Virtual Reality Sensor Glove Project Summary:

Robots are great for performing tasks that you don’t feel like doing or fixing problems without you being forced to supervise them. But robots are difficult to program and require a significant amount of computing power to perform the most basic of human functions. But what if you could simply drive the robot yourself? What if you could take control and perform all of the intricate human tasks but remotely, from the comfort of your living room? We believe that by mapping body movements with sensors we can create a control system for humanoid robots that will be able to follow your exact commands and interact with the world with human accuracy. This is the concept on which our project was initially built.

Our goal was to build a data glove that was capable of capturing human hand movements and simulating this control of a robotic hand using Gazebo ROS. This project would ultimately be used for more accurate video gaming, virtual reality, or real-life applications such as cleaning up toxic waste, remote surgery, or bomb disposal. Our idea for the glove was that it could gather input from a variety of sources, including a gyroscopic sensor, force sensors, flex sensors, and an RBG camera for motion confirmation. It would use a microcontroller to process the data before sending it back to a central computer in order to display and run the simulation. The glove would also give haptic feedback to the user’s fingertips based on the gripping force in the simulation.

Unfortunately, due to changes mid-semester, we were forced to seriously revise the scope of the project. In order to salvage the progress we had already made, we decided to keep the hardware components that had been built and instead focus our efforts on the development of a software simulation. Our revised project is more of a proof of concept that human-computer interaction devices can be useful for virtual reality, and will proper calibration, they can be used to simulate complex actions. Our project has thus morphed into a simple virtual reality demonstration using a few sensors to attached to a glove to cause reactions in a Unity game environment. Using a simulated robotic hand, we demonstrate that the glove can be used to send data to the game and pick up a block. With added sensor capability, it might even be possible to pick up the block and run with it or throw it as part of a virtual game like baseball.

Though we had initially wanted to also provide the use of our glove with haptic feedback by placing haptic motors on the fingertips of the glove to provide increasing vibrations based on the force with which an object in the simulation was being gripped, the transition to software has shelved that plan, however it would certainly be an area meriting additional research should a commercial version of this project ever exist. Additionally, we would have liked to integrate more sensors, like force and pressure sensors, in order to receive finer sensor readings for better precision.

As is, our design uses a Raspberry Pi 4 reading input with Python scripts from an analog to digital converter (ADC). The ADC, in turn, receives the analog signal output from our flex sensors by measuring the voltage drop across a voltage divider composed of the flex sensor and resistors. Our design for the sensor layout was to use two flex sensors for each finger. One sensor would track the relative change in motion between the back of the hand and the first joint of the finger. The second sensor would then span the space from the first joint all of the way to the end of the finger, as shown in Figure 1. This would provide simulated motion to capture the movement of each of the two tendons in human fingers. In addition, we wanted to map the motion of the wrist and arm in general. For this purpose, we used a combined accelerometer/gyroscope/magnetometer sensing unit. Our initial plan was to use the accelerometer to track the motion of our glove in space, while using the gyroscope to track the orientation of the hand, and even possibly use the magnetometer for reading the thumb motion A picture containing sitting, table, computer, orange

Description automatically generatedby placing a magnet on the end of the thumb. Using more Python code, we interacted with both the ADC and multi-sensor unit using I2C. During the first half of the semester, we were reading from the flex sensors, the gyroscope, and accelerometer. However, the accelerometer required additional processing in order to properly be used for position detection, and so the midsemester change put an end to that plan. The gyroscope, on the other hand, required little additional interpretation to be used, and currently returns the angle in degrees of each of the three axes of rotation.

Figure : Sensor placement design for finger motion capture

The software side of our project consisted of creating a robotic hand model within the Unity game engine. This involved taking apart existing open-source projects and re-purposing their parts to build a new model. Two designs for the model were built, with one resembling a human hand but with only three fingers, and another resembling a robotic gripper hand with three A picture containing indoor, table, sitting, water

Description automatically generatedfingers facing each other. Both models were designed to take input from either a keyboard or another external source to manipulate the upper and lower fingers, as well as the thumb and wrist. In order to make the hand more realistic, constraints in motion were added to match the general constraint of the human hand and as closely simulate a real robotic hand as possible, as well as force colliders to accurately show when the fingers gripped an object. The basic task of the simulation is to receive input from either a text file generated by the Raspberry Pi with the sensor data, or take in live input via a TCP server. In this manner, the glove could be used to manipulate the hand and grip a block in the simulated environment.

Figure : Simulated robotic hand showing force colliders

Despite the radical changes in the project scope and available resources, we were able to salvage our virtual reality data glove and implement a design that can serve as a proof of concept of the capability of human-computer interaction devices. Additional resources and time would certainly have helped increase the capabilities of this project, but as is it serves as a solid initial implementation that still leaves plenty of room for an expanded and improved version that could one day be released in a commercial market setting.