BME 252 Project

Gabriel Jabile (20967637)
Daniel Shim (20944050)
Andrew Shin (20948601)

PART A:

A3:

A butterworth notch filter with an order of 8 and a sampling rate of 2048 Hz was used to remove the 60 Hz noise in the emg signal. Notch filter was chosen due to its ability to attenuate a specific frequency without distorting the remaining signal. In this case a notch frequency of 60 Hz was able to eliminate the noise from the 60 Hz power line while preserving the rest. A narrow notch of 1 was used since the goal was to attenuate a specific frequency. Butterworth filters were chosen since they are stable and have a flat frequency response within the range of interest. This helps increase the accuracy of the signal by eliminating any oscillation or potential signal distortion. Sample rate was calculated using the find_sampling_rate function. To determine the order, trial and error was performed until the satisfied result was obtained. Order of 8 provided a steep roll-off and helped suppress unwanted frequency. It was noticed that anything greater than an order of 8 made little to no noticeable changes to the overall filter.

Transfer Function:

$$H(jw) = 1 / ((s - 15.699) * (s - 115.792) * (s - 532.637) * (s - 1710.210) * (s - 4064.252) * (s - 7394.751) * (s - 10507.657) * (s - 11784.638) * (s - 10466.428) * (s - 7336.834) * (s - 4016.598) * (s - 1683.525) * (s - 522.269) * (s - 113.093) * (s - 15.273) * (s - 0.969))$$

A4:

A butterworth band pass filter with an order of 8 and a sampling rate of 2048.2 Hz was used to remove the noise outside the range of 0.1-450 Hz. Bandpass filters can effectively remove all frequencies outside of the desired values. In this case the low and high cut-off was set at 0.1 Hz and 450 Hz respectively to discard any frequencies outside that range. The reasoning behind the usage of butterworth filter, and its order are similar to the notch filter from A3. An order of 8 resulted in a narrower bandwidth and steeper roll-off for both stopbands. Similarly, orders higher than 8 have no noticeable impact on the filter's performance.

Transfer Function:

$$H(jw) = 1 / ((s - 8.962) * (s - 37.133) * (s - 95.222) * (s - 170.922) * (s - 230.019) * (s - 241.669) * (s - 202.591) * (s - 136.808) * (s - 74.612) * (s - 32.754) * (s - 11.431) * (s - 3.100) * (s - 0.633) * (s - 0.092) * (s - 0.008) * (s - 0.000))$$

A5:

A butterworth notch filter and bandpass filter of order 8 was used to eliminate the 60 Hz (notch frequency) and 0.1/450 Hz (low/high cut-off) respectively. The filter, cut-off and order selection can be justified by the explanation in A3 and A4 respectively.

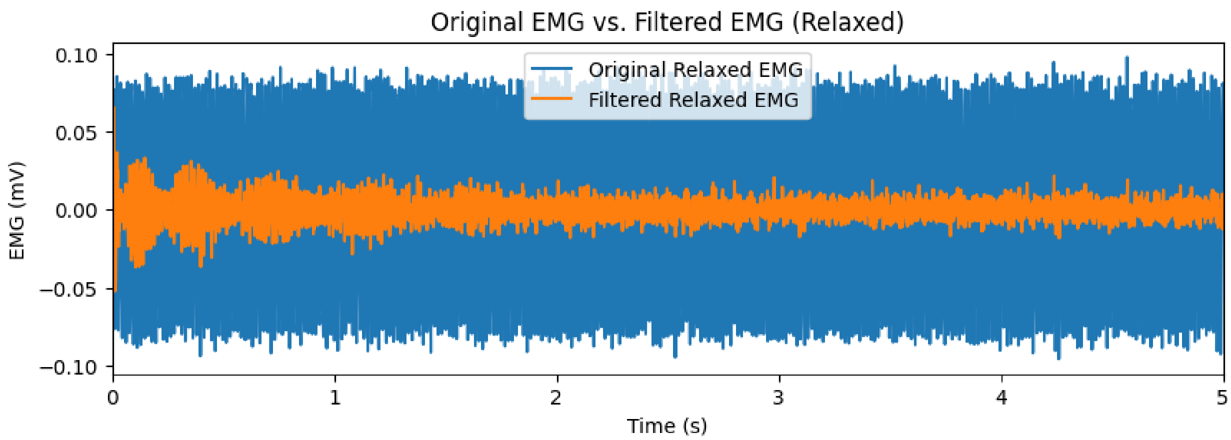

Figure 1: Original vs Filter Signal (Relaxed)

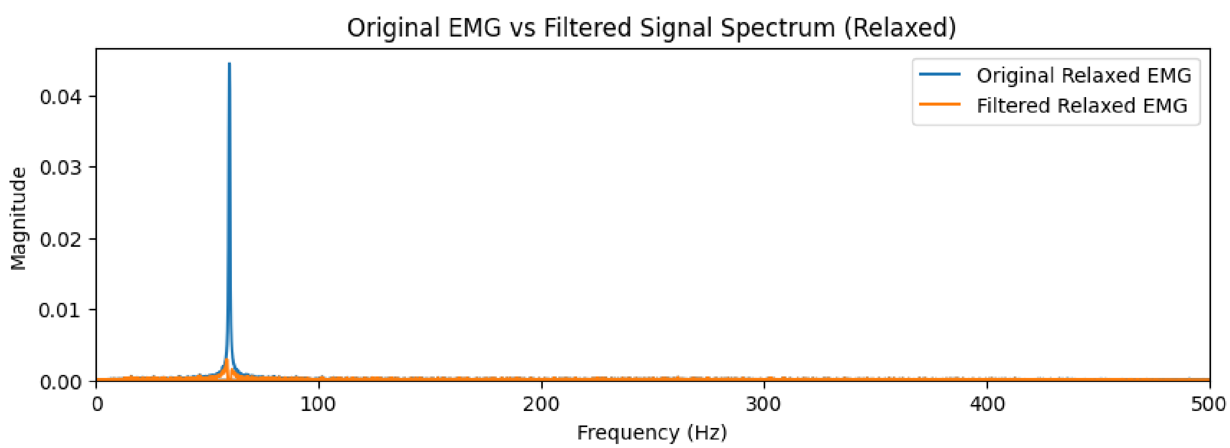

Figure 2: Original vs Filter Signal Spectrum (Relaxed)

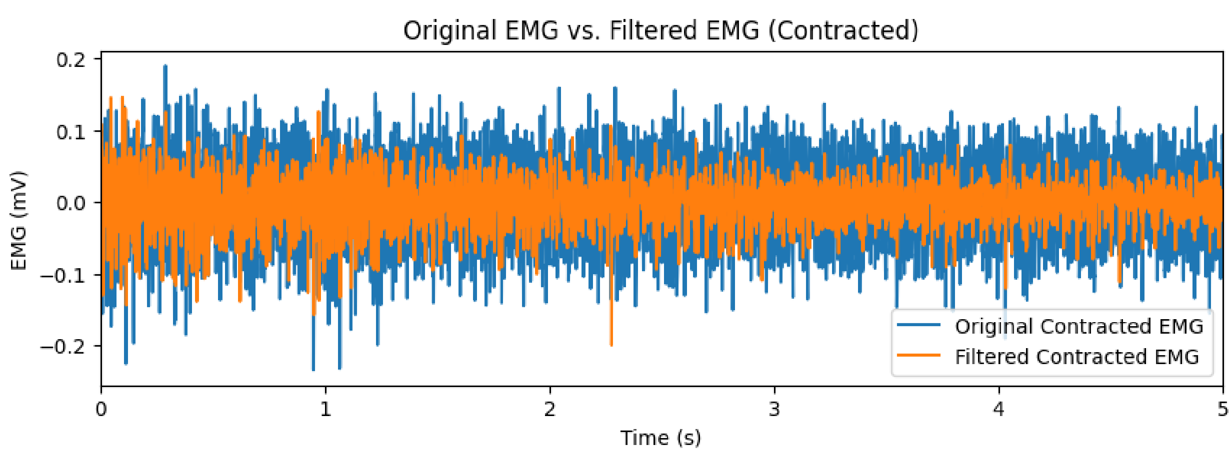

Figure 3: Original vs Filter Signal (Contracted)

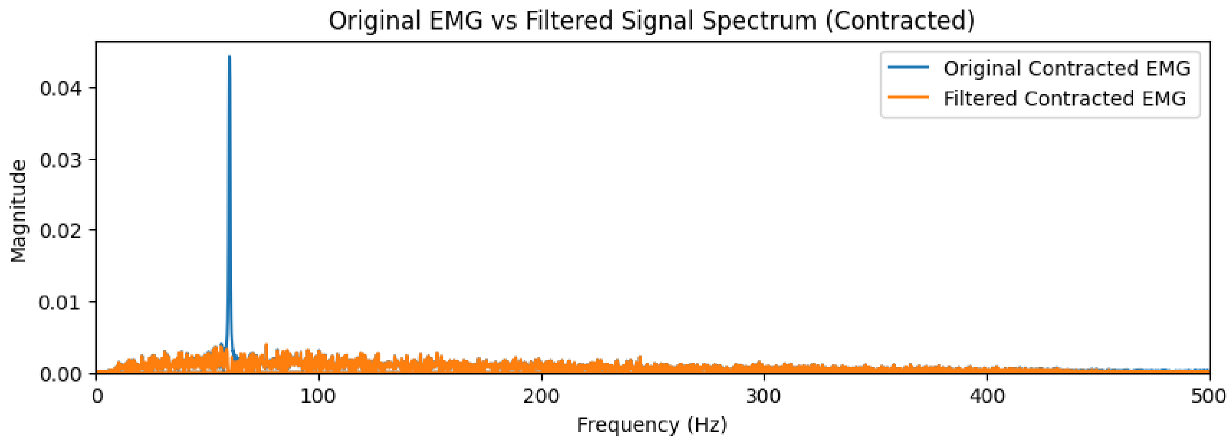

Figure 4: Original vs Filter Signal Spectrum (Contracted)

Original RMS Value (Relaxed): 56.13e-05
Filtered RMS Value (Relaxed): 7.806e-05
Original RMS Value (Contracted): 63.74e-05
Filtered RMS Value (Contracted): 31.16e-05

Transfer Function:

B(jw) = 1 / ((s - 24.661) * (s - 293.619) * (s - 2248.541) * (s - 12449.240) * (s - 53108.881) * (s - 181686.697) * (s - 512216.784) * (s - 1213677.983) * (s - 2452854.214) * (s - 4276328.864) * (s - 6488203.163) * (s - 8626110.992) * (s - 10102943.061) * (s - 10465392.292) * (s - 9615403.167) * (s - 7849750.254) * (s - 5698516.989) * (s - 3677963.509) * (s - 2108094.480) * (s - 1070679.125) * (s - 480253.418) * (s - 189369.461) * (s - 65235.445) * (s - 19473.782) * (s - 4983.868) * (s - 1078.146) * (s - 193.390) * (s - 27.997) * (s - 3.144) * (s - 0.257) * (s - 0.014) * (s - 0.000))

Upon filtering the two emg signals and calculating their respective RMS values, a threshold of 30e-05 was chosen. The is_cont function, compares the filtered RMS value to this threshold and reads “Muscle Contraction” if the RMS value is greater than the threshold.

The filtered signals in Figures 1 and 3 are noticeably cleaner after being filtered through the two filters. Figure 2 and 4 illustrates a clear attenuation of the 60 Hz frequency as well as the frequency outside of the desired 0.1-450 Hz range. Looking at the plots and RMS values, the 60 Hz did indeed corrupt our ability to decide if the muscle is contracted or not. Comparing the magnitude of the frequencies in Figure 2 or 4, the magnitude of 60 Hz is almost 15-20 times greater than any other frequencies. Therefore it is reasonable to predict the cause of the noise in the original relaxed signal is the 60 Hz. Looking at the RMS values of the relaxed data, the unfiltered signal’s value was nearly 7 times higher and is greater than the filtered contracted signal. Without filtering out the 60 Hz the RMS value will always exceed the contracted threshold of 30e-05 and corrupts the ability to detect contraction.

PART B:

B1:
Quote used:
“What's the difference between an oral thermometer and a rectal thermometer? . . . The taste mostly”

B3:
Overlapping chunks (50% overlap) enable smoother transitions, but may introduce artifacts at segment boundaries due to shared information. Non-overlapping chunks offer simplified processing but may cause gaps between segments, affecting the natural flow of speech. Consecutive chunks maintain continuity but could lead to blending artifacts. Introducing gaps aims to reduce artifacts, but it may affect naturalness due to pauses.

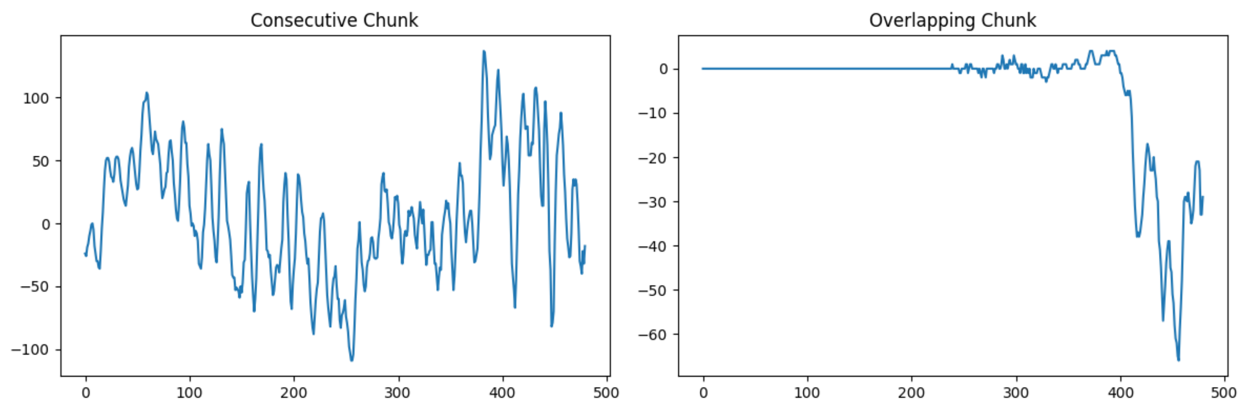

Figure 5: Consecutive vs Overlapping Chunks (Consecutive Chunks: 288 and Overlapping Chunks: 575)

BPFs analyze segmented data, creating parallel bands for automated analysis. Loop-based implementation resolves open questions on center frequency, filter type, order, bandwidth, and N filters. The chosen 50 Hz bandwidth ensures efficiency without overloading. Overlapping or non-overlapping bands can be selected based on specific needs. BPF outputs identify dominant frequencies for the robot-like voice synthesis.

In the synthesis phase, we calculate the RMS of each band (Ai) obtained from frequency-domain segmentation using band-pass filters. Using Ai as amplitude, we synthesize sine waves for each center frequency (fci) in a loop. By superimposing these waves from all bands, we create a single time stream representing the robotic voice. The process is repeated for each chunk, ultimately reconstructing the entire audio output.

B.4
Upon completing the robot-like voice synthesis process, the output stream was carefully evaluated for intelligibility. The sentence was successfully recognized, affirming the effectiveness of the segmentation and analysis techniques.

Figure 6: Input Audio Time Stream

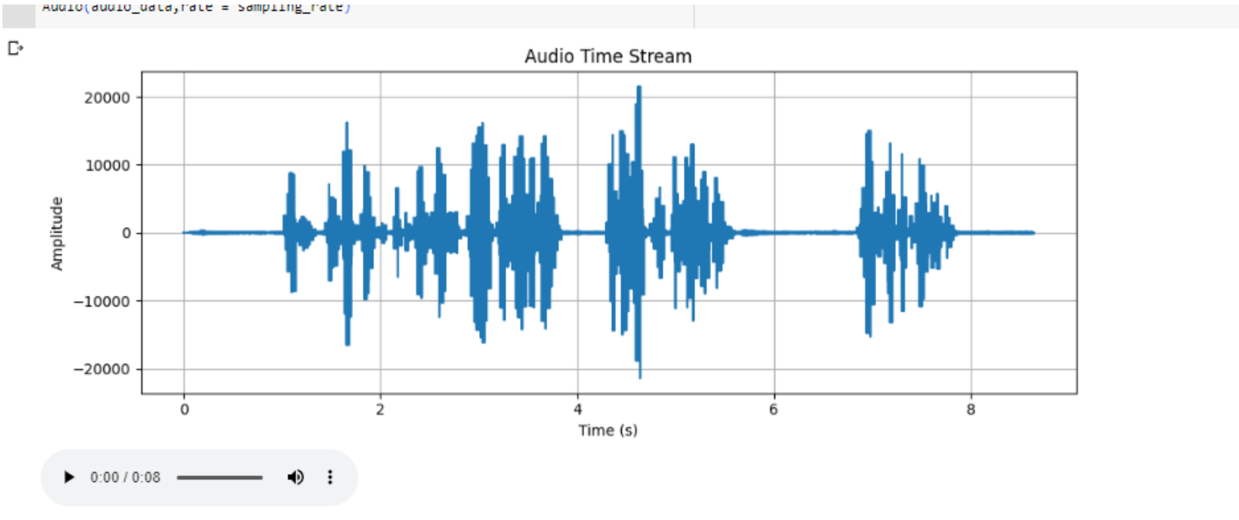

Figure 7: Output Audio Time Stream

Contributions:
Daniel worked on all of Part A. Gabriel and Andrew worked on Part B together. Gabriel worked on time segmentation and wrote the discussion parts for Part B. Andrew worked on Frequency Domain Analysis and Synthesis.