Abstract

Controlling a drone by moving your arm as if using The Force from Star Wars is an interesting idea in theory. With applications in prosthetic control and drone aerial aviation, advancement in this research has real-world implications. With the use of the Myo armband and the DJI Mavic Pro 2, this idea has come to fruition. The Myo armband detects arm movements then sends this data to a computer that is able to process the input and further send drone flight instructions to the DJI Smart Controller which further instructs the drone to fly accordingly. With all these moving parts working in harmony, a user can control a drone with a simple wave of their arm.

Intro + Background

This project works toward using a Myo armband to control a drone with simple arm/hand gesture control. Not only would this be an interesting project to complete, but it also has practical purposes as well. For example, being able to control a drone with limited motion could prove to be a precursor to helping an amputee control a prosthetic with more free motion [3]. Additionally, advancements in drone aviation would be applicable to a variety of fields including drone delivery services and unmanned surveillance. Due to the fruitfulness of this project, others have also completed similar projects. During the CES event in 2014, Thalmic Labs (manufacturer of the Myo armband) partnered with AR.Drone to present a demo of using the Myo to control a drone. However, this never appeared as a public product or service and, even in the demo video, it is clear that there was a significant lag between when the motion was completed and when the drone reacted [1]. A German graduate student, Andreas Degele, also attempted a similar project pairing a Myo armband and an AR.Drone 2.0 and releasing a free application called *MyoPilot,* published by Paul Bernhardt at Thalmic Labs [2]. However, this product has since become deprecated since Thalmic Labs had a complete overhaul in 2018, rebranding to North and stopping the sale of the Myo armband in favor of smart glasses (in fact, I could only find the GitHub repositories for *MyoPilot* of the code but not the commercial download for the application [4]). North has since been bought by Google in 2020 for their augmented reality division [6], further deprecating any work by Thalmic labs on the Myo armband.

The Myo armband itself contains multiple sensors that can enable gesture detection. It contains electromyograms to detect electrical waves from muscles. This allows the Myo armband to detect when flexed or relaxed at very precise levels. It also contains multiple inertial measurement units (IMUs) which act similarly to a gyroscope in detecting current orientation and change in orientation. The IMU records a continuous stream of orientation data. Together, these sensors enable the Myo armband to detect basic gestures such as if you make a fist with your arm. The complete list of gestures that the Myo can detect out-of-the-box (developed by Thalmic labs already): fist, fingers spread, wave in, wave out, double-tap.

Although the Myo armband is deprecated, Niklas Rosenstein graciously saved the downloads that were once available on the Myo webpage (no longer functioning) while working on his python SDK for Myo [7]. These SDKs include ones for Windows, Android, and Mac. The downloads also include a Windows and a Mac application to directly interact with the Myo armband without developing any of our own code (this is helpful for managing the armband). Niklas Rosenstein also made a Python library that utilizes the Windows SDK to interact with the

Myo armband using python code [8]. The Myo armband comes with a corresponding Bluetooth receiver (in the form of a USB plug) that plugs into your computer. This is necessary for the Python library and Windows SDK to work properly and connect with the Myo armband.

The drone used for this project is the DJI Mavic 2 Pro (with DJI Smart Controller). This is a state-of-the-art drone (though many of its capabilities are not used for the purposes of this project), and DJI provides extensive SDKs including Android, iOS, and Windows [9]. The DJI Smart Controller that comes with the drone runs some DJI Android skin and connects to the drone using DJI's proprietary Ocusync technology [10].

Methods

Connecting the Myo armband to the DJI Mavic Pro 2 is a multi-step process. First, the Myo armband uses its sensors to gather gyroscopic data on the user's arm. This data is then sent and received by a Windows computer that is connected to the armband. The data is read in using the available Python library and converted to drone flight instructions using Python code. These flight instructions are then sent to the DJI Smart Controller which connects directly to the DJI drone. These instructions are conveyed to the drone which acts upon them and flys accordingly. The drone is also able to send back its camera and sensor feed to the DJI Smart Controller which sends it back to the Windows computer so that it can more accurately determine the correct drone flight instructions. Additionally, the Windows computer can control the Myo armband (e.g. make it vibrate) for intuitive user controls. A summary of these connections is shown below, but the main point is that the Windows computer is taking in input from both the sensor and the drone, computing the correct flight path for the drone, and sending this information to the drone using the DJI Smart Controller as a middle-man. The DJI Smart Controller is necessary as the middle-man since a normal laptop does not contain Ocusync technology that is necessary to communicate with the drone.

To connect the Myo armband to the computer, the myo-python library by Niklas Rosenstein was used [11]. It is able to import the Windows C++ SDK and convert C++ to Python for ease of access to those familiar with Python. When the Myo armband is connected to the computer and the Python code is running, the armband is continuously sending a stream of data that the Python code is listening for. This data is then processed on the Windows computer to basic drone instructions.

These drone instructions must then be sent from the computer to the DJI Smart Controller. The DJI Smart Controller is running a custom-made Android app that needs to be able to listen for the continuous stream from the Myo armband. Using a web socket, the computer turns into a server that the Smart Controller is continuously sending web requests to [12]. These web requests can send information to the computer, but more importantly, the computer returns the current drone instructions at that moment. To save computation power, drone instructions are not constantly calculated but are rather only calculated when the Smart Controller requests instructions from the computer.

Using the DJI Android SDK, these flight instructions are then sent to the drone which flys accordingly. For more details on how the Myo sensor data is converted to drone instructions, please consult the code repository and email krishganotra@gmail.com with any further questions.

Results

Although simple in theory, connecting all the working parts that are only scarcely compatible due to proprietary technology was more difficult than anticipated; The Myo armband would only connect to the Windows computer since it needed the USB Bluetooth adapter, and the DJI Drone would only connect to the Smart Controller due to the proprietary Ocusync technology. This created the hurdle of connecting the laptop to the Smart Controller and transferring a continuous stream of data quickly without lag. This was solved with the web socket, though there are other viable solutions.

The struggle of using deprecated technology is also notable. Working with the Myo armband that is no longer supported was frustrating. All online sources are no longer available and older, deprecated versions of complementary technologies are used too, requiring them to be converted to up-to-date code. The work by Niklas Rosenstein made it feasible to work with the deprecated armband, but other deprecated technologies without such a resource should be avoided.

Extensions

With the drone now flying, more features of the advanced DJI drone can be utilized for advanced applications. The continuous stream from the camera and sensors on the drone can lead to interesting computer vision advancements including obstacle avoidance, intelligent path making, and object detection.

The technology used to convert arm input into drone instructions was also rushed and could be improved upon. Aerial aviation has many intricacies and making the flight of the drone more intelligent through artificial intelligence and machine learning would be a major advancement into making the experience of flying the drone with arm gestures more smooth.

Currently, there are 2 middlemen to process and send data from the Myo armband to the drone. There is a Myo SDK for Android that could potentially remove the need for the computer, making the transfer of data less convoluted and potentially faster. In this case, the Myo USB Bluetooth adapter would be plugged into the DJI Smart Controller and it would be able to connect directly to both the Myo armband and the DJI Mavic Pro 2.

A different sensor that is more widely supported could also be used over the Myo armband. This will enable a smoother experience for both the user and the developer.

Implementation

The Myo-python library was used. It works with python 3.7, so you may have to create a virtual environment with a lower level of python. This was run on a Windows laptop, though could very well be run on a Mac as well using the Mac SDK (the documentation uses the

Windows SDK so Mac compatibility is not ensured). The computational power of the computer should not matter too much as the code is not very computationally heavy.

The drone used is the DJI Mavic Pro 2. The sensor used is the Myo Armband.

To run the code, follow the instructions in the GitHub repository.

Works Cited

[1] Engadget. (2014, January 13). *Parrot AR.Drone meets MYO Armband at CES 2014 | Engadget* [Video file]. Retrieved from https://youtu.be/1gOZRg_74gI

[2] Bernhardt, P. (2018, August 24). *Flying Drones with MyoPilot*. Retrieved March 11, 2020, from https://developerblog.myo.com/flying-drones-with-myopilot/

[3] Focals by North. (2016, January 18). *Introducing the world's first Myo-controlled prosthetic arm* [Video file]. Retrieved from https://youtu.be/LSuzMxQDmzg

[4] Degele, A. (n.d.). *it12052/MyoPilot*. Retrieved from https://github.com/it12052/MyoPilot

[5] *How do I access the raw EMG data from the Myo armband?* (n.d.). Retrieved from https://support.getmyo.com/hc/en-us/articles/202536726-How-do-I-access-the-raw-EMG-data-from-the-Myo-armband-

[6] Bursztynsky, J. (2020, June 30). *Google acquires North, which makes smart glasses similar to Google Glass*. CNBC. https://www.cnbc.com/2020/06/30/google-acquires-north-augmented-reality-glasses-start-up.html.

[7] Rosenstein, N. (n.d.). *Releases · NiklasRosenstein/myo-python*. GitHub. https://github.com/NiklasRosenstein/myo-python/releases.

[8] Rosenstein, N. (n.d.). *NiklasRosenstein/myo-python*. GitHub. https://github.com/NiklasRosenstein/myo-python.

[9] DJI Developer Technologies. (n.d.). https://developer.dji.com/.

[10] *DJI Smart Controller*. DJI. (n.d.). https://www.dji.com/smart-controller.

[11] Rosenstein, N. (2019, April 30). *myo-python*. PyPi. https://pypi.org/project/myo-python/.

[12] *Socket - Low-level networking interface*. socket - Low-level networking interface - Python 3.9.5 documentation. (n.d.). https://docs.python.org/3/library/socket.html.