EEEE 536.60L1 and EEEE 636.60L1

Lab 1

01/03/2025

Majd Abo Kalam – 781003348

Kinan Ghaalrebat - 386006855

Rashed Alhajri – 346004413

EEEE 536.60L1 and EEEE 636.60L1 –Biorobotics/Cybernetics Lab 1: Data Acquisition and Analysis of EMG Signals using Myo Band

Objective:

The goal of this lab experiment is to introduce the student to biopotential and how they can be recorded using the Myo Armband. This experiment includes data collection and analysis of electromyograms EMG. Students are expected to learn how to use Lab Stream, Myo Armband, and EMG data analysis.

Learning Outcomes:

By completing this lab, students will:

1. Develop proficiency in using Python for data analysis and collection.
2. Learn to install different packages for myo arm band
3. Use myo arm band for muscle activity and basics of EMG signals.

Pre-Requisites:

1. Install anaconda navigator with python 3.8 only.
2. Follow the steps exactly in https://github.com/abhilasha2614/myo_ecn_airc

NOTE: Provide myo_ecn sdk local path in respective python code file.

Background:

The human body is very complex and consists of many systems including mechanical, electrical, and chemical systems. The electrical system within the human body includes electrical potentials that propagate down nerve cells and muscle fibers. Numerous events that occur the body such as brain functions, muscle movements and eye movements are invoked by these electrical potentials. Physiological potentials arise from the ionic currents that flow in an out of the nerve and muscle cells of the human body. These biopotentials are typical measured using various electrodes in combination with electronic instrumentation. Through the collection of data from biopotentials further insight can be acquired on how the different systems within the human body are functioning.

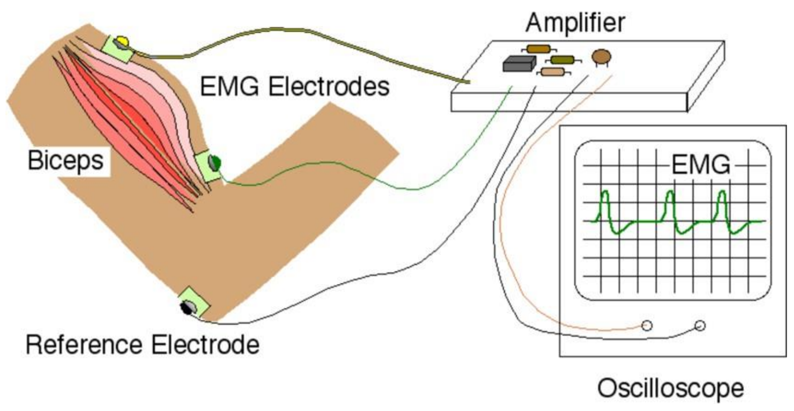
1.1 EMG Signals

There are three categories of muscles in the body including cardiac, smooth, and skeletal muscles. Focus will be placed on skeletal muscles are for they are the muscles attached to the bones and are under voluntary control. Muscles are attached to bones via tendons and during muscle contraction the tendon pulls the bone and movement occurs. Muscle contraction is invoked by an action potential, and it can be measured with electrodes on the surface of the skin. The measurement of these action potentials is called an electromyogram or EMG. Like EEG, EMG is also a summation of all the action potentials occurring in a muscle during a period. A single action potential, an average, lasts for 1- 3 milliseconds and can relate to a muscle contraction lasting, on average, 10-100 milliseconds. A contraction as a result of a single action potential is referred to as a twitch. Twitches occurring in high frequency are called a tetanus response and are further categorized as fused and unfused. In unfused tetanus the twitches are still noticeable, whereas in fused tetanus individual twitch can no longer be distinguished. Most muscle contractions correspond to action potentials with frequencies in the range of 8 – 25 Hz.

EMG can be recorded from the surface of the skin with a frequency ranging from 2–500Hz and amplitude between 50uV – 5mV. Movement artifacts within an EMG are generally lower in frequency and can be attenuated with a high pass filter. A simple method for processing an EMG signal is to rectify the signal (take the absolute value) and determine the average value of the waveform. Another method is to find the average power of the EMG signal by calculating the RMS power or root mean squared power of the waveform.

1.1 Electrodes

The type of electrodes used to measure biopotentials are dependent on the biopotentials being recorded and their application. The two electrodes used in this lab are the metal plate snap electrodes and the gold cup electrodes. The metal plate snap electrodes are composed of a small silver or silver plated with chloride disc surrounded by foam padding with an adhesive. The metal plate snap electrode can perform DC readings with low noise and small drift, making them suitable for ECG, EMG, and EOG recordings. The gold cup electrodes are most common is EEG reading for their small size and ease of placement. The gold cup electrodes require a paste to adhere the electrode to the skin.



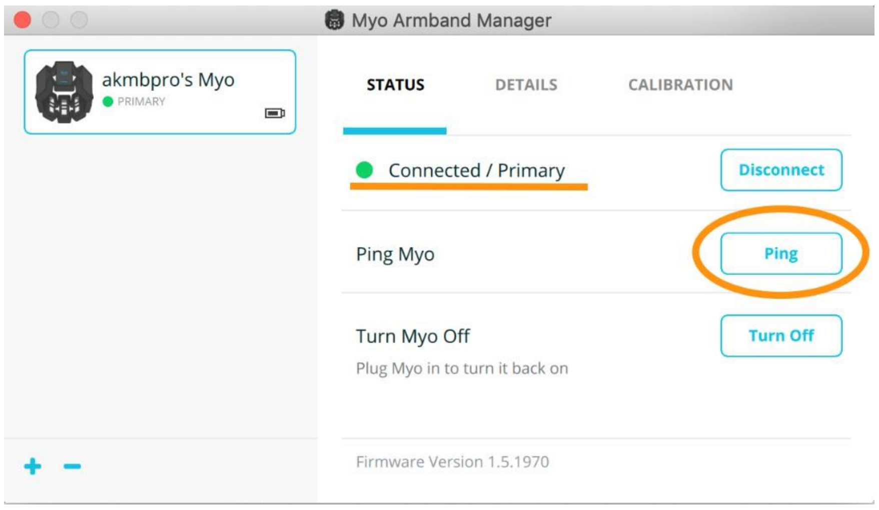
Applications of EMG

1. Medical Diagnostics: Detects neuromuscular disorders (e.g., ALS, carpal tunnel syndrome).
2. Rehabilitation & Prosthetics: Helps control robotic prosthetic limbs using muscle signals.
3. Human-Computer Interaction: Used in gesture recognition (e.g., Myo armbands).
4. Sports & Biomechanics: Monitors muscle fatigue and performance in athletes.

Myo Band Setup:

1. Download MyoConnect from https://myo-connect.software.informer.com/download/#downloading. Available for Windows and MacOS, a simple installation.
2. Insert MYO's Bluetooth dongle in your USB port.
3. Follow the instructions in the software and sync by waving your hand right or as per suggestions in the software.

4. Run MyoConnect, right-click on its icon in task bar, select Armband Manager ....
5. Approach the dongle with your armband. It should automatically get paired with MyoConnect.
6. In MyoConnect, press 'Ping' to make sure that it is not connected to some other armband nearby. Your armband should vibrate in response to the ping.



Myo Band Usage Images:

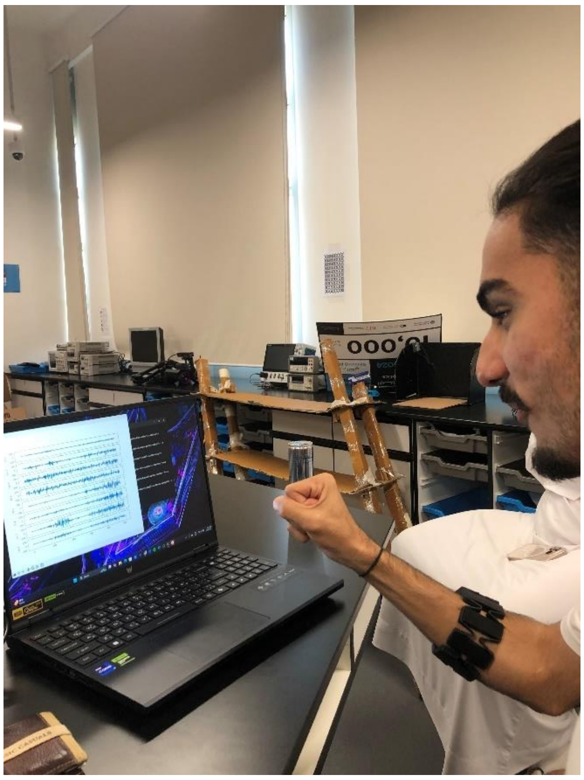
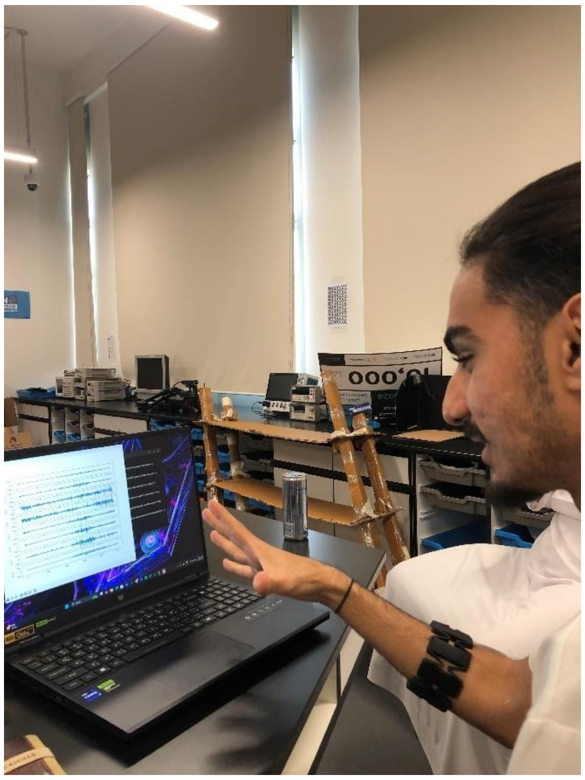



Figure 1 & 2: *Myo Armband placed on the forearm, ensuring proper contact for EMG signal acquisition.*

Data Streaming:

1. Once environment is created and all packages successfully installed.
2. Download the github file and unzip.
3. Activate the conda environment 'myo' or any name you created.

```
conda  activate  myo
```

4. Use cd command to navigate inside the project folder.
5. Navigate to the folder with this package, then to **./examples/streaming** and run a test script:

```
python  streaming.py
```

6. Script emg_streaming.py demonstrates a method to collect and plot EMG data from the armband in a real-time manner. Class MultichannelPlot provides a solution for fast plotting of multichannel signals.

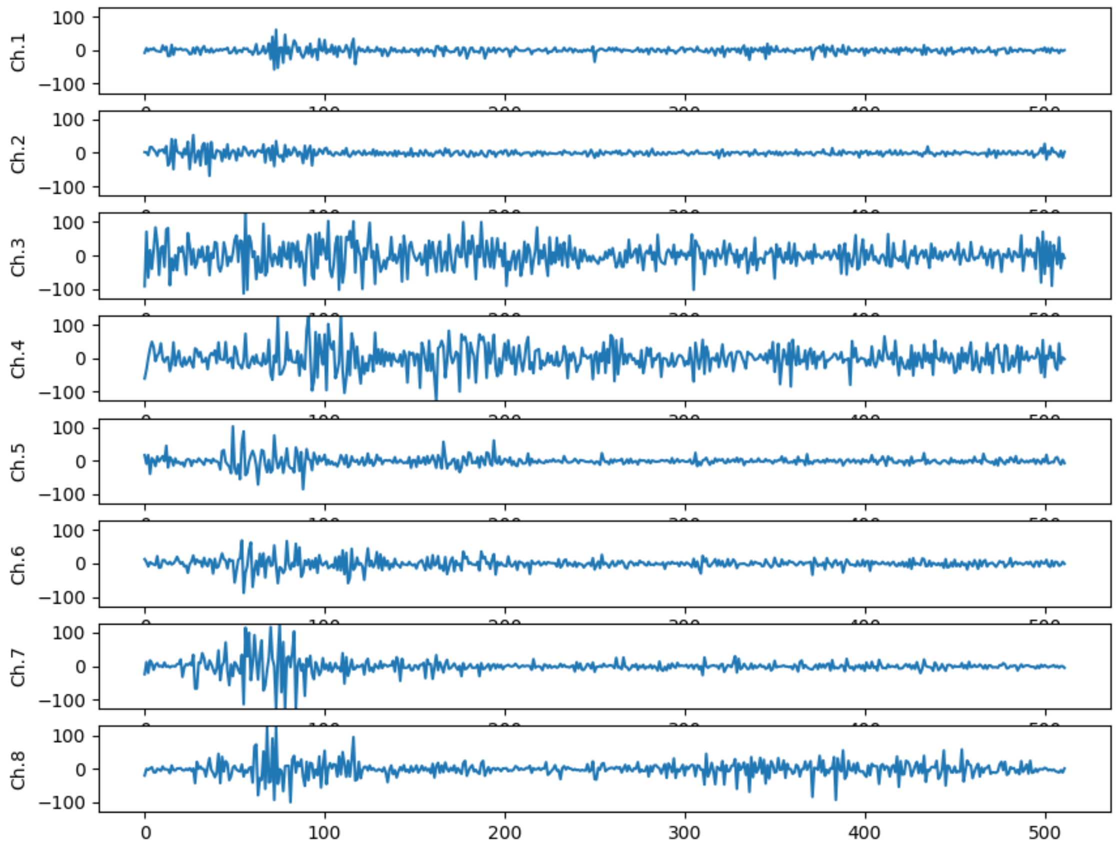


***Figure 3:** EMG signal data streaming from the Myo Armband across 8 channels. The signals represent muscle activity detected in real-time.*

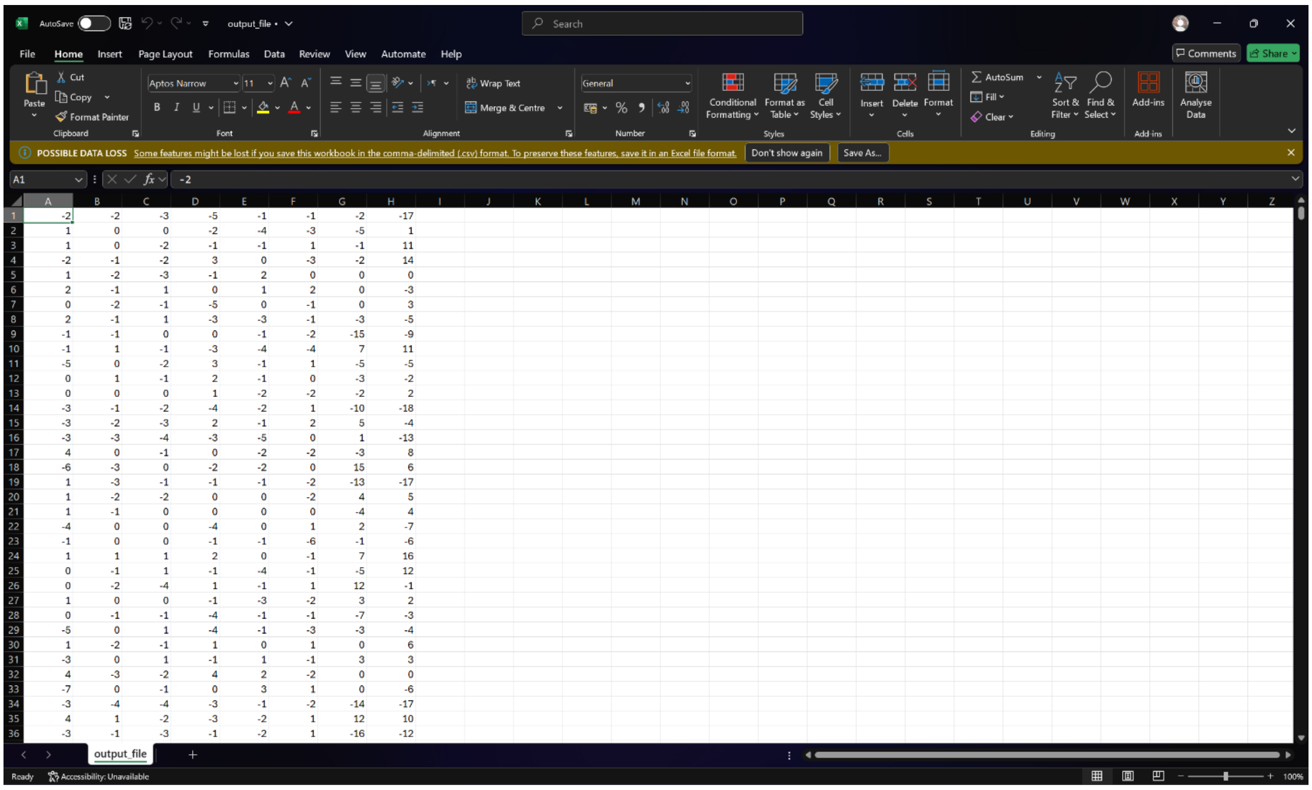
The image (Figure 3) shows the EMG signals recorded from the Myo Armband while streaming data. Each of the eight channels represents muscle activity detected from different parts of the arm. The wavy lines indicate muscle movements, where stronger signals mean more muscle activation, and flatter lines show less movement. This confirms that the Myo Band is successfully capturing real-time muscle signals, which will be used for further analysis.

Data Acquistion:

1. Use the following command to acquire data

```
python acquisition.py 10 filename
```

2. Script acquisition.py demonstrates a method to collect a specified amount of EMG data from the armband and save it to a file. It uses Class Collector. The following command will acquire the signal for 10 seconds and save it to file **filename.csv**.
3. The csv file contains muscle signals from 8 channels.



	A	B	C	D	E	F	G	H
1	-2	-2	-3	-5	-1	-1	-2	-17
2	1	0	0	-2	-4	-3	-5	1
3	1	0	-2	-1	-1	1	-1	11
4	-2	-1	-2	3	0	-3	-2	14
5	1	-2	-3	-1	2	0	0	0
6	2	-1	1	0	1	2	0	-3
7	0	-2	-1	-5	0	-1	0	3
8	2	-1	1	-3	-3	-1	-3	-5
9	-1	-1	0	0	-1	-2	-15	-9
10	-1	1	-1	-3	-4	-4	7	11
11	-5	0	-2	3	-1	1	-5	-5
12	0	1	-1	2	-1	0	-3	-2
13	0	0	0	1	-2	-2	-2	2
14	-3	-1	-2	-4	-2	1	-10	-18
15	-3	-2	-3	2	-1	2	5	-4
16	-3	-3	-4	-3	-5	0	1	-13
17	4	0	-1	0	-2	-2	-3	8
18	-6	-3	0	-2	-2	0	15	6
19	1	-3	-1	-1	-1	-2	-13	-17
20	1	-2	-2	0	0	-2	4	5
21	1	-1	0	0	0	0	-4	4
22	-4	0	0	-4	0	1	2	-7
23	-1	0	0	-1	-1	-6	-1	-6
24	1	1	1	2	0	-1	7	16
25	0	-1	1	-1	-4	-1	-5	12
26	0	-2	-4	1	-1	1	12	-1
27	1	0	0	-1	-3	-2	3	2
28	0	-1	-1	-4	-1	-1	-7	-3
29	-5	0	1	-4	-1	-3	-3	-4
30	1	-2	-1	1	0	1	0	6
31	-3	0	1	-1	1	-1	3	3
32	4	-3	-2	4	2	-2	0	0
33	-7	0	-1	0	3	1	0	-6
34	-3	-4	-4	-3	-1	-2	-14	-17
35	4	1	-2	-3	-2	1	12	10
36	-3	-1	-3	-1	-2	1	-16	-12

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Figure 4: *Snippet of the acquired EMG data saved in output_file.csv, showing muscle activity recorded over 2000 data points.*

The data acquisition step involved collecting EMG signals from the Myo Armband and saving them into a CSV file named "output_file.csv". The recording captured 2000 lines of data, representing muscle activity over time across eight channels. Each row in the file corresponds to a timestamped EMG reading, with values showing the electrical activity of different muscles. This data will be used for further signal processing and analysis to extract useful insights about muscle movements.

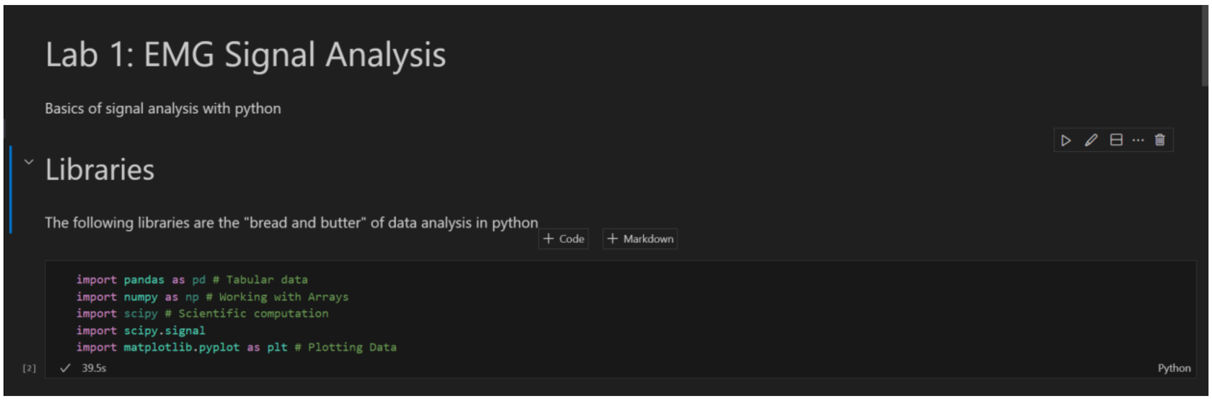
DATA ANALYSIS:

1. Download and extract zip file from https://github.com/BioRobotics- Spring-2020/Lab 1 2022.git

2. Please open the Lab_1_EMG_Analysis.ipynb jupyter notebook and follow the

directions there. You can download p18 emg.csv from MyCourses and put it in the Data folder.

3. Run each cell and note down the output.

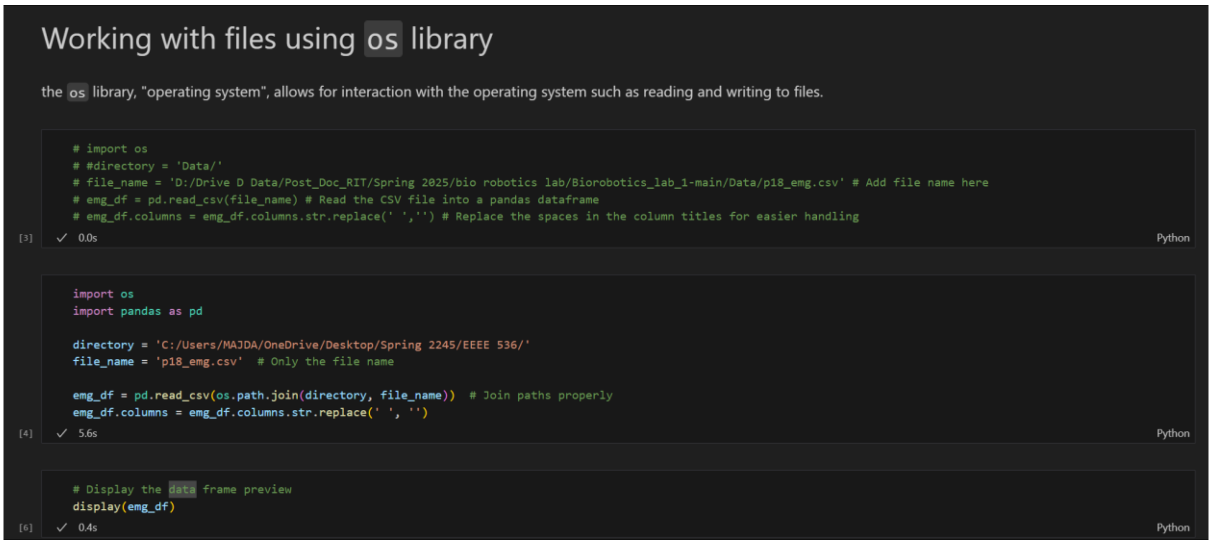

Lab 1: EMG Signal Analysis

Basics of signal analysis with python

Libraries

The following libraries are the "bread and butter" of data analysis in python

```python
import pandas as pd # Tabular data
import numpy as np # Working with Arrays
import scipy # Scientific computation
import scipy.signal
import matplotlib.pyplot as plt # Plotting Data
```
[2] ✓ 39.5s Python


Working with files using os library

the os library, "operating system", allows for interaction with the operating system such as reading and writing to files.

```python
# import os
# #directory = 'Data/'
# file_name = 'D:/Drive D Data/Post_Doc_RIT/Spring 2025/bio robotics lab/Biorobotics_lab_1-main/Data/p18_emg.csv' # Add file name here
# emg_df = pd.read_csv(file_name) # Read the CSV file into a pandas dataframe
# emg_df.columns = emg_df.columns.str.replace(' ','') # Replace the spaces in the column titles for easier handling
```
[3] ✓ 0.0s Python

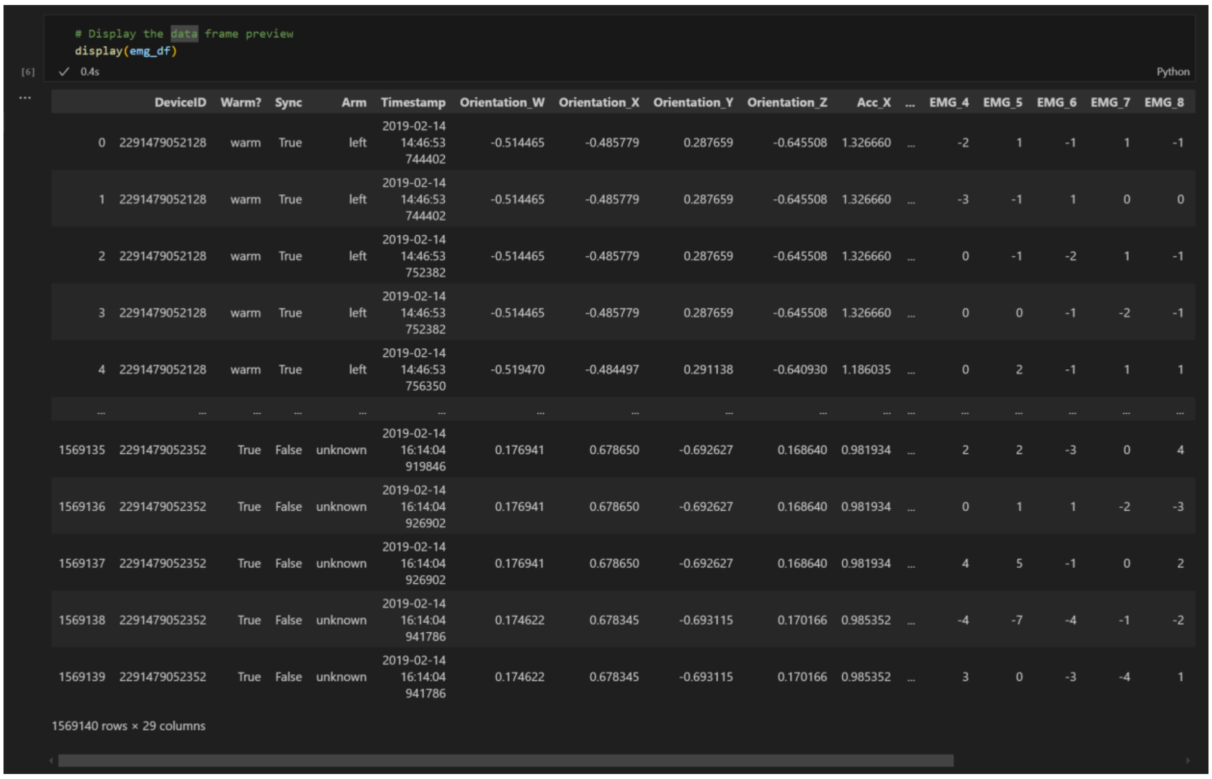
```python
import os
import pandas as pd

directory = 'C:/Users/MAJDA/OneDrive/Desktop/Spring 2245/EEEE 536/'
file_name = 'p18_emg.csv'  # Only the file name

emg_df = pd.read_csv(os.path.join(directory, file_name))  # Join paths properly
emg_df.columns = emg_df.columns.str.replace(' ', '')
```
[4] ✓ 5.6s Python

```python
# Display the data frame preview
display(emg_df)
```
[6] ✓ 0.4s Python

Figure 5 & 6: *Importing essential libraries for EMG signal analysis in Python, including pandas, numpy, scipy, and matplotlib. The script also demonstrates loading and reading the EMG data file using the os and pandas libraries, ensuring proper data handling for further analysis.*



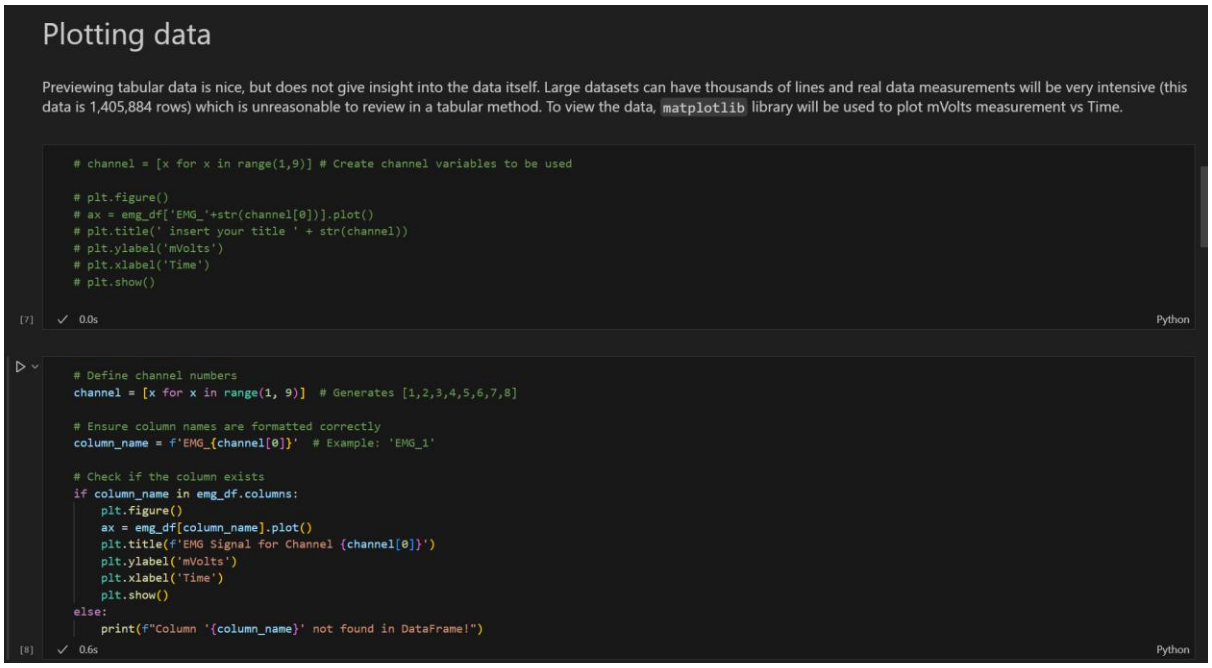
```
# Display the data frame preview
display(emg_df)
```

[6] ✓ 0.4s Python

	DeviceID	Warm?	Sync	Arm	Timestamp	Orientation_W	Orientation_X	Orientation_Y	Orientation_Z	Acc_X	...	EMG_4	EMG_5	EMG_6	EMG_7	EMG_8
0	2291479052128	warm	True	left	2019-02-14 14:46:53 744402	-0.514465	-0.485779	0.287659	-0.645508	1.326660	...	-2	1	-1	1	-1
1	2291479052128	warm	True	left	2019-02-14 14:46:53 744402	-0.514465	-0.485779	0.287659	-0.645508	1.326660	...	-3	-1	1	0	0
2	2291479052128	warm	True	left	2019-02-14 14:46:53 752382	-0.514465	-0.485779	0.287659	-0.645508	1.326660	...	0	-1	-2	1	-1
3	2291479052128	warm	True	left	2019-02-14 14:46:53 752382	-0.514465	-0.485779	0.287659	-0.645508	1.326660	...	0	0	-1	-2	-1
4	2291479052128	warm	True	left	2019-02-14 14:46:53 756350	-0.519470	-0.484497	0.291138	-0.640930	1.186035	...	0	2	-1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1569135	2291479052352	True	False	unknown	2019-02-14 16:14:04 919846	0.176941	0.678650	-0.692627	0.168640	0.981934	...	2	2	-3	0	4
1569136	2291479052352	True	False	unknown	2019-02-14 16:14:04 926902	0.176941	0.678650	-0.692627	0.168640	0.981934	...	0	1	1	-2	-3
1569137	2291479052352	True	False	unknown	2019-02-14 16:14:04 926902	0.176941	0.678650	-0.692627	0.168640	0.981934	...	4	5	-1	0	2
1569138	2291479052352	True	False	unknown	2019-02-14 16:14:04 941786	0.174622	0.678345	-0.693115	0.170166	0.985352	...	-4	-7	-4	-1	-2
1569139	2291479052352	True	False	unknown	2019-02-14 16:14:04 941786	0.174622	0.678345	-0.693115	0.170166	0.985352	...	3	0	-3	-4	1

1569140 rows × 29 columns

Figure 7: *Preview of the EMG dataset, displaying multiple columns, including device ID, timestamp, orientation data, acceleration, and EMG signals from eight channels. This confirms successful data loading for further analysis.*



Plotting data

Previewing tabular data is nice, but does not give insight into the data itself. Large datasets can have thousands of lines and real data measurements will be very intensive (this data is 1,405,884 rows) which is unreasonable to review in a tabular method. To view the data, `matplotlib` library will be used to plot mVolts measurement vs Time.

```
# channel = [x for x in range(1,9)] # Create channel variables to be used

# plt.figure()
# ax = emg_df['EMG_'+str(channel[0])].plot()
# plt.title(' insert your title ' + str(channel))
# plt.ylabel('mVolts')
# plt.xlabel('Time')
# plt.show()
```

[7] ✓ 0.0s Python

```
# Define channel numbers
channel = [x for x in range(1, 9)]  # Generates [1,2,3,4,5,6,7,8]

# Ensure column names are formatted correctly
column_name = f'EMG_{channel[0]}'  # Example: 'EMG_1'

# Check if the column exists
if column_name in emg_df.columns:
    plt.figure()
    ax = emg_df[column_name].plot()
    plt.title(f'EMG Signal for Channel {channel[0]}')
    plt.ylabel('mVolts')
    plt.xlabel('Time')
    plt.show()
else:
    print(f"Column '{column_name}' not found in DataFrame!")
```

[8] ✓ 0.6s Python

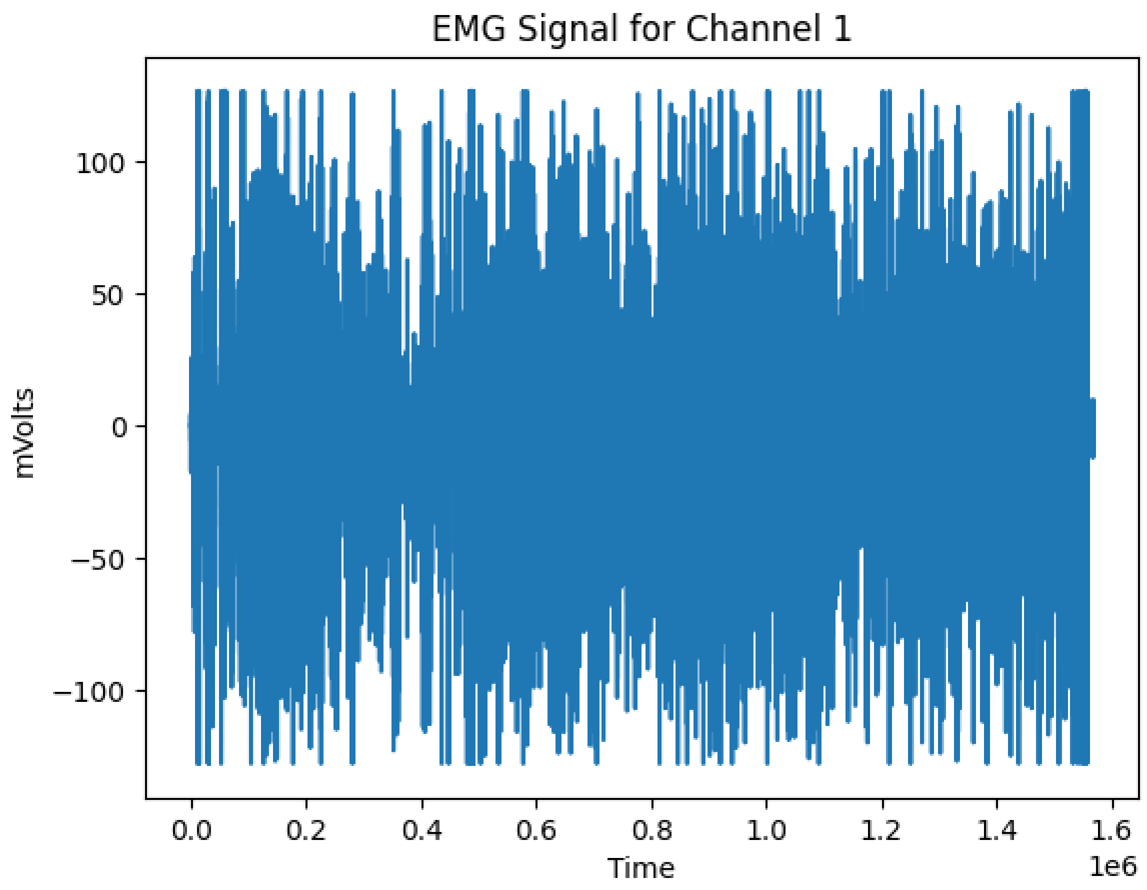
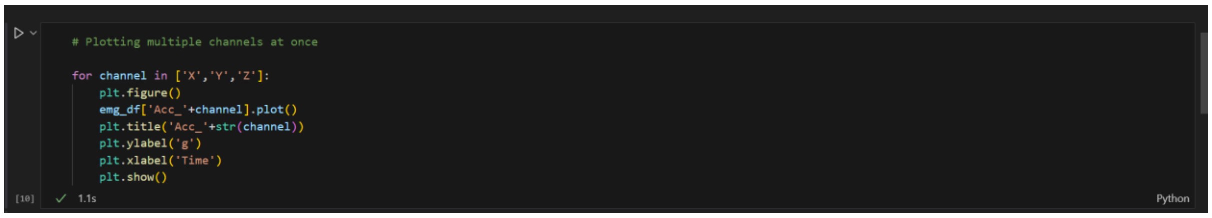
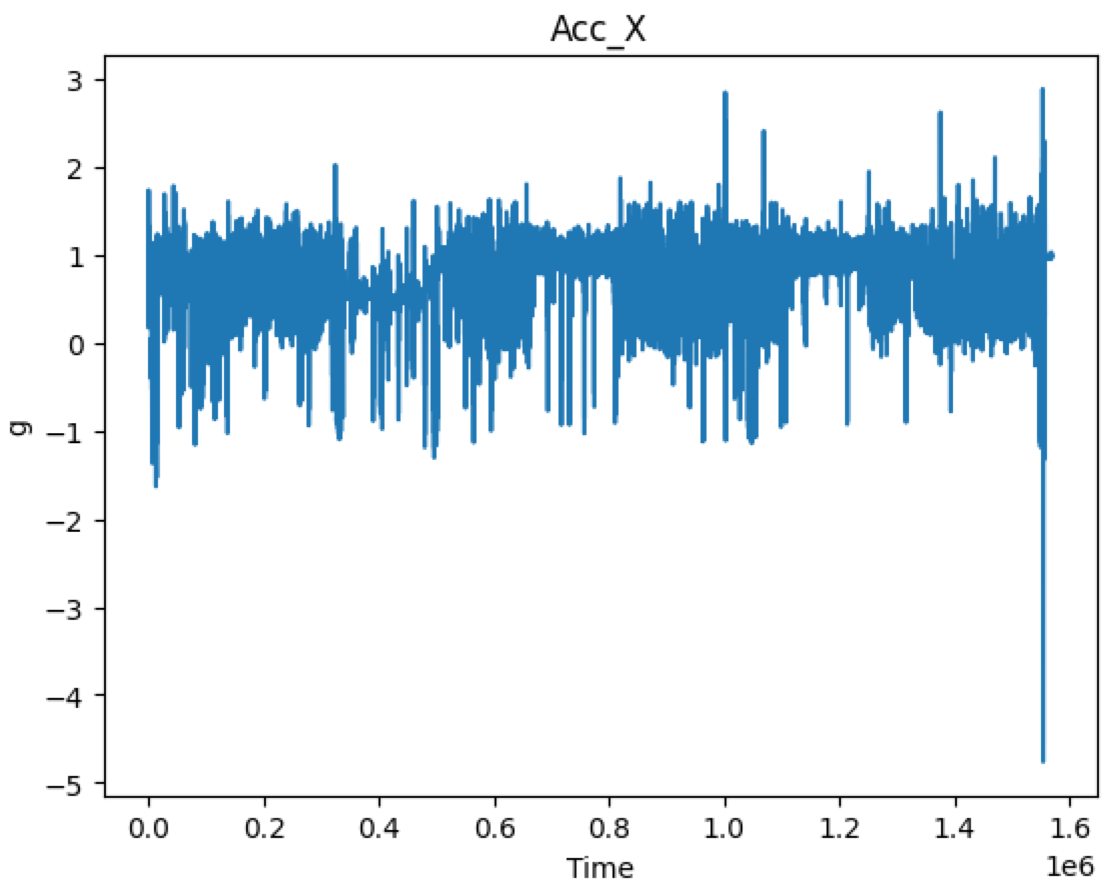


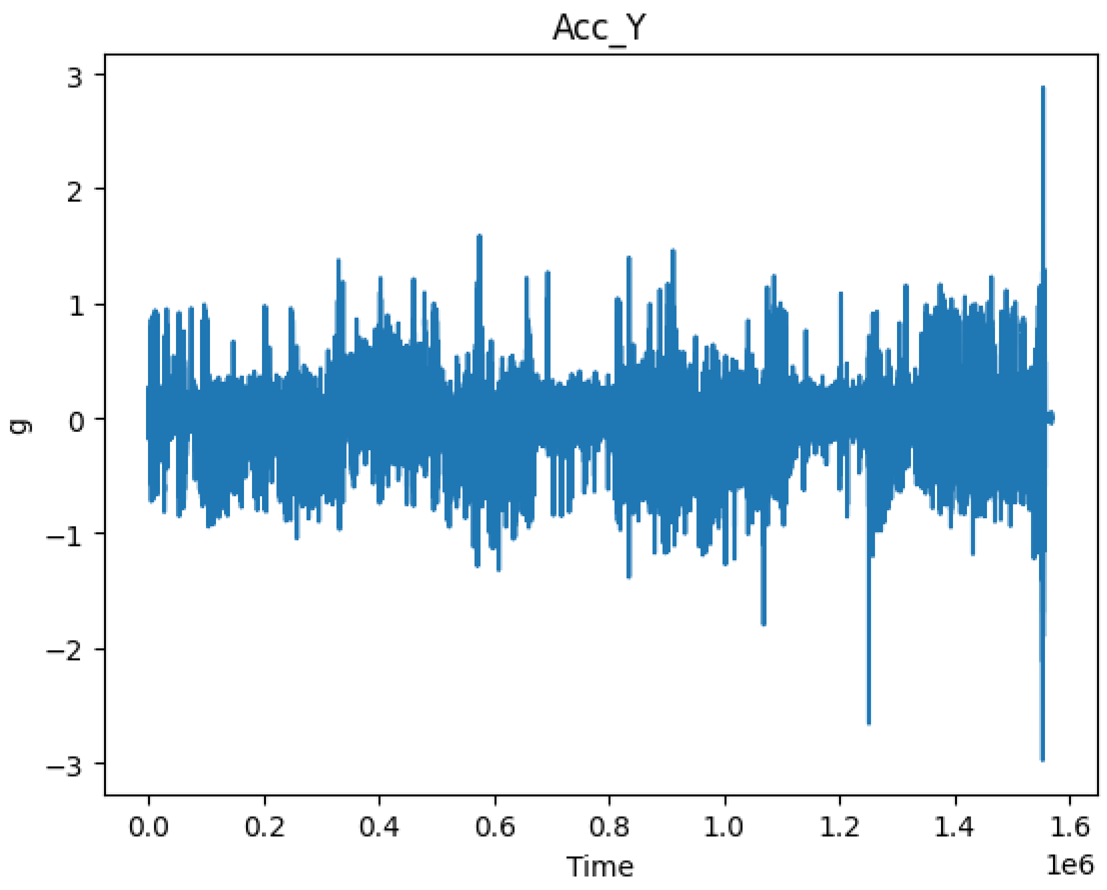
Figure 8 & 9: *Plotting EMG signals using the matplotlib library. The first image shows the Python script used to visualize the data, and the second image displays the EMG signal for Channel 1 over time, measured in millivolts (mV).*



```
# Plotting multiple channels at once

for channel in ['X','Y','Z']:
    plt.figure()
    emg_df['Acc_'+channel].plot()
    plt.title('Acc_'+str(channel))
    plt.ylabel('g')
    plt.xlabel('Time')
    plt.show()
```


Acc_X
g
3
2
1
0
−1
−2
−3
−4
−5
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Time
1e6


Acc_Y
g
3
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Time
1e6

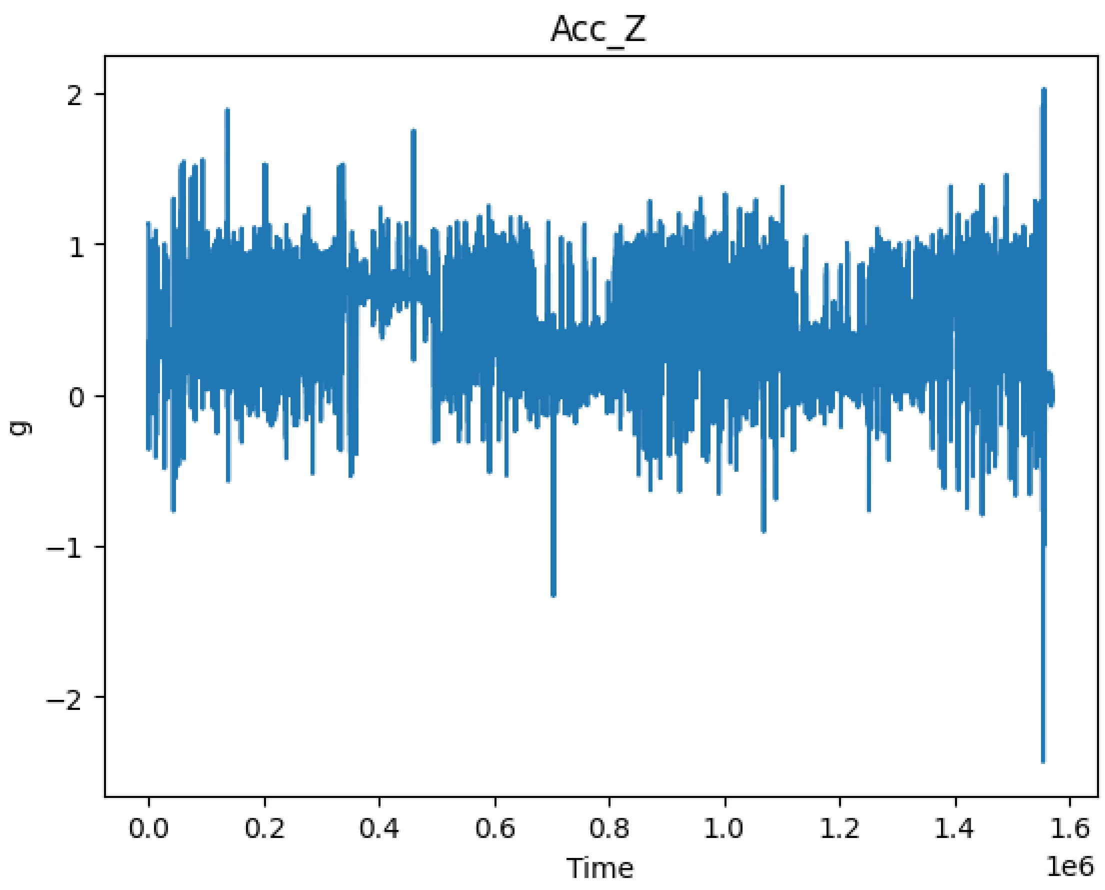
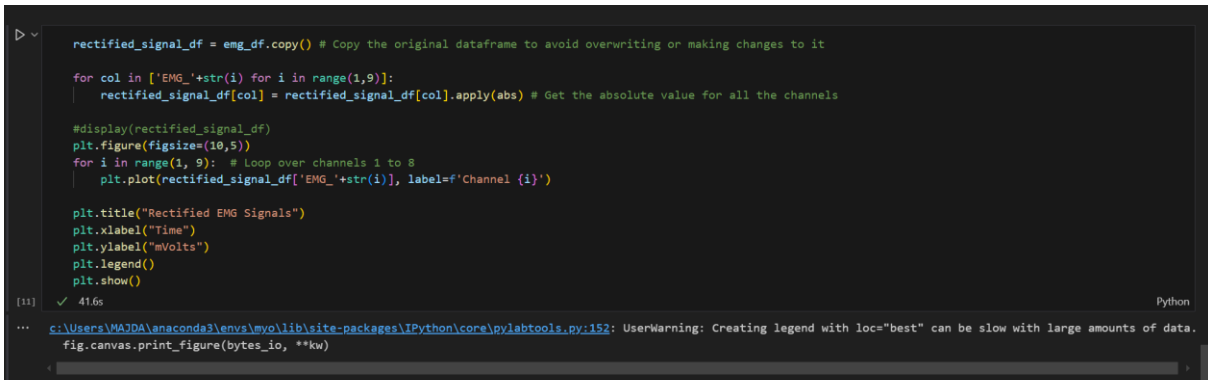


Figure 10–13: *Acceleration data visualization for the X, Y, and Z axes. The first image shows the Python script used to plot multiple acceleration channels, while the following three plots display the acceleration signals along each axis over time, measured in gravitational force (g).*



```
rectified_signal_df = emg_df.copy() # Copy the original dataframe to avoid overwriting or making changes to it

for col in ['EMG_'+str(i) for i in range(1,9)]:
    rectified_signal_df[col] = rectified_signal_df[col].apply(abs) # Get the absolute value for all the channels

#display(rectified_signal_df)
plt.figure(figsize=(10,5))
for i in range(1, 9):  # Loop over channels 1 to 8
    plt.plot(rectified_signal_df['EMG_'+str(i)], label=f'Channel {i}')

plt.title("Rectified EMG Signals")
plt.xlabel("Time")
plt.ylabel("mVolts")
plt.legend()
plt.show()
```

```
[11]  ✓ 41.6s                                                                 Python
```

```
c:\Users\MAJDA\anaconda3\envs\myo\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)
```

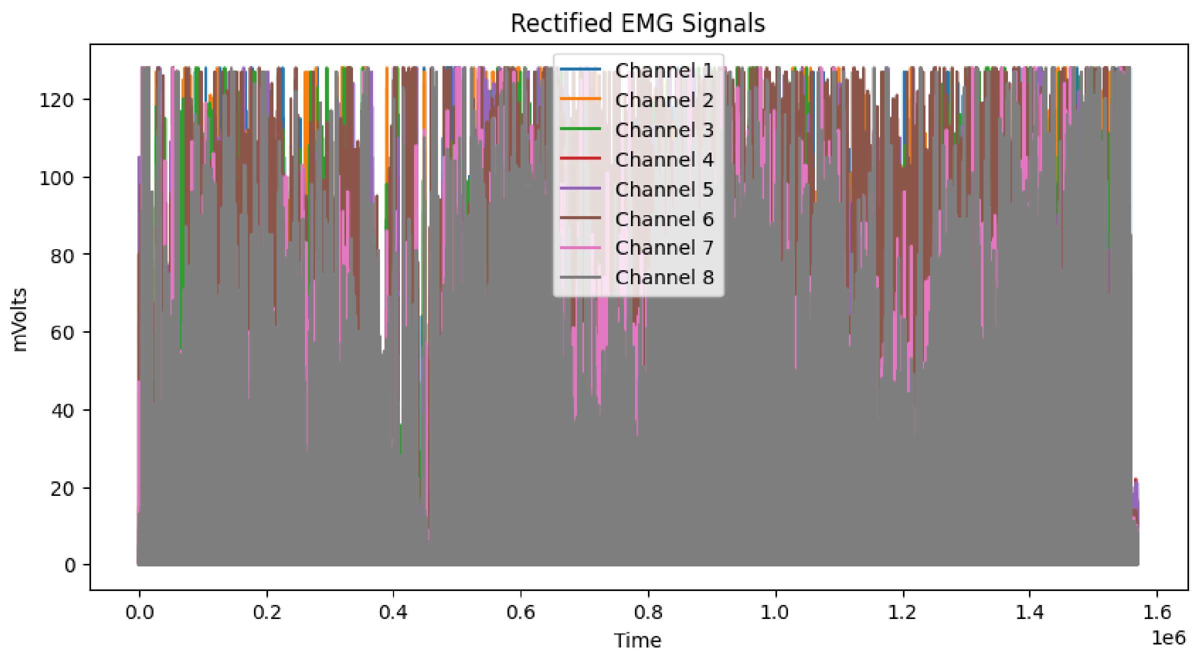
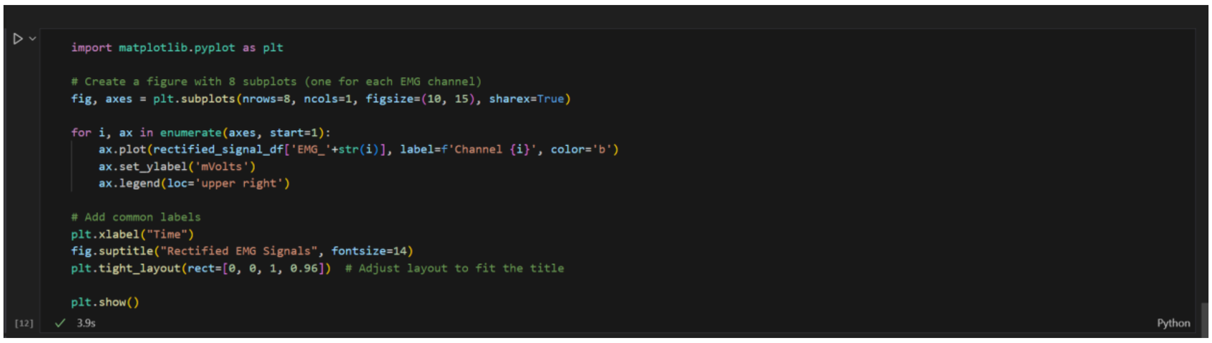


Figure 14 & 15: *Rectified EMG signals visualization. The first image shows the Python script applying rectification by taking the absolute value of EMG signals. The second image displays the rectified EMG signals for all eight channels, ensuring that all values are positive for better signal analysis.*



```
import matplotlib.pyplot as plt

# Create a figure with 8 subplots (one for each EMG channel)
fig, axes = plt.subplots(nrows=8, ncols=1, figsize=(10, 15), sharex=True)

for i, ax in enumerate(axes, start=1):
    ax.plot(rectified_signal_df['EMG_'+str(i)], label=f'Channel {i}', color='b')
    ax.set_ylabel('mVolts')
    ax.legend(loc='upper right')

# Add common labels
plt.xlabel("Time")
fig.suptitle("Rectified EMG Signals", fontsize=14)
plt.tight_layout(rect=[0, 0, 1, 0.96])  # Adjust layout to fit the title

plt.show()
```
[12] ✓ 3.9s Python

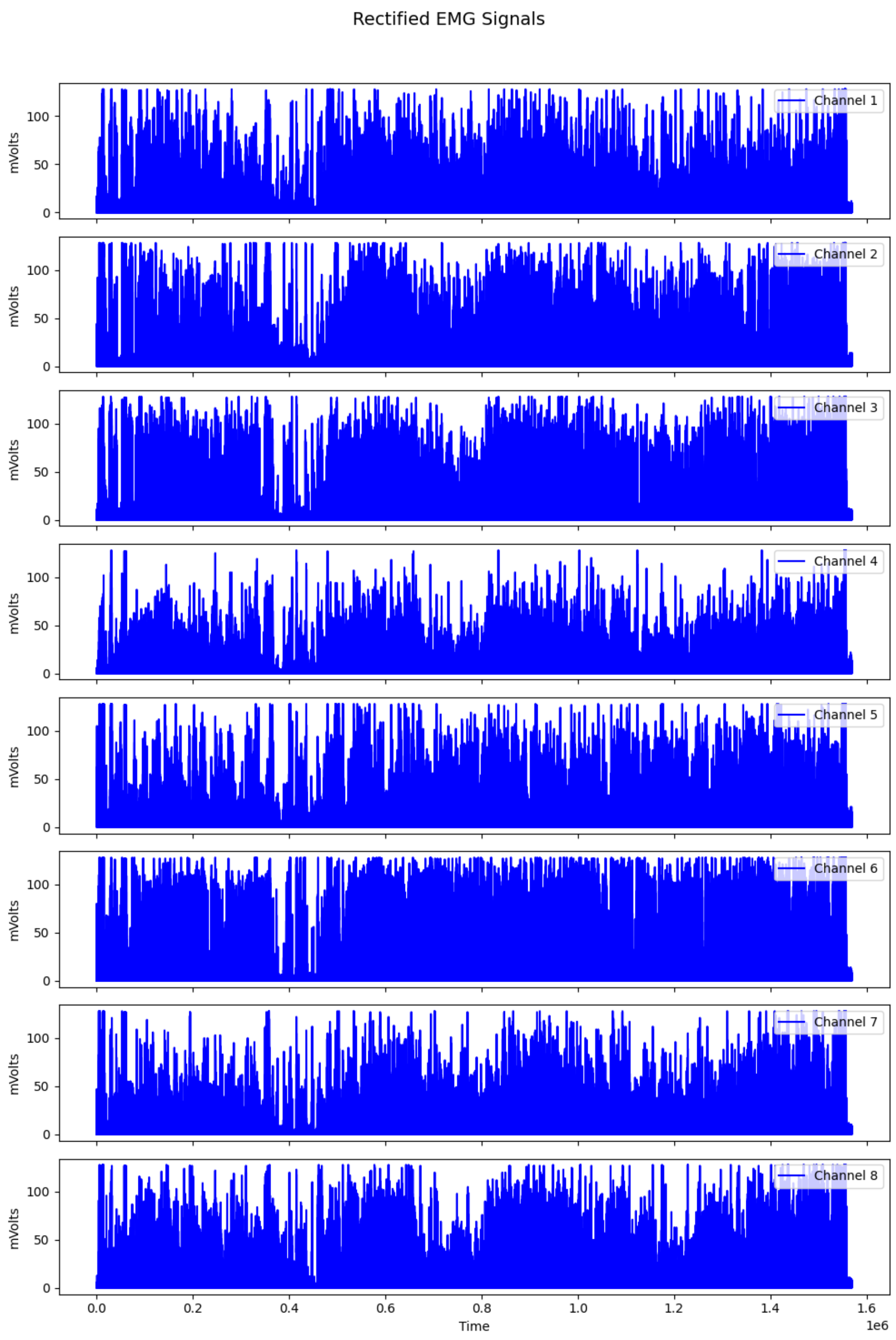


Figure 16 & 17: *Python script and resulting plot for rectified EMG signals across eight channels. The script generates subplots for each channel, allowing a clear visualization of muscle activity. The plotted signals display the rectified EMG data, highlighting variations in signal strength over time for all eight channels.*

Following data acquisition, the recorded EMG signals were processed and analyzed using Python in Jupyter Notebook. The analysis began with importing key libraries needed for handling and visualizing the data, such as pandas, numpy, scipy, and matplotlib (Figures 5 & 6). The EMG dataset was then loaded, and an initial preview was displayed to confirm the structure, which included details such as timestamps, muscle activity across multiple channels, and sensor orientation (Figure 7).

To better understand the signals, raw EMG data was plotted, allowing visualization of muscle activity patterns over time (Figures 8 & 9). Additionally, acceleration data along the X, Y, and Z axes was plotted to examine movement variations and external influences on the EMG signals (Figures 10–13). Since raw EMG data includes both positive and negative values, rectification was applied by converting all values to their absolute form, ensuring a more consistent representation of muscle activity (Figures 14 & 15).

To enhance clarity, the rectified signals were organized into subplots, displaying each EMG channel separately for easier comparison of muscle activity levels (Figures 16 & 17). This structured approach made it possible to identify differences between channels and observe patterns in muscle contractions, which can be useful for applications such as biomedical research, rehabilitation, and gesture recognition systems.

Group Reflection on the Lab Experience

This lab was a great learning experience for our group. At first, we faced some challenges with setting up the Myo Armband and installing the necessary software, but by working together, we solved these issues. Each member contributed by handling different tasks like data collection, coding, and signal analysis. We also learned the importance of cleaning and processing EMG signals, as raw data can be noisy. Seeing the real-time muscle activity reflected in our plots made the experiment more exciting. Overall, this lab helped us understand how EMG technology works and how it is used in real-world applications.

Discussion: The Importance of EMG in Real-World Applications

Electromyography (EMG) is very useful in different areas, including medicine, rehabilitation, technology, and sports. It helps measure muscle activity and provides valuable information about how muscles work.

- **Medical Diagnostics**
  - Doctors use EMG to find and diagnose muscle and nerve problems. It can detect

neuromuscular disorders like ALS (Amyotrophic Lateral Sclerosis), muscular dystrophy, and carpal tunnel syndrome. When a patient has muscle weakness or pain, an EMG test can show if the problem is in the muscles themselves or the nerves controlling them. This helps doctors make better decisions about treatment and therapy.

- **Rehabilitation & Prosthetics**
    - EMG plays a big role in helping people recover from injuries and improving prosthetic limb control. Moreover, they help in physical therapy and rehabilitation, helping individuals regain muscle strength and mobility after injuries or surgeries.
        - Rehabilitation: Patients recovering from injuries or surgeries often need therapy to rebuild their muscle strength. EMG helps therapists track muscle activity and progress over time. This is especially important for people who have had strokes or spinal cord injuries.
        - Prosthetics: For people who have lost a limb, EMG allows them to control robotic prosthetic arms or legs using their own muscle signals. When they try to move a missing limb, their muscles still generate signals, which the prosthetic can recognize and respond to. This makes bionic limbs more natural and easier to use.
- **Human-Computer Interaction (HCI)**
    - EMG is used in gesture-based technology, where people can control computers, robots, or virtual reality (VR) systems just by moving their muscles. Devices like the Myo Armband can read muscle movements and allow users to control a computer, play games, or operate a robotic arm without needing a mouse or keyboard. EMG also helps people with disabilities, as it allows them to control wheelchairs or assistive devices just by moving their muscles.
      This technology is useful for gaming, robotics, and smart home control, making it easier to interact with machines in new ways.
- **Sports & Biomechanics**
    - Athletes and coaches use EMG to study muscle performance and prevent injuries. EMG helps track muscle fatigue, which means it shows when an athlete's muscles are getting too tired or overworked. This can help prevent muscle strain or injuries. It also helps improve training techniques by showing which muscles are being used the most. Coaches can then adjust workouts to strengthen weaker muscles and improve

performance. Moreover, physical therapists also use EMG to check if an injured athlete’s muscles are healing properly before they return to sports.

- **Robotics and Exoskeletons**
    - EMG is used to control robotic suits (exoskeletons), which help people with paralysis or weak muscles walk again. These suits read muscle signals and provide extra support to help people stand, walk, or lift objects.
      Exoskeletons are used in hospitals to help paralyzed patients move again and in factories to help workers lift heavy loads without straining their muscles.
- **Assistive Technology for Disabled Individuals**
    - EMG-based systems allow people with limited mobility to control devices such as wheelchairs, home automation systems, and speech-generating devices using muscle movements. This enhances accessibility and improves the quality of life for individuals with severe disabilities.

In conclusion, EMG is a valuable tool with many real-world applications in medicine, technology, and sports.It helps doctors diagnose muscle problems, patients regain movement, and athletes improve performance. It also allows people to control computers, prosthetic limbs, and robotic devices using their muscles. This lab gave us hands-on experience with EMG data collection and analysis, helping us understand how muscle signals can be used in many different ways to improve lives.