VR Robot

Final Report

Zijian Song

Levi McAdams

Samuel Torres

Eric Hinds

Tyler Allen

12/03/2023

Fall 2023 – Spring 2024

Project Sponsor: Dr. Reza Razavian

Faculty Advisor: David Willy

Sponsor Mentor: Dr. Armin Eilaghi

DISCLAIMER

This report was prepared by students as part of a university course requirement. While considerable effort has been put into the project, it is not the work of licensed engineers and has not undergone the extensive verification that is common in the profession. The information, data, conclusions, and content of this report should not be relied on or utilized without thorough, independent testing and verification. University faculty members may have been associated with this project as advisors, sponsors, or course instructors, but as such they are not responsible for the accuracy of results or conclusions.

EXECUTIVE SUMMARY

For the Summer 2023 476C session, our team was tasked with developing a VR (virtual reality) Robot out of a previous capstone project called R.U.T. (robot utility tank). The goal was to take the previous project, implement a robotic arm, and complete VR controls. A product such as this has many applications. In summary, this project can be used to perform tasks in places where a human could not or should not be present. To achieve this, the team was given a $2500 base budget along with a list of measurable requirements to attain. The project was split into 3 areas. Area, one requires the team to make improvements to the R.U.T. project. The R.U.T. project fell short of its goals in terms of mobility. The team has addressed these shortcomings alongside other minor concerns. Secondly, the team needed to find a path for implementation of a robotic arm. We had the choice of improving a device on the market or developing one from scratch. Lastly, the implementation of the VR controls will tie the two mechanical components together including complete user control over both. This report's purpose is to discuss this process in all details of our capstone project's development and progress.

The design of our VR Robot is as follows:

RUT Base

This is the primary robot. The RUT is a unit handed down to our group from a previous capstone. The use of the RUT is primarily for movement and world interaction. This robot is what allows the project to function as a VR project in some regard. By allowing the robot to move within the real world, our team can attach cameras and use virtual reality headsets to visualize these movements from theoretically anywhere in the world.

Robotic Arm

The Robotic arm is a prebuilt one, known as the Braccio++. The Arduino based robotic arm is a basic budget arm designed for ease of use. Because of this the group has decided to use the Braccio++ in hopes that we can perfect our control method early on within the project. Additionally, to ensure adequate workspace with the arm the links were redesigned to be longer and stronger.

VR Control

The VR Control of this project is of the utmost importance. That said, it was also the most difficult part of the project. A full program was designed within the VR gaming engine known as Unity. This program communicates with the Uduino app to send commands to the braccio arm, while also displaying live video feed to the headset user. Complete remote control of all systems was achieved through this platform.

ACKNOWLEDGEMENTS

We would like to acknowledge staff members that were key to the success of this project by providing the team with helpful insight. We would like to thank David Willy for providing us with the tool necessary to successfully achieve every customer and engineering requirement. We appreciate all the technological help that we received from Cole. We thank our clients Dr. Armin Eilaghi and Dr. Reza Razavian for their support and time making sure that we were on the right track. Finally, we thank the machine shop staff for helping us manufacture the parts that were necessary to finish our project.

TABLE OF CONTENTS

Table of Contents

1 BACKGROUND

1.1 Introduction

For the Summer 2023 486C course, group 1 has been tasked with developing a VR (virtual reality) Robot. The goal of this project is to have control of a robot and robotic arm through a VR Headset. Given the world's industries push towards automation, hands off controls, and virtual reality, this project will serve as a platform for all parties involved to get familiar with the technology. A robotic device such as this could serve many useful purposes in later development. These include but are not limited to; surgery performed by a doctor who cannot be present, defusal of an explosive device, planetary exploration, service of machines in hazardous areas, and any other tasks needing human intervention where one cannot be present, or the risk of being present is deemed to have a high risk of injury or fatality.

1.2 Project Description

Following is the original project description provided by the sponsor.

"Virtual reality (VR) controlled robots are robots that can be controlled remotely by a human with a VR headset. These robots can be used for a variety of Mechanical Engineering applications such as gripping, welding, cutting, and lifting heavy objects while being controlled by a human operator. There are several advantages of using VR-controlled robots. One advantage is that VR-controlled robots can be used to perform tasks that are dangerous for humans. Another advantage is that VR-controlled robots can be used to perform tasks that are difficult for humans. For example, VR-controlled robots can be used to perform tasks in manufacturing that require precision and accuracy. Additionally, VR-controlled robots can be used to perform tasks in healthcare that require a high level of accuracy.

VR-controlled robots have many applications, including in manufacturing, healthcare, and education. For example, Covariant.AI uses VR, artificial intelligence, and various deep learning techniques to teach robots to perform certain tasks.

This project will be built on the existing capstone project finished in 2022 at the Mechanical Engineering Department. A full metal frame housing the electrical components and a track system made in-house was designed. This robot can carry a load of 200 lbs. and 2 mph.

The goal of this capstone project is to add a VR-controlled arm to the robot. This arm will be controlled with a VR Meta Quest 2 headset and mimics the movements of the individual hand. Here is a schematic of the remotely controlled arm using a VR.

The capstone team, in collaboration with the client, will decide on the tasks to be performed by the Robot. The components needed for a VR-controlled robot can vary depending on the type of robot and its intended use. For example, a VR-controlled robot designed to take the place of humans in emergency situations would require different components than a VR-controlled robot designed for manufacturing. Some components that may be needed for a VR-controlled robot include sensors, cameras, and actuators. The robotic arms are controlled by Raspberry Pis, which are running web-based servers designed to accept input from the VR controllers. This input is translated into movements for the robotic arm."

2 REQUIREMENTS

In this chapter, the reader will find five main aspects of the client's requirements. First of all, the customer requirements and their weights will be discussed showing the completion/progress of each one. The engineering requirements will also be studied in this chapter followed by the functional decomposition and house of quality. Finally, this chapter will conclude with an overview of the standard, codes, and regulations specific to this project. It is important to understand that all the information provided in this chapter is accurate to the current stage of the project.

2.1 Customer Requirements (CRs)

Table 1 discusses the customer requirements, their weights, and a brief justification of the weight. Weights are on a scale of 1-5.

Customer Requirements	**Weight**	**Justification**
Increase Maneuverability	4	Robot base should be able to move without limitation.
Accurate VR Control	5	Input should be reflected as an output. No random movement.
Easy to Use	2	A user should be able to learn controls quickly.
Large Workspace	2	Important to be able to perform tasks in front of the robot.
Rigid Design	3	Given the product's application, it is crucial for it to perform without failure.
Durable	3	The device should not break under regular use.
Fast Controls	3	It is important that the robot responds fast to the user's input.
Cost Effective	4	Maximum budget ceiling of $2500
Powerful	2	Robotic arm should be able to lift regular objects.
Multidirectional Control	4	The arm should be able to rotate.

Table 1: Customer Requirements

2.2 Engineering Requirements (ERs)

The written scope for this project did not include any engineering requirements, however, the team has made a list of engineering requirements that need to be met to successfully complete the customer requirements mentioned above. In the following table the ER's are listed along with the target and justification.

Engineering Requirement	Target Value	Justification
Decreased Turn Time	35 degrees/second	High turn speed for mobility
Low Visual Latency	<50 ms	User can see movements in real time
Large Workspace	0.5 meters	Robot can reach necessary task objective.
Decreased Turn Radius	<2 meters	Small footprint to turn for increased mobility.
Reasonable Factor of Safety	2	High factor of safety to minimize failures.
Program Speed	400 ms	Low program speed for quick input processing.
Minimum Payload	0.25 kg	Payload for light duty tasks like moving objects.

Table 2: Engineering Requirements

2.3 Functional Decomposition

The main function of this project is to allow controlling the robot tank with arm under a first person view of VR. The VR should provide the user a free view on the robot chassis then enable control based on that view.

The function of the project is decomposed into Camera, Control Board and Robotic Arm. Camera is used to allow the user to have a view around the robot and create a valid vision space. Control Board is to activate the movement combining RUT and robotic arm then enable solo operation. Robotic Arm is the core of the replacement operation of users, it is used to grab items and hold it for transfer. The purpose and relationship of these functions will be illustrated in the following sessions.

2.3.1 Black Box Model

This is the Black Box Diagram of the project. The input is the operation command of VR and handle, a signal input, operated at the user's will. The output is the robot motion, it is a signal and energy output, which needs to have the ability to grab items, and transfer it through a distance on the ground. The camera function does not show up for it forms a loop inside of the black box to help the user identify the position of item and RUT and find a way to transfer it.

The project's final purpose is to have the robot tank with arm grab items and transfer it the way the user wants through VR control. The Black Box model illustrated the necessary input and output. It clarified the final purpose and necessary operation mode, so the function model and designs are to activate this operation.

Figure 1: Black Box Model

2.3.2 Functional Model/Work-Process Diagram/Hierarchical Task Analysis

The work progress is based on a dual-centered system. The PC serves as a user portal and the control board serves as a robot portal. The PC allows the combination of viewing and controlling for the user, we use a software as a middleman (currently it is Unity), it receives the view space from a camera and sends it to VR thus forming a free view for the user. Also, the portal collects input from the VR handle, processes it to a valid operation command then transfers it to the control board. The control board receives the command and applies it on RUT and arm.

This diagram provides a vision of how each function connected physically and functionally. It indicates the necessary hardware connection method and programming. The PC needs to have a remote connection to the camera and control board transferring massive data from camera and minor data and command to the control board. Programming is necessary on PC to process the input from the VR handle and make it into a first stage signal then send to the control board. The signal received on the control board needs to be transferred into a valid operation command applying one robotic arm and RUT.

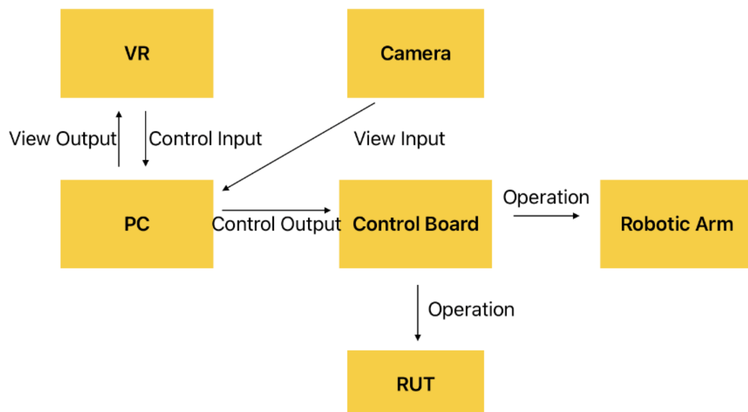


Figure 2: Work-Process Diagram

2.4 House of Quality (HoQ)

The House of Quality serves to relate the customer requirements, engineering requirements, and provide a benchmark of the current products. In the top section, the interaction between engineering requirements is shown using + or - to show a positive or negative correlation. Technical requirements are listed in the lower section along with their absolute and relative technical importance values. These values will help the group by showing the most key areas of the project to focus on. **See Appendix B for a larger view of the table.**

System QFD

Project: VR Robot
Date: 07-06-23

Technical requirement correlations	Decreased Turn Time	Increased Torque Advangtage	Low Program Speed	Low Latency	Increase Robotic Arm Length	High Network Speed
Decreased Turn Time						
Increased Torque Advangtage	+					
Low Program Speed						
Low Latency			-			
Increase Robotic Arm Length		-				
High Network Speed			-	-		
High Material Strength					+	

Legend

A	CALIBER Mk3 EOD
B	Andros F6
C	Wheelbarrow MK9

Customer Needs	Customer Weights (1-5)	Decreased Turn Time	Increased Torque Advangtage	Low Program Speed	Low Latency	Increase Robotic Arm Length	High Network Speed	High Material Strength	1 Poor	2	3 Acceptable	4	5 Excellent
		Technical Requirements							Customer Opinion Survey				
Cost Within Budget	4		6	4	8	4	6	4	ABC				
Durable and Robust Design	3	4	6					4					ABC
Reliable Design	3	6	6	4			4	6					ABC
Safe to Operate	4			6				4			BC		
Move in 3D	1	5	1										ABC
Large Work Space	2	3				9							ABC
Uniform Continuous Force Capacity	2		5	2									ABC
High Mechanical Stiffness	2		3					3					ABC
High Resolution Sensing	4				6		8			A			BC
Low-Latency Communications	3			3	7		8				AB	C	
Fast Update Rate	3			8	5		8				A	BC	
Technical Requirement Units		Seconds	Newtons	Seconds	ms	m	Mb	GPa					
Technical Requirement Targets		10	5<	0.5	50	0.5	30	55					
Absolute Technical Importance		41	69	89	92	30	116	68					
Relative Technical Importance		6	4	3	2	7	1	5					

Table 3: House of Quality

2.5 Standards, Codes, and Regulations

In order to be competitive in the market the project has to meet certain rules and regulations that adhere to ISO and ANSI regulations. Achieving the following standards and codes listed below will help the team design and build the project up to code. There are many codes and standards that have been taken into consideration that are not in the table below, this is because they have been followed and met by external entities like the manufacturer of a certain product that the team purchased, previous capstone team, welding gear used for the project, and machinery used to put together the project.

Standard Number or Code	**Title of Standard**	**How it applies to Project**
AISI S240	North American Standard for Cold-Formed Steel Structural Framing	Helps in the design of how the device with interface with the user in a safe manner.
AWS B2.1/B2.1M:2021	Standard Welding Procedure Specifications (SWPS)	Provides practices under ANSI regulations that will prevent any failure due to poor welding practices.
ISO 15072:2012	Hexagon Bolts with Flange with Metric Fine Pitch Thread - Small Series - Product Grade A	Provides reliability to all the parts that will be held by nuts and bolts. This will help choose the nuts and bolts appropriate for the project.

ISO/TC 270	Plastics and rubber machines	Provides insight when seeking to purchase the right PLA for 3D printing.
IEC 60034-1 +A1 and A2	Electric Motor Standards as Defined by the IEC and the Harmonized European Standard	Rotating electric motors. Part 1: Rating and design
IEC 60034-6	Electric Motor Standards as Defined by the IEC and the Harmonized European Standard	Rotating electric motors. Part 6: Cooling (IC code)
IEC 60034-14	Electric Motor Standards as Defined by the IEC and the Harmonized European Standard	Rotating electric motors. Part 14: Mechanic vibration for machines with drive shaft heights of 56mm or more. Measuring, estimate and vibration limits
IEC 60072-1	Electric Motor Standards as Defined by the IEC and the Harmonized European Standard	Dimensions and output power for rotating electric motors. Part 1: Frame size 56 to 400 and flange size 55 to 1080.
ISO/TR 19247	Camera Testing Guidelines	Guidelines for reliable testing of digital still cameras describes best practices for performing tests of digital cameras, including test criteria, conditions, protocols, and documentation, as well as the training of personnel for reliable testing.

Table 4: Standards of Practice

3 DESIGN SPACE RESEARCH

3.1 Literature Review

The team utilized many different resources to help benchmark and use ideas that would be beneficial for the project's progress. These sources included both websites, articles, and textbooks. Applications of these sources varied. Below is a list of 23 sources which aided in the development of the VR Robot capstone along with a brief description of each.

1. The Definitive Guide to VR Video Streaming and 360° Video, https://www.dacast.com/blog/virtual-reality-vr-live-streaming-and-360-video-a-primer/ This source provides guides of streaming views to VR. It is helpful in transferring the view from camera to VR.

2. VR in Unity: Managing Controller Input and Hand Presence, https://sneakydaggergames.medium.com/vr-in-unity-managing-controller-input-and-hand-presence-part-1-controller-set-up-792682dd024d This source provides an introduction of VR controller input settings. This will help build the control system allowing VR controller and including all VR controller functions.

3. Unity VR: How to get Controller Data. https://www.youtube.com/watch?v=Kh_94glqO-0 This source provides a method of getting VR controller data. It is directly helpful in coding.

4. How does VR controller position works?, https://discussions.unity.com/t/how-does-vr-controller-position-works/243775This article in a forum helps having an understanding of VR controller position. It is useful in achieving an accurate control mode by controller position apart from the joystick.

5. Sending data from Unity to Raspberry, https://stackoverflow.com/questions/38816660/sending-data-from-unity-to-raspberry The post and answer in the forum provides a method for sending data from unity 3D. It allows the control function to step into the next step.

6. T. Braccio, "Tinkerkit Braccio Robot | T050000," Arrow.com, 2023. https://www.arrow.com/en/products/t050000/arduino-corporation?gclid=CjwKCAjws7WkBhBFEiwAIi1681DYQ-TGPbyy8P58Ew9W3bvChbhh1ShSgY_97JfVlLpMje8MVtXb9hoCp-gQAvD_BwE&gclsrc=aw.ds (accessed Jul. 08, 2023).

7. "Free CAD Designs, Files & 3D Models | The GrabCAD Community Library," grabcad.com. https://grabcad.com/library/arduino-braccio-robotic-arm-1 (accessed Jul. 08, 2023).

8. "SOLIDWORKS - Editing Imported Models," www.youtube.com. https://www.youtube.com/watch?v=292CRndUJss (accessed Jul. 08, 2023)

9. "HS-311 Standard Economy Servo | HITEC RCD USA," hitecrcd.com. https://hitecrcd.com/products/servos/analog/sport-2/hs-311/product (accessed Jul. 08, 2023).

10. "SpringRC SR431 Servo Specifications and Reviews," servodatabase.com. https://servodatabase.com/servo/springrc/sr431 (accessed Jul. 08, 2023).

11. Mechanism design and motion ability analysis for wheel/track mobile ...,

https://journals.sagepub.com/doi/full/10.1177/1687814016679763 (accessed Jun. 18, 2023). This source will cover the basics for understanding how to design a track based system. It will act as an engineering handbook as well as a great source to pull suggestions from when it comes to making improvements to our current designs.

12. "Continuous track," *Wikipedia*, Aug. 12, 2021. https://en.wikipedia.org/wiki/Continuous_track This source will act as a reference guide for different track designs since their invention. While it is common knowledge that Wikipedia shouldn't be used as a scholarly source. The website can still be a great tool to introduce concepts and ideas to research in detail when the time comes for it.

13. P. Allen, "Models for the Dynamic Simulation of Tank Track Components Defence College of Management and Technology." Available: https://core.ac.uk/download/pdf/40081469.pdf In this report from the defence college of management and technology, the process of design, analysis, and testing of a tank track is covered. It works to introduce new concepts to the team to take into consideration when making changes to the current system.

14. Ulsoy, A. & Whitesell, Joseph & Hooven, M.. (1985). Design of Belt-Tensioner Systems for Dynamic Stability. Journal of Vibration Acoustics Stress and Reliability in Design. 107. 282. 10.1115/1.3269258. A tensioner for the track has also been an idea to help improve function. With this being considered, this source will act as a guide in order to perform analysis and create a design that will work for our application.

15. A. Olatunde, "DESIGN & ANALYSIS OF A TENSIONER FOR A BELT-DRIVEN INTEGRATED STARTER- GENERATOR SYSTEM OF MICRO-HYBRID VEHICLES." Accessed: Jul. 06, 2023. [Online]. Available: https://tspace.library.utoronto.ca/bitstream/1807/17209/1/Olatunde_Adebukola_O_200811_MASc_thesis.pdf This source will act as a second reference to have a successful design of our tensioning system for the track. The goal is to have the tensioner in place to increase the friction of the wheels to the track, and also act as a form of suspension.

16. .kekenai, "通过SSH远程连接树莓派(Remotely connect to Raspberry Pi via SSH)" ,2020, https://ost.51cto.com/posts/913 This source introduces a way connecting raspberry pi to PC using SSH, it is helpful to transfer files from PC to raspberry pi.

17. "如何使用 Windows 笔记本电脑作为 Raspberry Pi 的监视器(How to Use a Windows Laptop as a Monitor for Raspberry Pi)", https://cn.moyens.net/how-to/167315/ This article introduces the way of using a laptop as the screen of raspberry pi to help operating raspberry pi settings. It is helpful in the testing stage that it is not necessary to buy a screen for raspberry pi currently.

18. Kris Hauser,"Robotic Systems", 2023, https://motion.cs.illinois.edu/RoboticSystems/ This is a tuitional website about robotic systems, it is produced by a professor in UIUC. This source provides detailed information of how robotic systems work including theoretical aspects like inverse kinematics. It is a good reference during the whole process of the project.

19. Unity User Manual (5.6),https://docs.unity3d.com/560/Documentation/Manual This source is the manual of Unity, it contains details of coding manners for Unity-based programmes. It helps locate the input from the VR controller and enables creating a link to the robot control system.

20. Raspberry Pi Arduino Serial Communication – Everything You Need To Know,

https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/ This source introduces the way of building connections from raspberry pi to arduino board, which is the terminal board of the current robotic arm with a control program inside. The source helps close the final part of the loop in the control system.

21. "The less-IS-more approach to robotic cable management," igus, https://www.igus.com/info/robot-cable-management-solutions (accessed Jul. 7, 2023). Although the information is mostly about bigger robotic arms and their cabling this can still provide important to our project in giving advice on how to wire the arm effectively without having issues with the cable getting caught or not having enough length for full extension.

22. T. Flynn, "A Primer on Electronics & Wiring in FRC," Chief Delphi, https://www.chiefdelphi.com/ (accessed Jul. 7, 2023). Although a wiring guide for a completely different system it can still be extremely useful in properly wiring all the connections between the different pieces of the robot.

23. Zivid, 2D vs 3D machine vision, https://www.zivid.com/2d-vs-3d-machine-vision (accessed Jul. 7, 2023). An article to help with the explanation of 2D and 3D cameras and to better help choose what is best for our project.\

3.2 Benchmarking

Due to the project's scope, the team has had no need to conduct on-site visits, but a lot of online research has been done. The team has gathered all the information from reliable sources that include NAU staff members and online research. The team is reverse engineering most of the aspects/areas of the current project, the objective being to decrease research time, money spent, and increase in reliability of the final end product since the last thing the team wants to do is to reinvent the wheel. Currently the area that the team is trying to benchmark against the current products in the market is the VR to robot connection. Something to look forward to as the team moves forward is to implement existing augmented reality software to the project since the final goal is to be able to reproduce human motion on a robot remotely.

3.2.1 System Level Benchmarking

Below a list of products available in the market is shown. Each one of the existing designs meet most or at least one of the requirements for this project. These designs have been used to benchmark our design and have helped the team with the design of the final product.

3.2.1.1 Existing Design #1: Arduino Braccio ++

This robotic arm has the capability to connect to VR and has the ability to move in six axes. The team has bought an arm awfully close to this one since this arm performs the required tasks provided by the client.

[1] "Arduino Braccio ++," Arduino Online Shop. https://store-usa.arduino.cc/products/braccioplusplus?selectedStore=us (accessed Jul. 06, 2023).

3.2.1.2 Existing Design #2: Crazy Robot Using VR

This video shows a robotic arm that is completely controlled by VR, it uses an interface that allows the viewer to see in a 3D space similar to Unity. One of the downsides of this arm is that it uses a 2D camera which makes it difficult for the user to detect or feel depth. This system helps the team by showing potential issues that the team can run into by using 2D cameras.

[2] "I Remotely Control A Crazy Robot Arm Using VR!," www.youtube.com. https://www.youtube.com/watch?v=qoyIVGB8OOk (accessed Jul. 08, 2023).

3.2.1.3 Existing Design #3: Teleoperated Mobile Manipulator

UF Factory has developed a human-robot interface software for non-robotic experts to teleoperate and program robotic manipulators remotely for physical tasks. The team is aiming for something similar to this product since it meets most of the requirements provided by the client. This product will be helpful to benchmark our product against the given requirements.

[3] https://zh-hk.facebook.com/Ufactory2013, "Case Study - Extended Robotics | UFACTORY," Feb. 23, 2023. https://www.ufactory.cc/user-case-extended-robotics/ (accessed Jul. 08, 2023).

3.2.2 Subsystem Level Benchmarking

The following 3 subsystems show different designs that the team has considered and is still considering.

3.2.2.1 Subsystem #1: Cameras

Below are 3 existing designs that have been considered by the team. In this section 3 different designs with 3 different cameras will be discussed in more detail.

3.2.2.1.1 Existing Design #1: 2D Camera(s)

A 2D camera would meet the least of the requirements since the user will not be able to have depth perception. This design can be used for non-depth sensing applications. This existing design can be a backup plan for the team but considering that this design meets the least requirements then we will try to stay away from using this design as much as possible.

3.2.2.1.2 Existing Design #2: 3D Camera

A 3D camera will provide the user with not only depth perception but also will enable the user to see the environment in infrared. This option meets all the requirements of the project due to its depth sensing capabilities. A 3D camera will be attached to the top of the RUT looking towards the direction of the arm. This will provide the user with a view of not only the arm but also of the close surroundings around the arm.

3.2.2.1.3 Existing Design #3: 360 Camera

A 3D camera allows the user to see in every direction, this would help the user to be aware of the surroundings without having to be limited to a 2D/3D view. Unlike the 3D camera, the 360 Camera does not count with depth sensing capabilities. Due to its design a 360 camera will not meet all the requirements provided by the client.

3.2.2.2 Subsystem #2: Control Boards

In the following section 2 different designs with 2 different control boards will be discussed.

3.2.2.2.1 Existing Design #1: Raspberry Pi

This board consists of a microcomputer that has the power to process data at a speed of 700 MHz to 1.4 GHz. This board will be used as the middleman between the VR/computer to the robotic arm/RUT. Due to its versatility the raspberry pi board has proven to be the best choice so far for the project.

3.2.2.2.2 Existing Design #2: Arduino Uno

This board already includes libraries that are designed to work with the robotic arm purchased by the team. Due to the way Arduino handles data it might not be a good idea to use Arduino as the middleman between the VR and the arm, however, more research is being done to conclude as to whether or not a raspberry pi or an Arduino board would be the best option.

3.2.2.3 Subsystem #3: Robotic Arm

One of the main subsystems of the project is the robotic arm. The team is currently working on the connection between the VR and the arm. Below is a list of two potential ways to have the robot achieve the goal proposed by the client.

3.2.2.3.1 Existing Design #1: Two Arms

Having two arms attached on top of the RUT can help the user to be able to use both arms to control the two robotic arms. The only issue foreseen by the team is that we might run out of controllers to control the RUT if the two VR sticks are already being used to control both arms.

3.2.2.3.2 Existing Design #2: Single Arm

A single arm is capable of reaching to the ground and grabbing objects of a weight less than 0.5 kg. This arm will be attached on the center of the robot or closer to the front of the RUT. The position of the arm on the RUT is still under consideration and research.

Below is an image that best represents the subsystems and the existing designs.

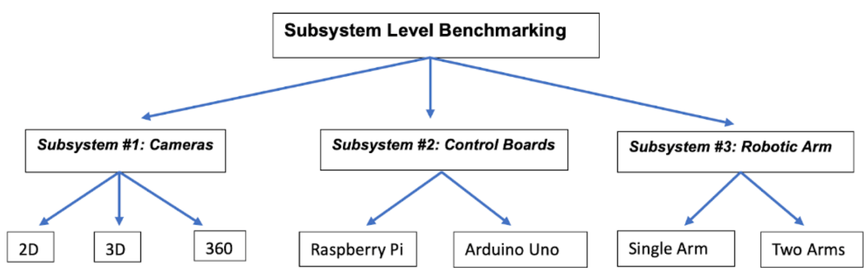


Figure 3: Subsystem Level Benchmarking

4 CONCEPT GENERATION

4.1 Full System Concepts

All full system designs would generally be using an Arduino to control the robotic arm while using a Raspberry Pi interacting directly with the Arduino while also controlling everything else and interfacing with the VR headset and a computer middleman if needed.

4.1.1 Full System Design #1: Singular Robotic arm with only 2D camera(s)

For the singular arm that only has 2D cameras the cameras would be the part that holds the design back the most. With 2D cameras they have poor depth perception as it really harms the ability to use the arm in a capacity that is required of the team. With multiple 2D cameras there can be multiple views of the robot and the surroundings. This design would work in most environments and since it is attached to RUT, it can be moved around wherever needed with a bigger workspace than a fixed arm.

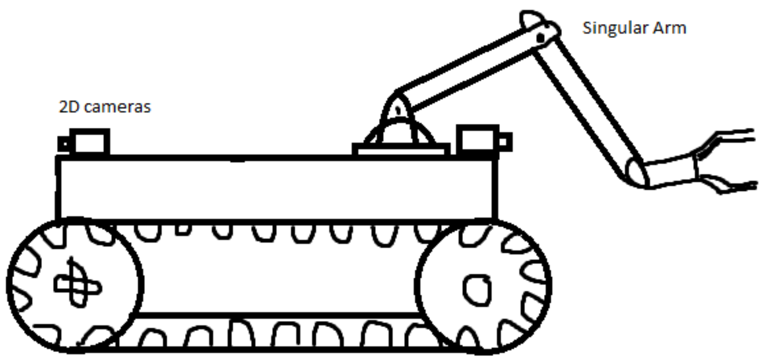


Figure 4: 2D Camera, Single Arm

4.1.2 Full System Design #2: Singular Robotic Arm with 3D Camera and Supporting 2D Camera(s)

Having a singular arm that has a 3D camera with 2D cameras is a great option as it helps mitigate a key issue with having only 2D cameras of which is depth perception. The 3D camera would help in showing the distance between the arm and the object that is being grabbed. With the 3D camera pointing towards the arm and its workspace it can also help with environments that have no light source as the 3D camera has no need for light. It also has the benefit of using the 2D cameras to move around the robot and workspace if there is light.

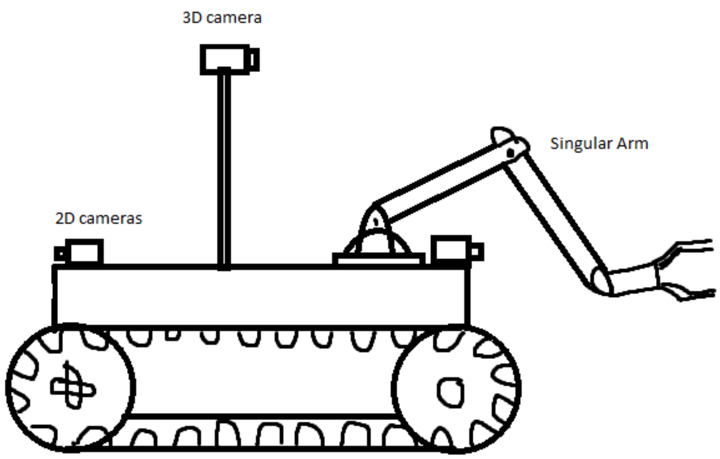


Figure 5: 3D Camera, Single Arm

4.1.3 Full System Design #3: Two Robotic Arms with 360 camera and 2D cameras

Utilizing two robotic arms would give a big increase in available workspace for the robot. The design would help meet the customer requirements with better effectiveness but has significantly more downsides than just the singular arm. The main issue holding back the two-arm design is the coding and controlling of the arms since there are limited available controls to be able to control both arms on top of RUT's movements. Additionally, the control of two arms depending on the mounting location could be confusing and hard for the user to control. This design would have the benefit of having more that it could achieve with two arms. For camera choice for this design a 360 camera with 2D cameras would be the cheapest and most viable solution for two arms, otherwise you would need even more 2D cameras or a second 3D camera for the other arm.

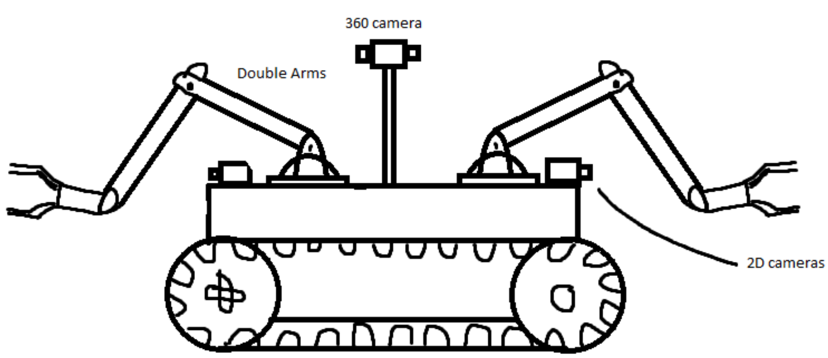


Figure 6: Two Arm, 360 Camera

4.2 Subsystem Concepts

Breaking down the designs into smaller sections is beneficial in showing the options and possibly finding a better solution. Along with each section or subsystem it can be shown that there are different designs that the team could have chosen and what purpose they would serve along with the pros and cons of each design.

4.2.1 Subsystem #1: Cameras

The camera is one of the most important parts of the robot as it allows the user who would be wearing a VR headset to view the robot, its surroundings, and most importantly the arm. Choosing a camera is difficult since each camera has different specifications and uses.

4.2.1.1 Design #1: 2D Camera(s)

A 2D camera or stereo camera would be a common camera that can be found almost anywhere that does not do anything particularly special. With the 2D camera there would more likely than not be multiple cameras to get multiple angles to be able to get all information needed. 2D cameras have the benefit of being easily integrated and are often cheaper than most other camera variants. On the other hand, 2D cameras often have really poor depth perception when viewing video data.

4.2.1.2 Design #2: 3D Camera

A 3D camera can also be known as a depth sensing camera that senses the area around it in infrared instead of the normal RGB like the 2D camera. This would be able to stream the live data of distance to an object in a color format where that color is a specific distance. A depth camera would be great for this application as it would help with the distance the arm has to travel while also being able to work in a pitch-black environment since it would view in infrared instead of viewable RGB. The main downside of a 3D depth camera is the increased price point over the 2D camera and the possible difficulty in integrating it into VR.

4.2.1.3 Design #3: 360° Camera

A 360° camera is typically two back-to-back 2D cameras with a heavily curved fisheye lens to be able to view in all directions. This would help with viewing the entire area around the robot, including behind without additional cameras. 360° cameras usually have worse depth perception than the 2D camera because of the fisheye lens distorting the image. Another massive issue with the 360° camera is that they can get fairly expensive for a good quality image.

4.2.2 Subsystem #2: Control Boards

The control board is the brains of the entire operation; without it the camera would not be able to connect with the VR headset and the arm would not be able to function. A proper board with sufficient operating power and usability is a necessity as controlling the arm, RUT, and streaming the live data back to VR can be intensive.

4.2.2.1 Design #1: Raspberry Pi

A Raspberry Pi is a small form factor computer used for many projects. It has an extremely high amount of modularity in what can be added and what can be done with the board. The Raspberry Pi is also a cheap option for control boards that has more refined options and since it is an actual minicomputer the operations that can be done with the Raspberry Pi can be very complex. There are thousands of already made codes for the Raspberry Pi that we can reference for ideas and solutions.

4.2.2.2 Design #2: Arduino

An Arduino is a super simple control board with a microcontroller that is not able to handle super complex programming. Arduino has the benefit of being easily programmable with super simple to use coding and like the Raspberry Pi has a vast library of scripts.

4.2.3 1.2.3 Subsystem #3: Robotic Arm

The robotic arm is a more straightforward subsystem as it was decided early on, the only design difference is whether only one arm is chosen or if two arms would be better.

4.2.3.1 1.2.3.1 Design #1: Singular Arm

For both the single arm and two arms the team would purchase the Tinkerkit Braccio robotic arm and 3D print some of the linkages at an extended length than what comes out of the box to reach the workspace requirement while also upgrading the servos to something more powerful to be able to pick up heavier objects at the extended length. Upgrading the servos from what comes with the kit is a necessity as with the original servos at the extended length it would be projected to not be able to pick up any decently sized objects as the torque requirements would be high. only having a singular robotic arm for the robot while having a workspace most likely only on the side the arm is attached to would be cheaper, easier to work with, and super simple to control.

4.2.3.2 1.2.3.2 Design #2: Two arms

As stated in the singular arm section the same arm would be used for the two-arm setup but with two arms give significantly more workspace to be able to use the arm. the main downside of having two arms for the system is the method of operating the robot is meant to be fully VR controlled so controlling both arms while also controlling the RUT bases movements all in VR with one controller would be immensely complex and is something that would be discussed on implementation if the team was interested after the integration and successful operation of the singular arm.

5 DESIGN SELECTED – First Semester

Listed below is the description following team 1's decision of our first semester "Final Design". Within section 5.1 will be an in-depth description of the final design, engineering calculations completed, 3D models, etc. Within section 5.2 the reader will encounter the future plan of our design. This future plan includes not only the schedule, but also the bill of materials and a detailed explanation of how the schedule is to be followed.

5.1 Technical Selection Criteria

Within the outline of the project, group 1 was given some technical criteria to abide by throughout the project. Some specific requirements set by our customer Dr. Reza, was that the robot has to: Move in a 3D space, have a uniform force capacity, to have a high mechanical stiffness and durability, to have low latency, and these are just a few. As the team inspected the RUT, we came up with some engineering requirements as well. These may include a better turn radius for the RUT, and quicker turning speed. We did the same for the robotic arm, and as the team became more familiar with the technology, we started coming up with quantifiable engineering requirements for this system as well. Some of these requirements may include a specific ultimate strength of the arm, minimum torque required, or even the arms available radius. All of these criteria were used in the creation of the Pugh Chart as well as the Decision Matrix. Listed below are the figures of the charts stated above.

Customer Requirements	**One-arm 2D Camera**	**One-arm 3D Camera**	**Two-arm 2D Camera**	**Two-arm 3D Camera**
Move in 3D	3	4	3	4
Large Workspace	3	3	4	4
High-resolution Sensing	3	4	3	4
Accurate VR control	4	5	2	3
Easy to use	4	4	2	2
Total	17	20	14	17

Table 5: Decision Matrix

Critical quality	Weight	One-Arm 2D Camera	One-Arm 3D Camera	Two-Arm 2D Camera	Two-Arm 3D Camera
Increase Manueverability	5	+	+	-	-
Accurate VR Control	5	0	+	-	0
Easy to use	5	+	+	-	-
Large Work Space	3	0	0	+	+
Rigid Design	2	+	+	0	0
Durable	2	+	+	0	-
Fast Controls	4	0	+	-	-
Powerful	3	0	0	+	+
Multidirectional Force	4	+	+	0	0
	Total +	5	7	2	2
	Total 0	4	2	3	3
	Total -	0	0	4	4
	Total	18	27	-13	-10

Table 6: Pugh Chart

5.2 Rationale for Design Selection

The team has decided to move along with Full Design Idea #2. This design consists of a singular robotic arm with a main 3D camera and supporting 2D cameras. This design was found to be the best for our group due to the connection which can be seen in the Decision Matrix and Pugh Chart. Using the Pugh Chart, the team was able to qualitatively narrow down the decision for each individual part. This was used to decide the importance of each component as well as likelihood of incorporation into the final result. Using the Decision Matrix, the team was able to use the narrowed down designs and look at the deep specifics of each part. What made each part important in their own way? How each part compares to one another in a very easy to visualize way. To break this down with existing data, in this section the goal is to explain the importance of each CR followed in the Decision Matrix specifically.

A masterclass provided by Professor Peter Corke which goes in depth on the understanding of 3D space [5]. For our project specifically this is incredibly important due to the interaction with the 3D environment around the robot.

A large workspace is self-explanatory. With the current size of the robot, it would only be of actual use in a large workspace. That being said, throughout the research and process of this project the group aims to be able to further implement this information into future projects.

High resolution sensing is becoming increasingly important in robotics. As the reader can see within source [6] by Science Network, the possibility of high-resolution sensing could result in "...more advanced robot manipulation…". This is important for our team because if we can find a robotic arm, or components for the robotic arm to help with the control, this would be a huge help.

Ease of use is another pretty self-explanatory topic. By incorporating parts that make the process not only simpler, but easier to use, this helps with the overall completion of the project. If we can incorporate a simpler design, maybe this will allow more perfection to the design.

Even though we decided that the best option was #2 we ended up utilizing a 2D camera instead of a 3D. This is because we ran into issues connecting the camera to the VR headset. This change can be seeing in any of the images of the final project in the pages below.

6 Project Management – Second Semester

6.1 Gantt Chart

Project Management Assignment - ME486

NORTHERN ARIZONA UNIVERSITY

Legend: On track | Low risk | Med risk | High risk | Unassigned

Project start date: 12/3/2023

ME 486C VR Robot Team 1

Milestone description	Category	Progress	Start	Days
RUT				
Create and implement a track tentioner system	Med Risk	100%	12/3/2023	0
Desing and manufacture wheel	Med Risk	100%	12/3/2023	0
Design a better ON/OFF system	Low Risk	100%	1/12/2024	0
Implement AI receiver inside the RUT and make necsessary connections	High Risk	100%	12/18/2023	0
Paint and work on asthetics (optional)	Low Risk	100%	1/12/2024	0
Robotic Arm				
Re-print custum arm links	On Track	100%	12/3/2023	0
Design and manufacture a RUT connection base	On Track	100%	12/3/2023	0
Upgrade joint servos (TBD)	Low Risk	100%	12/3/2023	0
Provide stress analysis	On Track	100%	12/10/2023	0
VR/FPV				
Make decison as to whether to stay with VR or switch to FPV	Goal	100%	12/12/2023	0
Connect headset to the RUT and have two way comunication	High Risk	100%	12/3/2023	0
Fully control RUT wirelessly	High Risk	100%	12/3/2023	5
Purchace required VR/FPV parts	Low Risk	100%	12/3/2023	0
Etras				
Complete all purchases	On Track	100%	12/3/2023	0
Assemble all secctions above together.	High Risk	100%	11/9/2023	0
Draft of poster	Goal	100%	11/2/2023	0
Final poster and PPT	Goal	100%	11/16/2023	1
Finalize testing plan	Med Risk	100%	11/3/2023	0
Final CAD Packet	High Risk	100%	12/3/2023	23
Final report and website check	Goal	100%	2/20/2024	1
Client Handoff - Spec Sheet & Operation/Assembly Manual	Goal	0%	12/9/2023	1
Hardware Status Update - 33+% build	Low Risk	100%	11/12/2023	1
Hardware Status Update - 67+% build	Med Risk	100%	10/19/2023	1
Hardware Status Update - 100% build	High Risk	100%	11/9/2029	1
Product Demo & Testing Results in class	High Risk	100%	12/3/2023	20

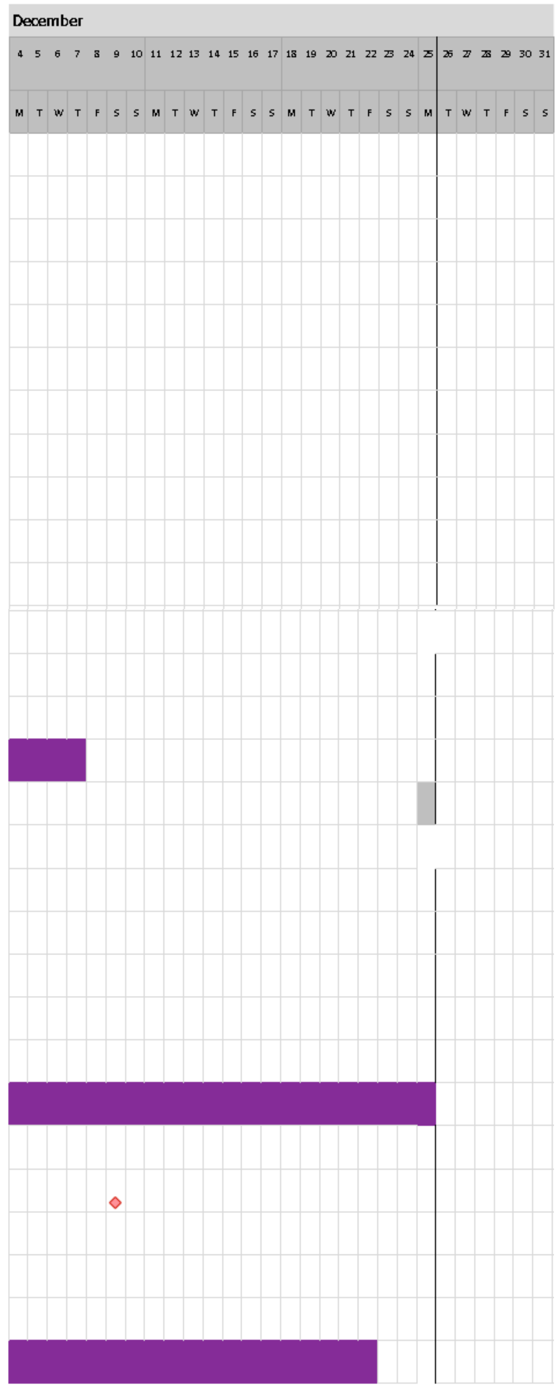


Table 7: Final Gantt Chart

When looking at the Gantt chart, it should be known that there are 4 main categories making up the graph. These are the same categories from the original Gantt chart: RUT, VR/FPV, Robotic Arm, and Extras. Each of these has subcategories corresponding to what the team had decided to move forward with throughout the semester. Some examples being: creating the tensioner system for the RUT, fully controlling the robot wirelessly, or even designing and manufacturing the roller brackets for the tracks. If we are to look at the original Gantt chart from the beginning of the semester, the largest difference is the completion of each task. Within the finalized chart, the only remaining task is to distribute the product to the client, or "Client Handoff". This includes a full spec sheet of RUT, as well as an Operation and Assembly manual, to allow the client ease of use for future implementations. Overall, the team did an incredible job completing each task, with little to no complications. This chart allowed the team to stay on task and move forward at an incredibly effective rate, thus giving the team more time for whatever problems we did have. The only thing that the team most likely could have done better was the implementation of the VR/FPV systems. This system took by far the longest, due to the incredibly difficult level of computer science application needed, and as such if we had provided more time, we may have been able to troubleshoot a bit more for a much more refined product. Our product was up to client standards and is in working order, so it is hard to say that this was a 'problem', but rather a hiccup that the team did not foresee requiring the amount of time it did.

6.2 Purchasing Plan / Manufacturing Plan

For the team to successfully complete this project and move along smoothly, purchasing parts was a major part of the process. The purchasing plan was fairly simple, as the purchasing manager (Eric) would oversee all purchase orders, to make sure there was little to no confusion. When a part was needed, the team would discuss and come to a consensus. After our decisions were made, it would then fall on Eric to create and finalize the purchase order. It was imperative that the team was on the same page, so before any emails were sent in regard to each PO, we would sit down and make any final changes as a team. Once the final copy was made, Eric would submit it to the class supervisor, clients, and dean's office for verification. Once verified by each party involved, the order would be seen through by the dean's office and parts were distributed to the team within a proper time frame. Below is a finalized BOM depicting each item purchased, as well as each item manufactured along with the materials used to do so:

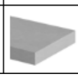
BOM Number	Image	Description	Link	Manufacturer	Quantity	Price ea.	Price Total:	Ordered (Y/N)	Manufactured (Y/N)	Recieved/Finished (Y/N)
1		Spring 22' Capstone Project	-	RUT Team	1	-	-	-	-	-
2		High strength PLA rolls for 3d printing	pre/l/pro-serie	MatterHack	4	57.00	228.00	Y	-	Y
2.1		Arduino compatible robotic arm	https://www.a	VR Robot	1	103.74	103.74	Y	-	Y
2.2		Robotic arm base	-	VR Robot	1	-	-	-	Y	Y
2.3		Arm mounting base plate	-	VR Robot	1	-	-	-	Y	Y
2.4		Arm mounting lock plate	-	VR Robot	1	-	-	-	Y	Y
2.5		Extended segment for roboarm	-	VR Robot	2	-	-	-	Y	Y
2.6		Tensioner bolt mounting block	-	RUT Team	2	-	-	-	Y	Y
2.7		Drive wheel	-	VR Robot	2	-	-	-	Y	Y
2.8		Driven wheel	-	VR Robot	2	-	-	-	Y	Y
2.9		Tensioner support bracket		VR Robot	2	-	-	-	Y	Y
3		Stock material for roller bracket	v.mcmaster.co	McMaster-Carr	1	96.86	96.86	Y	-	Y

Table continues on the next page.

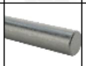
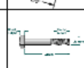
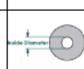
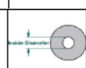
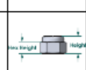
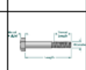
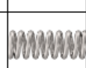
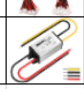
3.1		Roller bracket	-	VR Robot	2	-	-	-	Y	Y
4		Stock material for roller shaft	master.com/8	McMaster-Carr	1	38.09	38.09	Y	-	Y
4.1		Roller Shaft	-	VR Robot	4	-	-	-	Y	Y
4.2		Roller Mount	-	VR Robot	2	-	-	-	Y	Y
5		Roller bracket center bearing	2rs-ball-beari	Bearings Direct	2	3.75	7.50	Y	-	Y
6		Roller shaft bearing	2rs-ball-beari	Bearings Direct	4	7.40	29.60	Y	-	Y
7		Center bearing retention rings (pack)	mcmaster.com	McMaster-Carr	2	7.47	14.94	Y	-	Y
8		Shaft bearing retention rings (pack)	mcmaster.com	McMaster-Carr	4	15.38	61.52	Y	-	Y
9		Front wheel bearing retention rings (pack)	mcmaster.com	McMaster-Carr	4	8.27	8.27	Y	-	Y
10		Meta Quest 2 VR Headset	tent=67227933	Meta	1	300.00	300.00	Y	-	Y
11		Oil-Embedded sleeve bearing		McMaster-Carr	4	5.40	21.60	Y	-	Y
12		8x1.25x180mm bolt	om/Product-D	Bolt Depot	2	5.14	10.28	Y	-	Y
13		washer 8mm	om/Product-D	Boltdepot	2	0.11	0.22	Y	-	Y
14		wide washer 8mm	om/Product-D	Bolt Depot	4	0.19	0.76	Y	-	Y
15		8mm nuts	om/Product-D	Bolt Depot	8	0.17	1.36	Y	-	Y
16		8x1.25x35mm bolt	om/Product-D	Bolt Depot	4	0.65	2.60	Y	-	Y
17		Compression spring (pack)	.mcmaster.co	McMaster-Carr	2	13.88	13.88	Y	-	Y
18		Arduino UNO board	https://vilros.	Arduino	1	52.99	52.99	Y	-	Y
19		New primary arm servo	https://www.	BETU	1	64.99	64.99	Y	-	Y
20		FPV transmitter (Camera)	hvlocint&hvk	AKK	1	35.99	35.99	Y	-	Y
21		FPV reciever	https://shop.	MakerFire	1	25.99	25.99	Y	-	Y
22		Camera battery	https://www.	HRB	1	37.99	37.99	Y	-	Y
23		JST connector	WNJ&keywor	Smseace	1	7.98	7.98	Y	-	Y
24		Voltage stepdown	41188076193	EKYLIN	1	11.99	11.99	Y	-	Y
			Total Spent:		1177.14	Percent Ordered:	100%			
						Percent On-Hand	100%			
						Percent Manufactured:	100.00%			

Table 8: Finalized BOM

As you can see, the team has successfully purchased and manufactured everything that our project required. In total our team needed to manufacture a total of 10 total parts. These include the Robotic Arm Base, Arm Mounting Base Plate, Arm Mounting Lock Plate, Extended Arm Segments (x2), Tensioner Bolt Mounting Blocks (x2), Drive Wheels (x2), Driven Wheels (x2), Tensioner Support Brackets (x2), Roller Brackets (x2), Roller Shaft (x2), and finally, the Roller Mounts (x2). The team decided to stick to the same manufacturing plan as the original, which allowed us to ensure a smooth process throughout the semester and allowed the team to have parts when they were needed. This plan went as follows:

• Who will make the part

The team did not have a set person for who was manufacturing each part, and it depended primarily on what was being made and for what purpose. For example, 3D printed parts were either Samuel or Tyler as they have 3D printers to be able to print those parts. Metal parts, however, were manufactured by whoever was available and has the proper training for whatever machine needs to be used to be able to make the part. The designing of the part mainly fell on the person working on that section of the project with other members helping out where needed.

• When and how long will it take to make the part (start & duration or duration & expected h finish)

With the 3D printed parts they can be printed at almost any time if the team is using one of the printers a team member owns. With parts that need to be manufactured out of metal the team chose to go in and machine it themselves, have an employee of the machine shop on campus machine it for a price, or outsource the machining to another company. For any of those options a varying amount of time was needed, and the team had to decide on which option fit with when the part was needed and budget.

• What raw material will it be made of

Deciding the material that was to be used in the manufacturing really depended on what material would withstand all our scenarios. For certain parts that do not have high stress or load requirements 3D printed parts were used to lower costs and get the part much faster. Most of our metal parts were manufactured out of aluminum or some form of steel due to their ability to withstand large amounts of stress and load.

• Where will it be made

As mentioned above, some members of the team own 3D printers and parts that need 3D printing were made in their homes. The team's metal machined parts were done at the NAU machine shop on campus. In the case that a part needed to be outsourced for manufacturing it will be manufactured at a company that offers the best price for quality. This however was not necessary, other than bolts, bearings, and washers which were in fact outsourced.

7 Final Hardware

7.1 Final Hardware Images and Descriptions

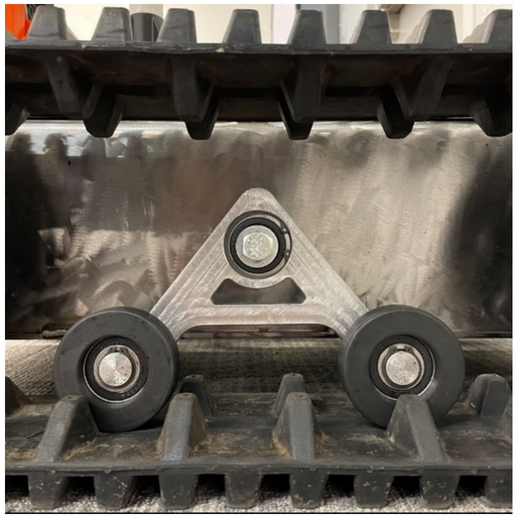


Figure 7: Track Roller

The track roller is to ensure that the vehicle does not get stuck by high centering itself. It does this simply by being a rigid object the track can roll over seamlessly to keep the track in contact with the surface it is driving over.

Figure 8: Track Tensioner

The tensioner is meant to resolve a huge issue with the movement capabilities of RUT. Before the installation of the tensioner there was not enough friction between the driven wheel and the track which would cause the wheel to freely spin and RUT to not move. With the tensioner installed it fixes that issue and allows RUT to be mobile.

Figure 9: Robotic Arm

The robotic arm is an integral piece as it is one of the main goals of this project. It is a purchased Braccio Arduino robotic arm that had some links between joints elongated to achieve the required workspace goal.

7.2 Design Changes in Second Semester

7.2.1 Design Iteration 1: Change in Remote Connection

Due to the unavailability of connecting Arduino Wi-Fi to Unity, the team turned to the use of an external Wi-Fi module. ESP8266-01 is selected to be the module functioning remote Wi-Fi connection. The data path is changed from direct connection to Arduino to transferring via the module. The connection between the module and Arduino board uses software serial. Unity now is streaming to ESP8266-01 through TCP/IP, then go to Arduino through serial.

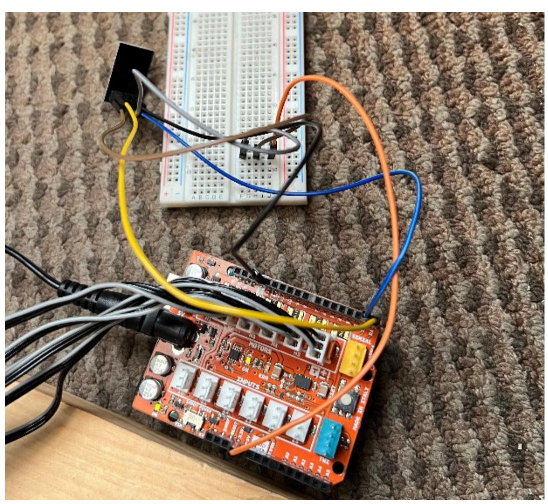


Figure 10: New Remote System

7.2.2 Design Iteration 2: Track Tensioner

The first iteration of the track tensioner was a super simple piston design that would be attached to the side of the robot between the two tracks that would house a spring that pushed a piston into the underside of the top level of track. This design was mainly scrapped because there was not going to be enough room for the tensioner and the track roller, and the team decided that the track roller was more important to have in that area and move the track tensioner. The track tensioner was instead changed to having a spring on a bolt pushing on the entire axle to tension both sides at once with significantly less chances for failure.

Figure 11: Section Cut of Original Piston Tensioner Design

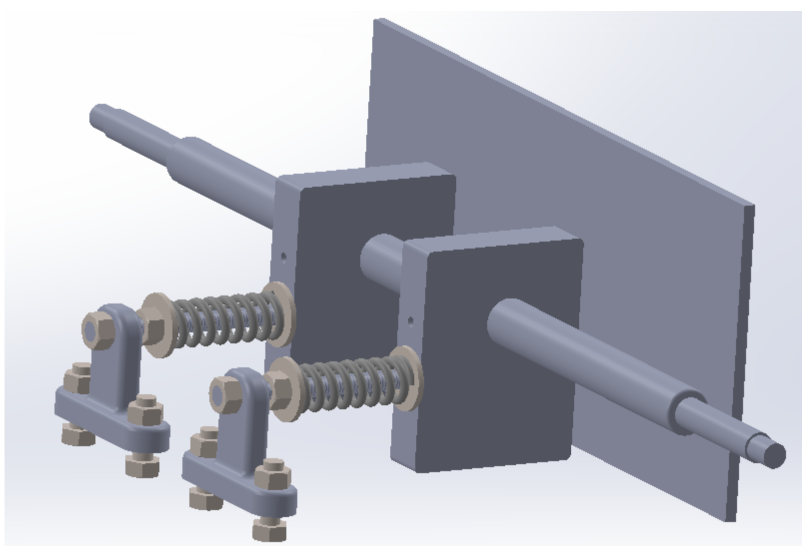


Figure 12: Tensioner Design Used

7.3 *Hardware Challenges Bested*

In almost every project that has ever existed there has been a multitude of problems that need to be addressed to have a successful project. The VR Robot project is just another one of those projects that has many problems, some arising immediately while other are surprises at the finish line. Getting to 100% for the team was filled with difficulties throughout the entire process. Some of the problems we faced was 3D print time, programming issues, and connectivity issues.

7.3.1 3D Print Time

3D printing takes a lot of time even for the simplest parts, and for our team the majority of parts being printed were of a decent size so they could only be printed out one at a time and took upwards of 20+ hours to print each. One of the solutions to this problem was utilizing the 2 printers that were available between the team members and ensuring that every second was being used to print.

7.3.2 Programming Issues

The programming issues for this project was extensive, mainly because the team had to basically learn a whole programming language from scratch to be able to complete the project. On top of learning the programming language the team had to figure out how to use Unity. The only valid solution was for the team to learn these languages to get the coding done.

7.3.3 Connectivity Issues

Getting the Arduino connected to Unity wirelessly was significantly more difficult than the team had initially expected and ended up being a huge issue the team barely solved in the end. The issue was solved simply by using an external Wi-Fi connection between Unity and the Arduino.

8 Testing

8.1 Testing Plan

Experiment 1: Turn Time

The purpose of this experiment is to test the time it takes for rut to spin 180 degrees. The only data collected is time.
The test will be operated with RUT and a timer. The time of 180 deg turn will be recorded on multiple terraces with one trial for each. The result will be calculated into deg/s as the final parameter.

Experiment 2: Turn Radius

This test is to determine how much room is required for the rut to perform a turning ranging from a few degrees all the way up to a 180 degree turn on different surfaces. These surfaces will range from rocky gravel, carpet, grass, and any other common surface the RUT will be expected to operate on. This test can simply be done by making a mark using something like tape at the beginning position and then making rut perform a 180 degree turn and seeing how big of a space is needed. The results from this experiment will simply just be measured in meters.

Experiment 3: Obstacle Course

In order to validate our engineering calculations, the RUT has to be able to perform well in an obstacle course. This experiment will test mainly the new wheels and the new tensioner system. The obstacle course will involve sharp turns, slopes, boulders and potholes.

Experiment 4: Lift Test

This experiment is to determine the amount of payload the robotic arm is capable of being picked up and properly held without dropping or having the arm struggle. The procedure for this experiment would simply have weights of varying size being picked up by the arm and slowly adding more weights until either the max is found, or the team is satisfied with the outcome. The measurements for this experiment will be in kg.

Experiment 5: Latency Test (Camera)

The purpose of this test is to evaluate the latency of the camera. 2 phones with the function of taking video are needed for the test. The only parameter tested is the latency in frames or ms.
Assuming the view in the phone is real time and the one in camera has latency. One of the phones is placed next to the camera with video on, the other phone is taking video recording both screens of the previous phone and the computer screen showing the camera view. An object is moved in slowly and hits a target. The latency can be tested by checking the video from the second phone by comparing the time it takes the object to hit the target in phone view compared to camera view checking with video software frame by frame.

Experiment 6: FEA Test

This test is run solely in SolidWorks. It will be a finite element analysis of any and all mechanical systems which we expect to see high loads on. This is to ensure that our minimum factor of safety of 2 is kept for all mechanical systems.

Experiment 7: Program Response Time

The purpose of this test is to evaluate the latency of the controller. The only parameter recorded is the frame of ms.

The procedure is similar to exp 5. One phone is needed in the test apart from the objects within the project. One person takes the video of the arm moving and stops with the controller, then using video software to find the frame difference between controller and arm to stop. The difference between frames divided by refresh rate is the latency.

Experiment 8: System Trials

This test will help the team find out if there are any code errors or interface issues. This will help bring up any debugging issues. The way the team will test this is by pushing the limits of the VR and Unity connection to the RUT by running the RUT through the obstacle course (Experiment 3).

8.2 Testing Results

Show images of testing as well any plots/figures/tables that summarize the results. Discuss thoroughly.

Show your finalizes Specification Sheet and describe how ERs/CRs were met or not met. Introduce information as needed.

In table 9, the results of the testing as planned will be listed in a format to include any and all values gathered in experiments.

Test #:	**Description**	**Value**
1	Turn Speed	73.78 deg/sec
2	Turn Radius	0.81m
3	Obstacle Course	0.32m climbing at 90 deg
4	Lift Test	0.23kg
5	Latency Test (Camera)	<50ms
6	FEA Test	Min Factor of Safety: 2.37
7	Program Response Time	0.5s
8	System Trials	Functioning

Table 9: Testing results

Table 10 is the summary of the testing result regarding to Engineering Requirements.

Engineering Requirement	**Target**	**Tolerance**	**Measured Value**	**Requirement Met**
ER1 Decreased Turn Radius	2 m	+ .25 m	0.81m	Yes
ER2 - Low Visual Latency	0.3s	+ 0.2s	<50ms	Yes
ER3 - Work Space Size	0.5 m	- 0.1 m	0.81m	Yes
ER4 Decreased Turn Time	35 degrees/s	- 5 degrees/s	72.78+/- 4 deg/s	Yes
ER5 Reasonable Factor of Safety	2	- 0	2.37	Yes
ER6 Program Speed	400 ms	+ 300 ms	500ms	Yes
ER7 Minimum Payload	0.25 kg	- 0.05 kg	0.23kg	Yes

Table 10: ER's

In general, all tests meet the Engineering requirements. For turn radius, turn speed and visual latency, the results are far beyond the requirements. Workspace and factor of safety are qualified but not beyond much. Payload and Program speed are not up to target but within tolerance, so they met the requirement. Based on the result, lifting strength and control latency, which is mainly program speed, are still needs to be improved in future work.

Table 11 is the summary of Customer Requirements.

Customer Requirement	Is Requirement Met
CR1 - Increase Maneuverability	Yes
CR2 - Accurate VR control	Yes
CR3 - Easy to use	Yes
CR4 - Large workspace	Yes
CR5 - Rigid design	Yes
CR6 - Durable	Yes
CR7 - Fast controls	Yes
CR8 - Cost effective	Yes
CR9 - Powerful	Yes
CR10 - Multi Directional force	Yes

Table 11: CR's

All client requirements are met.

8.3 Testing Challenges Bested

Throughout the testing process, there were many challenges that presented themselves. It was important to produce strategies for each testing plan to decide what was needed to ensure a successful test with usable data. An example of this can be seen with the latency test. Initially, we designed the latency test to be subjective. This was where an individual would use the device and give an opinion on if there was a delay noticed. This data, however, was decidedly flawed. The team met to discuss the best options to retest, and it was decided that we would measure the time it took for the device to move after an input was given. This was done by screen recording and counting frames from input to movement.

The latency test was the most notable challenge faced during testing. Most other tests conducted were very straight forward and more so a test of functionality to ensure our design worked. Critical tests to check for component failures occurred prior to manufacturing. Finite element analysis was a method used to limit the design iterations and get a component which is durable to first time to ensure enough time was had to complete the project.

9 RISK ANALYSIS AND MITIGATION

In order to ensure success of the VR Robot, an understanding of any and all potential risks needs to be evaluated by the team. An FMEA (Failure Mode and Effects Analysis) was created. This tool will allow us to understand all risks and their ranking through a calculated risk priority number. The top ten critical failures will be analyzed in section 4.1 and the full FMEA can be found in Appendix B.

9.1 Potential Failures Identified First Semester

Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Potential Causes and Mechanisms of Failure	RPN	Recommended Action
1. Frame	Cracking	Mobility loss + further damage	Crashing	50	Periodic visual checks
1. Battery	Dying	Power loss	Failure to charge	125	Charge after use
1. Motor Controller	Shorting out	Power loss	Water	50	Ensure enclosure is sealed
1. Power Switch	Wearing	Power loss	Overuse	40	Replace periodically
1. Lid Attatchment	Bolts Sticking	Inability to access internals	Overuse	280	Upgrade hardware
1. Electronics Housing	Leaking	Electronic Failure	Water	18	Ensure enclosure is sealed
1. Motor Mounts	Yielding	Loss of mobility	Torque	18	Periodic visual checks
1. Frame	Bending	Reduction of monility	Crashing	9	Periodic visual checks
1. Wheel Shaft Mounts	Yiedling	Loss of mobility	Crashing	24	Use of stronger materials
2. Track	Snapping	Mobility loss	Over tension	40	Periodic visual checks
2. Drive Wheel	Slipping	Mobility loss	Not enough tension	504	Use of stronger materials
2. Driven Wheel	Cracking	Mobility loss	Crashing	24	Use of stronger materials
2. Drive Wheel Shaft	Twisting	Mobility loss	Torque	6	Use of stronger materials
2. Driven Wheel Shaft	Bending	Mobility loss	Crashing	6	Use of stronger materials
2. Roller Bracket	Yielding	Mobility loss	Crashing	45	Use of stronger materials
2. Roller Wheel	Cracking	Mobility loss	Crashing	24	Use of stronger materials
2. Roller Mount	Yielding	Mobility loss	Crashing	36	Use of stronger materials
2. Tensioner	Yielding	Mobility loss	Over tension	36	Use of stronger materials
2. Tensioner Mount	Yielding	Mobility loss	Over tension	210	Use of stronger materials
3. Servo Gear	Stripping	Loss of arm controll	High payload	315	Use of stronger materials
3. Servo Motor	Overloading	Loss of arm controll	High payload	630	Use of stronger materials
3. Servo Mount	Yielding	Loss of arm controll	High payload	24	Use of stronger materials
3. Arm Link	Yielding	Loss of arm controll	High payload	280	Use of stronger materials
3. Arm Mount	Yielding	Loss of arm controll	High payload	60	Use of stronger materials
3. Arm Hand	Yielding	Loss of arm controll	High payload	60	Use of stronger materials
3. Arm Wiring	Wearing	Loss of arm controll	Wear from repeated motions	6	Secure wires
3. Arduino	Shorting out	Loss of arm controll	Water or high heat	8	Ensure enclosure is sealed
3. Servo Wires	Wearing	Loss of arm controll	Wear from repeated motions	6	Secure wires
3. Arm Base Plate	Yielding	Loss of arm controll	High payload	70	Use of stronger materials
4. Arduino	Shorting out	Loss of arm controll	Water or high heat	8	Ensure enclosure is sealed
4. Arduino	Failing	Loss of connection	Electronic failure	48	Find maximum range
4. VR	Failing	Loss of connection	Electronic failure	420	Find maximum range
4. Unity	Failing	Loss of connection	Electronic failure	378	Find maximum range
4. Camera	Failing	Loss of connection	Electronic failure	336	Find maximum range
4. Camera Mount	Yielding	Loss of view	Crashing	18	Use of stronger materials
4. Unity Code	Breaking	Loss of control	Code bugs	448	Code refinement
4. Controller	Disconnection	Loss of control	Electronic failure	120	Find maximum range
4. Controller Connection	Failing	Loss of control	Electronic failure	200	Find maximum range
4. Camera Body	Yiedling	Loss of view	Crashing	20	Use of stronger materials

Table 12: Previous FMEA

9.2 Potential Failures Identified This Semester

The team did not realize any new failures this semester. The previous FMEA was able to identify any and every issue we encountered throughout the development process. This semester was spent fixing failure that were listed in the previous FMEA.

9.3 Risk Mitigation

The team was able to accomplish design solutions which mitigated our major critical failures. The first major failure identified by the team was drive wheel power transmission. The previous design had been proven to be unreliable and broke to breaking. Figure 13 shows the design solution which implemented a lug system similar to an automobile to transmit power to the wheels.

Figure 13: Lug Bolts

Additionally, the issue of tracks slipping was mitigated by designing a ground up tensioning system. This ensure the proper amount of frictional force was made between the wheel and track to transmit power without slipping. Figure # shows the revised tensioning system.

Figure 14: Tensioner

A major failure in mobility was also the RUT getting stuck on large objects due to a phenomenon known as high centering. This is when an object makes contact with the middle of the body after already passing the front wheels over and getting stuck. The new roller system ensured that the track could not flex and allow an object to make contact with the bottom of the frame.

Figure 15: Roller System

The last mobility system risk was the rear wheels falling off their bearings. This failure was identified to be caused by a lack of retention rings withing the wheel to secure the bearings. Retention rings were implemented into the new design which solved this failure.

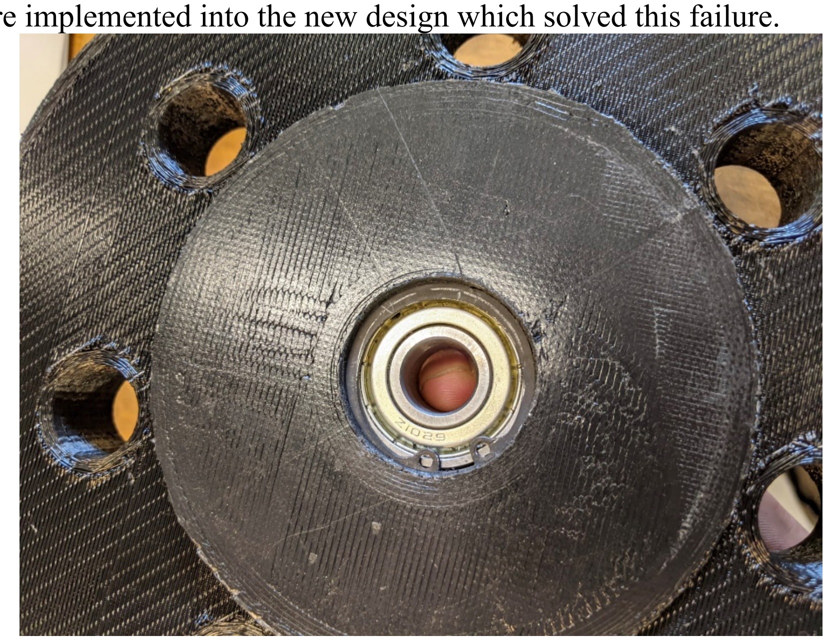


Figure 16: Retention Rings

Figure 17: New arm link

One of the customer's requirements was to be able to have a workspace of 0.8m in diameter. To achieve this the team printed new links to be able to reach further. Given that the length of the arm was increased the moment and shear forces on the arm also increased. To mitigate this problem the team used a pro series PLA with a yield strength of 40 MPa.

Figure 18: 70 kg Servo

Once these new links were attached the team realized that the base servo would not be able to have the necessary torque to lift the arm plus the payload. To mitigate this the team researched different types of servos and realized that the BETU 70kg servo had roughly the same dimensions but with more torque. This new servo helped mitigate the potential risk of not having the torque enough to lift the requited payload.

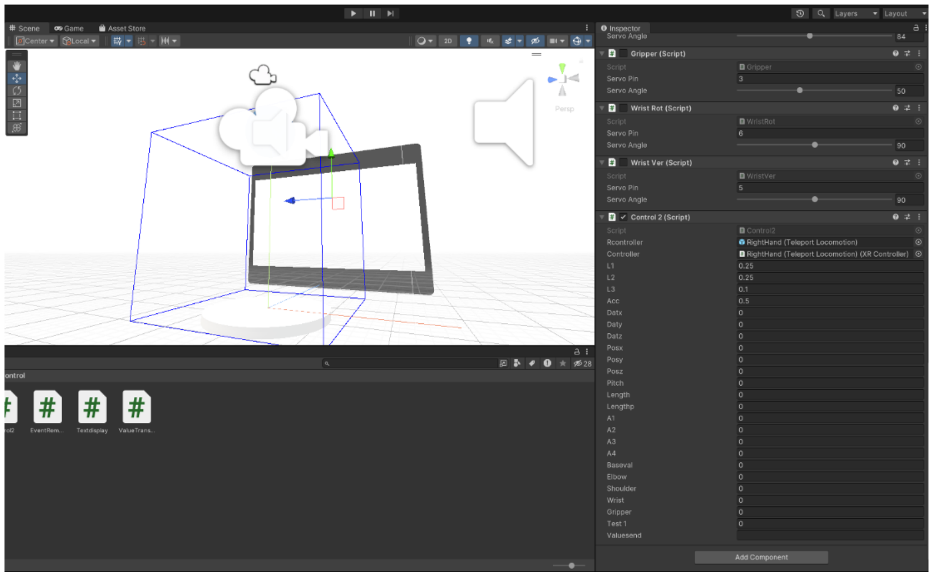


Figure 19: Unity Interface

The VR controller input has a risk of zeroing when the controller goes beyond the working space. The reason causing the problem is that inverse cosine and arctan function are used in the calculation of angles, so when the controller goes beyond, the parameter jumps to no value so it might cause a common value jumping to 0. One of the ways to reduce the problem is to decrease the accuracy value, 'acc' shown in the figure above.

The arm is very sensitive to the motion of VR controller, the fast movements of controller might break the connection between motor and arm, or it might hit something in a high speed then causing damage to itself. Also, the movement of one motor will cause a slight shaking of the other motors.

The system is tested with Oculus VR only, for other VR devices, it is not 100% ensured that the system works.

10 LOOKING FORWARD

10.1 Future Testing Procedures

Before proceeding with future testing, it is important to note that the wheels need to be machined or made of a material with a higher yield strength. Do not perform rough driving tests with the current wheels.

Another important thing to note is that the tensioners need to be limited to a free radius of spin of 90 degrees. Otherwise, the tensioner will flip over and stop applying tension to the track.

Once the wheels and tensioner have been properly fixed it is recommended that a tension test is performed by driving the RUT uphill over medium size boulders and to carefully observe if the tension provided by the tensioner is enough. If the wheels start slipping this could mean that the track needs more tension. To mitigate this a spring with a higher spring constant may be used to replace the current springs from the tensioner system.

Another aspect that will need more testing is the VR system. Due to the limited knowledge of computer science that the team has, the remote operations are lacking in accuracy and reliability. The team recommend future capstone teams to delegate this area of the project to a team more knowledgeable in this area.

Lastly a mayor area for improvement is the electronics organization inside RUT. Currently most wires are longer than needed. Soldering all the connections to a copper proto board would be the way to go for organizing all the wire connections.

Detail testing procedures yet to be accomplished or not included in the scope of your project.

10.2 Future Iterations

Iterations on control system will help improve the performance of the robot. The codes controlling motor have 2 parts in unity and Arduino. For Arduino part, iterations on the motor output functions will help with the self-protection of the arm, also increase the torque under serial input.

For Unity part, iterations on physical geometry of the arm will achieve a better control or provide multiple modes for different conditions.

Relocation of camera position will help with the control of robot. The current position is too high, and it is hard to see the arm and the ground from VR view. Relocating the camera will help with the sight, also solving the problem of the arm hitting the pole.

11 CONCLUSIONS

In May of 2023 the team was tasked with the mission of improving and incorporating an arm to the RUT project. The team was handed a list of customer and engineering requirements and a limited budget. Today, after seven months of hard work the team was able to not only successfully meet all customer and engineering requirements but to do so within the budget that was provided. The goal of having a working RUT has been achieved and the team is proud of their intellectual and physical work.

11.1 Reflection

Throughout the course of the 7 months the team took into consideration some important factors when coming up with engineering designs. One of the most important factors that the team took into consideration was safety. We understand how important safety is for the wellbeing of all stakeholders therefore the team design every part of the RUT with a factor of safety higher than 2 to ensure that this product will be safe to use and that will not yield when undergoing predetermined loads. The team sought to maximize the amount of material used when designing parts to prevent waste of materials.

11.2 Project Applicability

College is a great way to start one's career, but if you never work in teams, you will not be prepared to go out into the industry and perform well. It is of no use if we have a great mind and are unable to accurately communicate our ideas. In the same way it is of no use if we are unable to work with others when it comes to big engineering projects. Look at the golden gate bridge, the Eiffel tower, Atlas from General Dynamics, all these could not exist if only one person worked on it. This alone highlights the importance of teamwork and how indispensable it is to achieve great things. As the team reaches the end of our capstone we can look back and see how each one of us was there to help each other, some of us had ideas that benefited the team and some ideas that delay the project, yet even with the highs and lows we have learned to bear with the weaknesses of one another and to encourage one another to move forward and don't look back. We have learned the importance of teamwork and moving forward each team member will have no issue thriving in team environments, as we have learned in these past four years how to successfully work in teams. "Talent wins games, but teamwork and intelligence win championships" *Michael Jordan.*

12 REFERENCES

[1] “Arduino Braccio ++,” Arduino Online Shop. https://store-usa.arduino.cc/products/braccioplusplus?selectedStore=us (accessed Jul. 06, 2023).
[2] “I Remotely Control A Crazy Robot Arm Using VR!,” www.youtube.com. https://www.youtube.com/watch?v=qoyIVGB8OOk (accessed Jul. 08, 2023).

[3] https://zh-hk.facebook.com/Ufactory2013, “Case Study - Extended Robotics | UFACTORY,” Feb. 23, 2023. https://www.ufactory.cc/user-case-extended-robotics/ (accessed Jul. 08, 2023).

[4] Kris Hauser,”Robotic Systems”, 2023, https://motion.cs.illinois.edu/RoboticSystems/
[5] “Masterclass - 3D Geometry” https://robotacademy.net.au/masterclass/3d-geometry/?lesson=88 (accessed Jul. 07, 2023).
[6] I. Fadelli, “Digit: A high-resolution tactile sensor to enhance robot in-hand manipulation skills,” Tech Xplore - Technology and Engineering news, https://techxplore.com/news/2020-07-digit-high-resolution-tactile-sensor-robot.html (accessed Jul. 7, 2023).

13 APPENDICES

13.1 Appendix A: Codes

Arduino Codes for remote control

```
#include <SoftwareSerial.h>
SoftwareSerial esp(0,1);


// Servo
#include <Servo.h>
#define MAXSERVOS 8
Servo base;
Servo shoulder;
Servo elbow;
Servo wrist_rot;
Servo wrist_ver;
Servo gripper;


void setup()
{
  Serial.begin(19200);
  esp.begin(9600);

  esp.listen();

  Serial.setTimeout(25565);

  pinMode(12,OUTPUT);
    digitalWrite(12,HIGH);

  base.attach(11);
  shoulder.attach(10);
  elbow.attach(9);
  wrist_rot.attach(6);
  wrist_ver.attach(5);
  gripper.attach(3);// put your setup code here, to run once:
  gripper.write(73);
  shoulder.write(90);
  elbow.write(85);
  wrist_ver.write(90);
  wrist_rot.write(90);
  base.write(70);
```

```
}



void loop() {
  // put your main code here, to run repeatedly:

        String cmd = "";
        if(esp.available() > 0){

        String cmd = esp.readStringUntil('\n');
        const char* cmdd = cmd.c_str();
        //Serial.print(cmd);
        //Serial.println();

        ServoControl(cmdd);
   }

}

int sangles[] = {70, 90, 85, 90, 90, 73};
int vangles[6];

void ServoControl(const char* cmdd){
  char* token;
      int index = 0;
      token = strtok(cmdd, ",");
      while (token != NULL && index < 6) {
      vangles[index++] = atoi(token); // Convert token to integer
      token = strtok(NULL, ",");
      }

      pinMode(12,OUTPUT);
      digitalWrite(12,HIGH);

      if (vangles[0] < 0) vangles[0]=0;
  if (vangles[0] > 180) vangles[0]=180;
  if (vangles[1] < 15) vangles[1]=15;
  if (vangles[1] > 165) vangles[1]=165;
  if (vangles[2] < 0) vangles[2]=0;
  if (vangles[2] > 180) vangles[2]=180;
  if (vangles[3] < 0) vangles[3]=0;
  if (vangles[3] > 180) vangles[3]=180;
  if (vangles[4] > 180) vangles[4]=180;
  if (vangles[4] < 0) vangles[4]=0;
```

```
      if (vangles[5] < 10) vangles[5] = 10;
  if (vangles[5] > 73) vangles[5] = 73;

      //One step ahead
      if(abs(vangles[0]-sangles[0])>1){
        base.write(vangles[0]);
        sangles[0] = vangles[0];
      }

        if(abs(vangles[1]-sangles[1])>1){
        shoulder.write(vangles[1]);
        sangles[1] = vangles[1];
        }
        if(abs(vangles[2]-sangles[2])>1){
        elbow.write(vangles[2]);
        sangles[2] = vangles[2];
        }

        if(abs(vangles[5]-sangles[5])>1){
        gripper.write(vangles[5]);
        sangles[5] = vangles[5];
        }

}
```

ESP8266-01 Codes

```
#include <SoftwareSerial.h>
SoftwareSerial Ard(2,3);
#include<Uduino_Wifi.h>
Uduino_Wifi uduino("uduinoBoard"); // Declare and name your object

// Servo
#if defined(ESP32)
// Install esp32servo library => Sketch>Include Library>Manager Libraries>Esp32Servo
#include <ESP32Servo.h>
#else
#include <Servo.h>
#endif

#define MAXSERVOS 8

void setup()
{
  Serial.begin(115200);
  Ard.begin(9600);
#if defined (__AVR_ATmega32U4__) // Leonardo
```

```
  while (!Serial) {}
#elif defined(__PIC32MX__)
  delay(1000);
#endif

  // Optional functions,  to add BEFORE connectWifi(...)
  uduino.setPort(4222);   // default 4222
  uduino.connectWifi("SSID", "pswd");

  uduino.addCommand("s", SetMode);
  uduino.addCommand("d", WritePinDigital);
  uduino.addCommand("a", WritePinAnalog);
  uduino.addCommand("rd", ReadDigitalPin);
  uduino.addCommand("r", ReadAnalogPin);
  uduino.addCommand("br", BundleReadPin);
  uduino.addCommand("b", ReadBundle);
  uduino.addCommand("Control", Control);

  uduino.addInitFunction(InitializeServos);
  uduino.addDisconnectedFunction(DisconnectAllServos);
}

void ReadBundle() {
  char *arg = NULL;
  char *number = NULL;
  number = uduino.next();
  int len = 0;
  if (number != NULL)
    len = atoi(number);
  for (int i = 0; i < len; i++) {
    uduino.launchCommand(arg);
  }
}

void SetMode() {
  int pinToMap = 100; //100 is never reached
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
  {
    pinToMap = atoi(arg);
  }
  int type;
  arg = uduino.next();
  if (arg != NULL)
```

```
  {
    type = atoi(arg);
    PinSetMode(pinToMap, type);
  }
}

void PinSetMode(int pin, int type) {
  //TODO : vérifier que ça, ça fonctionne
  if (type != 4)
    DisconnectServo(pin);

  switch (type) {
    case 0: // Output
      pinMode(pin, OUTPUT);
      break;
    case 1: // PWM
      pinMode(pin, OUTPUT);
      break;
    case 2: // Analog
      pinMode(pin, INPUT);
      break;
    case 3: // Input_Pullup
      pinMode(pin, INPUT_PULLUP);
      break;
    case 4: // Servo
      SetupServo(pin);
      break;
  }
}

void WritePinAnalog() {
  int pinToMap = 100;
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
  {
    pinToMap = atoi(arg);
  }

  int valueToWrite;
  arg = uduino.next();
  if (arg != NULL)
  {
    valueToWrite = atoi(arg);
```

```
    if (ServoConnectedPin(pinToMap)) {
      UpdateServo(pinToMap, valueToWrite);
    } else {
      analogWrite(pinToMap, valueToWrite);
    }
  }
}

void WritePinDigital() {
  int pinToMap = -1;
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
    pinToMap = atoi(arg);

  int writeValue;
  arg = uduino.next();
  if (arg != NULL && pinToMap != -1)
  {
    writeValue = atoi(arg);
    digitalWrite(pinToMap, writeValue);
  }
}

void ReadAnalogPin() {
  int pinToRead = -1;
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
  {
    pinToRead = atoi(arg);
    if (pinToRead != -1)
      printValue(pinToRead, analogRead(pinToRead));
  }
}

void ReadDigitalPin() {
  int pinToRead = -1;
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
  {
    pinToRead = atoi(arg);
  }
```

```
  if (pinToRead != -1)
    printValue(pinToRead, digitalRead(pinToRead));
}

void BundleReadPin() {
  int pinToRead = -1;
  char *arg = NULL;
  arg = uduino.next();
  if (arg != NULL)
  {
    pinToRead = atoi(arg);
    if (pinToRead != -1)
      printValue(pinToRead, analogRead(pinToRead));
  }
}

void loop()
{
  uduino.update();
  //delay(10);
}

void printValue(int pin, int targetValue) {
  uduino.print(pin);
  uduino.print(" "); //<- Todo : Change that with Uduino delimiter
  uduino.println(targetValue);
}



/* SERVO CODE */
Servo servos[MAXSERVOS];
int servoPinMap[MAXSERVOS];

void InitializeServos() {
#if defined(ESP32)
  ESP32PWM::allocateTimer(0);
  ESP32PWM::allocateTimer(1);
  ESP32PWM::allocateTimer(2);
  ESP32PWM::allocateTimer(3);
#else
#endif

  for (int i = 0; i < MAXSERVOS - 1; i++ ) {
    servoPinMap[i] = -1;
```

```
    servos[i].detach();
  }
}

void SetupServo(int pin) {
  if (ServoConnectedPin(pin))
    return;

  int nextIndex = GetAvailableIndexByPin(-1);
  if (nextIndex == -1)
    nextIndex = 0;
#if defined(ESP32)
  servos[nextIndex].setPeriodHertz(50);
#endif
  servos[nextIndex].attach(pin);

  servoPinMap[nextIndex] = pin;
}


void DisconnectServo(int pin) {
  servos[GetAvailableIndexByPin(pin)].detach();
  servoPinMap[GetAvailableIndexByPin(pin)] = 0;
}

bool ServoConnectedPin(int pin) {
  if (GetAvailableIndexByPin(pin) == -1) return false;
  else return true;
}

int GetAvailableIndexByPin(int pin) {
  for (int i = 0; i < MAXSERVOS - 1; i++ ) {
    if (servoPinMap[i] == pin) {
      return i;
    } else if (pin == -1 && servoPinMap[i] < 0) {
      return i; // return the first available index
    }
  }
  return -1;
}

void UpdateServo(int pin, int targetValue) {
  int index = GetAvailableIndexByPin(pin);
  servos[index].write(targetValue);
  delay(10);
```

```
}

void DisconnectAllServos() {
  for (int i = 0; i < MAXSERVOS; i++) {
    servos[i].detach();
    digitalWrite(servoPinMap[i], LOW);
    servoPinMap[i] = -1;
  }
}

void Control(){
String angles =
String(uduino.getParameter(0))+","+String(uduino.getParameter(1))+","+String(uduino.getParameter(2))+","+S
tring(uduino.getParameter(3))+","+String(uduino.getParameter(4))+","+String(uduino.getParameter(5));
Ard.print(angles);
Ard.println();
delay(100);
}
```