DextArm: Advanced Robotics Control System Through AR/VR, 6DOF and Stereoscopy Vision

By

Muhammad Yusuf bin Mohamad Razip
17009102

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Information Technology (Hons)
(BIT)

JANUARY 2022

Supervised by
Dr Suziah binti Sulaiman

Universiti Teknologi PETRONAS
32610 Seri Iskandar
Perak Darul Ridzuan

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MUHAMMAD YUSUF BIN MOHAMAD RAZIP

CERTIFICATE OF APPROVAL

DextArm: Advanced Robotics Control System Through AR/VR, 6DOF and Stereoscopy Vision

By

Muhammad Yusuf Bin Mohamad Razip

17009102

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
In partial fulfilment of the requirement for the
Bachelor of Information Technology (Hons)
(BIT)

Approved by,

Suziah

Dr Suziah Sulaiman
Senior Lecturer
Computer & Information Sciences Departm
Universiti Teknologi PETRONAS

____

(Dr Suziah binti Sulaiman)

UNIVERSITI TEKNOLOGI PETRONAS
SERI ISKANDAR, PERAK
January 2022

TABLE OF CONTENTS

Contents

LIFT OF FIGURES

ABSTRACT

The continuous emergence of the use of articulated robotics in the recent decade has greatly resulted industries especially in the replacement of worker from hazardous environments. Though, articulated robotics has its limitation in terms of manual control and feedback.

This study sets out to address the limitation of control and feedback of articulated robotics. This study includes the roles of six degrees of freedom (6DOF), stereoscopy vision, augmented reality (AR) and virtual reality (VR) to increase the efficiency, ease of control, and feedback by allowing reduction of complexity of control and replicating human-like depth of perception vision for feedback in articulated robotics.

The findings suggested that six degrees of freedom of control can help with manually controlling an articulated robot. Furthermore, stereoscopy vision can enhance visual feedback in robotics, increasing its effectiveness of monitoring. Finally, augmented reality will allow the possibility of adapting with input and feedback transmission latency though visual assistance. Hence, this final year project will be focused on developing a system that implements 6DOF control, AR/VR, and stereoscopy vision into articulated robotics and together with conducting tests and evaluation of the concept.

CHAPTER 1: INTRODUCTION

1.1 Background of Study

For decades, articulated robotics have become a norm in many industries as a method of replacing human labour. It can be due to the repetitiveness of the task, or the hazard posed by the environment of which, human presence would be too risky. Since the first patented articulated industrial robot in 1954 by George Devol, it has been the choice of multiple industries around the world for its broad and flexible applications. What makes the articulated robot very desirable is its high number of axes, varies in different sizes, and has the dexterity that can sometimes resemble a human arm. Thus, it is most used for complex task such as assembly and welding in a manufacturing line. For a far more complex, highly variable, and non-repetitive task such as bomb disposal, deep-sea exploration, and surgery, industries would have to rely on a specialized type of articulated robotics. This type of robot commonly requires manual human control. They are effective in performing specific tasks, but they can sometimes be very limited in terms of control and feedback. As a result, some industries that performs highly complex and risky task such as bomb disposal and hyperbaric welding would still rely on direct human interaction. This leads to the necessity to develop an intuitive method to achieve effective feedback and high controllability of a robotic arm that can be applied in multiple applications. This highly universal robotics controlling method could be the solution to these problems as a flexible and affordable method in assisting highly variable and complex task that are too risky for human presence.

1.2 Problem Statement

A manually controlled articulated robot requires 2 important functions, which is control and feedback. When controlling the robot, the operator needs to generate an input in which, the signal would be sent to the robot. The most common method of input are joysticks. A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. This is highly effective when controlling the tip of the articulated robot to move laterally in a plane, which is forwards, backwards, left, and right. Adding another joystick can increase its degrees of movement such as up and down. The problem occurs when robots, with highly complex range of motions, requires an input of six degrees of movement, which is forward, backward, left, right, up, down, yaw, pitch, and roll. Although it can be solved by adding more joysticks, having more than 2 is not a viable solution since it would increase complexity of control and require extensive muscle memory training. Furthermore, performing a task with this method is highly inefficient since the operator would require switching their hands between different joysticks. To solve this problem, we can implement a controller that can sense six degrees of movement, translating the movement of the operator's hand into usable data that can later be reported to the robot it is controlling. This method will solve the problem of control by reducing the complexity of manoeuvring the robot.

In conjunction with control, feedback is one of the most important parts when operating a manually controlled articulating robot. The method of feedback stimulation to the operator can be through haptic feedback, tactile feedback, kinaesthetic feedback, visual feedback through camera, or visual feedback through direct eye-to-device viewing. The most common is visual feedback through camera. They are used as a method to monitor the movement of the robot relative to the input generated by the operator. The operators will observe the object they are manipulating with a screen. It is an effective way to remotely control the robot if the task occurs in an environment that are hazardous or too risky for human presence. The only limiting factors of this method is that it does not provide depth of perception to the operator. Meaning, the operator cannot differentiate the distance of objects that are seen through the camera of the robot. In a critically sensitive task such as bomb disposal, the operator may accidently alter an object behind or in front of the object they intended to disable. This may lead to undesirable outcomes. To solve this problem, we can implement a method that allow human-like stereoscopy vision to the operator. In theory, this can be solved using 2 cameras with a certain distance lateral to each other and display it to each eye of the operator. It will give the operator a sense of depth similar to a human eye and allow them to perform tasks in a much precise manner. This project sets out to test this method.

Lastly, when operating a remotely controlled articulating robot, transmission latency is something that can never be avoided. Latency can be defined as the delay before a transfer of data begins following an instruction for its transfer. In the context of robotics, the operator will expect a delay in feedback of the movement of the robotic arm relative to the time when the input is given. This can create a lot of problem. For an example, the operator gives an input to move the tip of the robotic arm to slide left, expecting it to return feedback of sliding to the left. The delay between the input and the feedback may take up to a few second depending on the distance or medium by which the signal travels through. Extended delay may give the operator a false assumption that the robotic arm is not receiving enough input, therefore the operator will give more input than necessary. As a result, the robotic arm will move more than what the operator desires. Although this case may occur subtly, it can create major problems in a critically sensitive task. To solve this problem, this project will not focus on reducing the delay but to create a method to adapt with the delay through the use of augmented reality and virtual reality.

1.3 Objectives

The main goal of this project is to develop a controlling system for high-dexterity robotics that will assist people in performing highly complex task in hazardous and risky environments. This project will be implementing a motion controller to achieve six degrees of freedom for robotics control. This project will also implement virtual reality technology to allow stereoscopy vision and depth of perception. In addition to virtual reality, this project will also implement augmented reality technology to cope with the delay that occurs between user input and feedback. Following are the objectives of this project to achieve the mentioned goals:

- To investigate how motion controller can be used as a method of six degrees of freedom control for high-dexterity robotics
- To investigate how human stereoscopy vision works and how it can be implemented into the system

- To develop a software that implements AR/VR technology for 6DOF robotics control, visual assistance for delay, and live stereoscopy video delivery to the user
- To evaluate the effectiveness of the overall system

1.4 Scope of Study

The scope of the project refers to the extent of which the research area will be explored. The scope of this project will involve with studying six degrees of freedom in movement and its related technologies. The project will extend further to study on creating usable data from the movement of the user's hand that can then be implemented into robotics actuation. In addition, the scope of this project will also be focused on stereoscopic vision, how to mimic it, and how to implement it into the software. Virtual reality related technology will be used to integrate six degrees of freedom in movement and stereoscopic vision into a single interactive system. In addition, augmented reality will also be implemented to create visual assistance to cope for input and feedback transmission latency. A software will also be developed to operate all the required functionalities of this system. Lastly, the scope of the project will cover the test and evaluation of the system.

CHAPTER 2: LITERATURE REVIEW

The objective of this literature review is to develop an understanding and discover the occupations that involves complex task in hazardous environments where this system can be applied. Furthermore, the objective is to develop an understanding on the current technologies applicable for this project.

2.1 Risky Operations and Its Implication.

Task that requires complex object manipulation that occurs in a hazardous environment can range broadly. An example of this type of occupation is electrical related occupations, where it involves high risk and complex tasks. Tasks like these can and has caused high number of occupational fatalities.

(“Workplace Injury & Fatality Statistics”, 2019) shows that in North America, the number of occupational fatalities related to electrocution has increased to its highest in 2019. According to their record and statistics, there were 166 electrical related fatalities in 2019 alone, which was statistically a 3.75% increase over 2018 and the highest number of fatalities since 2011. Furthermore, the records for each source of data are retrieved from industries that follow the Occupational Safety and Health Act. This indicates that although industries follow the rules and protocols of safety in a hazardous environment, it is not enough to mitigate the increasing number of cases involved with occupational fatalities.

Another example of work that is similarly hazardous is Underwater Welding. (Hofmann, n.d) mentions that underwater welding can be an extremely deadly task since it combines all the hazard of hot work with the potential of drowning. He further supports his claim with data by OSHA. The data revealed the total death of 116 among 3000 full-time commercial divers in the United States alone. The data shows that underwater welding has a death rate nearly 40 times the national average across all industries. This further solidifies that an alternative solution must be developed in order to reduce the increasing number of fatalities involved with hazardous occupations.

2.2 Articulated Robotics Manipulator as a solution for Hazardous Task.

For environments that are hazardous, many industries rely on specialized articulated robotics to perform a broad range of highly complex task. One such environment is underwater (or subsea). The pressure of these environments can reach extremes high depending on the depth in which the operation is performed. The operations performed by this specialized articulating robotics commonly takes place in different application within offshore oil and gas, marine renewable energy (MRE) and marine civil engineering industries as well as in marine science and military applications. They are a very effective solution for performing these specific tasks.

As mentioned by Sivcev et al. (2018), some of the tasks underwater manipulators are designed to execute include pipe inspection, salvage of sunken objects, mine disposal (bomb disposal), cleaning surfaces, opening and closing valves, drilling, rope cutting, cable laying and repair, clearing debris and fishing nets, biological and geological sampling, archaeological work, etc. This proves that, in general, that manipulators are an essential for multiple applications related to complex underwater tasks that bares hazardous factors.

2.3 General Use of Articulated Robotics

Articulated robotic is commonly used as industrial robots. They look like human arms, hence the name they are usually called robotic arm or manipulator arm. Their articulations with several degrees of freedom allows for a wide range of movement and applications (Guarana, 2020). The task that these robots perform includes welding, painting, assembly, disassembly, pick and place for printed circuit boards, and many other tasks: all accomplished with high endurance, speed and precision. They are effective in handling and assisting with material, replacing human labour.

Industrial articulated robots have become more common in use as of 2020. They are used in many industries worldwide and had become the common choice in performing intricate tasks. According to International Federations of Robotic, in the year 2017, an estimated 1.64 million industrial robots were in operation worldwide (“Worries about premature industrialization”, 2017).

2.4 Six Degrees of freedom

Six degrees of freedom (6DOF) refers to the freedom of movement of a rigid body in three-dimensional space. It describes the freedom of change in position as translation in three perpendicular axes combined with changes in orientation through rotation about three perpendicular axes (Jordi, 2013). Furthermore, the six degrees of freedom is a term most commonly used in the context of virtual reality, referring to the tracking of either the user's hand or the head ("Degrees of freedom | Google VR", 2019).

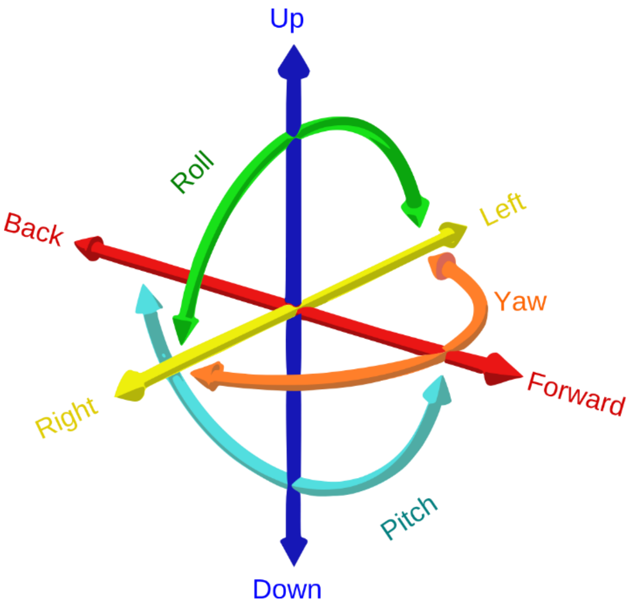


Figure 1: Six Degrees of Freedom

In the context of articulated robotics, the term 6DOF is important in defining its range of motion. Degrees of freedom is a practical metric in contrast with the abstract definition of DOF which measures the aggregate positioning capability of a system (Paul et al., 1981)

Motion tracking hardware devices are defined by their ability to sense motion in six degrees of freedom. Motion tracking device such as TrackIR and software-based apps like Eyeware Beam are used for 6DOF head tracking. This device is often used in flight simulators and other simulators that require looking around the cockpit or simply avoiding collision in a virtual environment. Similarly, a 6DOF controllers also does the same thing but it is implemented for the tracking of the user's hand and is commonly used for virtual reality applications.

Figure 2: Oculus Quest 2 6DOF Controllers

2.5 Stereoscopy Vision

Stereoscopy (or stereopsis) is defined by the perception of depth and three-dimensional structure obtained on the basis of visual information derived from two eyes. Because the eyes of humans are located at different lateral position on the head, it produces two slightly different images projected to the retinas of the eyes (Howard, 1995).

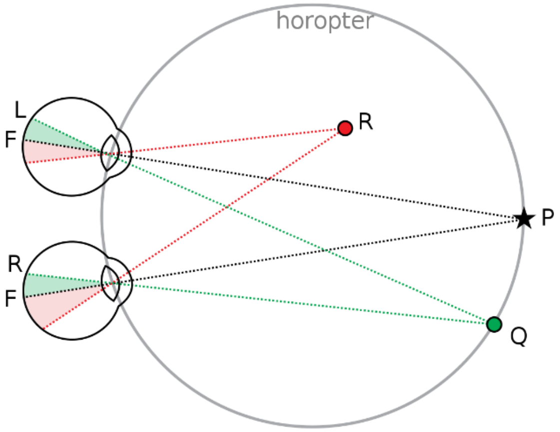


Figure 3: Stereopsis Resulting from Binocular Vision

Stereopsis was first explained in 1838 by Charles Wheatstone. He explains that the mind perceives an object of three dimensions by means of the two dissimilar pictures projected by it on two retinas (Wheatstone, 1838). He then created a device to replicate this effect which later known as a stereoscope. The device consists of 2 mirrors pointed towards 2 images that are slightly different than each other. The device produces an illusion of 3D stereoscopy image which in today’s technology, can be replicated with cameras and screen. Computer stereo vision is part of the field of computer vision. It is sometimes used in mobile robotics to detect obstacles. Example applications includes the ExoMars and surgical robotics (Leshem, 2010). The main concept of this device is an inheritance of the concept proposed by Wheatstone nearly 2 centuries ago.

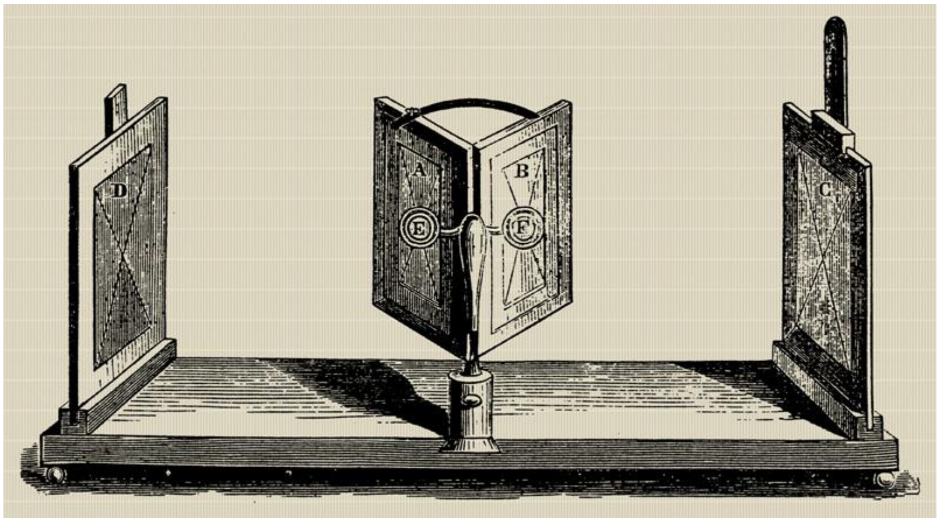


Figure 4: Wheatstone's Mirror Stereoscope

CHAPTER 3: METHODOLOGY

This chapter will explain the approach that will be used and the detail of each phase. The methodology that will be used in this project is the Agile method. The Agile Software Development Life Cycle methodology is a way to manage a project by breaking it up into several phases. This approach is chosen because it can accommodate the change and the need to produce software faster. As to compare with Waterfall, Agile is well equipped to handle the complexity and variable involved in the development cycle. The methodology consists of phases which are planning, designing, develop, testing, deploying, reviewing and launch but in no particular order. Which means that the project is not limited to a specific order, can stop at any phase, iterate previous phase, and repeat the entire cycle continuously. This is especially useful since the project may go through multiple cycle of redesigning and development after each testing phase. The reason this project may repeat multiple cycles of phases is to ensure that the final product produce the best possible result within the timeframe of this project. All results will be produced within the testing phase.

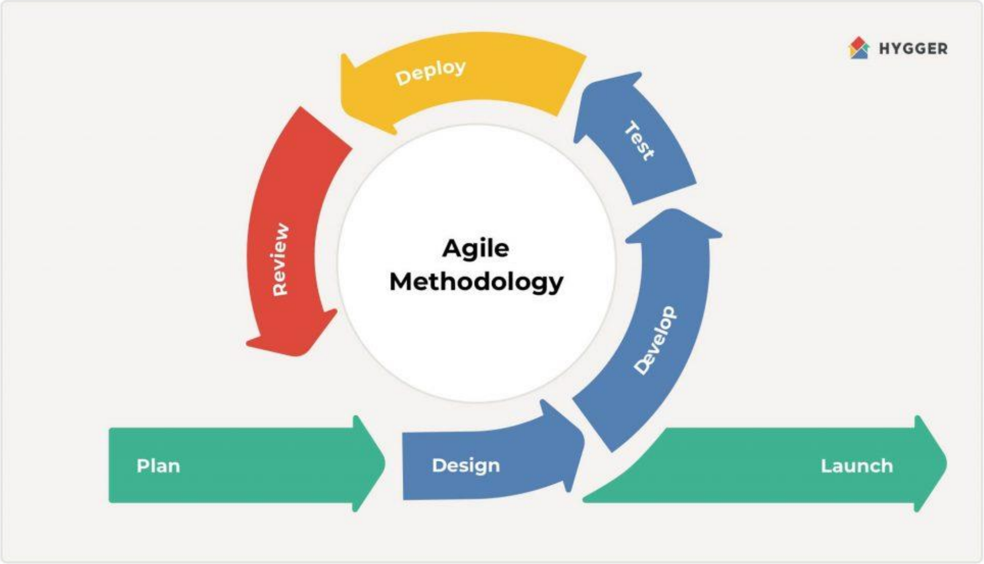
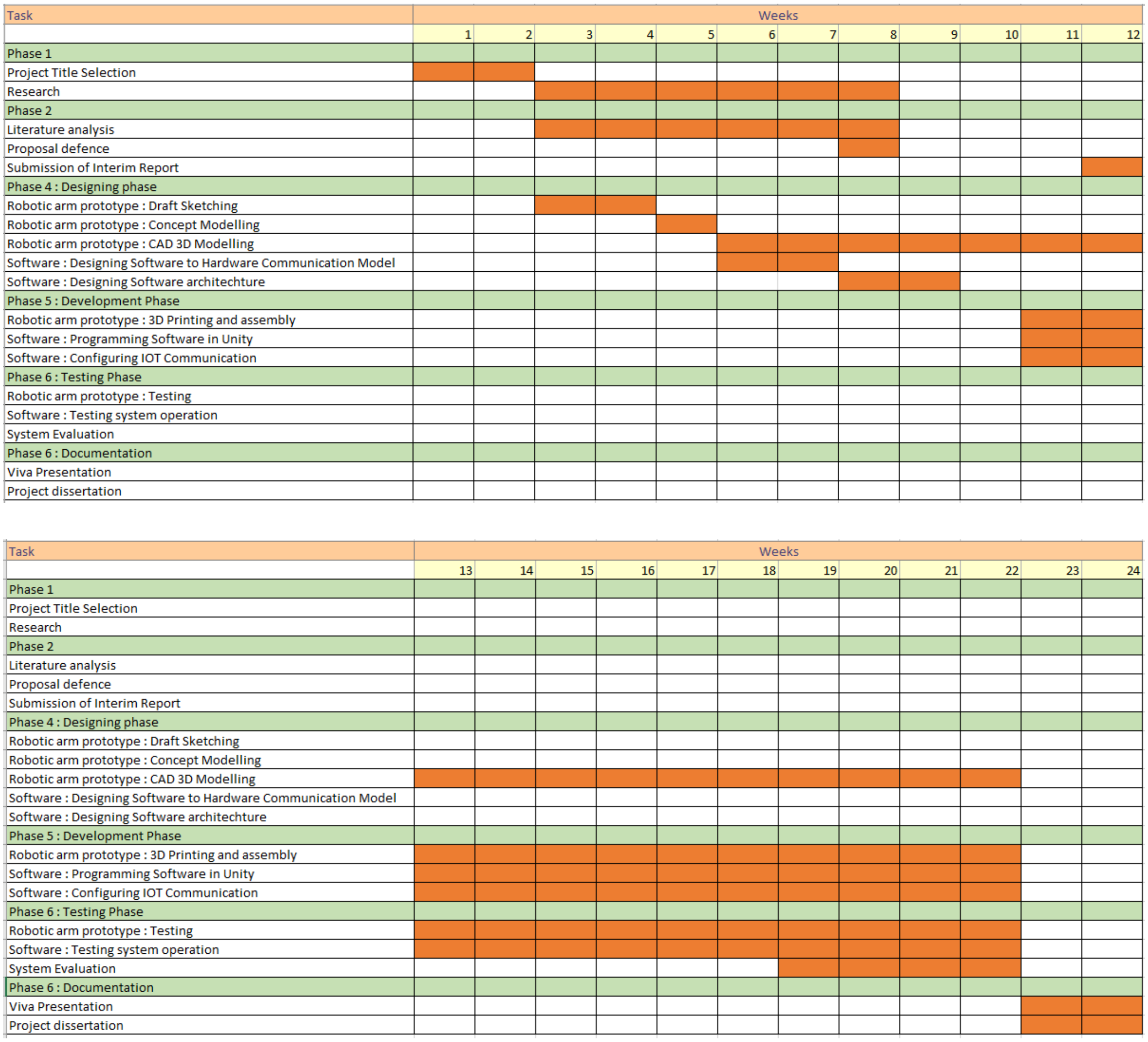


Figure 5: Agile Methodology

3.1 Planning Phase

The planning phase plays a big role in ensuring that the project will finish successfully in time. The focus of the planning phase is the approach and scheduling of each phase that will be performed. The project will be done within 2 semesters which is Final Year Project 1 and Final Year Project 2 and totals up to 24 weeks. Therefore, the duration of each task in each phase are as follows:



Task	Weeks											
	1	2	3	4	5	6	7	8	9	10	11	12
Phase 1												
Project Title Selection	🟧	🟧										
Research			🟧	🟧	🟧	🟧	🟧	🟧				
Phase 2												
Literature analysis			🟧	🟧	🟧	🟧	🟧	🟧				
Proposal defence								🟧				
Submission of Interim Report												🟧
Phase 4 : Designing phase												
Robotic arm prototype : Draft Sketching			🟧	🟧								
Robotic arm prototype : Concept Modelling					🟧							
Robotic arm prototype : CAD 3D Modelling						🟧	🟧	🟧	🟧	🟧	🟧	🟧
Software : Designing Software to Hardware Communication Model						🟧	🟧					
Software : Designing Software architechture								🟧	🟧			
Phase 5 : Development Phase												
Robotic arm prototype : 3D Printing and assembly											🟧	🟧
Software : Programming Software in Unity											🟧	🟧
Software : Configuring IOT Communication											🟧	🟧
Phase 6 : Testing Phase												
Robotic arm prototype : Testing												
Software : Testing system operation												
System Evaluation												
Phase 6 : Documentation												
Viva Presentation												
Project dissertation												

Task	Weeks											
	13	14	15	16	17	18	19	20	21	22	23	24
Phase 1												
Project Title Selection												
Research												
Phase 2												
Literature analysis												
Proposal defence												
Submission of Interim Report												
Phase 4 : Designing phase												
Robotic arm prototype : Draft Sketching												
Robotic arm prototype : Concept Modelling												
Robotic arm prototype : CAD 3D Modelling	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
Software : Designing Software to Hardware Communication Model												
Software : Designing Software architechture												
Phase 5 : Development Phase												
Robotic arm prototype : 3D Printing and assembly	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
Software : Programming Software in Unity	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
Software : Configuring IOT Communication	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
Phase 6 : Testing Phase												
Robotic arm prototype : Testing	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
Software : Testing system operation	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		
System Evaluation							🟧	🟧	🟧	🟧		
Phase 6 : Documentation												
Viva Presentation											🟧	🟧
Project dissertation											🟧	🟧

Figure 6: Gantt Chart

Some tasks in the designing, development, and testing phase overlap with each other due to the Agile method. Each task proceeds to the other after it is done, but once it finishes with the testing phase, it may cycle back to the designing phase if issues were found and require iterations. Initially, the planning phase includes the development of a physical and working robotic arm as a test bed for the system. But the scope of this project has been reduced to only include the development of software due to time constraint. Additionally, the project will also discard the plan to establish an IOT communication between the robotic arm and the software due to the absent of the robotic hardware. Therefore, in the listed task shown above, “Robotic arm prototype: 3D Printing and assembly” and “Configuring IOT Communication” will be ignored throughout this project.

3.2 Requirement Phase

The project requirements are conditions that must be met or completed to ensure the success and completion of the project. The requirement in this project focuses on the effectiveness of the user interacting with the software. This must directly align with the objectives of this project, which is to implement six degrees of freedom in the controls of robotics, augmented reality to adapt with input and output transmission latency, and stereoscopy vision to allow depth of perception to the operator.

Regarding six degrees of freedom in control of robotics, the software produced in this project must be able to produce usable data from the motion of the user’s hand. This data can later be used for robotics actuation. The project will require testing and evaluation of this concept to prove that the required data can be produced.

Afterwards, the project will require the development and implementation of augmented reality into the system with the goal of adapting with the input and output transmission delay. The project will require testing and evaluation to prove that this concept is feasible in effectively solve the mentioned problem statement.

Lastly, the project will require the integration of stereoscopy vision into the system. This project must prove its capability of allowing depth of perception by testing the effectiveness of users to differentiate object between different distances.

3.3 Designing Phase

The processes involved in the design phase for producing the complete system will involve 3D modelling and software designing. This phase identifies the essentials such as the tools and software required to design and build the complete system.

In this project, the software involved are as follows:

1. Autodesk Fusion 360: To create a precision CAD model of the robotic arm that will be used as a model for producing angular data and visual assistance model for augmented reality implementation.
2. Blender: To rig bones to the virtual robotic arm with the purpose of allowing Unity engine software to understand how the visual assistance model move and its range of motions.
3. Unity engine: To create a software that operates the entire system and as a virtual environment interface for the user to interact with.
4. Visual Studio Code: To script objects, interface, and interaction in the Unity game engine through C# language.
5. Oculus Software Development Kit (SDK): To be utilized for the functionalities of virtual reality.

3.4 Development Phase

Prior to the start of the project, the author will need to first understand and map out the communication model between the software and hardware. This model can later be used as a foundation and reference to the development of the software. The figure below shows the communication model of the system:

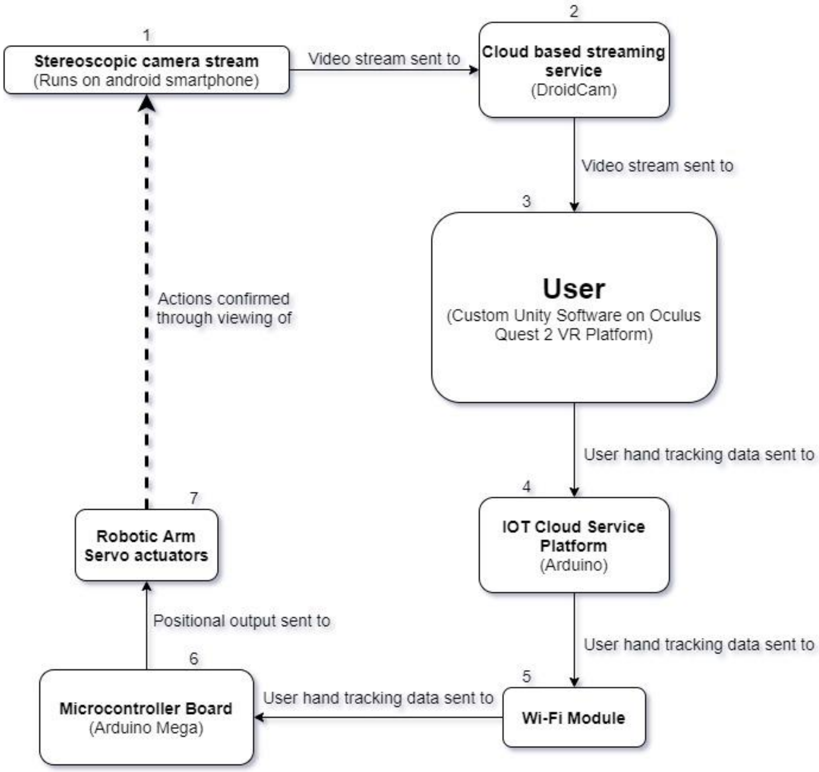


Figure 7: System Communication Model

Since the development of the actual robotic arm and establishing IOT communication is not within the scope of this project, the development phase will only prioritize on two modules from the system's communication model. This includes custom unity software for virtual reality and stereoscopy camera stream. The prioritization of the two modules aims to produce usable data for general robotics control from user's 6DOF hand control, augmented reality for input and feedback transmission delay, and depth of perception to the user through stereoscopy vision.

The project proceeds towards the following:

1. 6DOF Control and Augmented Reality in Unity Software

Six Degrees of Freedom (6DOF) is defined by the freedom of movement of a rigid body in a three-dimensional space. To be precise, the body is free to change its position as forward/backward (surge), up/down (heave), left/right (sway) translation and yaw, pitch, and roll rotation. In the context of this study, it refers to the tip part of the robot arm.

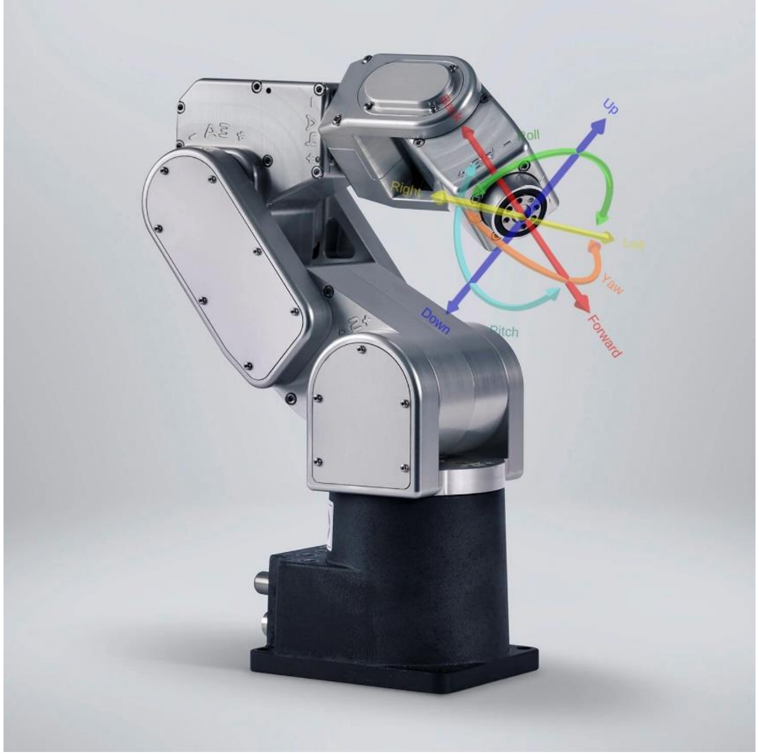


Figure 8: Articulated Robot with Six Degrees of Freedom

To sense the six degrees of freedom motion of the human hand that can then be translated to the motion of the robotic arm, the author must use a controller that is capable of sensing six degrees of movement. The Oculus Touch controller is a 6DOF controller that sense the position of each of the user's hand relative to the position of the user's head. The controllers are paired with the Oculus Quest 2 headset, a virtual reality rig that has four sensors in the front side to sense the position of the controllers.

Figure 9: Oculus Quest 2 Virtual Reality rig

To get the virtual reality rig to work in the Unity engine environment, all necessary software development kit and plug-ins must first be added into the system. This is to allow for the game engine to have all the related functionalities of Virtual Reality. The SDK and Plug-ins are as follows:

- Oculus XR Plugin

Oculus XR Plugin Release

Unity Technologies

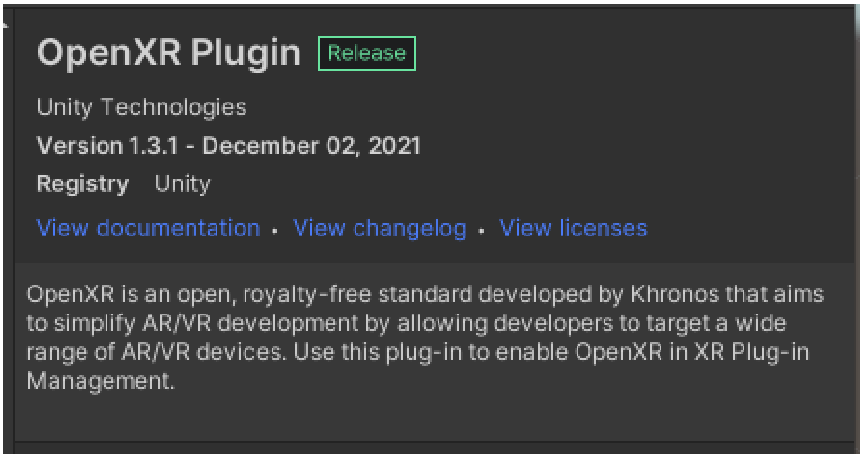
Version 1.11.2 - December 15, 2021

Registry Unity

View documentation · View changelog · View licenses

Provides display and input support for Oculus devices.

- OpenXR Plugin



OpenXR Plugin Release

Unity Technologies

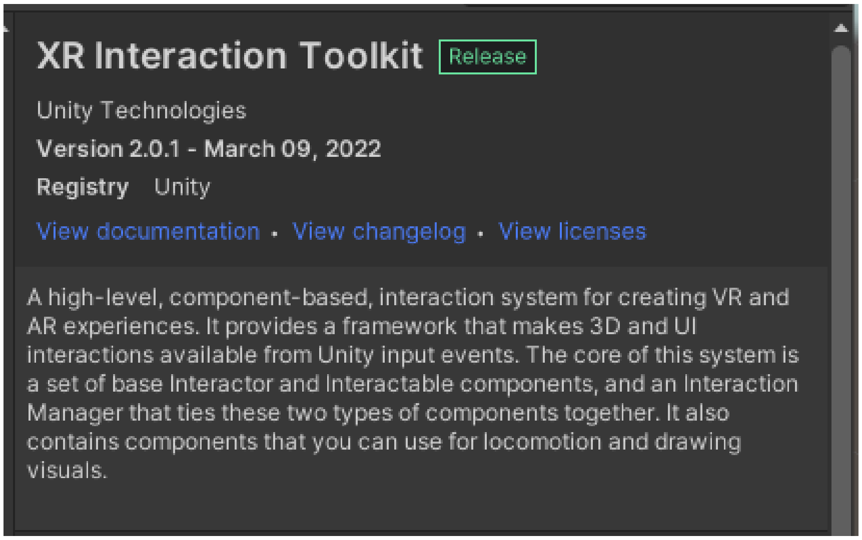
Version 1.3.1 - December 02, 2021

Registry Unity

View documentation · View changelog · View licenses

OpenXR is an open, royalty-free standard developed by Khronos that aims to simplify AR/VR development by allowing developers to target a wide range of AR/VR devices. Use this plug-in to enable OpenXR in XR Plug-in Management.

- XR Interaction Toolkit



XR Interaction Toolkit Release

Unity Technologies

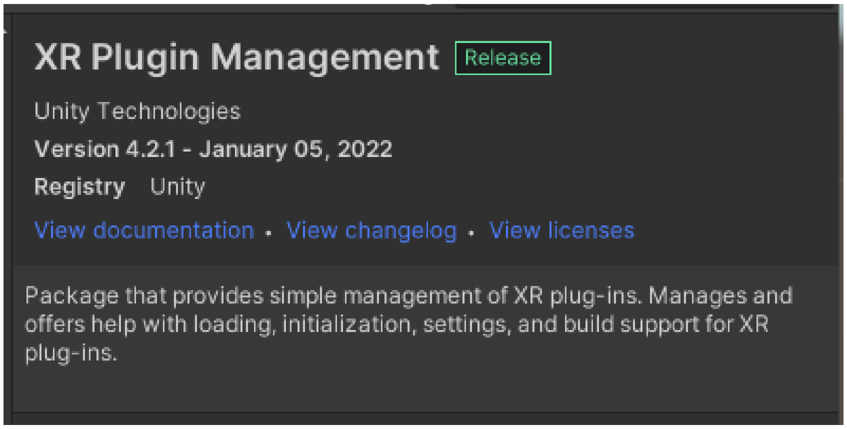
Version 2.0.1 - March 09, 2022

Registry Unity

View documentation · View changelog · View licenses

A high-level, component-based, interaction system for creating VR and AR experiences. It provides a framework that makes 3D and UI interactions available from Unity input events. The core of this system is a set of base Interactor and Interactable components, and an Interaction Manager that ties these two types of components together. It also contains components that you can use for locomotion and drawing visuals.

- XR Plugin Management



XR Plugin Management Release

Unity Technologies

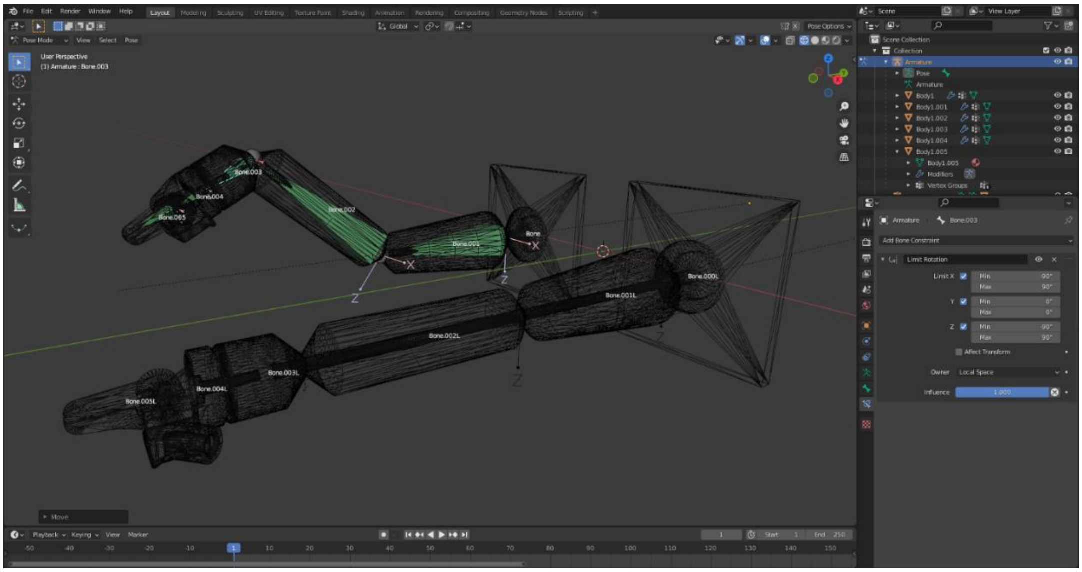
Version 4.2.1 - January 05, 2022

Registry Unity

View documentation · View changelog · View licenses

Package that provides simple management of XR plug-ins. Manages and offers help with loading, initialization, settings, and build support for XR plug-ins.

Once the game engine can interact with all necessary virtual reality functions, a 3D model of a robotic arm is created. The 3D model is produced in Autodesk Fusion 360. The 3D model is then transferred into Blender software to be rigged with armatures (bones). The purpose of rigging the armature to the model is to allow for Unity engine to understand how the model moves and its constraint of movement. Without the armature, the 3D model will appear as an object or mesh that does not have any moving parts. Therefore, this step is essential.



Limit Rotation

Limit X ☑ Min 90° Max 90°

Y ☑ Min 0° Max 0°

Z ☑ Min -90° Max 90°

☐ Affect Transform

Owner: Local Space

Influence: 1.000

Figure 10: 3D Robotic Arm Model and Armature in Blender

The robotic arm model serves two functions in the overall system. It is to allow for the software to produce angular data for general robotics application and to simulate an augmented reality visual assistance for input and feedback transmission latency.

To further explain the first function, the 3D model acts as a chain of joints in which, through the use of inverse kinematics in Unity engine, allow for the user to move the tip of the robotic arm while allowing for the forearm and upper arm of the model to follow its path. The inverse kinematics functionality in Unity engine changes the joints of the model according to the rotation of its parent and so the end point of a chain of joints can be determined from the angles and relative position of the individual joints it contains.

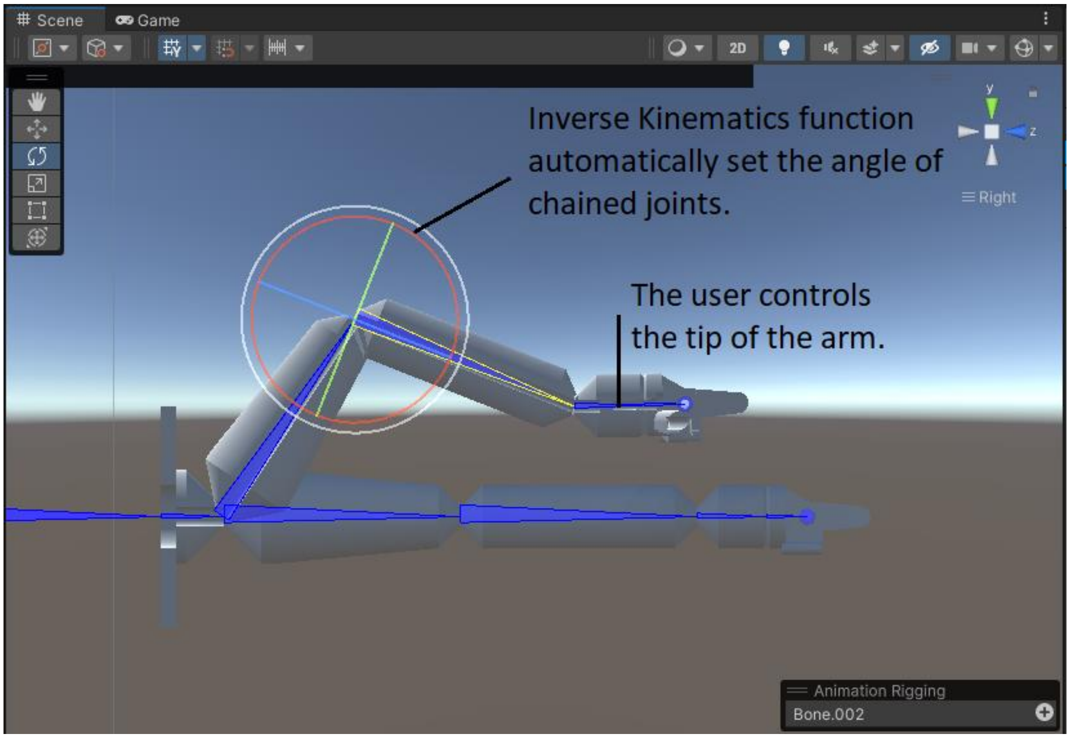


Figure 11: Inverse kinematics in Unity

In each of the bones of the chained joints (marked blue in the figure above), exist angular data that is accessible. This data can later be converted into usable angular data for robotics application. The data is extracted through a script that is written in C# in Visual Studio Code. The script takes every angular data of each chained joint of the 3D model and update it into the console log of Unity engine. The resulting data of this method will prove that the system is able to produce usable data from the six degrees of motion of the user's hand.

For the second function of the 3D robotic arm model, it serves the purpose of simulating an augmented reality visual assistance for input and feedback transmission latency. The 3D model will appear above the stereoscopy video stream and above the actual robotic arm that is seen through the video stream. This will create a "ghosting" visual effect where, above the real robotic arm, appears a translucent replica of itself. The 3D robotic arm model will follow the movement of the user's hand without any delay while the real robotic arm will later move to the position of the 3D robotic arm model. In theory, this "Ghosting" visual effect will let the user know where the final position of the real robotic arm by referencing to the position of the 3D robotic arm. Thus, this visual effect will prevent the user from giving more input than necessary even if the input and feedback transmission delay is major. Unfortunately, this theory cannot be proven physically without a working robotic arm in which, will not be produced during this project. To accommodate the result to prove this theory, a mock up simulation is done within Unity engine. A copy of the 3D robotic arm model is placed in the same position as the original model. The copy of the model mimics the movement of the original model with a predetermined delay. This method replicates input and feedback transmission delay. The detail on the method of testing this theory will be discussed within the testing phase section of this report.

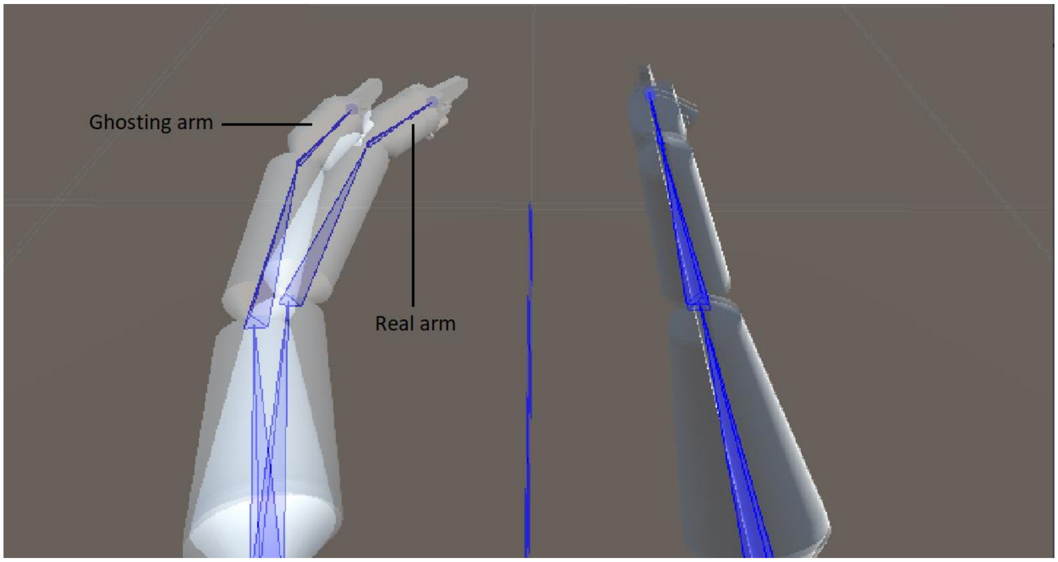


Figure 12: Mock-up of Visual Assistance for Input and Feedback Transmission Latency

2. Stereoscopic Vision

Stereoscopy (or stereopsis) is defined by the perception of depth and three-dimensional structure obtained on the basis of visual information derived from two eyes. Because the eyes of humans are located at different lateral position on the head, two slightly different images are projected to the retinas of the eyes. We can replicate this effect by displaying 2 images that are slightly different than each other. The source of these two images is from 2 cameras that are positioned at a distanced laterally from each other. For this project, to reduce cost, a single camera from a webcam will be used. A mirror device will be put in front of the camera to split the image before it enters the lens of the camera. The mirror will have two reflectors that are place at a distance lateral to each other. While in the middle, another set of mirrors will be used to redirect the angle of the images into the camera, combining it into 2 images in a single frame. The video stream of the camera can then be transmitted to each eye of the user through a virtual reality headset. This replicates the stereoscopic vision where it allows depth of perception.

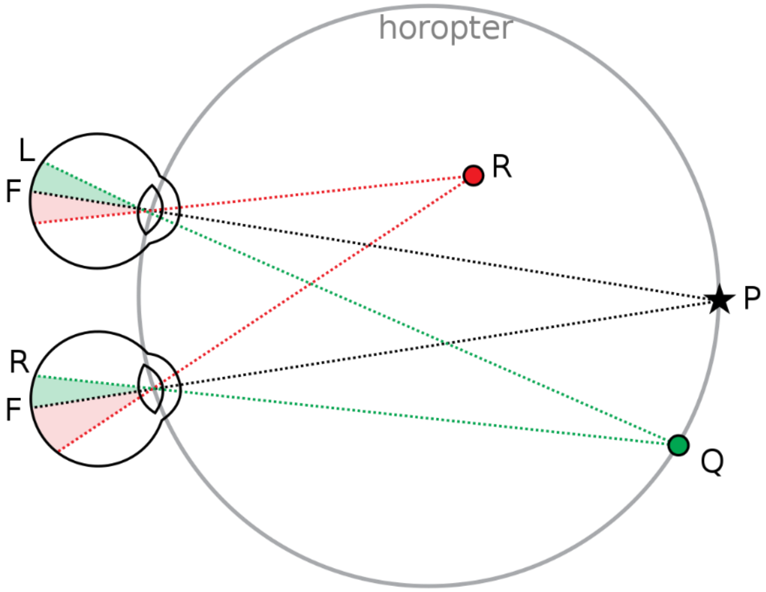


Figure 13: Stereoscopic Vision

Figure 14: Universal Stereo Lens for Camera

Figure 15: Webcam and Mirror Device

Within the Unity game engine, the dual image stream of the webcam will be displayed near the eyes of the user. The stream will be calibrated to match with the interpupillary distance of the eyes. To be precise, the centre of each of the dual image produced by the webcam stream is aligned with the centre of each eye of the user. To isolate the surrounding view that are outside of the main video stream, black planes are placed to block the peripheral vision of the user. This is to improve visual focus of the user when operating the system. The 3D robotic arm model in the system is configured where, the user can still see it above the video stream and the black planes. Once the system has been set up, the project proceeds with testing the mentioned theory of achieving depth of perception. The method of testing the effectiveness of this concept will be discussed within the testing phase of this report.

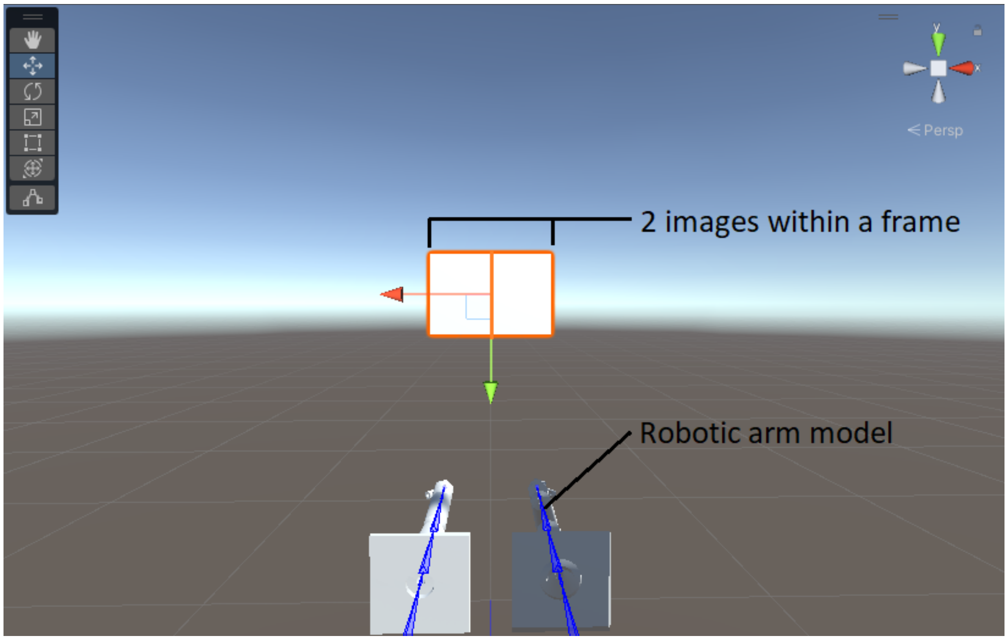


Figure 16: Stream Position

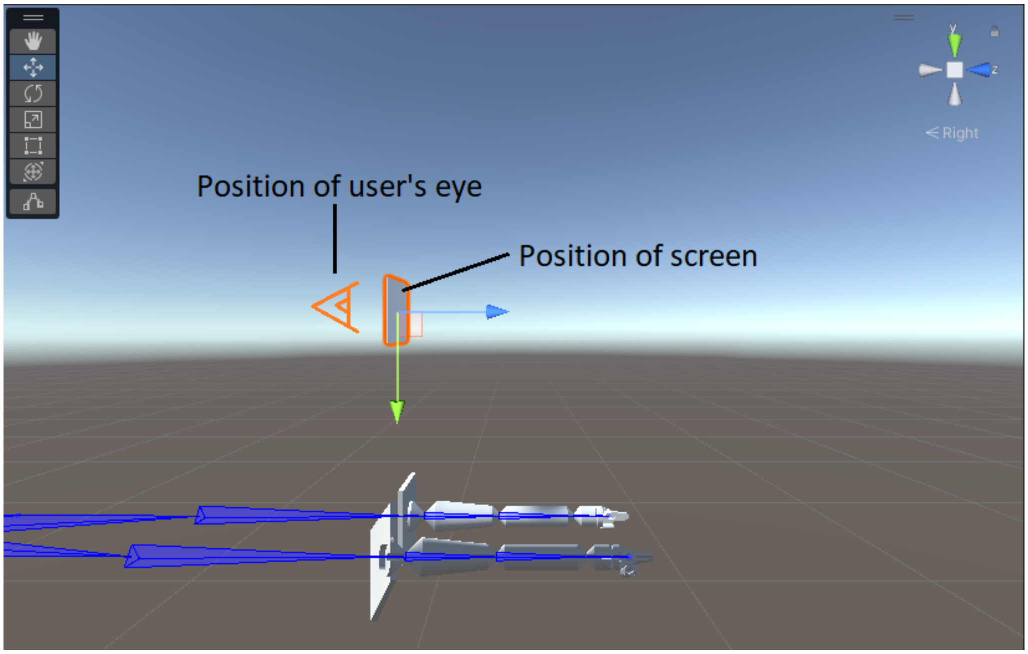


Figure 17: Stream Position

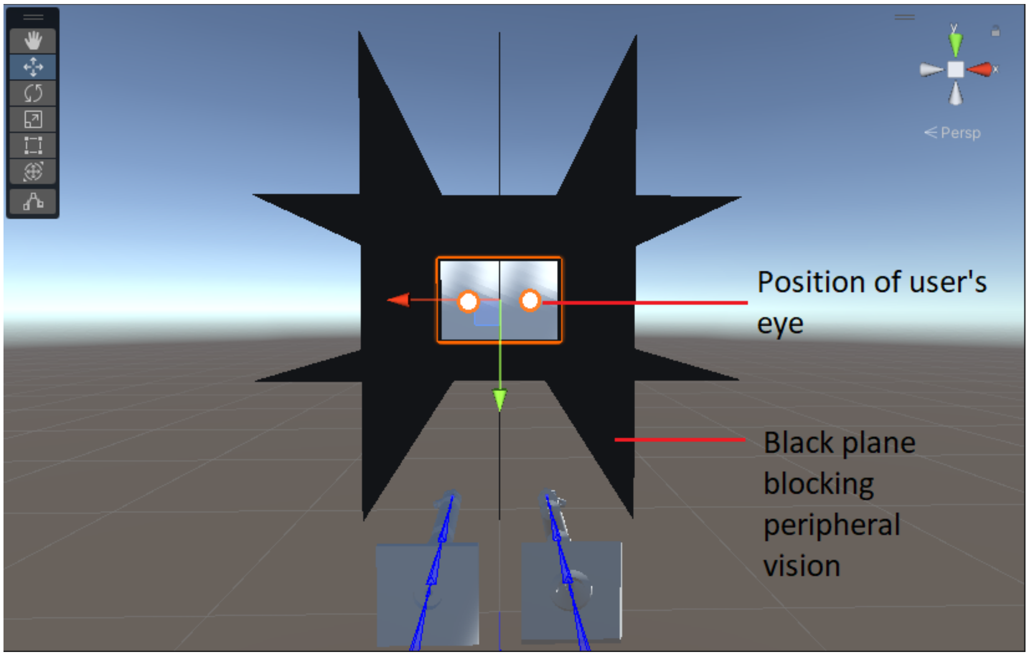


Figure 18: Blocking Peripheral View

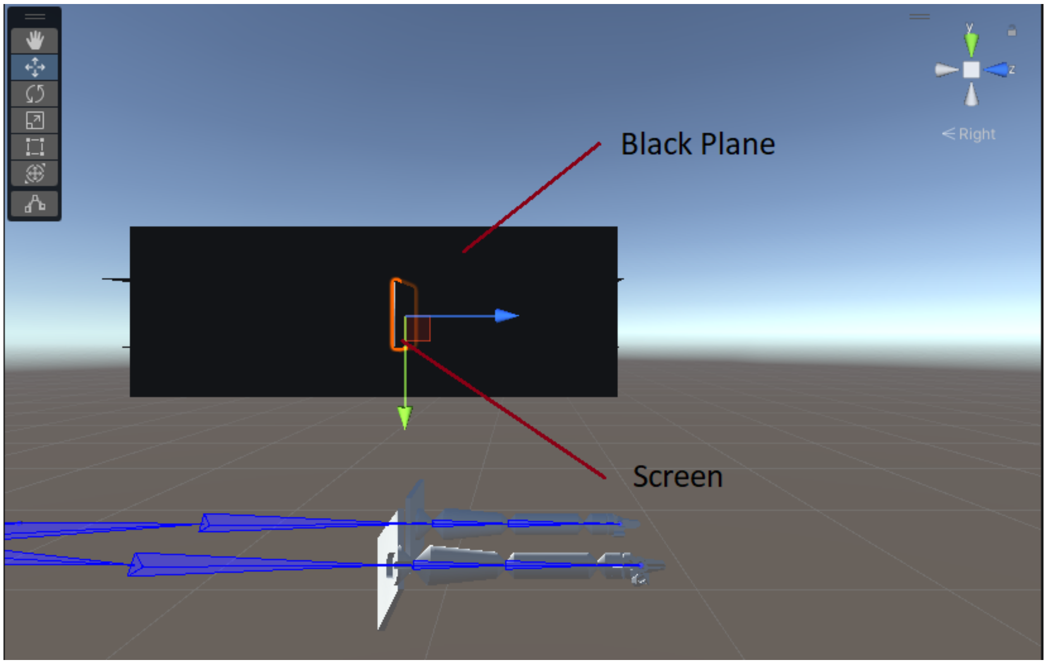


Figure 19: Blocking Peripheral View

3.5 Testing Phase

Four tests will be conducted in the testing phase. The first one focuses on the functionality of the system. This refers to the final product where the user is able to interact with the system. The second focuses on the ability to produce usable data for robotics application. This refers to the production of angular data in the console log of Unity engine. The third test focuses on the effectiveness of the augmented reality functionality. This refers to the effectiveness of the system to prevent over stimulated input to the robotic arm. The fourth test focuses on the effectiveness of the implementation of stereoscopy vision. This refers to the ability to produce depth of perception to the user. In all four tests, only the third and the fourth test is conducted on 10 human subjects. The subjects are required to wear a virtual reality headset and perform specific tasks. The subject's performances are then recorded for evaluation.

Figure 20: Test Subjects Wearing the Headset

The first test focuses on the functionality of the system. This phase has been conducted multiple times in parallel with the designing phase and the development phase. The test identifies issues and any potential issue or errors made. Once the problem has been identified, the project cycles back through the designing and development phase. The final result produced is an interactable interface where the user is able to view the dual image camera stream while manipulating the 3D robotic arm:

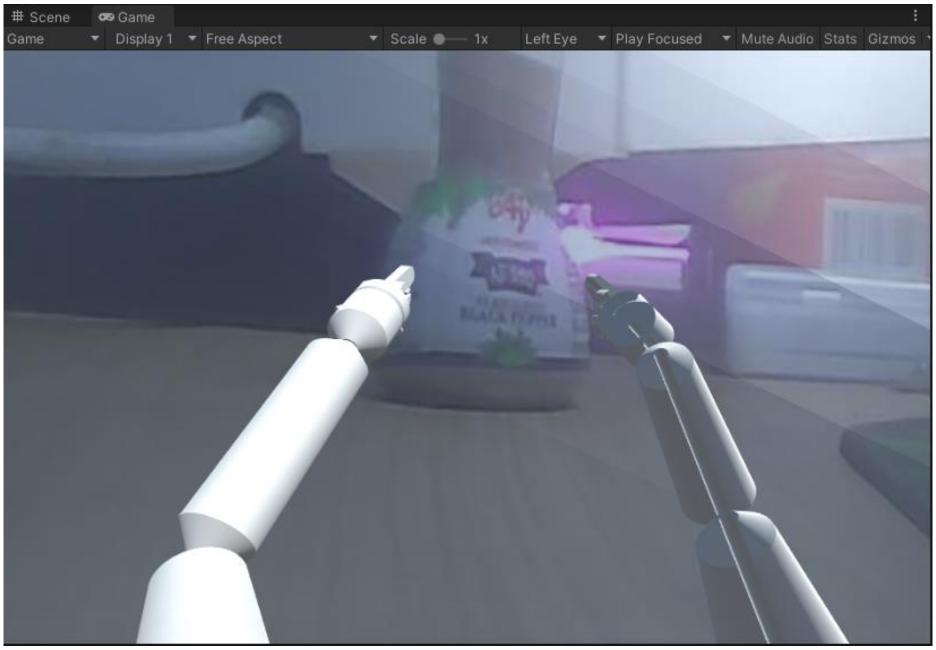
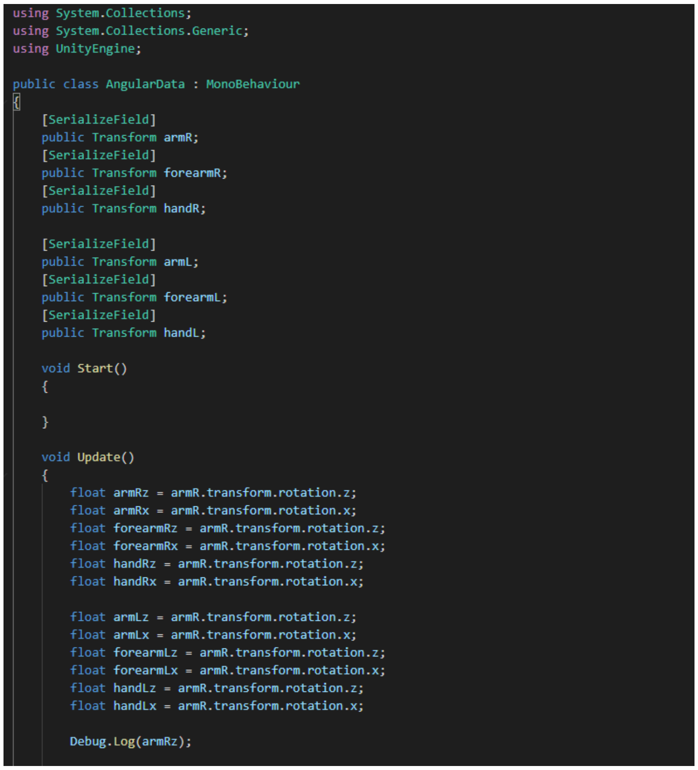


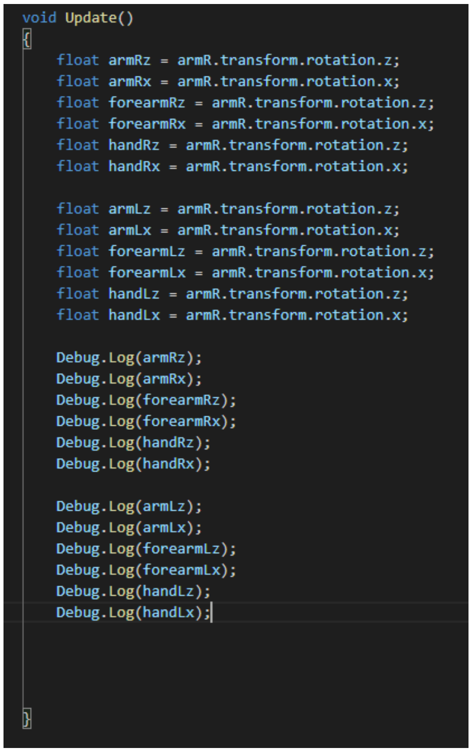
Figure 21: User's View of the Interface

The second test focuses on the ability of the system to produce usable angular data for robotics application. The test required the system to consistently output float data that indicates the current angle for two axes for each of the joint of the 3D robotic arm. The output of this data is viewed through the console log of Unity engine. The first part of this test will only prioritize on one axis from one joint of the arm model. For this part, the joint that is chosen is the upper arm. The required data will have a float range of -1.00 to 1.00, indicating the current angle of the joint in one axis. The script written to extract the data is as shown below. Afterwards, the second part of this test will include the output of data from all axes and joints of the robotic arm.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AngularData : MonoBehaviour
{
    [SerializeField]
    public Transform armR;
    [SerializeField]
    public Transform forearmR;
    [SerializeField]
    public Transform handR;

    [SerializeField]
    public Transform armL;
    [SerializeField]
    public Transform forearmL;
    [SerializeField]
    public Transform handL;

    void Start()
    {

    }

    void Update()
    {
        float armRz = armR.transform.rotation.z;
        float armRx = armR.transform.rotation.x;
        float forearmRz = armR.transform.rotation.z;
        float forearmRx = armR.transform.rotation.x;
        float handRz = armR.transform.rotation.z;
        float handRx = armR.transform.rotation.x;

        float armLz = armR.transform.rotation.z;
        float armLx = armR.transform.rotation.x;
        float forearmLz = armR.transform.rotation.z;
        float forearmLx = armR.transform.rotation.x;
        float handLz = armR.transform.rotation.z;
        float handLx = armR.transform.rotation.x;

        Debug.Log(armRz);
```

Figure 22: Script to Extract Data from One Axis of One Joint



```
void Update()
{
    float armRz = armR.transform.rotation.z;
    float armRx = armR.transform.rotation.x;
    float forearmRz = armR.transform.rotation.z;
    float forearmRx = armR.transform.rotation.x;
    float handRz = armR.transform.rotation.z;
    float handRx = armR.transform.rotation.x;

    float armLz = armR.transform.rotation.z;
    float armLx = armR.transform.rotation.x;
    float forearmLz = armR.transform.rotation.z;
    float forearmLx = armR.transform.rotation.x;
    float handLz = armR.transform.rotation.z;
    float handLx = armR.transform.rotation.x;

    Debug.Log(armRz);
    Debug.Log(armRx);
    Debug.Log(forearmRz);
    Debug.Log(forearmRx);
    Debug.Log(handRz);
    Debug.Log(handRx);

    Debug.Log(armLz);
    Debug.Log(armLx);
    Debug.Log(forearmLz);
    Debug.Log(forearmLx);
    Debug.Log(handLz);
    Debug.Log(handLx);




}
```

Figure 23: Script to Extract Data from All Axis of All Joints

The third test focuses on the effectiveness of the augmented reality functionality. To conduct this test, the author must produce a controlled environment and compare it to the environment where the augmented reality functionality is implemented. From the previously mentioned method of producing the "Ghosting" visual effect, there are two robotic arm models. The first model works as an augmented reality visual assistance where the tip of the arm follows the position of the user's hand without any delay. The second one is a simulation of a real robotic arm where it follows the final position of the first model with a predetermined delay set to 0.5 seconds. This is to simulate real input and feedback transmission latency. In the controlled environment, the first model is removed, forcing the test subject to control the arm model with the predetermined delay. The test subject is told to open their arm as wide as possible and within 1.5 seconds, close and touch the tip of both arm model. The test subjects are told to touch the tip of both the arm to as close as possible and not overlap.

Figure 24: Test Subject Performing the Action (Open)

Figure 25: Test Subject Performing the Action (Close)

The subjects are given 10 tries to perform this action. In each try, the following condition of the arm model is recorded.

- If the tip of the arm overlaps each other, it means that the subject failed to perform the action. This simulates over-stimulated input. The threshold to assume this condition is when more than 50% of the width of the tip of the arm model overlap with each other.

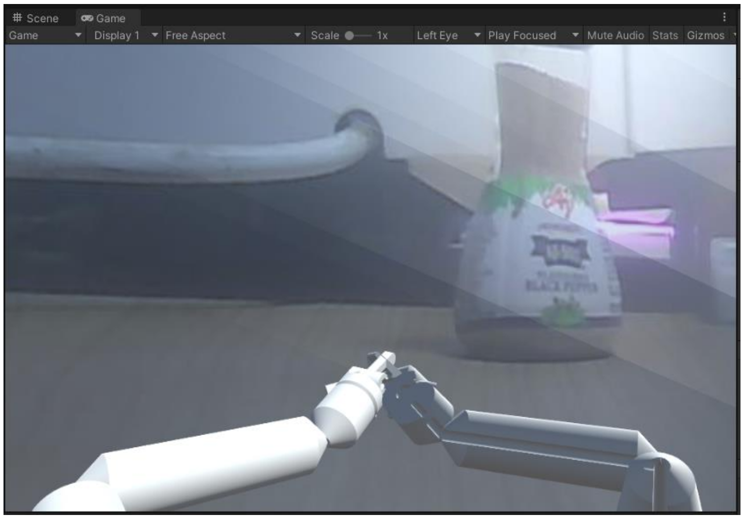


Figure 26: The Tip of the Arm Model Overlaps

- If the tip of the arm does not touch each other, it means that the subject failed to perform the action. This simulates under-stimulated input. The threshold to assume this condition is when the distance between the tip of the arm model is more than 50% of the width of the tip of the arm model.

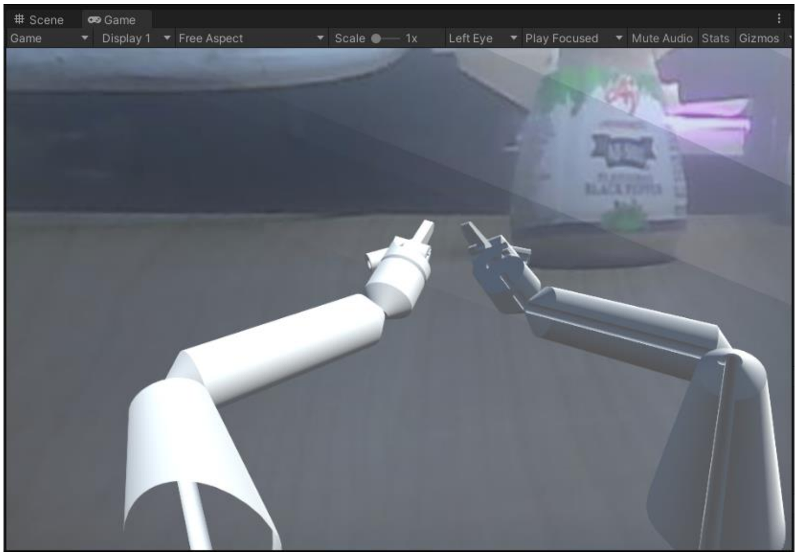


Figure 27: The Tip of the Arm Model Does Not Touch

- If the tip of the arm touches with each other, it means that the subject successfully performed the action.

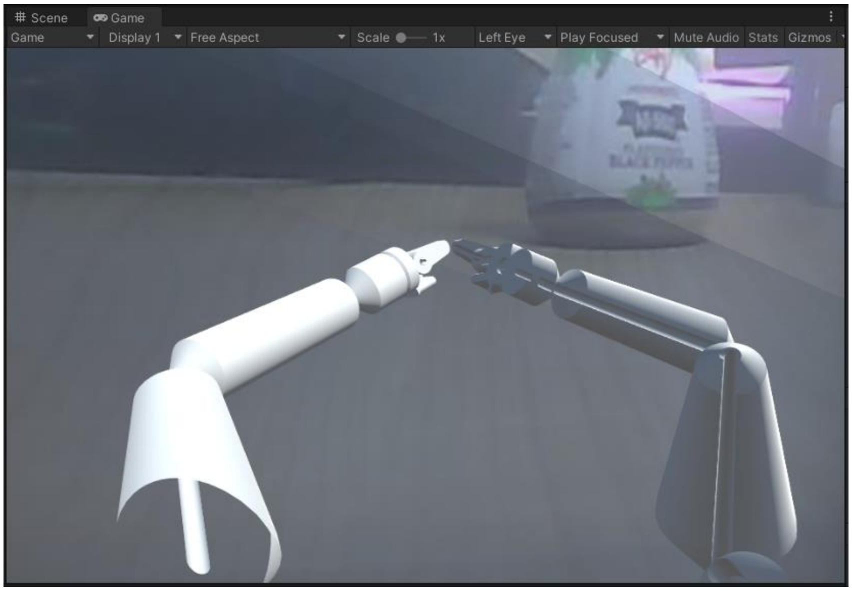


Figure 28: The Tip of the Arm Model Touches

Afterwards, the subjects are told to perform the same action but with the "Ghosting" visual effect enabled. The data gathered between the controlled environment and the "Ghosting" visual effect enabled are then compared and analysed.

The fourth test focuses on the effectiveness of implementation stereoscopy vision. This refers to the ability to produce depth of perception to the user. To conduct this test, the author must also produce a controlled environment and compare it to the environment where stereoscopy vision is applied. To perform this, three square paper cut-outs are made with each of the squares having different sizes but with the same ratio of width and height.

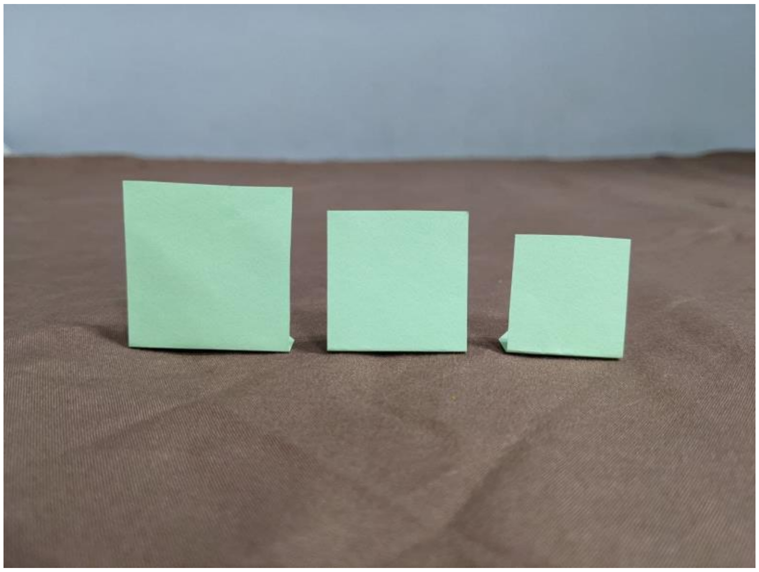


Figure 29: Paper Square with Different Sizes

The three paper squares are then placed at three different distances lateral to the position of the webcam. The three paper squares are specifically placed at a distance in which, through a non-stereoscopic camera, will appear as if all three squares have the same size and distance. The test subjects (while wearing the headset) are told to differentiate which of the three paper squares are the furthest, the middle-most, and the closest to the camera. The test subject's performance will only be considered successful if they manage to differentiate all three correctly. The test is repeated 10 times with each of the distance of the three paper squares swapped with each other between all 10 tries.

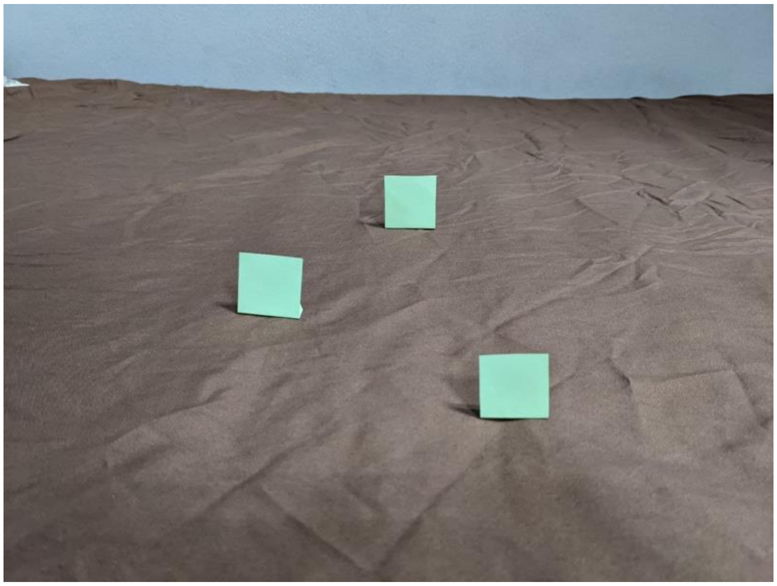


Figure 30: Paper Squares Placed at Different Distances

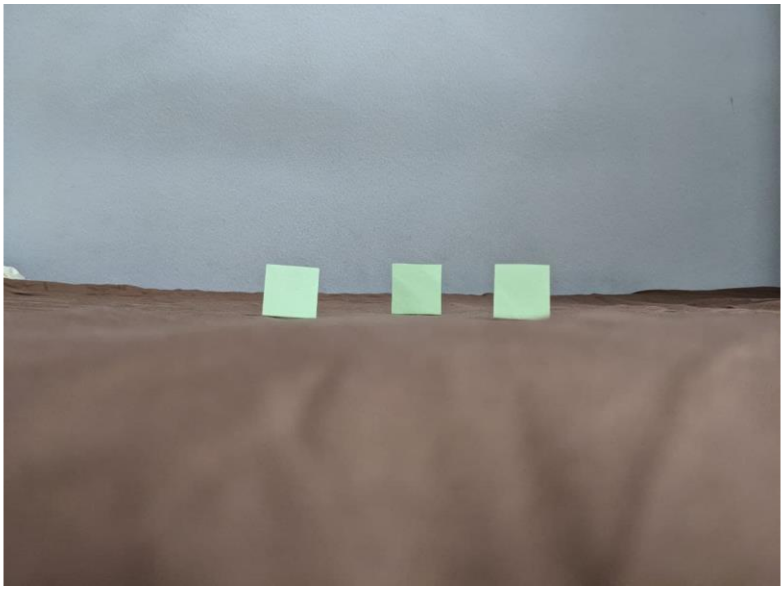


Figure 31: Paper Squares Viewed from a Non-Stereoscopic Camera

For the controlled environment, the author will disable the left display of the dual image camera stream, forcing the test subjects to perform the test while only using the right display of the stream. This simulates non-stereoscopic camera vision. Afterwards, after the test is conducted in a controlled environment, the author re-enables the left display of the dual image camera stream. This allows for the test subjects to view the three paper squares with stereoscopic vision. The test subjects are then told to perform the same actions again. The data gathered between the controlled environment and the stereoscopy vision enabled are then compared and analysed.

Figure 32: Stereoscopic Vision Disabled

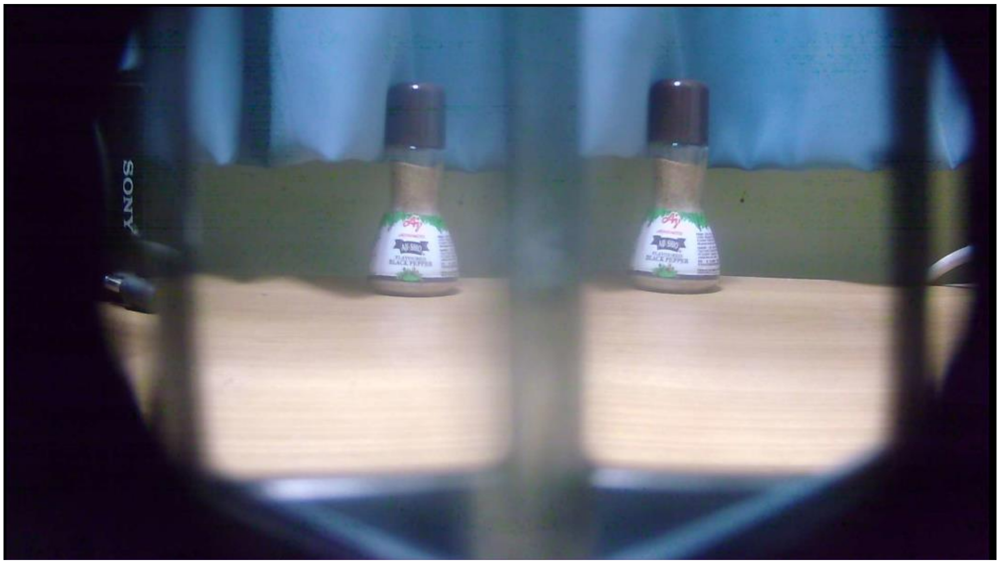


Figure 33: Stereoscopic Vision Enabled

CHAPTER 4: RESULTS AND DISCUSSION

4.1 User Interface

This section will discuss on the final results of the user interface. According to the testing phase in the previous chapter, the desired outcome on this section is the ability for the user to interact with the system. This refers to the ability of the user to view the dual image camera stream while manipulating the 3D robotic arm model. The ability of the user to interact with the system is essential for conducting other tests and producing results. Without a working interface, test subjects are not able perform the desired task mentioned in the testing phase section in the previous chapters. The following figure proves the interactability of the system by showing the user interface being used by a test subject:

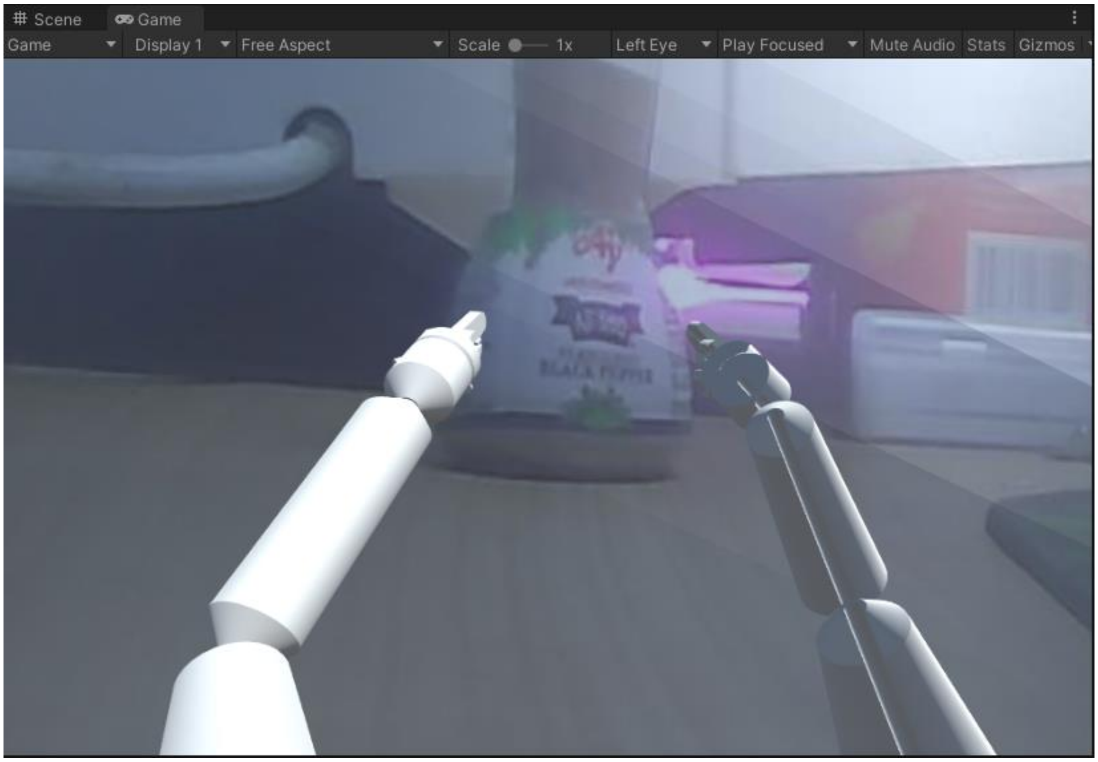


Figure 34: User Viewing the Dual Image Camera Stream

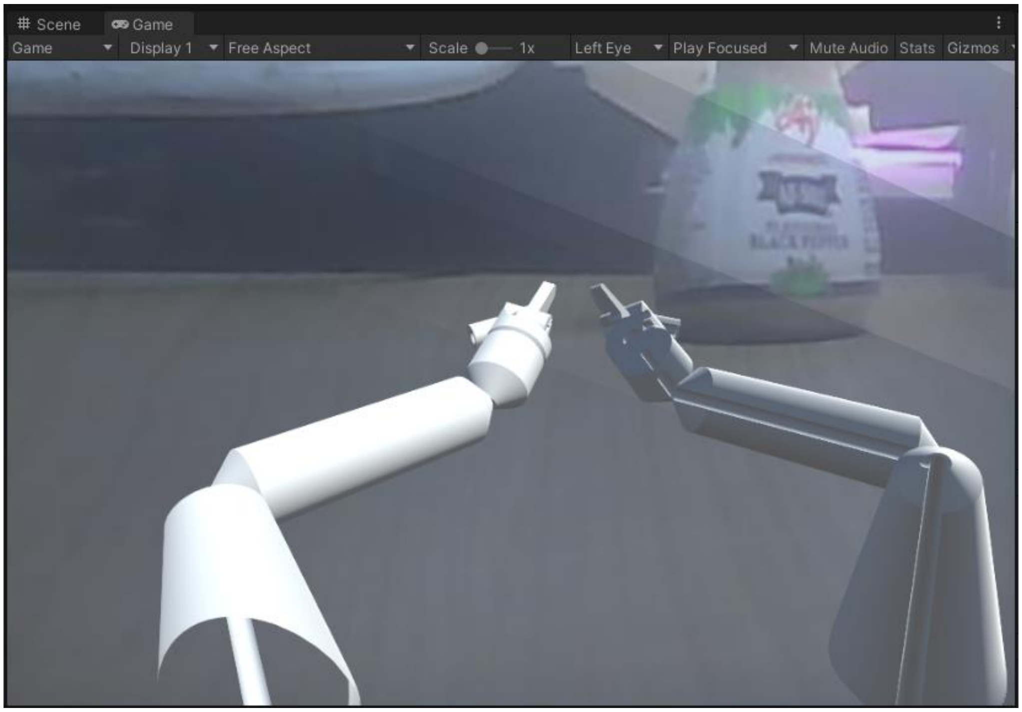


Figure 35: User Manipulating the 3D Robotic Arm

4.2 Producing Angular Data

This section will discuss on the ability of the system to produce data from the motion of the user’s hand. There is two parts in this section. The first part prioritizes on producing angular data from only one axis from one joint of the robotic arm. The result from this part of the test shows that the data received from moving one axis of the joint produce float data ranging from -1.0 to 1.0, indicating its current angle. The figure below shows the resulting data from moving the robot arm.

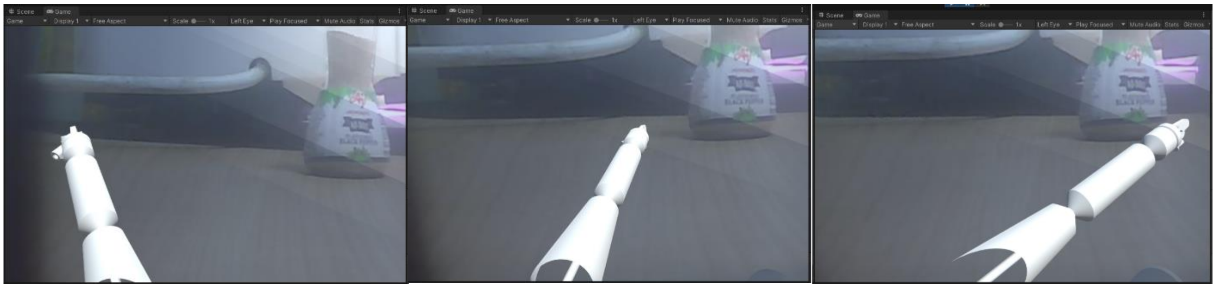
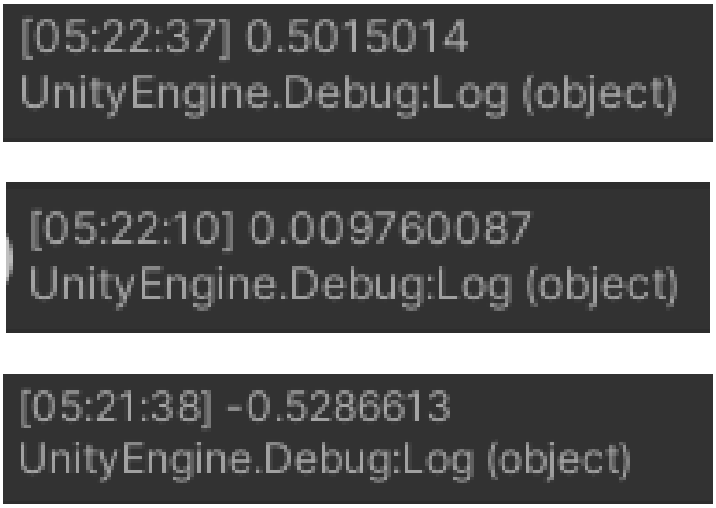


Figure 36: Moving the Arm Model Max left, centre, and Max Right



```
[05:22:37] 0.5015014
UnityEngine.Debug:Log (object)
```

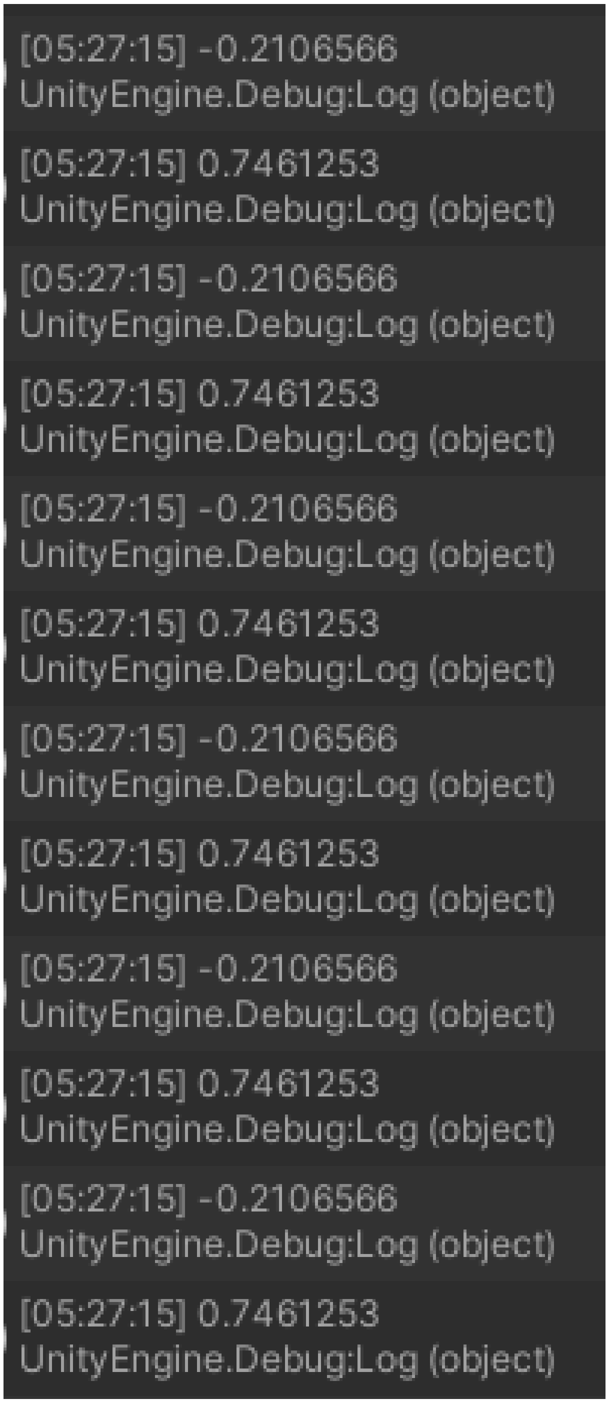
```
[05:22:10] 0.009760087
UnityEngine.Debug:Log (object)
```

```
[05:21:38] -0.5286613
UnityEngine.Debug:Log (object)
```

Figure 37: Data Received for Arm Model Max left, centre, and Max Right Respectively

From the first part of this section, the system is able to produce the desired data. Successfully performing this action means that the script used to extract this data can be repeated for every axis on every joint of the robotic arm. The results prove that the system is able to perform the mentioned requirements in the previous chapter. The following figure shows the data produced for all axes in every joint in the sequence of the following:

a. Right arm Z axis rotation
b. Right arm X axis rotation
c. Right forearm Z axis rotation
d. Right forearm X axis rotation
e. Right hand Z axis rotation
f. Right hand X axis rotation
g. Left arm Z axis rotation
h. Left arm X axis rotation
i. Left forearm Z axis rotation
j. Left forearm X axis rotation
k. Left hand Z axis rotation
l. Left hand X axis rotation



```
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
[05:27:15] -0.2106566
UnityEngine.Debug:Log (object)
[05:27:15] 0.7461253
UnityEngine.Debug:Log (object)
```

Figure 38: Data Produced

4.3 Visual Assistance for Input and Feedback Transmission Delay

This section will discuss on the effectiveness of the augmented reality functionality. From the test conducted, the author compared the data retrieved from the controlled environment with the environment by which the augmented reality functionality is enabled. The "F" in the table below marks that the test subject failed to perform the action while "S" indicates that the subject managed to perform the action successfully.

Human Subjects	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trail 8	Trial 9	Trail 10	Score	Success rate
Subject A	F	F	F	F	S	S	F	F	S	F	3/10	30%
Subject B	F	F	F	S	S	F	F	S	F	F	3/10	30%
Subject C	F	F	F	S	S	F	F	F	F	F	2/10	20%
Subject D	S	F	F	F	S	S	S	F	F	F	4/10	40%
Subject E	F	F	F	F	F	F	S	S	F	F	2/10	20%
Subject F	F	F	F	F	F	S	S	F	F	F	2/10	20%
Subject G	S	S	S	F	F	S	S	F	F	F	5/10	50%
Subject H	S	F	F	F	F	F	F	S	S	F	3/10	30%
Subject I	F	F	F	F	F	S	F	F	F	S	2/10	20%
Subject J	F	F	F	F	S	F	F	F	F	F	1/10	10%

Figure 39: Result from Controlled Environment

Human Subjects	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trail 8	Trial 9	Trail 10	Score	Success rate
Subject A	S	S	S	F	S	S	F	S	S	S	8/10	80%
Subject B	F	F	S	S	S	S	S	S	S	S	8/10	80%
Subject C	F	S	S	S	S	F	S	S	S	S	8/10	80%
Subject D	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject E	S	S	F	S	F	S	S	S	F	S	7/10	70%
Subject F	S	S	S	S	S	S	S	F	S	S	9/10	90%
Subject G	S	S	S	F	S	S	S	S	S	F	8/10	80%
Subject H	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject I	F	S	S	F	S	S	S	S	S	S	8/10	80%
Subject J	S	S	S	S	S	S	S	S	S	F	9/10	90%

Figure 40: Result with Augmented Reality Enabled

From the table above, results from the controlled environment have an average success rate of 27%. With the augmented reality enabled, the result shows an average success rate of 85%. This result shows an increase of 58% when the augmented reality is enabled, proving that the concept of using augmented reality as a visual assistance is effective in adapting to the simulated input and feedback transmission latency.

4.5 Stereoscopic Vision

This section will discuss on the effectiveness of implementing stereoscopy vision in achieving depth of perception to the user. From the test conducted, the author compared the data retrieved from the controlled environment with the environment by which the stereoscopy vision is enabled. The “F” in the table below marks that the test subject failed to perform the action while “S” indicates that the subject managed to perform the action successfully.

Human Subjects	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trail 8	Trial 9	Trail 10	Score	Success rate
Subject A	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject B	F	F	F	F	F	F	S	F	F	F	1/10	10%
Subject C	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject D	F	F	F	F	F	F	F	S	F	F	1/10	10%
Subject E	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject F	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject G	F	F	F	F	F	F	F	S	F	F	1/10	10%
Subject H	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject I	F	F	F	F	F	F	F	F	F	F	0/10	0%
Subject J	F	F	F	F	F	F	S	F	F	F	1/10	10%

Figure 41: Result from Controlled Environment

Human Subjects	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trail 8	Trial 9	Trail 10	Score	Success rate
Subject A	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject B	S	F	S	S	S	S	S	S	S	S	9/10	90%
Subject C	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject D	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject E	S	S	S	S	F	S	S	S	S	S	9/10	90%
Subject F	S	S	S	S	S	S	S	F	S	S	9/10	90%
Subject G	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject H	S	S	S	F	S	S	S	S	S	S	9/10	90%
Subject I	S	S	S	S	S	S	S	S	S	S	10/10	100%
Subject J	S	S	S	S	S	S	S	S	F	S	9/10	90%

Figure 42: Result with Stereoscopy Vision Enabled

From the table above, results from the controlled environment have an average success rate of 4%. With the stereoscopy vision enabled, the result shows an average success rate of 95%. This result shows an increase of 91% when the stereoscopy vision is enabled.

In the controlled environment, subject B, D, G and J managed to successfully perform the task of distinguishing the distance between all three paper squares with a successful rate of 1 out of 10. The author speculates that the test subjects may have distinguished the distance of the paper squares due to the change of focus from the limitation of the camera resolution. Objects tends to appear pixelated when placed at a further distance. The author also speculates that this success rate may have been due to luck. Therefore, the author assumed that all the test subjects are not able to differentiate the distance of all the three paper squares while in a non-stereoscopic view.

Comparing the result between the controlled environment and the stereoscopy vision enabled. The author concludes that implementing stereoscopy vision into the system successfully produced depth of perception to the user. This is proven by the success rate from the test with stereoscopy vision enabled compared to the controlled environment where it is disabled.

CHAPTER 5: CONCLUSION AND FUTURE WORK

Overall, the project was developed successfully in addressing the problem statements and meeting the author’s requirements and objectives. In this project, the implementation of virtual reality proved that it is a feasible medium for robotics control. The project was able to prove the effectiveness of implementing six degrees of freedom (6DOF) to reduce the complexity of control by producing data that is usable for robotics application. Furthermore, implementing augmented reality proved to be effective in adapting with the delay that occurs in an input and output transmission. Lastly, the project successfully replicated depth of perception by implementing stereoscopy vison into the system.

Initially, the scope of the project was to include the development of an articulating robotic arm as a test bed for the system. The scope has been reduced to only include the development related to software. To cope with the lack of a physical testbed, the author carefully devised an alternative method to test the concept within a software environment (some hardware are used). The results achieved from this test proved the effectiveness and feasibility of implementing 6DOF, AR/VR and Stereoscopic Vision into articulating robotics control.

If there is a potential chance by which this study will continue its path, future work will include testing with a real articulating robotic arm. The success of future works may potentially bring a revolutionary control system where it is applicable in an unlimited number of applications. The utmost goal of the study of this technology is the applications of robotic telesurgery, hazardous task performance and interplanetary exploration. The author is open of further study if there is demand for this system.

REFERENCES

1. Scott, N. (2016, August 10). *Is Da Vinci Robotic Surgery a Revolution?* Healthline. Retrieved December 5, 2021, from https://www.healthline.com/health-news/is-da-vinci-robotic-surgery-revolution-or-ripoff-021215

2. Workplace Injury & Fatality Statistics. (2019), Retrieved from https://www.esfi.org/workplace-safety/workplace-injury-fatality-statistics/

3. Hofmann, P. (n.d). *What are the biggest risks to divers who perform underwater welding?* Hofmann & Schweitzer. Derived from https://www.hofmannlawfirm.com/faqs/risks-of-underwater-welding-accidents-for-maritime-workers.cfm

4. Sivcev, S., Coleman, J., Omerdic, .E, Dooly, G.,& Toal, D. (2018) *Underwater manipulators: A review*. Ocean Engineering, 2018.

5. Guarana-DIY (2020) *"The Top Six Types of Industrial Robots in 2020"*. DIY-robotics. Retrieved December 5, 2021.

6. "Worries about premature industrialisation". *The Economist.* Retrieved on December 5, 2021 from https://web.archive.org/web/20171021014710/https://www.economist.com/news/special-report/21729864-automation-less-threat-workers-emerging-world-it-made-out

7. Batalie, J. (2013) *"An introduction to Positional Tracking and Degrees of Freedom (DOF)"*. Road to VR. Retrieved December 6, 2021.

8. “Degrees of Freedom | Google VR”. *Google Developers.* Retrieved December 6, 2021.

9. Paul, Richard P. (1981) *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, 1981.

10. Howard, I.P., Rogers, B.J. (1995) *Binocular Vision and Stereopsis.* Oxford University Press, New York.

11. Wheatstone, C. (1838) *Phenomena of Binocular Vision.* King’s College, London.

12. Levanda, R., Lesham, A. (2010) *“Synthetic aperture radio telescope”.* IEEE Signal Processing Magazine.