Functional Specification

Year: 2022 Semester: 2 Team: 2 Project: VRms

Creation Date: 1/20/2022 Last Modified: January 21, 2022

Member 1: Emma Clary Email: clarye@purdue.edu

Member 2: Brian Latimer Email: blatimer@purdue.edu

Member 3: Matthew Wen Email: wen101@purdue.edu

Member 4: James Donnelly Email: donnell9@purdue.edu

Assignment Evaluation:

| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| --- | --- | --- | --- | --- |
| **Assignment-Specific Items** | | | | |
| **Functional Description** |  | x3 |  |  |
| **Theory of Operation** |  | x3 |  |  |
| **Expected Usage Case** |  | x3 |  |  |
| **Design Constraints** |  | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

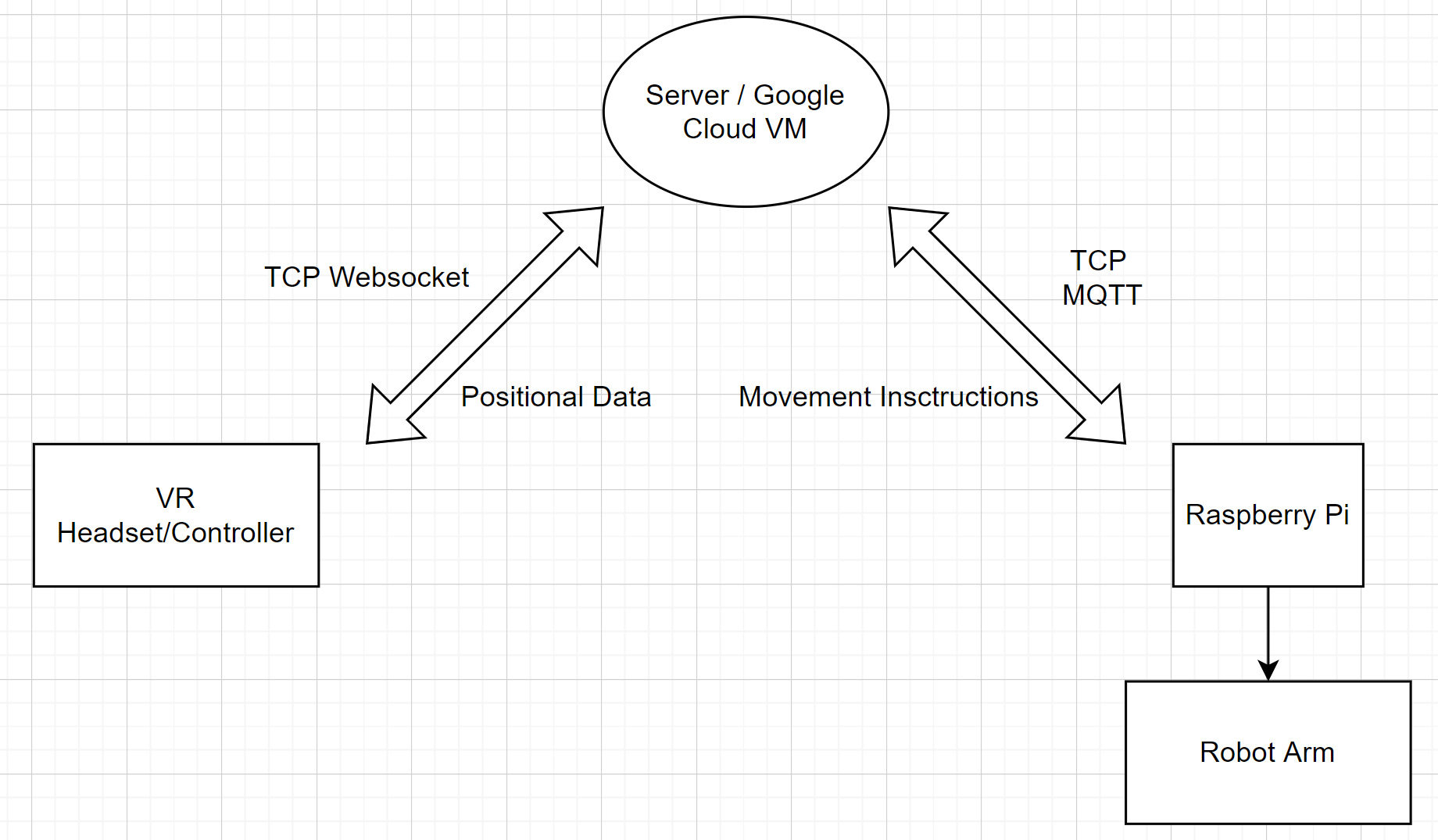
5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

1.0 Functional Description

VRms is a VR (virtual reality) program that allows a user to control a robot arm via a VR simulation to complete tasks in an environment that is unsafe and unsuitable for direct human interaction. The movement of the VR controller will result in the movement of the robot arm to mimic the movement of the VR user’s hand with the movement of the robotic hand. This translation of movements will allow our robotic arm to perform certain actions based on the movements of the VR controller, giving workers the ability to complete unsafe tasks in a safer location.

2.0 Theory of Operation



This system diagram shows the general flow of data throughout the components of our system. As shown, the VR headset and controller will communicate with the arm through the assistance of a Google Cloud virtual machine and a Raspberry Pi. The Google Cloud VM will take positional data from the VR headset and convert it to movement instructions for the robot arm, while simultaneously streaming video from the arm to the VR headset. TCP Websockets will be used to transfer video streaming data, while TCP MQTT will be used to transmit movement data, as MQTT is better suited for efficient transfer of smaller data sizes. The Raspberry Pi will be used to interface the hardware on the arm with the wireless connection to the Google Cloud server.

The VRms system will utilize at least one motor that is controlled by a PWM signal. One of the biggest challenges with speed control is balancing the ability to calculate and set position fast enough to correctly replicate the VR user’s movements while also preventing these transitions from resulting in damage to the robot from high speed transitions. This speed control is done by controlling the duty cycle of the PWM signal. The duty cycle is the percentage of digital high to digital low during a PWM period. The speed of the motor is proportional to the average voltage of the PWM signal over that period [4].

3.0 Expected Usage Case

The target user of VRms is workers that are required to work in unsafe conditions. Therefore the company employing these workers would be the target buyer of this product. VRms is meant to be a versatile product that is portable and lasts for several years. Because of this, many people could use each individual VRms arm due to the company owning it and utilizing it whenever the need arises. The environment that the VRms arm will be placed in is likely to be dangerous for human workers, such as traffic directors and other deadly occupations. Since the VRms arm is meant to be versatile in its usage, it must be able to withstand a number of environmental hazards such as being outdoors.

4.0 Design Constraints

4.1 Computational Constraints

The primary computational function of VRms is the translation between the unity provided angles and the positional data sent to servo 1 through 6. This translation will require floating point calculations since the angle precision is +/- 0.01°. The microcontroller will need to communicate the unity angles using a TCP connection between the VR simulation and the server and then through a TCP connection from the server to the Raspberry-Pi. The Raspberry-Pi will then communicate the Unity provided angles through a serial connection to the microcontroller. This data transfer and calculation will need to happen multiple times a second to correctly update the positions to match the VR user’s controller movement.

As for memory, the data will be stored on an SD card (non-volatile memory), but there will also need volatile memory to store all of the incoming data that needs to be used in the servo position calculations.

4.2 Electronics Constraints

The interfaces we expect to use are UART and/or SPI. The components in the current design are 6 LewanSoul Digital Servos, 6-channel Bluetooth Servo Controller, a couple of switches/buttons, LED indicators, Wifi module, and a 360° Motor. The Raspberry-Pi to microcontroller communication protocol will most likely be UART, similar to the UART communication between the microcontroller and the 6-channel Bluetooth Servo Controller.

4.3 Thermal/Power Constraints

Since VRms is likely to be in an outdoor location or somewhere away from a wall outlet, it will have an onboard battery. Operating voltage will be around 5V-7.5V and should operate at a range of 0.5-2A at any time. Target battery life should exceed an 8 hour work day. It should also be able to be recharged if the battery has a shorter shelf life than a couple years.

4.4 Mechanical Constraints

Physical constraints will vary for the different jobs VRms are used for, but as a general rule, it shouldn’t be any bigger than the average human in both height and width. Differing jobs will require different interactions with the environment, so the end piece of the arm will have different attachments available.

VRms is used in environments that could be **potentially** harmful for humans, but it is not meant to withstand those hazards. For example, crane operators have been killed from crane parts, crane loads, or crane collapses (632 deaths total) [5]. VRms is not meant to withstand a crane from collapsing, but instead is saving human lives from these hazards while getting the job done. Therefore it has no extreme durability standards past the ability to operate for years barring injury to the arm.

4.5 Economic Constraints

VRms should be a low-cost solution to teleoperation of a robot arm. The arm along with the parts to make it able to be remotely controlled would have a price range of $450 - $500 not including the VR headset. Any VR headset would be capable of running this software and controlling any number of arms, and would have a price of $300 for the cheapest headset, the Meta Quest 2. [3] The closest competing company's teleoperated robot arm, Reachy, costs $9,000 [2] for the base, single-arm version of their robot. It is clear that our arm will cost much less than similar products on the market. Our robot arm will not contain any import or export constraints limiting the parts we can use, either.

5.0 Sources Cited:

[1] “Discover Reachy, a Robotic Platform Based on Ai – Reachy by Pollen Robotics, an Open Source Programmable Humanoid Robot.” *Discover Reachy, a Robotic Platform Based on AI – Reachy by Pollen Robotics, an Open Source Programmable Humanoid Robot*, 2016, <https://www.pollen-robotics.com/reachy/>.

[2] Tarantola, Andrew. “Reachy Is an Expressive, Open-Source Robot.” Engadget, 6 Mar. 2020, <https://www.engadget.com/2020-01-05-reachy-is-an-expressive-open-source-robot.html>.

[3] “Oculus Quest 2: Our Most Advanced New All-in-One VR Headset.” Oculus, <https://www.oculus.com/quest-2/>.

[4] Goolsbey, Jan. “Improve Brushed DC Motor Performance.” Adafruit Learning System, <https://learn.adafruit.com/improve-brushed-dc-motor-performance/duty-cycle>.

[5]“Crane-Related Deaths in Construction & Recommendations for Their Prevention.” *Elcosh*, <https://www.elcosh.org/document/1781/d000823/Crane-Related+Deaths+in+Construction+%2526+Recommendations+for+Their+Prevention.html#:~:text=The%20revised%20average%20is%2042,as%20to%20type%20of%20crane>.