ECE4220

EMG Signal Extraction

Final Project

Author
Emily Mixon

Instructor
Guilherme Desouza

December 2010

Abstract

The desired outcome of this project is to extract the window of an EMG signal. Often times it is necessary to analyze recorded electrical activity produced by muscles to study how muscles react in order to replicate the movement with robotic assisted technology. It can be distracting and misleading if there is excess noise at the beginning and the end of the recorded signal. The goal of this project is to discard the data that is most likely noise and extract only the portion of data where the active region is located. In order to find where the active region is located, it is shown that a wavelet transform must be performed on the given data set in order to calculate a maximum noise threshold that can be used to detect the active signal and extract its position. Finally, the graphed results are given in order to confirm the success of the algorithm.

Table of Contents

Table of Figures

1 Introduction

With the development of improved robotic assisted technology, it is often necessary to analyze actual muscle activity. In order to examine the signal data more accurately, the window of muscle activity for a given signal must be extracted to eliminate noise at the signal's peripherals. The engineers working to develop this technology will benefit from more readable data which they can use to develop technology that will improve the standard of living for those who are limited by their own functionality. This project implements an algorithm that looks at the signal where the active region is and extracts the window that the corresponding positions. It starts off by using an A/D converter to extract a set number of data points and saves these to a file. Theoretically, the signal would then filter the signal, but that is beyond the scope of this project. If we assume that the given data is already filtered, the window of detection can be extracted by calculating a threshold of the maximum noise and saving all the positions where the signal value exceeds the threshold. The algorithm smoothes out the detection interval and then finds the center of the detection window. Once the center has been found, an interval can be created comprising of a predetermined number of values centered in the middle found in the previous step. Currently, this algorithm has only been implemented in Matlab, but for this project its functionality has been implemented in C. Also, as previously mentioned, it is assumed that the signal data has already been filtered before it is extracted by the detection algorithm. Other boundaries within the scope of this project include saving the final output data to a file instead of outputting it back through a D/A converter, and that this project is implemented using the TS-7250 board with the MAX197 option enabled rather than using a separate microprocessor with additional floating point software or hardware. It is concluded that the A/D conversion algorithm and the EMG detection algorithm both successfully work within their respective blocks in the system. The result is an extracted signal with the only valid values, the ones that were within the detection interval.

2 Problem Statement

The main problem this project solves is the need to extract a detection window of active signal data. This problem cannot be solved however, until other outstanding issues need to be addressed. Before a signal window can be extracted, the A/D converter needs to acquire signal data. This has to be done on a board where the A/D option is installed in configured. It takes approximately 12 μs in order for the converter to read in a digital value to the register. This case can utilize a real time task where the period of the real time task is made to be slightly larger than the approximate conversion time. Once all the recorded data is obtained, then it can be processed to find the extraction window. Implementing this algorithm in C required the use of a wavelet friendly library, which was found in the GNU scientific library, in order to extract the coefficients of a wavelet transform. A wavelet transform was implemented rather than a Fourier transform because the time resolution needed to be saved in addition to the frequency resolution, which is necessary since the desired output is a window in the time-scale. The original Matlab algorithm used a continuous wavelet transform in order to extract the data, but this implementation in C was not supported by the library. A discrete wavelet transform, however, was supported by the utilized library. Research needed to be done to figure out if a discrete wavelet transform could replace a continuous wavelet transform. In a continuous wavelet transform, the number of transform levels is given and the number of coefficients that is returned by the transform function is a 2-D array with a length equal to the number of original data points and height the size of the number of levels given. This would have needed to be boiled down into a 1-D array the size of the number of data points containing the maximum coefficient value out of each level in the array. A discrete wavelet transform eliminates some of the redundancy of a continuous transform and returns an array the length of the number of data points. The number of data points however, must be equal to a power of 2, so that it fits the equation that the number of resolution levels is equal to the logarithm base 2 of the number of data points in the set. The returned array was organized by level so that the index of the level ranged from 0 to one less than the maximum number of levels found in the previously stated equation, and the number of coefficients within each level ranged from 0 to one less than the number two raised to the current level. While the discrete wavelet transform provided less resolution than the continuous wavelet transform, for the application of this project, the discrete wavelet transform provided usable coefficients in a more efficient and time saving manner. Once the wavelet coefficients were obtained, a threshold could be calculated that allowed the user to detect and optimize a detection window to display the EMG signal.

3 Proposed Approach

As it can be read from the block diagram, the first step of this project is to acquire the data from the EMG signal. This piece of the system utilizes the MAX197 A/D converter on the TS-7250 board to extract a set of data. The software for this piece maps the necessary registers on the A/D converter to memory so that they can be written to and read from. Next a real time task is initialized to handle reading the values after waiting a period to allow the conversion take place. The software checks a bit on one of the initialized registers to make sure that the MAX197 chip is installed. Next it enters a loop that initializes the register to start the conversion, waits a period then reads a word-sized value off the corresponding registers and saves that value to a buffer. Once the buffer has been filled with the specified size of data the buffer is copied to a file so it can be utilized in the detection phase of the system. The file containing the data set is copied into an integer array. As stated in the previous section, a discrete wavelet transform was performed on the data set to extract a set of coefficients the size of the original data set. The algorithm finds the maximum coefficient from a known time frame of noise points at the beginning of the signal. The algorithm then multiplies this value by a given constant greater than one to calculate a threshold value. A detection vector can then be formed by saving a one in every position where the coefficient value exceeds the threshold value. After the detection vector is formed the algorithm goes back to extract the interval over which the majority of the ones occur. This is done by taking the density of ones immediately to the left and right of each one and comparing the sum of the two densities to a calculated minimum value. This gives us a main active interval, but small gaps are possibly left over that should be included, so the algorithm goes back through to find potential gaps and smoothes out the signal to make it one long continuous signal. Next the algorithm seeks to find the beginning and the end of the main parts of the signal. Sometimes there are two main parts of the signal, so the algorithm has to account for two beginning positions and two end positions. If there are two valid main parts, the beginning of the first part and the end of the second part is considered to be the interval. If the first main part is equal to zero, then the beginning and the end of the final interval are both set to zero. Otherwise, if there is just one main part, the beginning and end are equal to the first interval's beginning and end. After these two points are found, a center can be calculated. The final interval consists of zeros to the point where the interval starts, then ones for a given number of desired output points, and then zeros again till the length of the data set is reached. Finally, the output signal is found by multiplying the final detection interval values with the original signal values at the corresponding points. This output signal is saved to a file, so it can be analyzed. A block diagram showing the interaction between the separate pieces of the system is included for reference.

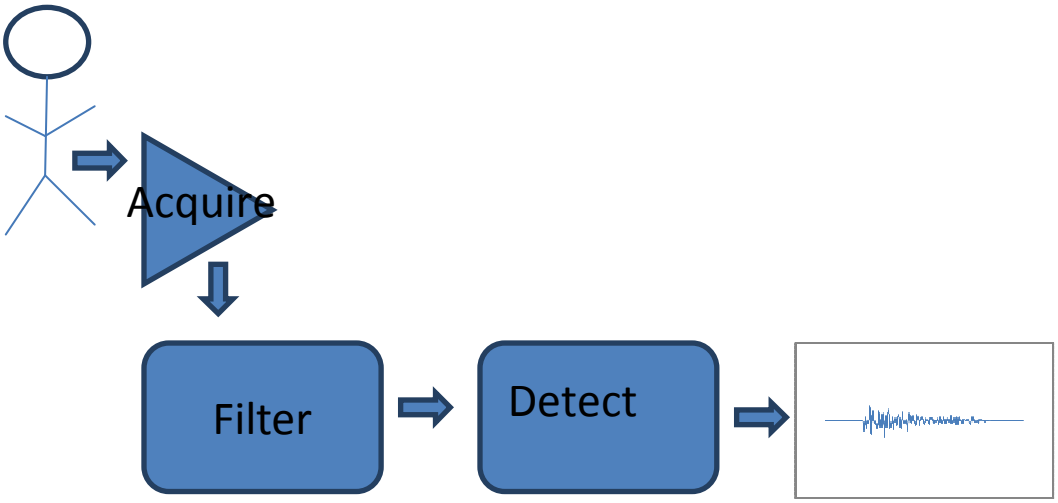

Figure 1: Project Block Diagram

4 Results

As it can be seen in the figures below, the both the A/D conversion algorithm and the detection algorithm were successful in extracting the required data. In order to obtain the data, the TS-7250 was connected to a power supply and the voltage was varied in order to test the voltage over a desired range. On the A/D conversion algorithm it can be seen that a specified number of discrete floating points were extracted. These values were saved to a file which was used to form the graph below.

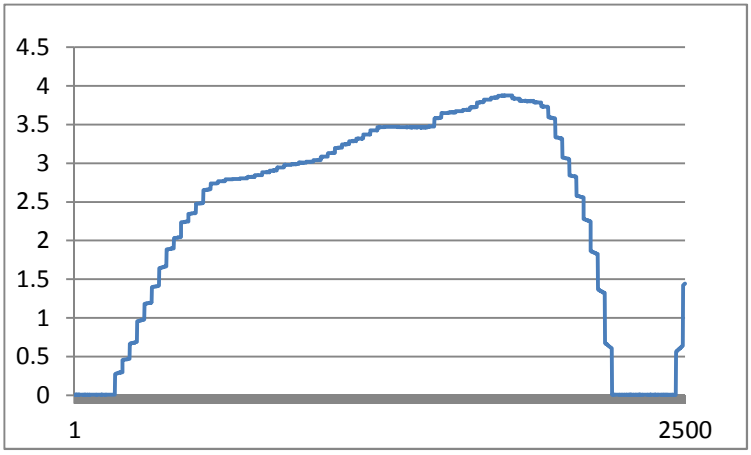

Figure 3: Acquired Data

In order to obtain the detection window for the EMG extraction algorithm, the signal from Figure 3 was read in and saved to a buffer.

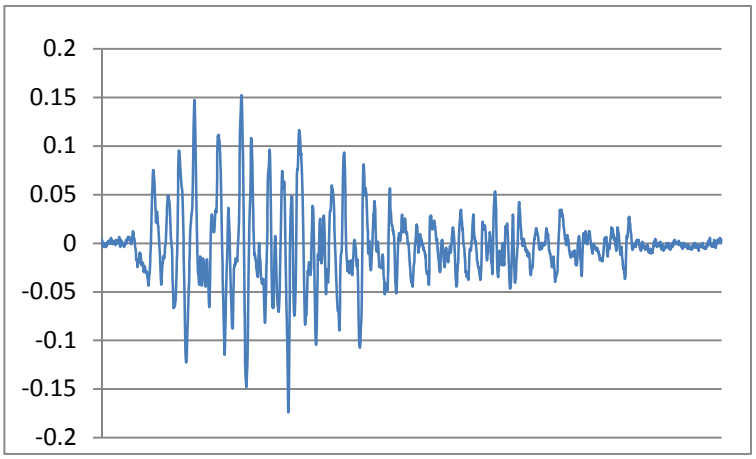

Figure 2: Recorded Signal

Finally, Figure 4 shows the detection window containing the portion of the signal with active muscle activity. It can be seen that the values of the detected signal on the window's peripherals is zero.

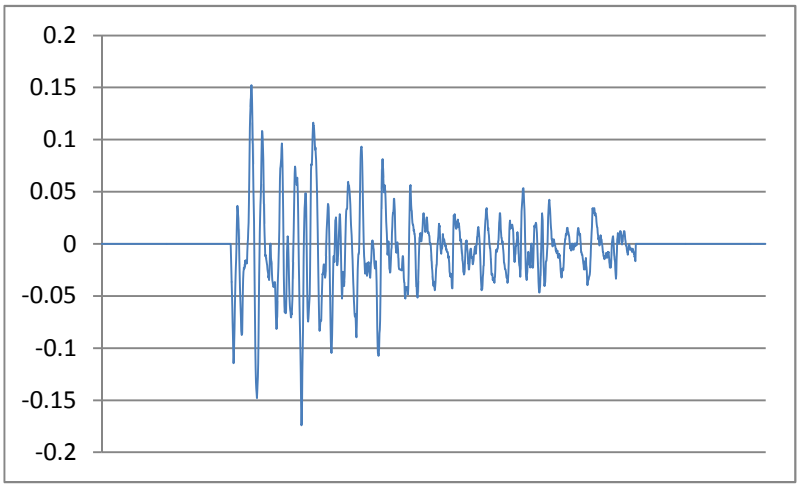

Figure 4: Detected Data

5 Conclusions and Future Work

From the Results section, it can be concluded that the algorithms work to specification and can be implemented accordingly. If more resources were available to allow the continuation of this project, the filtration block of the system should be designed and implemented. This would allow the same set of data to be recorded, filtered, and detected all in one sitting versus implementing the individual components separately with different data sets. Once the whole system is working as desired, the whole idea of the project could then be implemented onto an independent embedded device rather than relying on a computer and the TS-7250 board. This implementation could potentially require either extra hardware or software floating point support depending on the specifications of the utilized microcontroller. On a smaller scale, the given implementation could be improved with the LAPACK library to improve the transition of the extraction algorithm from Matlab to C. Overall, the system performed as desired for the scope of this project, and more research should be done in order to increase the scope in improve the overall design of the chosen implementation.

Bibliography

[1] C. Valens, "A Really Friendly Guide to Wavelets," Feb. 2010. [Online]. Available: http://polyvalens.pagesperso-orange.fr/clemens/wavelets/wavelets.html. [Accessed Nov. 28, 2010].

[2] L.A. Rivera, "Algorithm for sEMG Signal Detection/Extraction," pp. 1–10.