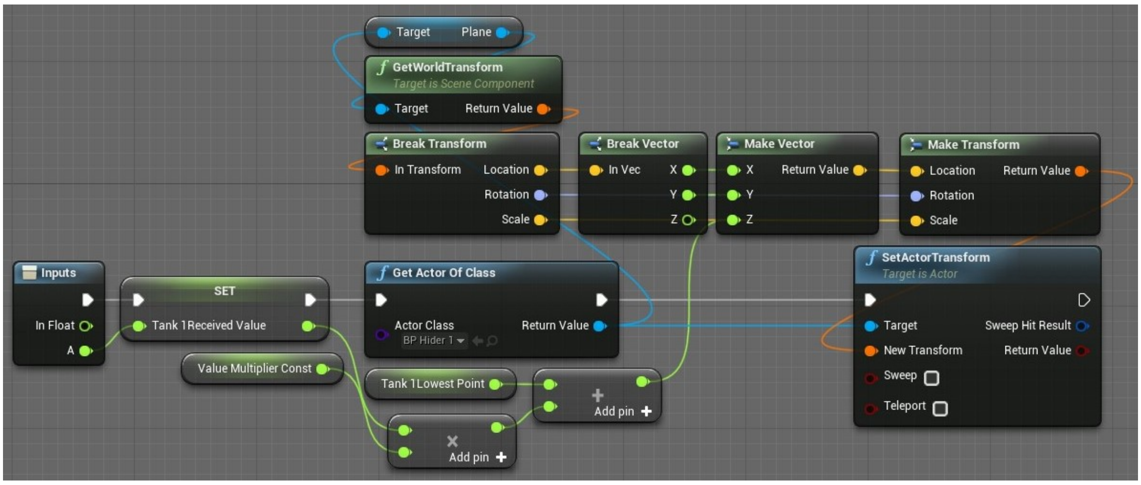


Figure 34. Tank Level Change function.

6.1.2. UDPSender

Once the UDPSender is initialized, it is ready to send the values calculated in the previous sections to MATLAB using UDP C++ functions that are exposed to the blueprint. More precisely, custom set points Tank1H, Tank2H, Tank3H and custom K_p and K_i values are sent to MATLAB.

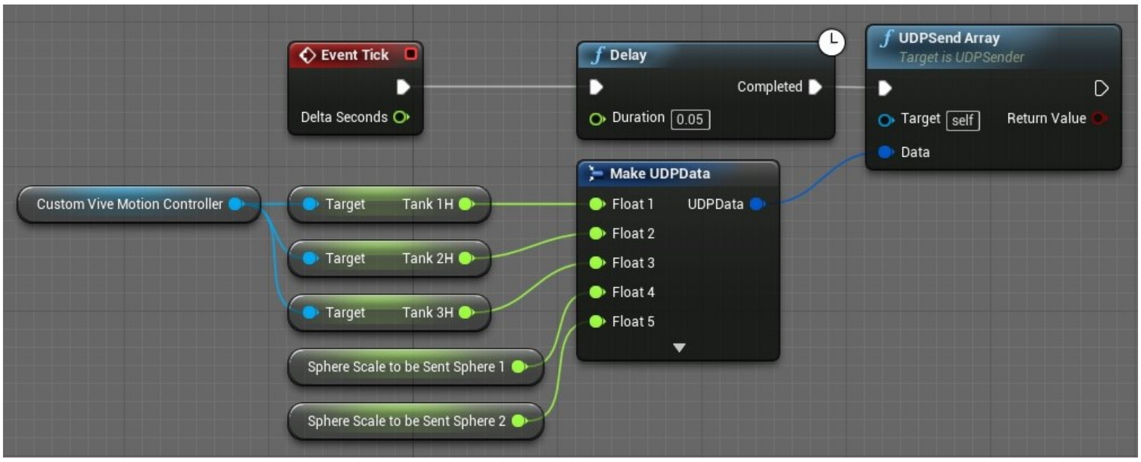


Figure 35. UDPSender.

7. Data visualization

Since there are many parameters and values that are being constantly changed, it is essential to provide the user with visualization of this data. A reasonable way to achieve this is to use a widget chart that is being updated in real-time.

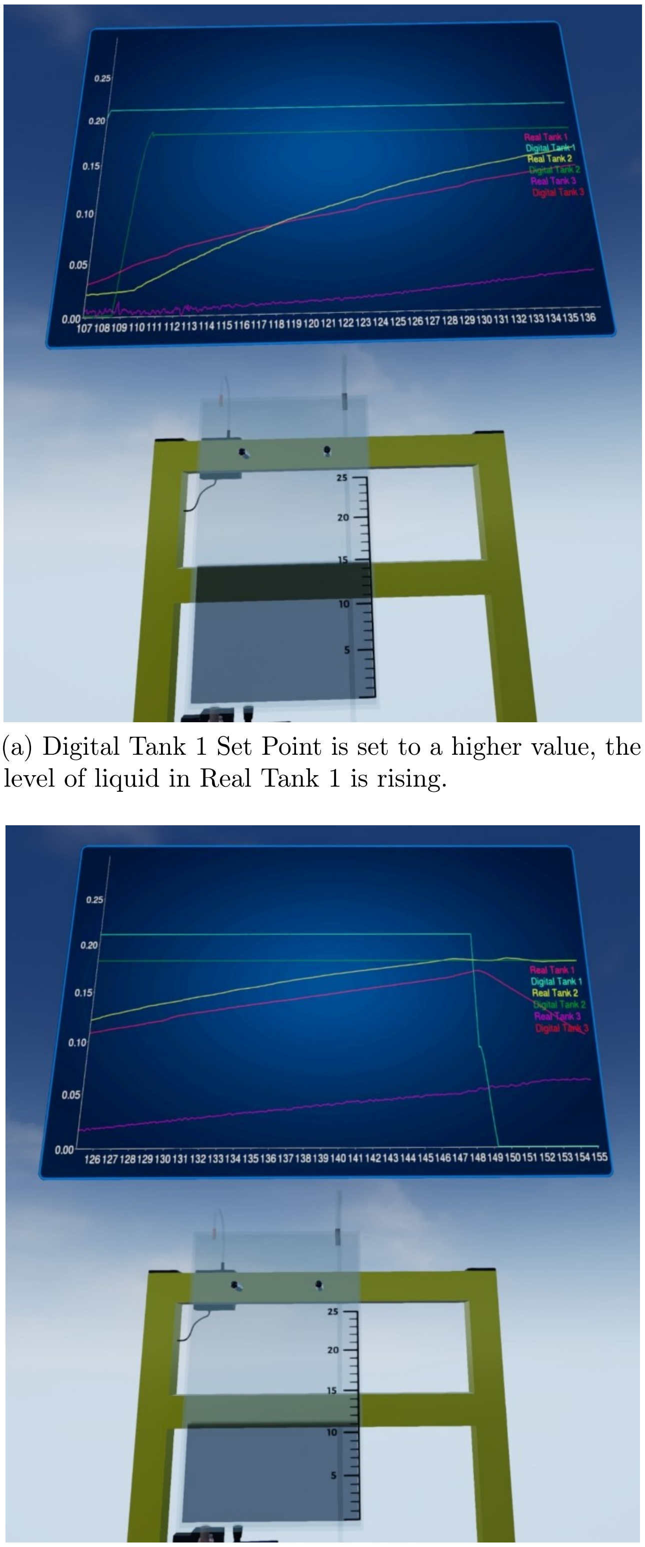
7.1. KantanCharts

In the present solution a KantanCharts plugin is used which is a package of UMG (Unreal Motion Graphics) chart widgets fully integrated with the UMG widget designer. It is possible to configure chart display properties and then add the data to be displayed via Blueprints or C++ as preferred [38].

The widget is located at the top of the Digital Twin and is attached to it. There are 6 values that are being displayed on the chart:

- Real Tank 1
- Digital Tank 1
- Real Tank 2
- Digital Tank 2
- Real Tank 3
- Digital Tank 3

Real values show levels of liquid in real tanks while Digital values show Set Points defined by the user. Since the data is displayed in real-time and all values are shown simultaneously, it is convenient to monitor whether real values correspond to Set Points.



(a) Digital Tank 1 Set Point is set to a higher value, the level of liquid in Real Tank 1 is rising.

(b) Digital Tank Set Point 1 is set to a lower value, the level of liquid in Real Tank 1 is decreasing.

Figure 36. Real-time data visualization.

Every frame a new data point that corresponds to a passed value is drawn to the chart which makes the chart update in real-time. Values for Real data points are acquired from MATLAB via UDPReceiver while values for Digital data points are Tank1H, Tank2H and Tank3H calculated in Section 5.4.4.

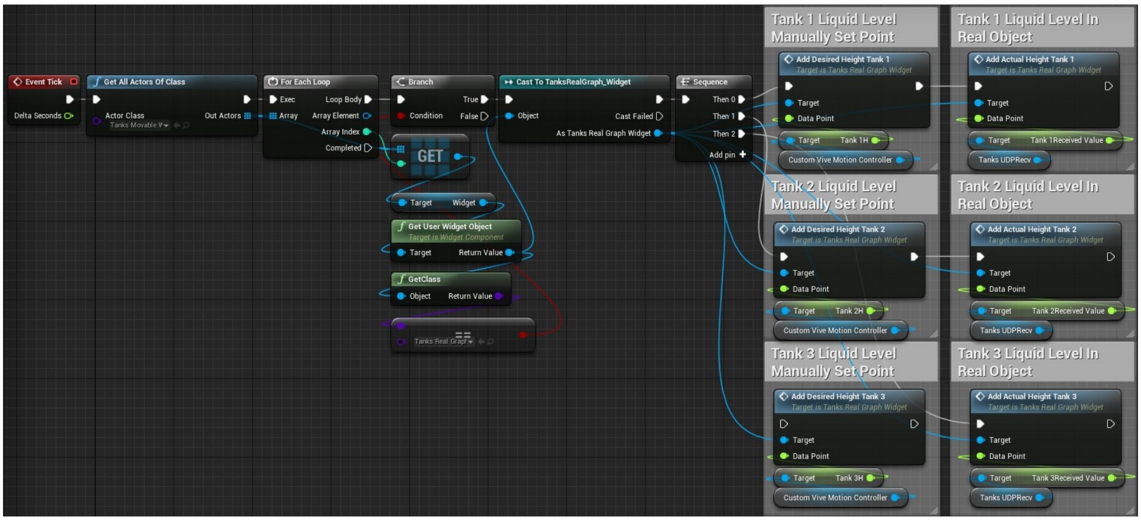

Event Tick
Delta Seconds
Get All Actors Of Class
Actor Class
Tanks Movable V
Out Actors
For Each Loop
Exec
Loop Body
Array
Array Element
Array Index
Completed
Branch
Condition
True
False
Cast To TanksRealGraph_Widget
Object
Cast Failed
As Tanks Real Graph Widget
Sequence
Then 0
Then 1
Then 2
Add pin
GET
Target
Widget
Get User Widget Object
Target is Widget Component
Target
Return Value
GetClass
Object
Return Value
Tanks Real Graph
Tank 1 Liquid Level Manually Set Point
Add Desired Height Tank 1
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 1H
Custom Vive Motion Controller
Tank 1 Liquid Level In Real Object
Add Actual Height Tank 1
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 1Received Value
Tanks UDPRecv
Tank 2 Liquid Level Manually Set Point
Add Desired Height Tank 2
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 2H
Custom Vive Motion Controller
Tank 2 Liquid Level In Real Object
Add Actual Height Tank 2
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 2Received Value
Tanks UDPRecv
Tank 3 Liquid Level Manually Set Point
Add Desired Height Tank 3
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 3H
Custom Vive Motion Controller
Tank 3 Liquid Level In Real Object
Add Actual Height Tank 3
Target is Tanks Real Graph Widget
Target
Data Point
Target
Tank 3Received Value
Tanks UDPRecv

Figure 37. Chart is filled with data.

8. MATLAB/Simulink solution

In many natural sciences and engineering disciplines such term as mathematical modeling is involved. It represents a process of developing mathematical models - a description of a system using mathematical concepts and language to facilitate proper explanation of a system or to study the effects of different components and to make predictions on patterns of behaviour [39].

Nowadays, a considerable number of professional software that supports the creation of mathematical models of different structure and complexity has been developed. Among them is MATLAB—a programming platform developed by MathWorks that is designed specifically for engineers and scientists [40]. At the heart of it is the proprietary MATLAB language, which is a matrix-based language that allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python [40,41]. Embedded with MATLAB, Simulink is a block diagram environment for multi-domain simulation and Model-Based Design that provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. When MATLAB and Simulink are used together, textual and graphical programming are combined to design systems in a simulation environment [42].

In the present thesis MATLAB and Simulink environment is used for the implementation of control logic and an interface to the real object. A non-linear mathematical model is created in Simulink and used in order to define and control the behavior of the multi-tank hydraulic system.

When speaking about real-life control systems in the context of modeling and identification, two situations are typically encountered [2]:

1. Most frequently, there is no model of the controlled plant, so the controllers are either auto-tuned, or tuned manually following very simple tuning rules and operator intuition.

2. Less frequently, a model of the plant dynamics exists, which allows to employ, e.g., model based control design.

In the present work, the second case is dealt with, which is a crucial factor since having a coherent model of the plant dynamics is essential in case of training applications development [2].

8.1. System control

In terms of system control, a classical PI controller is used to control the given plant. In particular, the parallel form of the PI controller which operates on the negative unity feedback-produced error signal

$$e(t) = y_{SP} - y_{real},$$

where y_{SP} is the set point and y_{real} is the controlled plant output, to produce a control law $u(t)$ that in the time domain has the form

$$u(t) = K_p e(t) + K_i \int_0^\tau e(t)\mathrm{d}t,$$

where K_p is the proportional gain, and K_i is the integral gain is considered. In industrial applications, K_p and K_i are the usual "tuning knobs"—appropriate values of these gains ensure stable and efficient operation of the control system. Unfortunately, in many cases the tuning of the PI controller is far from an optimal one. Thus, learning the procedure of tuning these gains to yield adequate control performance is essential in control engineering practice to improve tracking performance, safety, efficiency and reliability of control systems and also to reduce wasted energy which occurs due to improper behavior of the control law [2].

8.2. Experiment

In order to consolidate the developed method, a real-time experiment that includes two-way Digital Twin–real object communication and focuses on the control of the first tank of the multi-tank system is conducted. This section provides a description of the aforementioned experiment as well as a consequent overview of Simulink model segment related to the control of the first tank and data transmission between Simulink and UE4.

First of all, in order to perform real-time hardware-in-the-loop experiments, the Simulink model has to be connected to a real plant. This is achieved with the Real-Time Windows Target, which is a PC solution that provides a real-time kernel for executing Simulink models on a Windows or Mac computers and includes library blocks that connect to a range of I/O devices [42].

Packet Input block is used to read binary data or a CAN (Controller Area Network) message from a communication channel [42]. In the present case, data is being received from UE4 via UDP. More precisely, values of set points (Tank1H, Tank2H and Tank3H) calculated in Section 5.4.4 and control parameter values (SphereScaleToBeSentSphere1 and SphereScaleToBeSentSphere2) calculated in Section 5.4.5 are received.

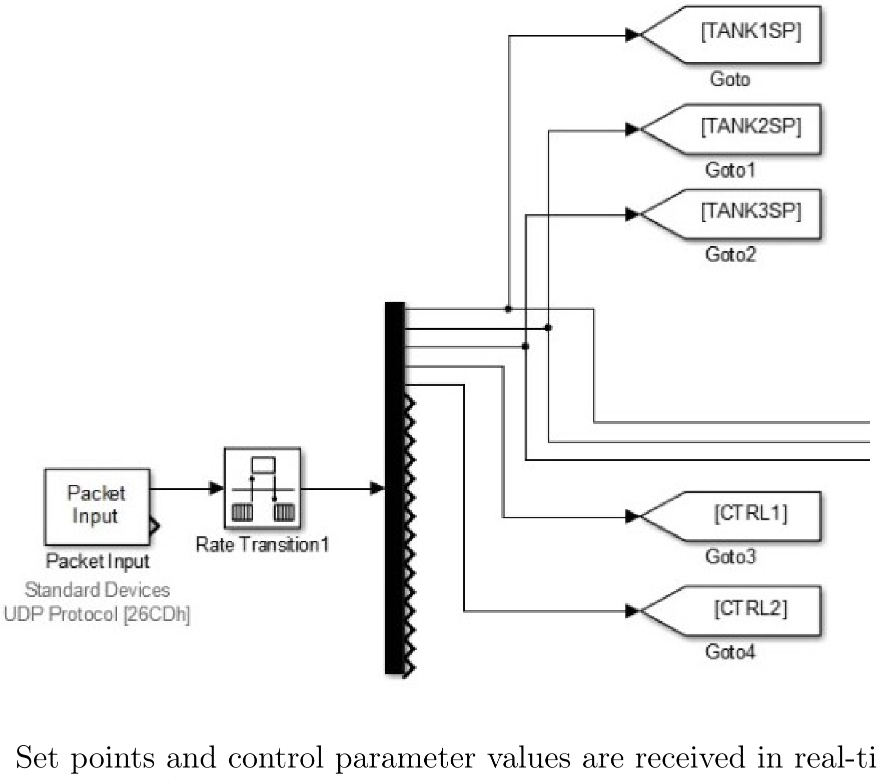


Figure 38. Set points and control parameter values are received in real-time from UE4 via UDP protocol.

Then, error signal $e(t)$, set point and K_p, K_i parameters are passed to the PI controller as inputs. The controller produces two outputs: control law value $u(t)$ and an reset integrator. The latter is obtained in order to discard the accumulated error (which is useful when conduction multiple experiments in a row).

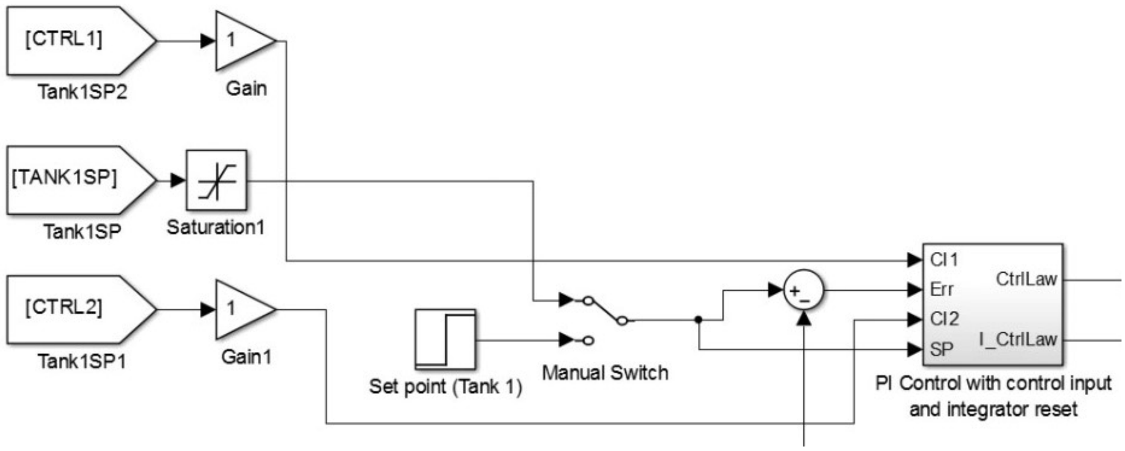


Figure 39. Error signal, set point and control parameters are passed to PI controller as inputs.

Afterwards, the control law value $u(t)$ is passed to two different blocks: the software driver of the real object as the first input, which corresponds to pump engagement proportion, and as one of two inputs to the Kalman filter—an algorithm which uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone [43]. The second input to the Kalman filter is the value that needs to be “filtered” which is, in this case, the level of liquid in the first tank. The “filtered” value is then subtracted from the set point value to produce a new error signal and also is passed to the Packet Output block that works similarly to Packet Input to be sent to UE4.

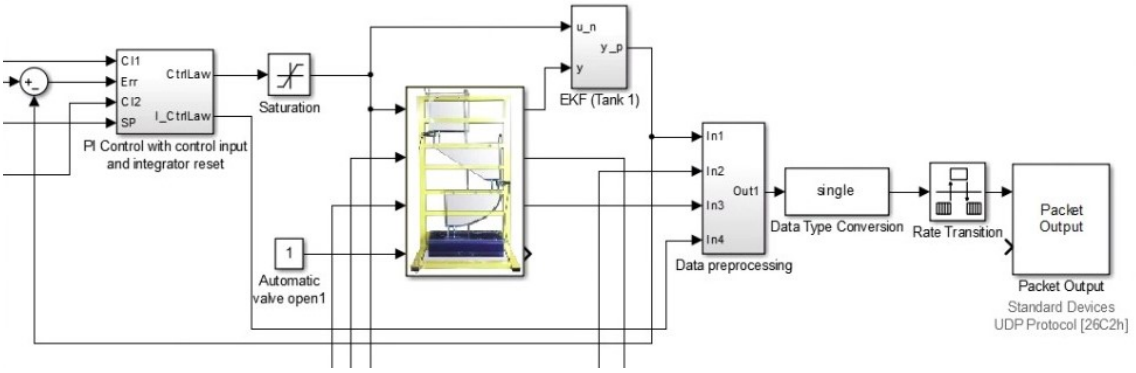


Figure 40. Control law value is passed to the software driver, the level of liquid in the tank is filtered and sent to UE4 and a new error signal is calculated.

Thus, the description of Simulink model is provided. The following is the description of the experiment, its related data and analysis of the results.

The given experiment is focused on the control of the first tank of the multi-tank system. The schematic diagram depicting the first two out of three tanks and their interconnection is shown in the figure below.

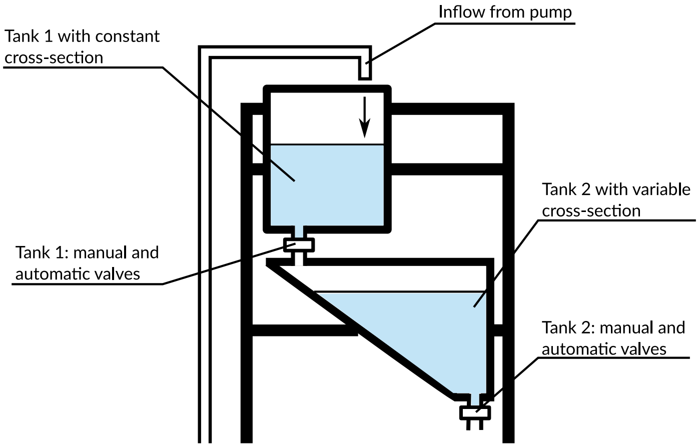

Inflow from pump
Tank 1 with constant cross-section
Tank 2 with variable cross-section
Tank 1: manual and automatic valves
Tank 2: manual and automatic valves

Figure 41. Configuration of the first two tanks of the multi-tank system.

The paper [2] describes the process of choosing metrics for identifying the range of PI controller parameters based on simulation. The cost functions are obtained by simulating the nonlinear model of the first tank across the PID controller gain range $K_p \in [0, 10] \wedge K_i \in [0, 5]$ considering the set point of 0.15m. The comparison of obtained functions allows to identify the region of acceptable performance which is formed if parameters K_p and K_i belong to ranges $[0, 10]$ and $[0, 0.5]$ respectively, thus providing theoretical predictions of system behavior depending on the values of the controller gains. Three points of the form (k_p, k_i) that belong to the identified region are then chosen. During the experiment the values of controller gains that correspond to these points will be used.

Table 1. Chosen Points.

	k_p	k_i
Point 1	2.0	0.25
Point 2	6.0	0.5
Point 3	6.0	0.25

The real-time experiment is then conducted. The XR user, wearing the HMD to observe and manipulate the perceived environment, sets up the digital twin and proceeds to tune the gains of the PI controller while performing step experiments $SP = 0.00 \rightarrow 0.15$ in a single attempt lasting 500 seconds. With each new attempt, the level in the tank is reset to zero, and integral component state is also reset.

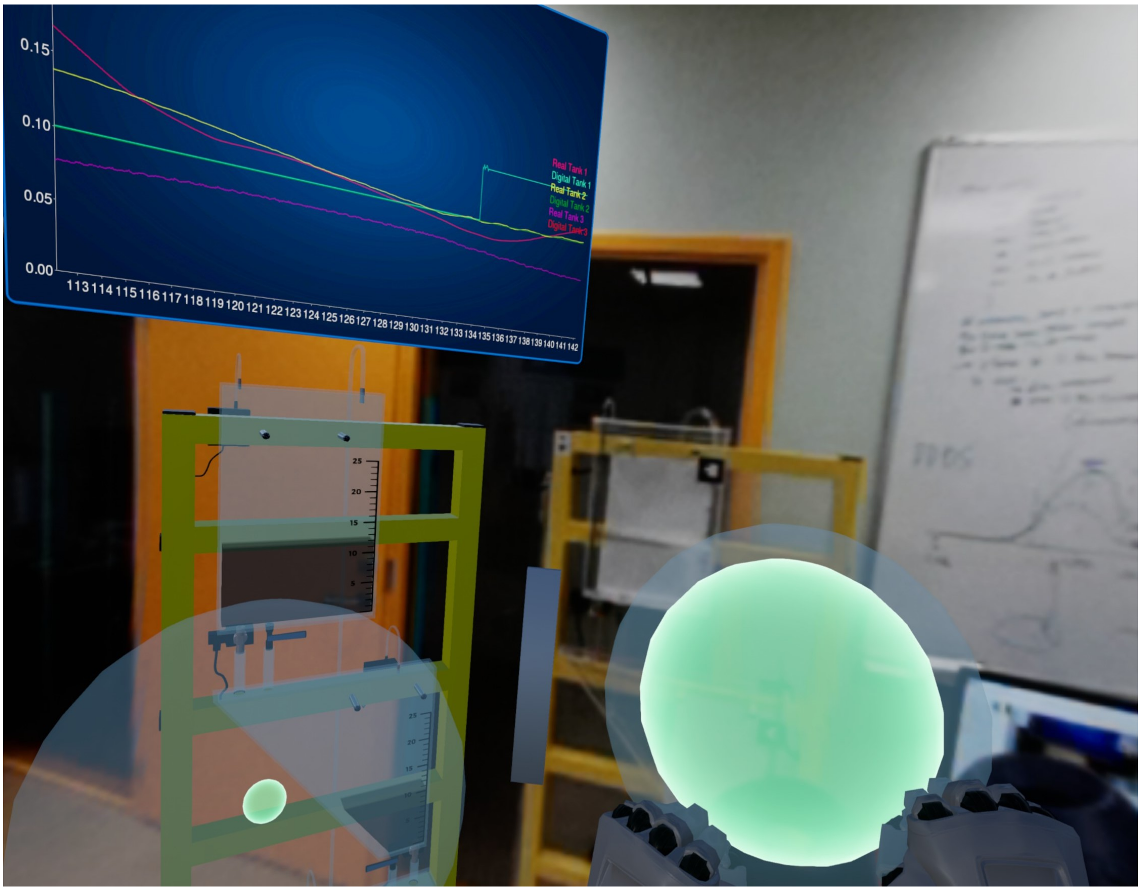


Figure 42. The XR environment featuring the digital twin, real object, and the controller gain setting spheres. The user is changing the size of the I sphere effectively updating the K_i gain setting of the PI controller in Simulink.

During the first attempt it can be clearly seen that there is a considerable overshoot when the level of liquid is being raised and also a noticeable undershoot when it decreases. Therefore, the given controller tune-up is not suitable for efficient system control.

During the second attempt values of both K_p and K_i controller gains are increased and are equal to 6.0 and 0.5 respectively which results in a slightly smaller overshoot, decreased time during which the overshoot is eliminated and a completely eliminated undershoot.

During the third attempt the K_p does not change, whereas the K_i is half as small as in the previous attempt which results in an almost completely eliminated overshoot.

The given controller tune-up is the most efficient of all three attempts.

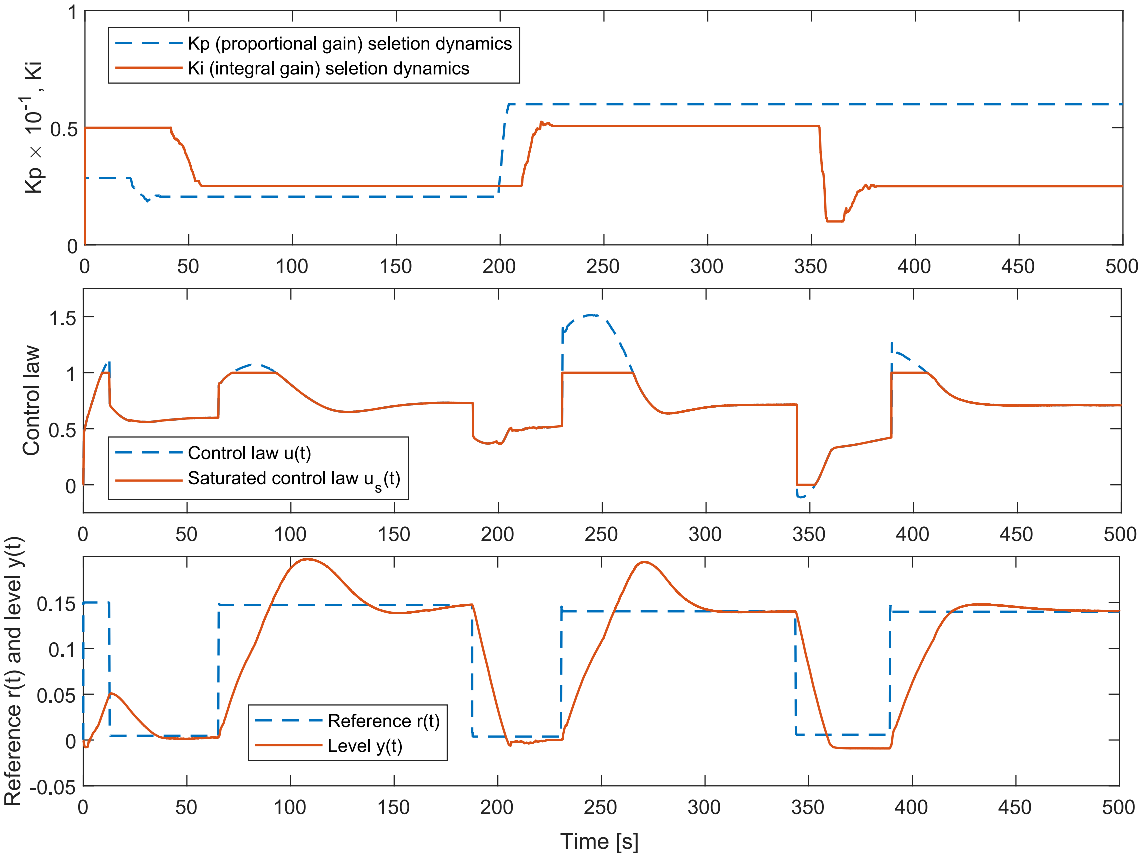


Figure 43. Results of three consecutive step experiments with different PI controller gains.

The results of the experiment show that theoretical predictions are confirmed with a practical experiment which means this type of visualization can be used for creating effective training applications and also for advanced SCADA system prototyping.

9. Conclusions

At the beginning of the present thesis specific objectives were defined. The main pursued goal was to develop a pilot prototype, which, after being tested to improve the quality of training on real examples at the university, could also be transferred to the industrial context and used for training engineers of automatic control systems, thus facilitating industrial use of XR and DT.

The achievement of stated goal required the following problems to be solved:

- A literature overview had to be performed, since any technical thesis should draw from previous R&D results.
- A 3D digital representation of the real object had to be created.
- Real-time system control using PI controller had to be implemented.
- Real-time change of control parameters and subsequent change of system behavior had to be implemented.
- The validity of system control had to be verified by comparing theoretical predictions and practical results.
- Two-way communication between the real object and the DT had to be implemented.
- The ability to influence the real object through interactions with the DT (setting custom set points) had to be implemented.
- Data visualization in an intuitive way had to be implemented.

Each aforementioned problem has been successfully solved.

Throughout this work, all the necessary steps involved into creating a pilot prototype have been covered. More precisely, a comprehensive literature overview has been conducted, thus forming the necessary base, enabling further research and development.

The creation of a 3D model and implementation of all the functional aspects of the digital twin in Unreal Engine 4 has been thoroughly covered. Not only the principle of work of all functional components like setting custom set points, moving the digital twin in the perceived environment, displaying the level of liquid in digital tanks etc. has been covered, but also the mechanics this functionality stands on have been explained in detail.

Real-time interactions along with intuitive system control also have been successfully implemented, which is indicated by satisfactory results of comparison of theoretical expectations and actual performance.

As a result, an immersive environment for a novel supervisory control and data acquisition based on the concept of digital twins in extended reality geared towards laboratory training and controller tuning has been successfully created. As for the future, the research involves subject based studies to improve UI elements making them more intuitive and informative along with porting given application to a professional-quality head-mounted device.

Acknowledgments

The author of the present thesis would like to exspress his deepest gratitude to his supervisor Dr. Aleksei Tepljakov for his support, responsiveness, advice and guidance throughout the whole process of working on the present thesis. This experience has been useful and fascinating in large part owing to mutual assistance, good work atmosphere and mutual respect.

The author would also like to express his sincerest gratitude to his loved ones for endless support, faith, patience and love. Even in moments of despair they helped me find strength to move on and without them finishing this project would have been much more difficult and exhausting, if not impossible.

References

[1] T. Reichenbach, G. Vasiljevic, and Z. Kovacic, "Virtual reality control systems," *Advanced Knowledge Application in Practice*, 2010.

[2] S. Jersov and A. Tepljakov, "Digital Twins in Extended Reality for Control System Applications," *43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, accepted for presentation.

[3] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[4] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. V. den Hof, "Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges," *Annual Reviews in Control*, vol. 43, pp. 1–64, 2017.

[5] "Reality - virtuality continuum," *Wikipedia*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Reality%E2%80%93virtuality_continuum

[6] "Virtual reality," *Wikipedia*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Virtual_reality

[7] K. Irvine, "XR: VR, AR, MR - What's the difference?" 2017. [Online]. Available: https://www.viget.com/articles/xr-vr-ar-mr-whats-the-difference/

[8] "Augmented Reality," *Wikipedia*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Augmented_reality

[9] "Mixed Reality," *Wikipedia*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Mixed_reality#Augmented_virtuality

[10] "Extended Reality," *Wikipedia*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Extended_reality

[11] M. Schluse, L. Atorf, and J. Rossmann, "Experimentable digital twins for model-based systems engineering and simulation-based development," in *2017 Annual IEEE International Systems Conference (SysCon)*. IEEE, 2017.

[12] J. Scheibmeir and Y. Malaiya, “An API development model for digital twins,” in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2019.

[13] P. Wu, M. Qi, L. Gao, W. Zou, Q. Miao, and L. lan Liu, “Research on the virtual reality synchronization of workshop digital twin,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. IEEE, 2019.

[14] T. Mikkonen, K.-K. Kemell, P. Kettunen, and P. Abrahamsson, “Exploring virtual reality as an integrated development environment for cyber-physical systems,” in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2019.

[15] U. Dahmen and J. Rossmann, “Experimentable digital twins for a modeling and simulation-based engineering approach,” in *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2018.

[16] V. Kuts, T. Otto, T. Tähemaa, and Y. Bondarenko, “Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality,” *Journal of Machine Engineering*, vol. 19, no. 1, pp. 128–144, 2019.

[17] V. Makarov, Y. Frolov, I. S. Parshina, and M. Ushakova, “The design concept of digital twin,” in *2019 Twelfth International Conference "Management of large-scale system development" (MLSD)*. IEEE, 2019.

[18] A. O’Dwyer, “Handbook of Pi and Pid Controller Tuning Rules,” 2009.

[19] N. D. Finkelstein, W. K. Adams, C. J. Keller, P. B. Kohl, K. K. Perkins, N. S. Podolefsky, S. Reid, and R. LeMaster, “When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment,” *Physical Review Special Topics - Physics Education Research*, vol. 1, no. 1, 2005.

[20] S. B. Nolen and M. D. Koretsky, “Affordances of virtual and physical laboratory projects for instructional design impacts on student engagement,” *IEEE Transactions on Education*, pp. 1–8, 2018.

[21] J. Pirker, I. Lesjak, and C. Guetl, “Maroon VR: A room-scale physics laboratory experience,” in *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2017.

[22] M. J. Callaghan, K. McCusker, J. L. Losada, J. Harkin, and S. Wilson, “Using game-based learning in virtual worlds to teach electronic and electrical engineering,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 575–584, 2013.

[23] G. Sziebig, B. Takarics, and P. Korondi, “Control of an Embedded System via Internet,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3324–3333, 2010.

[24] B. Balamuralithara and P. C. Woods, “Virtual laboratories in engineering education: The simulation lab and remote lab,” *Computer Applications in Engineering Education*, vol. 17, no. 1, pp. 108–118, 2009.

[25] M. S. Y. Jong, J. Shang, F. Lee, and J. H. M. Lee, “An Evaluative Study on VISOLE - Virtual Interactive Student-Oriented Learning Environment,” *IEEE Transactions on Learning Technologies*, vol. 3, no. 4, pp. 307–318, 2010.

[26] M. Callaghan, K. McCusker, J. L. Losada, J. Harkin, and S.Wilson, “Integrating virtual worlds & virtual learning environments for online education,” *2009 International IEEE Consumer Electronics Society’s Games Innovations Conference*, pp. 54–63, 2009.

[27] T. de Jong, M. C. Linn, and Z. C. Zacharia, “Physical and virtual laboratories in science and engineering education,” *Science*, vol. 340, no. 6130, pp. 305–308, 2013.

[28] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, “Virtual laboratories for education in science, technology, and engineering: A review,” *Computers & Education*, vol. 95, pp. 309–327, 2016.

[29] S. Yuan, J. Hu, and X. Li, “Application of virtual reality technique to real-time control system,” in *2007 2nd IEEE Conference on Industrial Electronics and Applications*. IEEE, 2007.

[30] J. Bathelt and A. Jönsson, “How to implement the virtual machine concept using xPC target,” Swiss Federal Institute of Technology Zurich, Institute of Mechanics, Tech. Rep., 2003.

[31] A. Jönsson, J. Wall, and G. Broman, “A virtual machine concept for real-time simulation of machine tool dynamics,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 795–801, 2005.

[32] M. Bacidore, “Digital twin to enable asset optimization,” *Smart Industry*, 2015. [Online]. Available: https://www.smartindustry.com/articles/2015/asset-optimization-courtesy-of-digital-twin/

[33] A. E. Saddik, “Digital twins: The convergence of multimedia technologies,” *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, 2018.

[34] C. J. Parris, J. B. Laflen, M. L. Grabb, and D. M. Kalitan, “The Future for Industrial Services: The Digital Twin,” *Infosys Insights*, pp. 42–49, 2016. [Online]. Available: https://www.infosys.com/insights/iot/future-industrial-digital.html

[35] A. Kose, A. Tepljakov, S. Astapov, D. Draheim, E. Petlenkov, and K. Vassiljeva, “Towards a synesthesia laboratory: Real-time localization and visualization of a sound source for virtual reality applications,” *Journal of Communications Software and Systems*, vol. 14, no. 1, 2018.

[36] Epic Games, “Unreal Engine Documentation,” 2020, retrieved 14.05.2020. [Online]. Available: https://docs.unrealengine.com/

[37] HTC Corporation, “Simulink, Simulation and Model-Based Design,” 2020, retrieved 22.05.2020. [Online]. Available: https://www.mathworks.com/products/simulink.html

[38] C. Angus, “Kantan Charts,” 2020. [Online]. Available: http://kantandev.com/free/kantan-charts

[39] M. Abramowitz, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* New York: Dover Publications, 1965.

[40] The MathWorks, Inc., “MATLAB Computational Software Platform,” 2020, retrieved 22.05.2020. [Online]. Available: https://www.mathworks.com/products/matlab.html

[41] C. Underwood, *The MATLAB Handbook.* North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2016.

[42] The MathWorks, Inc., “Simulink, Simulation and Model-Based Design,” 2020, retrieved 22.05.2020. [Online]. Available: https://www.mathworks.com/products/simulink.html

[43] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng*, vol. 82, no. 1, pp. 35–45, 5/25. [Online]. Available: https://doi.org/10.1115/1.3662552

A. Appendix: Associated Publications

Reference

S. Jeršov and A. Tepljakov, “Digital Twins in Extended Reality for Control System Applications”. In Proc. of 2020 43rd International Conference on Telecommunications and Signal Processing (TSP) (Virtual conference originally to take place in Milan, Italy). July 7-9, 2020. Accepted for presentation.

Abstract

In this paper, we investigate potential applications of digital twins (DT) in extended reality (XR) to control systems. Specifically, we are interested in facilitating industrial use of XR and DT by initially implementing training applications that can be used in the classroom and then moving into the development of supervisory control and data acquisition (SCADA) applications suitable for industrial use. A comprehensive literature review is performed providing the basis and motivation for this research endeavor. In the pilot study, we use a real-life laboratory model of a multi-tank system as a use case. We develop a digital twin for this control object in extended reality with all necessary interaction mechanics, and connect it to the real plant for performing real-time hardware-in-the-loop (HIL) experiments. The theoretical bases and the technical development are detailed in the paper, and results of an illustrative experiment are provided and commented upon.

Digital Twins in Extended Reality for Control System Applications

Stanislav Jeršov[†] and Aleksei Tepljakov[‡]

[†]Department of Software Science, Tallinn University of Technology, Tallinn, Estonia
[‡]Department of Computer Systems, Tallinn University of Technology, Tallinn, Estonia
Email (corresponding author): aleksei.tepljakov@taltech.ee

Abstract—**In this paper, we investigate potential applications of digital twins (DT) in extended reality (XR) to control systems. Specifically, we are interested in facilitating industrial use of XR and DT by initially implementing training applications that can be used in the classroom and then moving into the development of supervisory control and data acquisition (SCADA) applications suitable for industrial use. A comprehensive literature review is performed providing the basis and motivation for this research endeavor. In the pilot study, we use a real-life laboratory model of a multi-tank system as a use case. We develop a digital twin for this control object in extended reality with all necessary interaction mechanics, and connect it to the real plant for performing real-time hardware-in-the-loop (HIL) experiments. The theoretical bases and the technical development are detailed in the paper, and results of an illustrative experiment are provided and commented upon.**

Keywords—**digital twin; extended reality; control system; PID control; training; supervisory control and data acquisition; advanced user interface; industrial application**

I. Introduction

Over the last years there have been substantial advances in computer science and technology and, consequently, many areas of applications have benefited from their development and progress. Production design, manufacturing, education and engineering as a whole are capable of raising the efficiency and productivity to a new level due to technological progress in both hardware and software directions. An exponential growth in computational power gave rise to new methods, algorithms and theories. Unsolvable problems, or more precisely, formerly lengthy problems, became solvable in real-time. Thereby, a technology of creating copies of virtual systems and environments emerged—Virtual Reality (VR) [1].

Contrary to popular belief, VR is not being primarily used in the field of entertainment. Owing to its exceptional visualization capability and provision of immersive experiences and advanced interaction mechanics, VR has found applications in many forms. Among notable examples are virtual reality control systems, virtual laboratories and simulation/visualization of processes and products.

Given the fact that a lot of information about the surrounding world is received through vision, visualization of processes can be very useful and cognitive.

The concept of VR, together with the concept of augmented reality (AR) which allows digital objects to be superimposed over a real-life environment, give rise to the concept of extended reality (XR) in which both techniques are used.

On the other hand, digital realization of objects as both mathematical and three-dimensional models has given rise to the concept of *digital twins* which also have obvious industrial applications in many areas [2]. One such area is control systems with many open problems [3].

In this paper, we tie together digital twins, extended reality and control systems to create a novel application suitable for training purposes and for development of cutting-edge SCADA systems.

A. Contribution

The contribution of this paper is manifold. First, a comprehensive literature review concerning XR, digital twins, and control systems is provided. Second, a method is proposed that is suitable for both training applications in tuning PI controllers in XR with DT and also for implementing an advanced SCADA system. Third, the structure of the experimental system is described. Finally, real-time HIL experiments are conducted that illustrate the efficiency of the proposed approach.

B. Outline

The paper is organized as follows. In Sec. II, a literature review is provided forming the basis of our study. In Sec. III the proposed methodology is detailed. In Sec. IV, the experimental setup is described, necessary system parameters are computed using the proposed method, results of a real-time experiment are provided and discussed. In Sec. V, some items for discussion are given. Finally, in Sec. VI, conclusions are drawn.

II. Prior Art

In this section, we review research related to XR, digital twins, control systems, and virtual laboratories. This knowledge must guide further research and development efforts where the ultimate goal is to not only provide advanced XR training facilities, but also implement these techniques for developing novel SCADA systems. In other words, we start at laboratory

This research endeavor was partially supported by a starting research grant awarded to Dr. Aleksei Tepljakov in 2019 by School of Information Technologies, TalTech. It was also supported by the Estonian Research Council grant PRG658.

scale and gradually move in the direction of appropriate industrial applications that utilize cutting edge technology.

Over the last years virtual laboratories have made a significant contribution to education, because computer simulations that are properly designed are useful for a variety of contexts that can promote student learning [4]. In its most common instructional design, the virtual laboratory provides a simulation that mirrors a physical laboratory at the university [5]. In addition, it allows students to experiment with the physical phenomena in a safe environment and may provide students with information not available as well as inclusion of visual cues or alternative representations not possible to observe in physical laboratories (e.g. visualization of emergent molecular interactions) [5], [6].

Whereas many virtual laboratories are only accessible within the university, there are also remote experimentation laboratories (although not always supporting VR devices like Oculus or HTC Vive—a noteworthy development nevertheless) that are accessible as part of a web-based learning approach. Expedited by advances in web applications and related technologies, remote experimentation laboratories have evolved in recent years, setting the accurate recreation of on-campus laboratory-based student experience with an equivalent level of user access, functionality and flexibility as the end objective [7]. Such remote experimentation facilities afford several critical benefits, augmenting and complementing web-based learning material by facilitating distant access to based in campus physical resources [8]. This is consistent with the statement given in the paper [9], which is: "Simulation based labs cannot provide a 'feel' for real things. Students need to use real devices and execute commands on real tools to gain necessary practical skills.". For engineering related distance education courses the use of a web-based delivery mechanism is the only realistic method of providing practical hands-on experience, allowing remotely located students to complete laboratory assignments, unconstrained by time or geographical considerations while developing skills in the use of real systems and instrumentation [10], [11]. Presumably, the support for VR head mounted devices and controllers will be introduced in most remote experimentation laboratories in foreseeable future.

The paper [12] presents the description of virtual laboratory experiments in different conditions: physical only instruments, virtual only instruments and combination of both. Observation results show that in the combination condition students profited from comparing two potentially different representations of the same phenomena and using abstract reasoning to analyze the differences.

The paper [13] presents the virtual laboratory, where students can, among other experiments, participate in control system experiment which involves a Magnetic Levitator. The main concepts illustrated by this experiment are: PID controllers, root locus controller design and ITAE (Integral Time Absolute Error) optimal controller design.

Virtual reality control systems are broadly used in system prototyping, design and simulation providing such benefits as reduced costs of development, mitigation of nonconformity and failure risks and greatly reduced prototyping and design time [1]. With respect to control systems, it is beneficial to display the virtual world and the real world synchronously. Comparing the real system with its virtual representation allows to get a profound understanding of control effectiveness of the whole system [14].

The paper [15], with a view to analysis of the dynamic behavior of the water jet cutting machine, presents a virtual machine, which consists of a real machine control bi-directional connected to a machine simulation, which is bi-directional connected to 3D visualization of the machine. This setup enables realistic system simulations, since the simulation input comes from a real control. Following this approach, it is possible to test large numbers of variants and optimize the product with the aid of a minimum of physical prototypes. The paper [16] brings the virtual machine approach further. The concept facilitates total systems simulation, including the dynamics of the control and the electrical and mechanical parts of the machine. This makes it possible to assess systems design suggestions at an early stage of development and to perform systems optimization. Thereby, the virtual machine concept is moving towards becoming a useful tool for supporting machine tool systems development.

The combination of immersive visuals and wide functionality led to growing interest and development of Digital Twins (DT). This concept originates from the Industry 4.0 developments and describes a one-to-one representation of a real world artefact (such as machine, component or part of the environment) comprising its visual representation, its behavior and its communication facilities [17]. The paper [18] defines DTs as software implementations of their physical counterparts that act through application program interfaces (APIs) to the physical devices they monitor, engage with and possibly control. In paper [19] DTs are defined as integrated multi-physics, multi-scale, multi-disciplinary attributes with real-time synchronization, faithful mapping, high fidelity, and the ability to implement the technical means of interaction and integration between the real world and the information world. Combining this with appropriate simulation algorithms enables to execute the model, making the Digital Twin "experimentable". Therefore, it becomes possible to test and evaluate the performance of newly developed overall-systems under different boundary conditions, interactions between components and subsystems in various operational scenarios and intended modifications without making any changes to the actual system [20], [21]. DTs can be controlled and programmed in real-time from the computer and are, if necessary, capable of making operational changes in the real object [22], [23]. As a product of two-way communication between the digital and the real object, the physical twin can use knowledge from digital twins to improve its characteristics [23].

For instance, General Electric (GE), builds cloud-hosted digital twins of its machines that process information collected from sensors using artificial intelligence, physics-based models, and data analytics to better manage those machines [24], [25].

The paper [22] presents the development of the DT of industrial robot, which is made as a precise digital copy of the real object. Aside from the functionality of the real object (e.g. movement of moving parts) such valuable concept as collision detection is described. Given that the DT is an exact digital copy of the real robot, as an alternative to providing the real object with proximity sensors it is possible to use the DT's geometry to monitor the position of the robot in its workspace and prevent it from accidental collisions with the environment. This collision detection approach provides almost the same level of reliability but does not require any extra physical equipment or significant setup time.

In this way, many fields of applied sciences have benefited from technical progress and development of computer science. Technologies such as Extended Reality bring long existing developments to a whole new level. New research efforts are constantly in progress, thereby opening doors to new, more complex and powerful systems.

To give an example of our previous work related to digital twins, and, in part also to control systems in XR, the paper [26] presents the description of the project aimed at providing means for including voluntary synesthetic experiences (*synesthesia* is the act of experiencing one sense modality as another) through the VR environment. The application of an acoustic localization method to the problem of locating the sound source in a room is investigated. Afterwards, sounds are processed and turned into visual signals, in other words visualized. In this project, the Kalman filter (an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements) is applied in order to reduce noise generated by the uncertainty of sound source location prediction. This is an important addition, since The Kalman filter has numerous applications in technology, including guidance, navigation, control of vehicles, signal processing, system control and many others.

While this project was not directly related to control systems, its development provided necessary insights into immersive aspects of XR as well as interaction mechanics. In the present effort, this knowledge is fully leveraged to develop a digital twin of a control object, make it interactive, and provide facilities for the control system aspect, namely, process monitoring, control, and management. This is detailed in the following sections.

III. Methodology

In the field of control systems where extended reality is concerned, industrial partners are mostly interested in three applications:

1) Training;
2) Process monitoring and analysis;
3) Controller design.

In case of industrial applications, the digital twin can act like a gateway to understanding process dynamics. A mathematical model can represent the latter, and if this model is sufficiently accurate, then seamless switching between the real object and the model with the digital twin should be possible inside an appropriately designed SCADA application. Therefore, the items listed above should be successfully covered by a suitable application of the digital twin for any given process.

In this pilot study, we investigate the case of liquid level control in a laboratory-scale replica of a typical multi-tank hydraulic system. The following discussion will focus on two major items: the proposed method and the technical implementation, wherein the digital twin is realized in an extended reality environment using Unreal Engine 4 for visualization and SCADA system development and MATLAB and Simulink environment for the implementation of control logic and an interface to the actual real-life multi-tank system.

We focus on PI control since this is arguably the most popular industrial control method [27]. It should be noted, however, that the proposed methodology can be tailored to an arbitrary control technique.

A. Proposed Method

We assume that a classical PI controller will be used to control a given plant; in particular, we consider the parallel form of the PI controller which operates on the negative unity feedback-produced error signal $e(t) = y_{SP} - y_{real}$, where y_{SP} is the set point and y_{real} is the controlled plant output, to produce a control law $u(t)$ that in the time domain has the form

$$u(t) = K_p e(t) + K_i \int_0^{\tau} e(t) \mathrm{d}t, \tag{1}$$

where K_p is the proportional gain, and K_i is the integral gain. In industrial applications, K_p and K_i are the usual "tuning knobs"—appropriate values of these gains ensure stable and efficient operation of the control system. Unfortunately, in many cases the tuning of the PI controller is far from an optimal one. Thus, learning the procedure of tuning these gains to yield adequate control performance is essential in control engineering practice to improve tracking performance, safety, efficiency and reliability of control systems and also to reduce wasted energy which occurs due to improper behavior of the control law.

When speaking about real-life control systems in the context of modeling and identification, two situations are typically encountered:

1) Most frequently, there is no model of the controlled plant, so the controllers are either autotuned, or tuned manually following very simple tuning rules and operator intuition.
2) Less frequently, a model of the plant dynamics exists, which allows to employ, e.g., model based control design.

In this work, we deal with the second case. Indeed, having a coherent model of the plant dynamics is essential in case of training applications. Where the first case is concerned, an accurate model of the dynamics can also be obtained semi-automatically by means of appropriate identification methods.

Thus, assuming that a model of a certain plant exists and is accurate, we propose the following method that is suitable for creating an application for training in controller parameter tuning where single-input, single-output control is concerned:

1) Construct a model of the negative unity feedback control system consisting of the plant and the controller in (1).

2) Parameterize the model with respect to the controller gains $\theta = \left[\begin{array}{cc} K_p & K_i \end{array}\right]$.
3) Construct a grid in the (K_p, K_i)-plane, and for every point in the plane with $K_p \in \left[K_p^{min}, K_p^{max}\right]$ and $K_i \in \left[K_i^{min}, K_i^{max}\right]$ with fixed traversal step sizes K_p^{step} and K_i^{step}, respectively, compute a suitable metric mapping that to a cost function $\mathbf{z}_{pi}$. This typically means performing a large number of simulations of the control system in question.
4) Use the obtained $(\mathbf{k}_p, \mathbf{k}_i, \mathbf{z}_{pi})$-surface to obtain suitable ranges of controller gains K_p and K_i and/or display the surface to the user so that upon visually inspecting it, the user will make an educated decision about a controller parameter setting that results in a desirable performance of the control system.
5) Provide the user with means of exploring this data in an extended reality SCADA system and/or setup the "tuning knobs" of the PI controller with respect to obtained parameter ranges. When an accurate mathematical model of the plant dynamics is available, the user can then first experiment with controller tuning on the model, and then switch to the actual plant, setting the controller parameters accordingly.

Note that a cruicial step of this method is to choose appropriate cost metrics—this choice will depend on the control system specifications and may include, for example, tracking performance, robustness to gain variations, desired frequency domain specifications such as specific gain margin G_m and phase margin φ_m specifications, etc.

B. Technical implementation

Unreal Engine 4 was chosen as the basis for visualization and functional implementation of the digital twin. UE4 is a powerful multipurpose real-time 3D creation platform. Despite the fact that it was originally designed for development of video games, it is used for other purposes as well. UE4 is also being widely used in film and television production, architecture, automotive and transportation, training and simulation, and others [28]. Owing to high-grade VR support, easy-to-use Blueprint visual scripting system and the ability to render highly detailed graphics, UE4 became a suitable choice for this project.

In general terms, the project involves The multi-tank System (further referred to as "Real object") that is connected to a desktop computer with a PCI DAQ board and its digital twin that is rendered in UE4 on another desktop computer with an nVidia GTX 1080 graphics card. The real object includes three tanks that are interconnected via valves and tubes. With the help of valves and water pump the system raises the level of water in accordance with given set points (SP). This process is controlled by a controller implemented in MATLAB/Simulink environment. The two-way communication between the real object and the digital twin is established through UDP socket. Then, VR HMD (Head Mounted Device) and motion controllers are used in order to observe and manipulate the perceived environment. The complete architecture is depicted in Fig. 1.

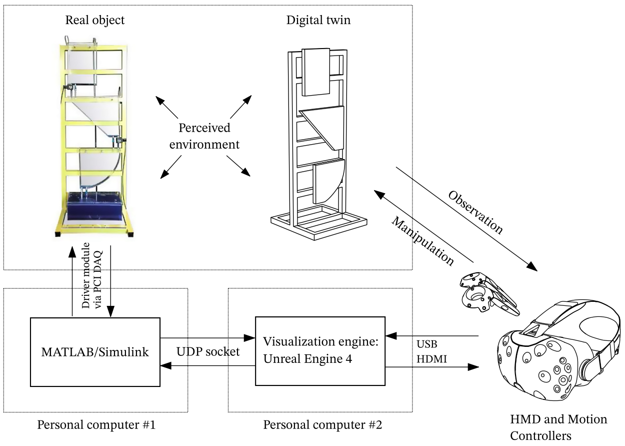


Fig. 1. Architecture of the developed experimental system including the real object, its digital twin, and necessary hardware components.

As it was mentioned previously, the PI controller is used in order to control the given plant. Being an essential element of efficient control performance, the correct adjustment of proportional gain K_p and the integral gain K_i parameters is a matter of high importance. With a view to providing the user with the opportunity to control K_p and K_i values and see the result of these changes in real-time, the "Control Spheres" are introduced. They represent an intuitive way to scale K_p and K_i values in the desired range by "stretching" movable corresponding spheres with both motion controllers. Values are then sent to MATLAB via UDP socket and processed for subsequent change in system behavior in accordance with the obtained parameter changes.

The workflow of two-way (Real object – Digital Twin) system can be represented as follows:

1) Initialization of MATLAB/Simulink model for the real object and real-time communication;
2) MATLAB model listens on UDP port, expecting the values for set points (SP) and PI controller gains;
3) Initialization of Digital Twin solution in UE4;
4) Initialization of communication of the Digital Twin solution in UE4;
5) Values of SPs and controller gains for real tanks are sent to MATLAB via UDP socket;
6) MATLAB receives values of SPs and controller gains from UE4 and processes them;
7) The system raises the level of water in real tanks in accordance with SPs using the controller updated with the received K_p and K_i gains;
8) Every time step (defined in the model) the water level values are sent to UE4 via UDP socket;
9) UE4 gets real water level values from MATLAB and visualizes the water in digital tanks at given level;
10) The user sets custom SPs in digital tanks by pointing at them with motion controllers and the gains of the

controller by resizing two spheres ("P" and "I", respectively);

11) Custom SP values and controller gains are sent to MATLAB via UDP socket;
12) Back to step (6).

The VR HMD used in this project is HTC Vive Pro. Being equipped with front-facing cameras, it allows to use a see-through module, enabling the user to simultaneously see both the real environment and the digital twin. Therefore, a more immersive experience is created providing the user with the ability to seamlessly affect the real object through the digital twin and instantly see the result of these interactions.

IV. Experimental results

In this work, we consider the control of the first tank of the multi-tank system. The schematic diagram depicting the first two out of three tanks and their interconnection is shown in Fig. 2.

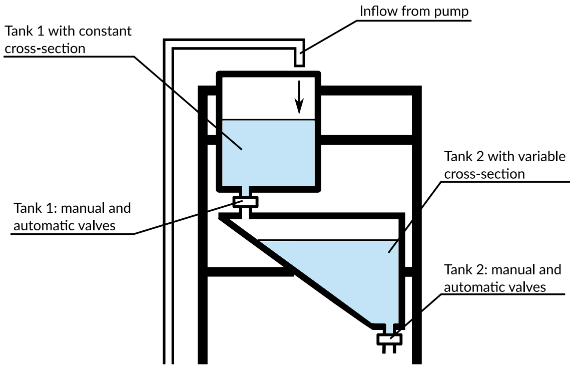

Inflow from pump
Tank 1 with constant cross-section
Tank 2 with variable cross-section
Tank 1: manual and automatic valves
Tank 2: manual and automatic valves

Fig. 2. Configuration of the first two tanks of the multi-tank system

The fluid dynamics in the first tank can be described by the following state-space model:

$$\dot{x}_1 = \frac{1}{\eta_1(x_1)}\left(u_p(v) - C_1 x_1^{\alpha_1} - \zeta_1(v_1) x_1^{\alpha_{v1}}\right), \qquad (2)$$

where x_1 is the level in the first tank, respectively, $\eta_1(x_1) = A = aw$ is the constant cross-sectional area of the tank, $u_p(v)$ is the pump capacity that depends on the normalized control input $v \in [0,1]$, $\zeta_1(v_1)$ is the variable flow coefficients of the automatic valve controlled by normalized input signal $v_1(t) \in [0,1]$. The fluid level is taken as the output of the system, i.e. $y_1 = x_1$. The normalized pump input v_1 is used to control the liquid level in the first tank y_1.

Following the discussion in Sec. III, we choose two metrics for identifying the range of PI controller parameters based on simulation:

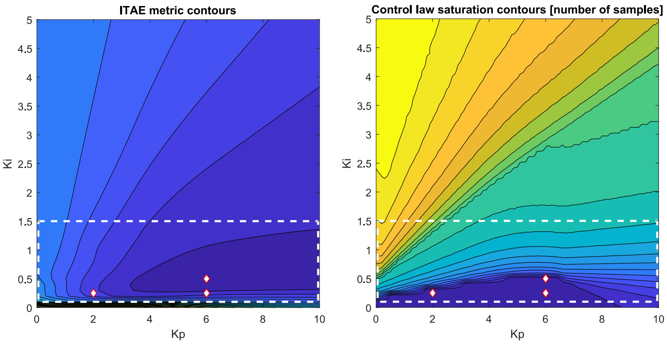
$$\mathrm{ITAE} = \int_0^T t e(t) \mathrm{d}t \qquad (3)$$

and

$$N_{u_{sat}} = \left|\{u_k \mid u_k \geqslant 0.99\}\right|, \qquad (4)$$

where $|\cdot|$ denotes the cardinality of the set of all such samples $u_k, k = 1, 2, \ldots, N$ that result in control saturation.

Based on simulating the control system consisting of (1) and (2), we obtain the contour plots depicted in Fig. 3. We then proceed to choose three points of the form (k_p, k_i) on the (K_p, K_i, f)-surface, where f denotes the combined metrics, for setting up the experiment. The points are chosen as $(2.00, 0.25)$, $(6.00, 0.50)$, and $(6.00, 0.25)$ which all fall into the identified region of acceptable performance.



(a) ITAE metric contours (b) Control law saturation contours

Fig. 3. Comparison of cost functions obtained by simulating the nonlinear model of the first tank across the PID controller gain range $K_p \in [0, 10] \wedge K_i \in [0, 5]$. The white dashed rectangle signifies the region which offers acceptable performance without causing control law saturation due to integral windup. In both cases, the set point of 0.15m is considered.

The real-time experiment then takes place. The XR user, wearing the HMD, sets up the digital twin, and proceeds to tune the gains of the PI controller while performing step experiments SP $= 0.00 \rightarrow 0.15$ in a single attempt lasting 500 seconds. This is shown in Fig. 4. With each new attempt, the level in the tank is reset to zero, and integral component state is also reset. The results of the experiment are depicted in Fig. 5.

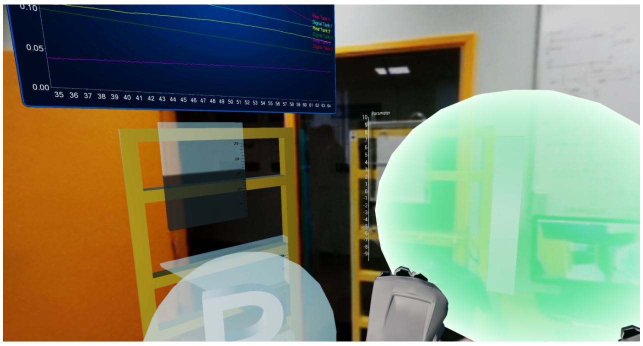


Fig. 4. The XR environment featuring the digital twin, real object, and the controller gain setting spheres. The user is changing the size of the I sphere effectively updating the K_i gain setting of the PI controller in Simulink.

It can be seen that the theoretical predictions in Fig. 3 are confirmed with a practical experiment which means this type of visualization can be used for creating effective training applications and also for advanced SCADA system prototyping.

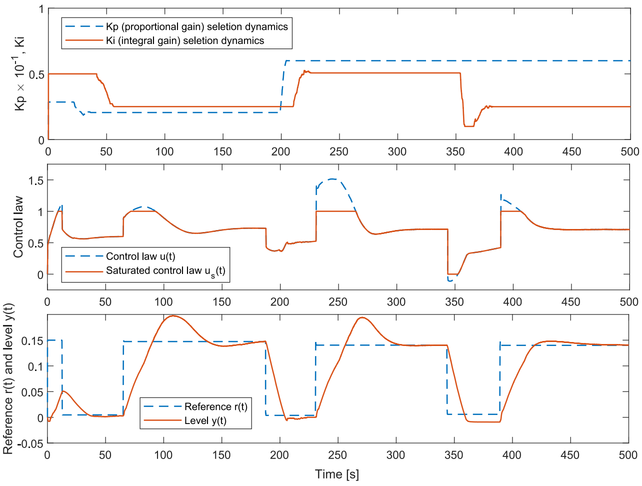


Fig. 5. Results of three consecutive step experiments with different PI controller gains.

V. Discussion

In this pilot study, we have discussed only PI control. However, it would also be interesting to include other control techniques, such as fractional-order and nonlinear control. Furthermore, subject-based studies have to be performed to assess the implementation of the immersive UI elements.

VI. Conclusions

In this paper, we have created an immersive environment for a novel supervisory control and data acquisition based on the concept of digital twins in extended reality geared towards laboratory training and controller tuning. The effectiveness of the prototype was successfully confirmed by performing a real-time control experiment with a real object and its digital twin. Future research involves subject based studies to improve UI elements making them more intuitive and informative.

References

[1] T. Reichenbach, G. Vasiljevic, and Z. Kovacic, "Virtual reality control systems," *Advanced Knowledge Application in Practice*, 2010.

[2] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[3] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. V. den Hof, "Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges," *Annual Reviews in Control*, vol. 43, pp. 1–64, 2017.

[4] N. D. Finkelstein, W. K. Adams, C. J. Keller, P. B. Kohl, K. K. Perkins, N. S. Podolefsky, S. Reid, and R. LeMaster, "When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment," *Physical Review Special Topics - Physics Education Research*, vol. 1, no. 1, 2005.

[5] S. B. Nolen and M. D. Koretsky, "Affordances of virtual and physical laboratory projects for instructional design impacts on student engagement," *IEEE Transactions on Education*, pp. 1–8, 2018.

[6] J. Pirker, I. Lesjak, and C. Guetl, "Maroon VR: A room-scale physics laboratory experience," in *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2017.

[7] M. J. Callaghan, K. McCusker, J. L. Losada, J. Harkin, and S. Wilson, "Using game-based learning in virtual worlds to teach electronic and electrical engineering," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 575–584, 2013.

[8] G. Sziebig, B. Takarics, and P. Korondi, "Control of an Embedded System via Internet," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3324–3333, 2010.

[9] B. Balamuralithara and P. C. Woods, "Virtual laboratories in engineering education: The simulation lab and remote lab," *Computer Applications in Engineering Education*, vol. 17, no. 1, pp. 108–118, 2009.

[10] M. S. Y. Jong, J. Shang, F. Lee, and J. H. M. Lee, "An Evaluative Study on VISOLE - Virtual Interactive Student-Oriented Learning Environment," *IEEE Transactions on Learning Technologies*, vol. 3, no. 4, pp. 307–318, 2010.

[11] M. Callaghan, K. McCusker, J. L. Losada, J. Harkin, and S.Wilson, "Integrating virtual worlds & virtual learning environments for online education," *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*, pp. 54–63, 2009.

[12] T. de Jong, M. C. Linn, and Z. C. Zacharia, "Physical and virtual laboratories in science and engineering education," *Science*, vol. 340, no. 6130, pp. 305–308, 2013.

[13] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, "Virtual laboratories for education in science, technology, and engineering: A review," *Computers & Education*, vol. 95, pp. 309–327, 2016.

[14] S. Yuan, J. Hu, and X. Li, "Application of virtual reality technique to real-time control system," in *2007 2nd IEEE Conference on Industrial Electronics and Applications*. IEEE, 2007.

[15] J. Bathelt and A. Jönsson, "How to implement the virtual machine concept using xPC target," Swiss Federal Institute of Technology Zurich, Institute of Mechanics, Tech. Rep., 2003.

[16] A. Jönsson, J. Wall, and G. Broman, "A virtual machine concept for real-time simulation of machine tool dynamics," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 795–801, 2005.

[17] M. Schluse, L. Atorf, and J. Rossmann, "Experimentable digital twins for model-based systems engineering and simulation-based development," in *2017 Annual IEEE International Systems Conference (SysCon)*. IEEE, 2017.

[18] J. Scheibmeir and Y. Malaiya, "An API development model for digital twins," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2019.

[19] P. Wu, M. Qi, L. Gao, W. Zou, Q. Miao, and L. lan Liu, "Research on the virtual reality synchronization of workshop digital twin," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. IEEE, 2019.

[20] U. Dahmen and J. Rossmann, "Experimentable digital twins for a modeling and simulation-based engineering approach," in *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2018.

[21] T. Mikkonen, K.-K. Kemell, P. Kettunen, and P. Abrahamsson, "Exploring virtual reality as an integrated development environment for cyber-physical systems," in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2019.

[22] V. Kuts, T. Otto, T. Tähemaa, and Y. Bondarenko, "Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality," *Journal of Machine Engineering*, vol. 19, no. 1, pp. 128–144, 2019.

[23] V. Makarov, Y. Frolov, I. S. Parshina, and M. Ushakova, "The design concept of digital twin," in *2019 Twelfth International Conference "Management of large-scale system development" (MLSD)*. IEEE, 2019.

[24] A. E. Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, 2018.

[25] C. J. Parris, J. B. Laflen, M. L. Grabb, and D. M. Kalitan, "The Future for Industrial Services: The Digital Twin," *Infosys Insights*, pp. 42–49, 2016. [Online]. Available: https://www.infosys.com/insights/iot/future-industrial-digital.html

[26] A. Kose, A. Tepljakov, S. Astapov, D. Draheim, E. Petlenkov, and K. Vassiljeva, "Towards a synesthesia laboratory: Real-time localization and visualization of a sound source for virtual reality applications," *Journal of Communications Software and Systems*, vol. 14, no. 1, 2018.

[27] A. O'Dwyer, "Handbook of Pi and Pid Controller Tuning Rules," 2009.

[28] Epic Games, "Unreal Engine," Available online: https://www.unrealengine.com/, 2020, retrieved 06.03.2020. [Online]. Available: https://www.unrealengine.com/