CHAPTER 5: DETAILED DESIGN

5.1. INTRODUCTION

This chapter will focus on the detail design of the sEMG sensor. A concept of the developed system is shown in figure 56. The system comprises of a sEMG sensor wrist band, and sEMG platform PCB. The wrist band contains the electrodes used to sense muscle activity, and passes this signal unto the sEMG platform. The platform communicates this information to the Matlab®/Simulink® environment where the raw sEMG signals are decoded into useful information. The Matlab®/Simulink® output is communicated back to the sEMG platform to control a PDM servo.

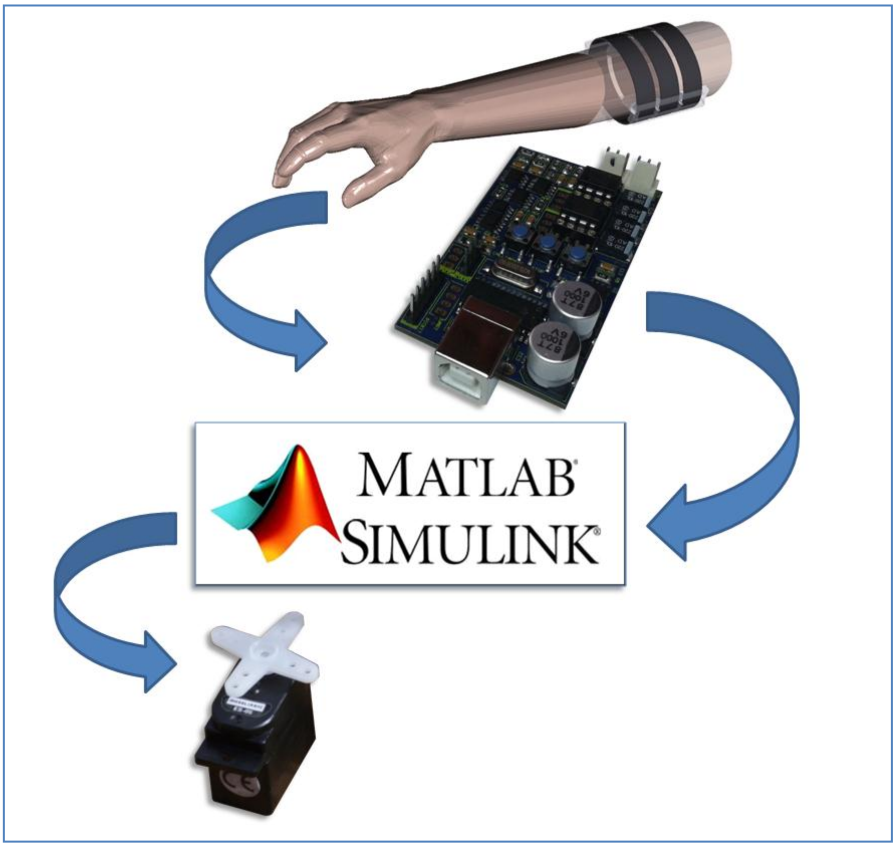

Figure 56: Summary of system components

The PDM servo is connected to the sensing platform. The PDM servo could be operated directly from the Matlab®/Simulink® environment, as the platform supports bi-directional communication. The PDM servo will be used in the testing/experimental setup for visual output of the system, since there is not an existing prosthetic hand to control yet.

The detailed design is the process of converting the conceptual design into a hardware device [18]. The "design for" – criteria is given in chapter 4 to add specifications to a functional design [18]. The requirements given in chapter 4 are used as specifications. The specifications are derived from the issues addressed in the problem statement, found in the introduction chapter. For the final circuit design and schematic, please refer to the schematics under the hardware design section in Appendix B.

5.2. DESIGN FOR CRITERIA

5.2.1. FUNCTIONAL CAPABILITY

Experimental results show that the voltage of a sEMG sensor ranges between 500 µV to 5 mV for upper limb concentric muscle contractions. The SNR of the sEMG sensor ranges from 5:1 to 50:1. These values are required for the amplifier design. The addition of the USB port and a serial emulator firmware programmed microcontroller creates an interface between the sEMG sensors and Matlab®/Simulink®.

5.2.2. DESIGN FOR RELIABILITY

Common mode shields are often used in sEMG systems if the electrodes and amplifier are some distance apart. This is omitted in this design as the design is intended to be used as an active sensor and the wire connections between the amplifier and electrodes are short. Proper power and ground reference planes are included in the PCB layout. EMI is a major issue in amplifier designs and the addition of EMI suppressor components to the circuit improves the EMC of the design. From experimental results, the most of a sEMG signal is found in the frequencies between 100 Hz and 450 Hz. The addition of a band-pass filter with $f_{cut\text{-}off\ low}$ = 100 Hz, and $f_{cut\text{-}off\ high}$ = 500 Hz would serve as an anti-aliasing filter for the (ADC $f_{sampling}$ = 1 ksps). Power-line noise (50 Hz) may also contaminate the EMG signal and DC-offsets in data will also be filtered to some extent by the band-pass filter.

5.2.3. DESIGN FOR USABILITY

A Pickit® programming port allows a user to change firmware on the PIC18F2550 microcontroller, and is added to the circuit. The addition of a UART allows future functional expansion. LEDs and pushbuttons provide status to the user and allow the user to select functions. SIL connectors allow the user to use sEMG amplifier on other platforms by bypassing certain modules on the platform. For functional expansion, a second PDM servo output and a UART header is added to the final design.

5.2.4. DESIGN FOR AFFORDABILITY

The cost of a PCB increases as its complexity (in terms of number of layers) increases. The design is too large and complicated for a single-sided, single layer PCB. The layout is done on a double-sided, single layer PCB, allowing enough area for proper power and ground planes.

5.2.5. DESIGN FOR TESTABILITY

Test pins (SIL connectors) used for functional expansion are available to test each of the modules on the PCB. The test points are clearly marked on the sEMG platform with silk-screen labels to ensure that the correct pins are used for testing purposes.

5.3. HARDWARE PRELIMINARY DESIGN

5.3.1. DETAILED FUNCTIONAL BLOCK DIAGRAM

The functional block diagram from the functional analysis in the conceptual design phase is revised to include the specification added by the "design for" criteria. The experimental sEMG recordings shows that a total amplifier maximum gain of approximately 10 000 is required to use the full range of the ADC (0-5V). Figure 57 shows the revised functional block diagram. The sEMG signal amplification is performed in two stages, with a digital potentiometer used as a passive automatic gain control (AGC) between the two stages. The reason for this configuration is to prevent signal clipping from occurring. The passive AGC can only attenuate signals, and must attenuate the signal to a lower level before it enters the final amplification stage. Should the AGC be placed after the second stage amplifier, and the second stage amplifier amplifies the signal beyond the 0 -5 V range, signal clipping would occur and the AGC's attenuation would not be able to recover the original signal.

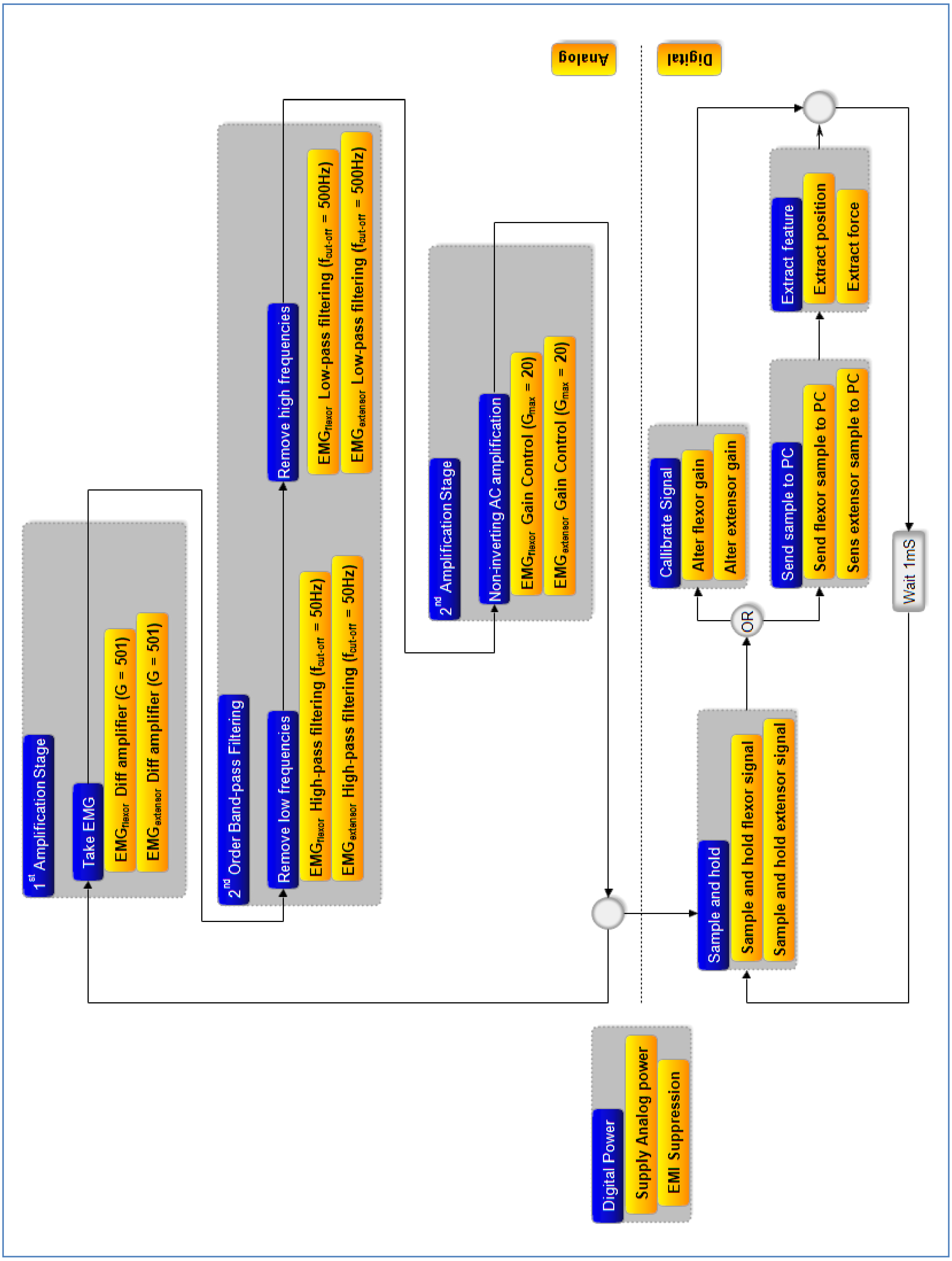

Figure 57: Revised functional block diagram for sEMG sensor

5.3.2. SENSING ELECTRODE PROTOTYPING

A temporary method is required to keep the sEMG electrodes in contact with the skin, across any set of antagonist muscles. The reference electrode also requires a proper contact to a bony joint. The design is intended for the flexor and extensor muscle groups of the hand, but it is not limited to those muscles.

The biceps-triceps muscle groups have also been tested, and the strap design fits to both the upper arm and lower arm of the patient. Recall that any bony joint in the human body is a good ground reference measuring point as there is minimum muscle and nerve activity in those areas [26]. The bony region below the elbow is chosen as a reference point.

Figure 58 shows a Solidworks® model of the wristband. The three black bands are elastic bands tipped with Velcro® to cater for multiple arm diameters. The electrodes are supported by clear Perspex®, and their position can be adjusted to ensure the correct position of the electrodes.

The location of the wristband on a patient's arm should be across the thickest part of the forearm. Note that in the case of a wrist amputation, the forearm's shape and functionality remains comparable to that of a normal person. This allows the sEMG platform to be tested on normal patients and amputees.

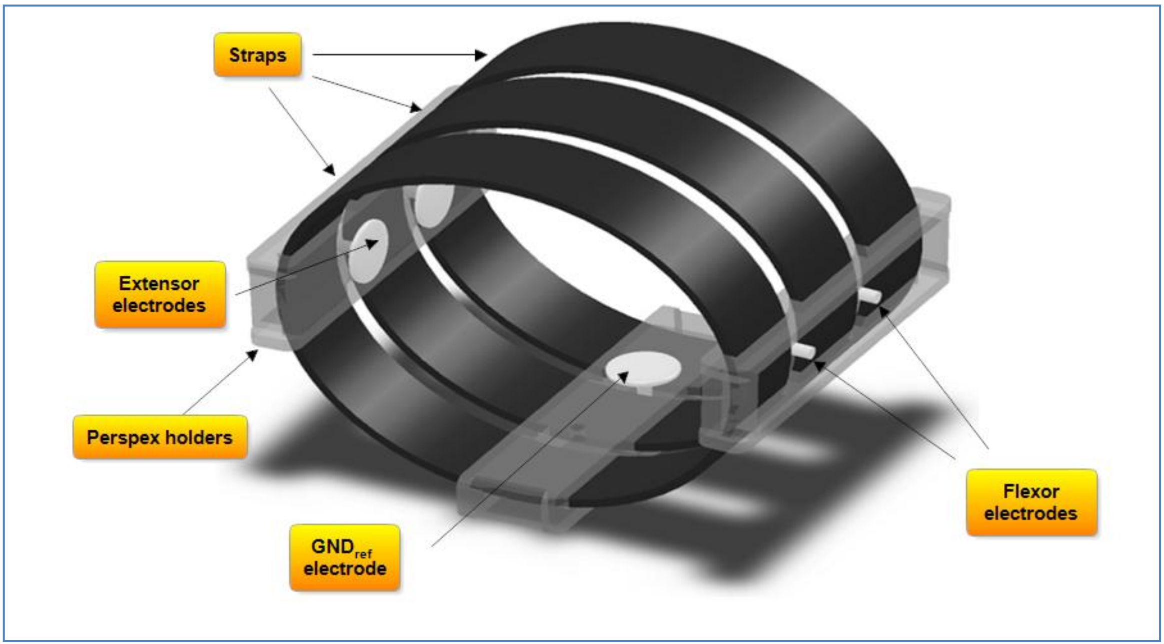

Figure 58: 3D model of the wristband design

The circuit diagram for the 5 electrodes is illustrated in figure 59. The electrodes names are the same as in the muscle model derived in the conceptual design phase. The GND_{ref} electrode's position is important, and fits close to the patient's elbow.

5.3.3. PCB PROTOTYPING

In order to the keep development cost to a minimum, it was decided to develop two hardware versions. The first version is a functional prototype for conceptual testing of the platform. The second version is a final layout, taking care when placing components and routing the connections.

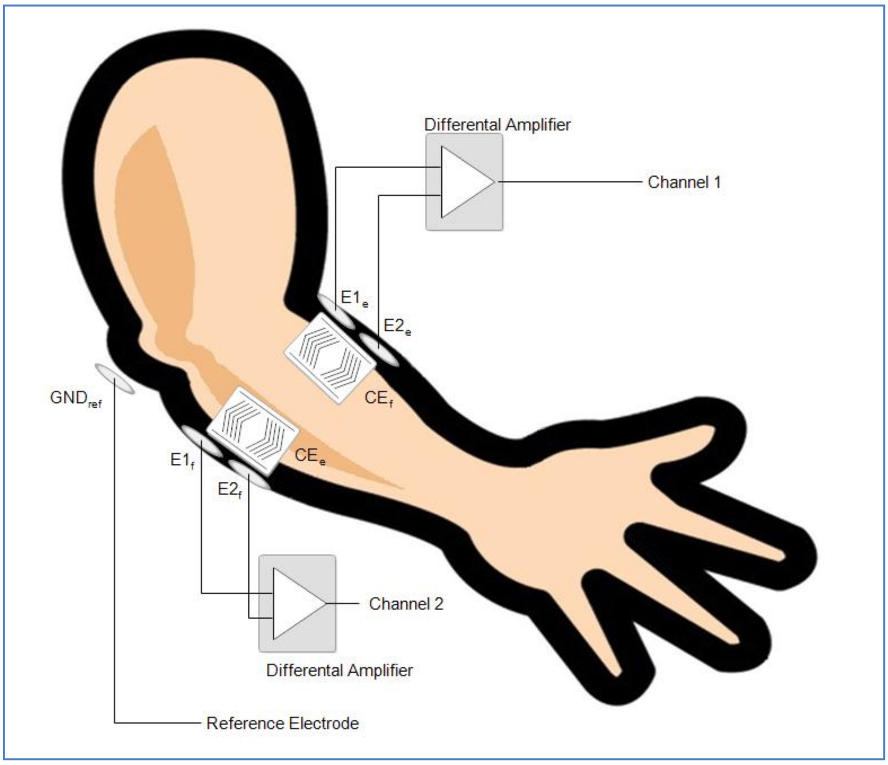

Figure 59: Electrode circuit diagram

The sEMG amplifier, band-pass filter, and AGCs and USB microcontroller are built on prototyping board to verify the concepts. The shortfalls of the prototype platform, with the improvements made on the final PCB layout, are discussed in the results section. A photograph of the double-layer prototype board and the final PCB layout can be viewed in figure 71, showing noticeable improvements in performance, aesthetics and physical size.

5.3.4. FIRMWARE FUNCTIONAL LAYOUT

Firmware refers to the code programmed onto the sEMG platform, used to control the hardware. The basic functionality of the firmware is listed below. The algorithm in figure 60 illustrates the functionality of the firmware.

The algorithm loops every 1ms to ensure that the sampling frequency of the sEMG ADC process occurs at 1 ksps. The USB stack needs to be serviced regularly. The push buttons on the sensor needs to be serviced regularly, although it is not critical to perform this operation at a constant interval. The PDM servo only needs an updated pulse (to calculate the desired position of the servo horn) every 20 ms. If this updated pulse is sent too soon, some PDM servo doesn’t function correctly.

The algorithm loop starts with the USB tasks, to check for information Matlab®/Simulink®. Then the three buttons are sensed to check whether the user pressed a button. The “calibration button” would initiate a branch into the calibration algorithm (this is a separate algorithm explained in the next section). The “start button” and “stop button” would start and stop the sending of information to Matlab®/Simulink®. The Windows® operating system assumes a COM port (emulated by the platform) is occupied already when data is transferred before the connection is established. The start and stop of information transfer is required as and Matlab®/Simulink® cannot establish a connection.

The servo is then updated with new information from the Matlab®/Simulink® information received only if 20 ms has elapsed. This is achieved by counting to 20 and only execute this instruction every 20th loop through the algorithm.

The sEMG ADC process is done, and the dual channel information is sent through the USB to Matlab®/Simulink®. The algorithm returns to the top again with the USB tasks process.

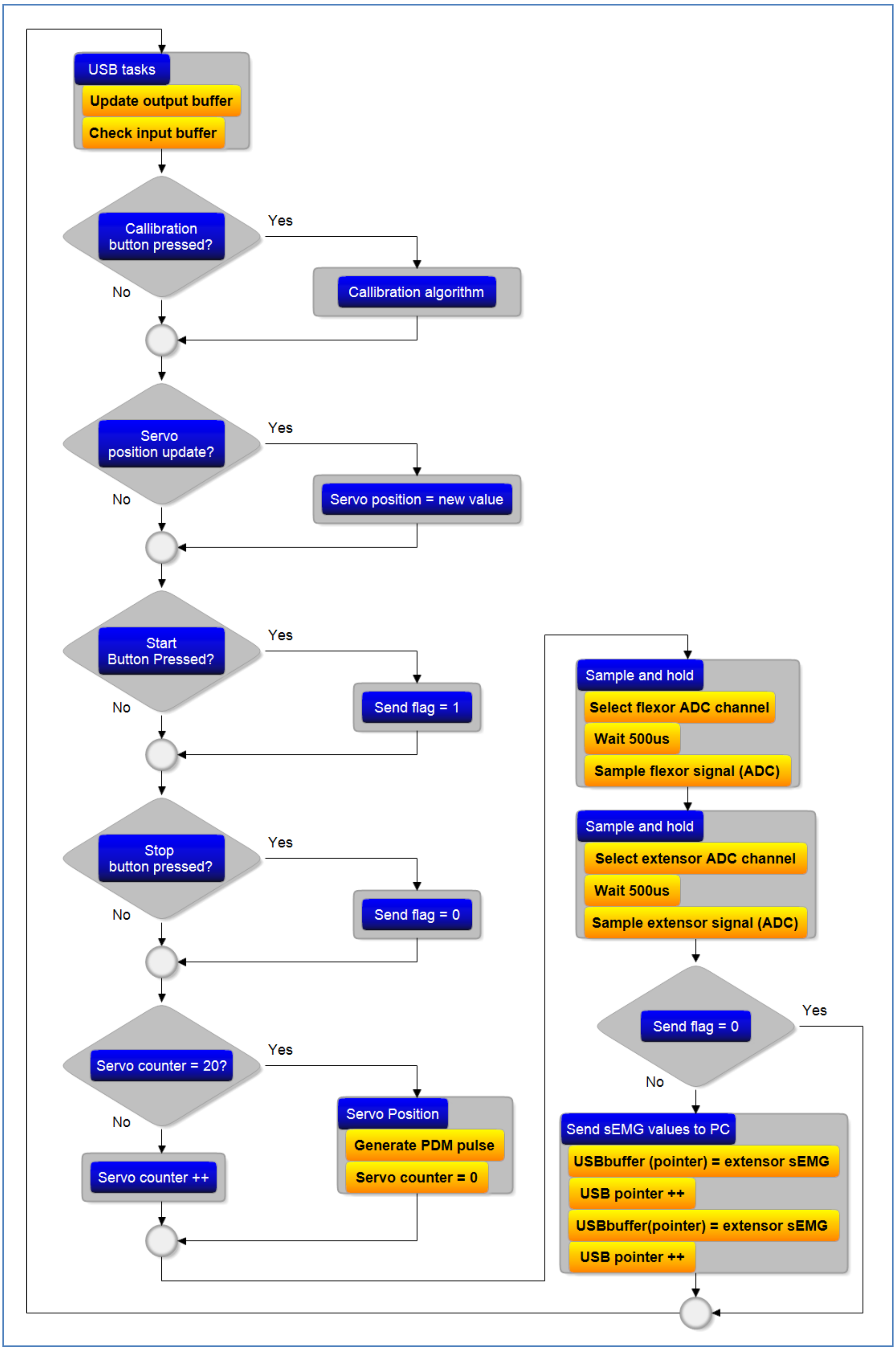

Figure 60: sEMG platform firmware algorithm

USB

The Microchip® USB-stack is used to enable the USB port on the PIC18F2550. The USB-stack does all the necessary USB housekeeping in terms of send/receive buffers, buffer pointers etc. The USB product descriptor is "HRL EMG SENSOR" and the description of its functionality in its .inf file is a "Communications Port". The sEMG platform's USB is basically a serial emulator and the Windows® operating system therefore recognises the sEMG sensor platform as a serial port. This configuration enables the user to connect the sEMG sensor to Matlab®/Simulink® by selecting a COM port functional block from the Matlab®/Simulink® library browser.

The firmware is not interrupt driven, as the firmware loop is guaranteed to take less than 4 ms to execute in the worst case. The USB-stack allows up to a few milliseconds before the USB-tasks will be serviced again (experimental results show a lag of 10 ms is acceptable).

USER I/O

The following input/output (I/O) devices are included in the platform design with a description of its functionality

Table 2: User inputs and outputs

Name of I/O	Functionality
Send LED	Indicates USB communication status. On indicate connected, flashing indicate connection error
Calibrate LED	Indicates when the firmware is currently running calibration algorithm
Start button	Used to give the instruction to the platform to start sending the sEMG data over the USB serial emulator. This was added to the platform as it was found that it is impossible for an application to establish a serial connection to the platform while the platform sends data through the USB
Stop button	Used to give the instruction to the platform to stop sending the sEMG data over the USB serial emulator
Calibrate button	Used to give the instruction to the platform to toggle between the main firmware program and the calibration algorithm

ADC

The ADC has a minimum acquisition time of a few us for which the input must be sampled before the input settles and an accurate conversion can be done. The accuracy is ensured by, instead of performing two conversions per 1 ms (AN0 and AN1 multiplexed for antagonist muscle sampling), alternating conversions are done every 500 µs. This allows the maximum time for the ADC's analog front end to settle. 10-bit conversions are done, and sent as two sets of two bytes in little-endian format through the serial port.

PDM SERVO INTERFACE

A PDM servo's position is communicated as a pulse with duration, ranging between 1 ms and 2 ms, and expect an instruction pulse every 20 ms. Matlab®/Simulink® determines the desired position for the PDM servo, and send the information to the platform. The platform's firmware generates these instruction pulses from the received information.

GAIN CONTROL INTERFACE

Digital potentiometers are used to change the amplifiers' gain. They are controlled by the microcontroller using an I2C interface to each of the two digital potentiometers.

CALIBRATION ALGORITHM

The calibration algorithm is initiated by pressing the calibration button on the platform. The heart of the calibration algorithm is to compare the sEMG signals differentially. This requires the normalisation of signal amplitudes. Should the antagonist muscle sEMG recordings' noise levels be different, a DC-offset would result from the difference in noise level. By adjusting the noise output of the 2 sEMG amplifiers to the same level, using the digital gain control, the differential comparison would have an output of zero.

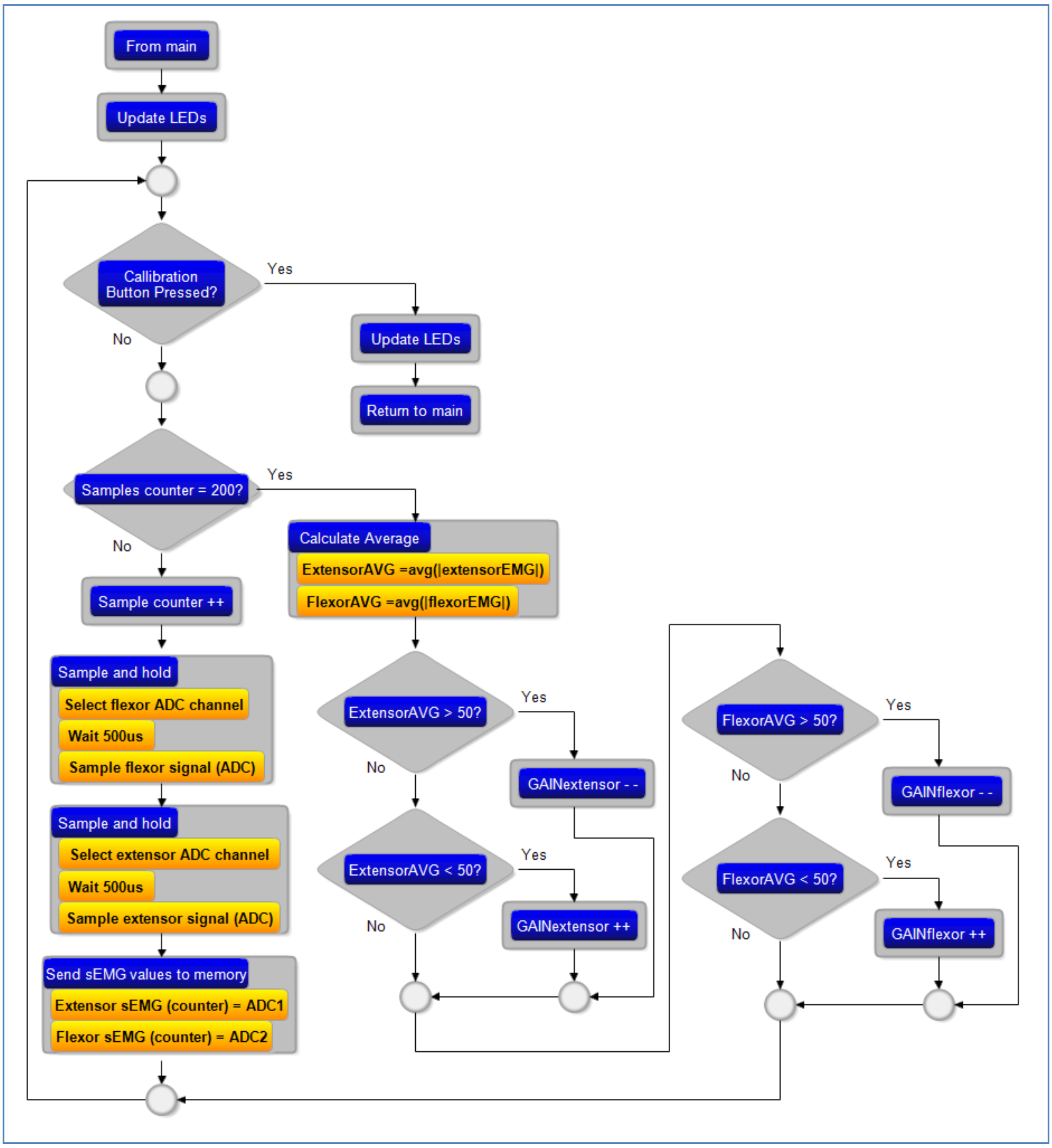

Figure 61: Proportional control calibration algorithm

Experimental results show that the SNR of most sEMG recordings are around 10:1 (the SNR value ranges from 5:1 in the worst case to 50:1 in the best case). The ADC on the sensing platform has a range of 0 V to 5 V, represented by a 10-bit value ranging from 0-1023, but the analog circuit on the platform adds 2.5V DC-offset to the amplified sEMG to prevent signal clipping of the AC sEMG signal. This means that the ADC value should be seen as a value ranging between -512 to +511, by subtracting the value of 512 from the raw data in Matlab®/Simulink®.

The average SNR of 10:1 suggests that a sEMG signal sampled by the ADC with an average noise level of 10% (average ADC value of ±50) of the ADC total range (maximum value of ±500) would suggest that the complete range of the ADC is used, and the noise amplitude is known. The digital gain of each channel is altered until the average noise level of each channel are equal to one another, and equal to ±10% (average ADC value of 50) of the range of the ADC.

5.3.5. SOFTWARE (MATLAB®/SIMULINK®)

Simulink® is a modeling and simulation tool in the Matlab® environment. It contains a library browser with block diagram components. The DSP toolbox is used in this application.. The following functions described are implemented in Simulink®, with a screenshot of the file in figure 64. The file MATLAB_USB_EMG_Sensor6.mdl can be found on the data disk.

COM PORT

The COM port functional block in figure 64 is the interface with the sEMG platform. Settings for the serial port are 115200 baud rate with the byte order as little-endian. The data size of the ADC is 10-bit or 2 bytes. The bit settings are 8 data bits and 1 stop bit.

REMOVE DC-OFFSET

The analog circuit on the platform adds a DC-offset of 2.5V or the value of 512 to each of the samples. This means that an ADC value of 512 represents zero, ADC values below 512 are negative and ADC values above 512 are positive values. This DC-offset is subtracted as soon as the data type of both sEMG samples are changed to double.

RECTIFY SIGNALS

The absolute value block is used to take the absolute value (rectify) of both sEMG samples. This step is required because sEMG signals are AC signals and the average value of the original signal would result in zero.

MOVING AVERAGE FILTERING OF SIGNALS

From experimental results, a moving average filter of length 200 removes the high frequency noise due to irregular firing of MUAPTs in the sEMG signals, without a significant change in the response time of the output. A moving average block with averaging period of 200 ms is used for this step.

ADDITION OF SIGNALS (FORCE)

An array-vector adding block adds the antagonist averaged samples for the force exerted by the patient whose sEMG is recorded. This step is derived from experimental results that showed that the amplitudes from both sEMG signals increases, the moment more effort is applied to move and hold the joint in a specific position.

SUBTRACTION OF SIGNALS (POSITION)

The differential comparison of the averaged antagonist muscle sEMG samples is done using an array-vector subtracting block. The output of this functional block is an approximation of the proportional position of the patient’s hand gesture.

AMPLITUDE MODULATION

Previous prosthetic sensor algorithms use an amplitude-coded output. This idea is implemented in this project, but the number of levels is increased to 16 levels instead of 2 or 3. More levels could be added, but the number of decision boundaries is sufficient to approximate smooth proportional control. The position decoded from the sEMG samples is compared to level thresholds using threshold switches. The thresholds of the switches and the decision bins are specified in table 2. Signal scaling of the output should be done before it could be compared to a position table or thresholds on table 2 used to control a PDM servo. This is done manually in Matlab®/Simulink®, since the sensitivity level of the controller is personal choice and is different for each patient. In figure 64, the "sliders gain" are used to implement this feature.

POSITION SETTING OF PDM SERVO

The sEMG platform checks for any incoming capital letters ranging from "A" to "P". The PDM servo's pulse varies from 1 ms to 2 ms in 16 increments according to a corresponding letter it receives. The PDM servo has a swing of 180°. The "send" serial block in Matlab®/Simulink® enables data to be returned to the sEMG platform to control the servo's position. Figure 62 is an example of amplitude coding used for a modulation technique. The x-axis represents both the time-dimension and the letter sent through the serial port to the PDM servo. The y-axis shows the normalised position. In Matlab®/Simulink®, the output ranges from ±1, and the servo translates the information code received into an angle between ±90º.

Table 3: Bin names and thresholds

Name of Bin	Range of Bin
A	pos < -0.7
B	-0.7 ≤ pos < -0.6
C	-0.6 ≤ pos < -0.5
D	-0.5 ≤ pos < -0.4
E	-0.4 ≤ pos < -0.3
F	-0.3 ≤ pos < -0.2
G	-0.2 ≤ pos < -0.1
H	-0.1 ≤ pos < 0
I	0 ≤ pos < 0.1
J	0.1 ≤ pos < 0.2
K	0.2 ≤ pos < 0.3
L	0.3 ≤ pos < 0.4
M	0.4 ≤ pos < 0.5
N	0.5 ≤ pos < 0.6
O	0.6 ≤ pos < 0.7
P	pos ≥ 0.7

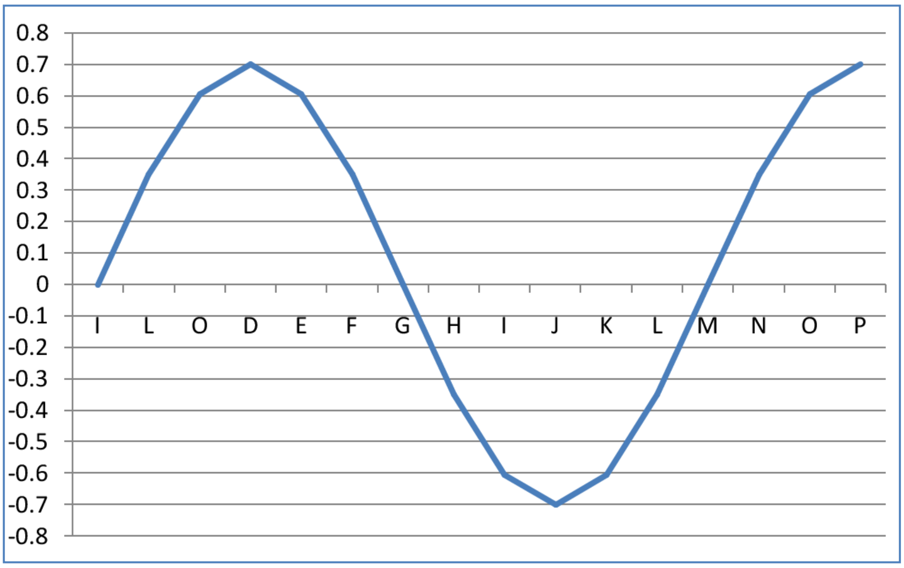

Figure 62: Sine wave sample signal amplitude coded

PLOT DATA

The incoming antagonist muscle sEMG signals, the intermediate step results, and final output are plotted using the plot functional blocks from the Matlab®/Simulink® library browser.

ANIMATION FOR POSITION REFERENCE

The solution to measure precision and accuracy in human hand position movement is by using an animation as reference and to ask the patient to mimic a specified movement. The difference between accuracy and precision is illustrated by figure 71. Accuracy is the measure that indicates the distance the actual value differs from the reference value (or position error). Precision is the measure for repeatability, or how close multiple samples' values corresponds.

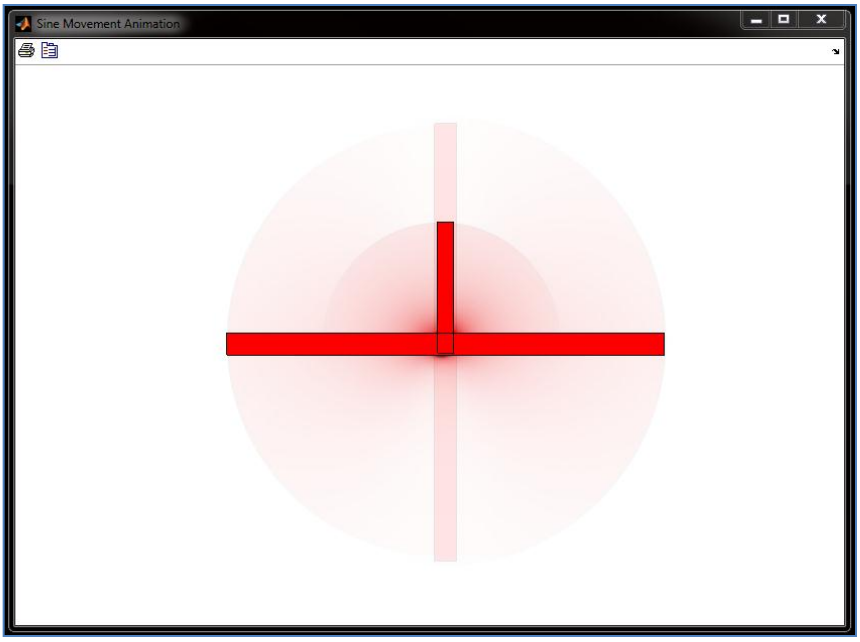

Figure 63: Position reference animation

The animation is realised in Simulink® using the 3Dof animation functional block from the Aerospace block-set in the library browser. A Sine function generator (shown in figure 64) is connected to the animation block to enable a slow oscillating ruler-shaped object used as reference. Figure 63 is a screenshot of the animation. The decoded position values of a patient will be compared to the sine reference wave in the results chapter.

DATA STORAGE

The recording of data is important, and Simulink® has a functional block that enables a user to save arrays and matrix in a .mat file. Table 3 describes how the data saved as a single matrice in the .mat files is structured using concatenate functional blocks also from the library browser. The 100 second aquisition with animation per experiment is captured and saved in the .mat file.

Table 4: Datastructure of .mat file

Name of data (rows)	**Size of array (columns)**
position output	100000x1
force output	100000x1
sine reference	100000x1
flexor muscle sEMG samples	100000x1
extensor muscle sEMG samples	100000x1

Figure 64 shows the complete Simulink® model. All the software functions described above can be viewed in the model. A functional algorithm follows in figure 65.

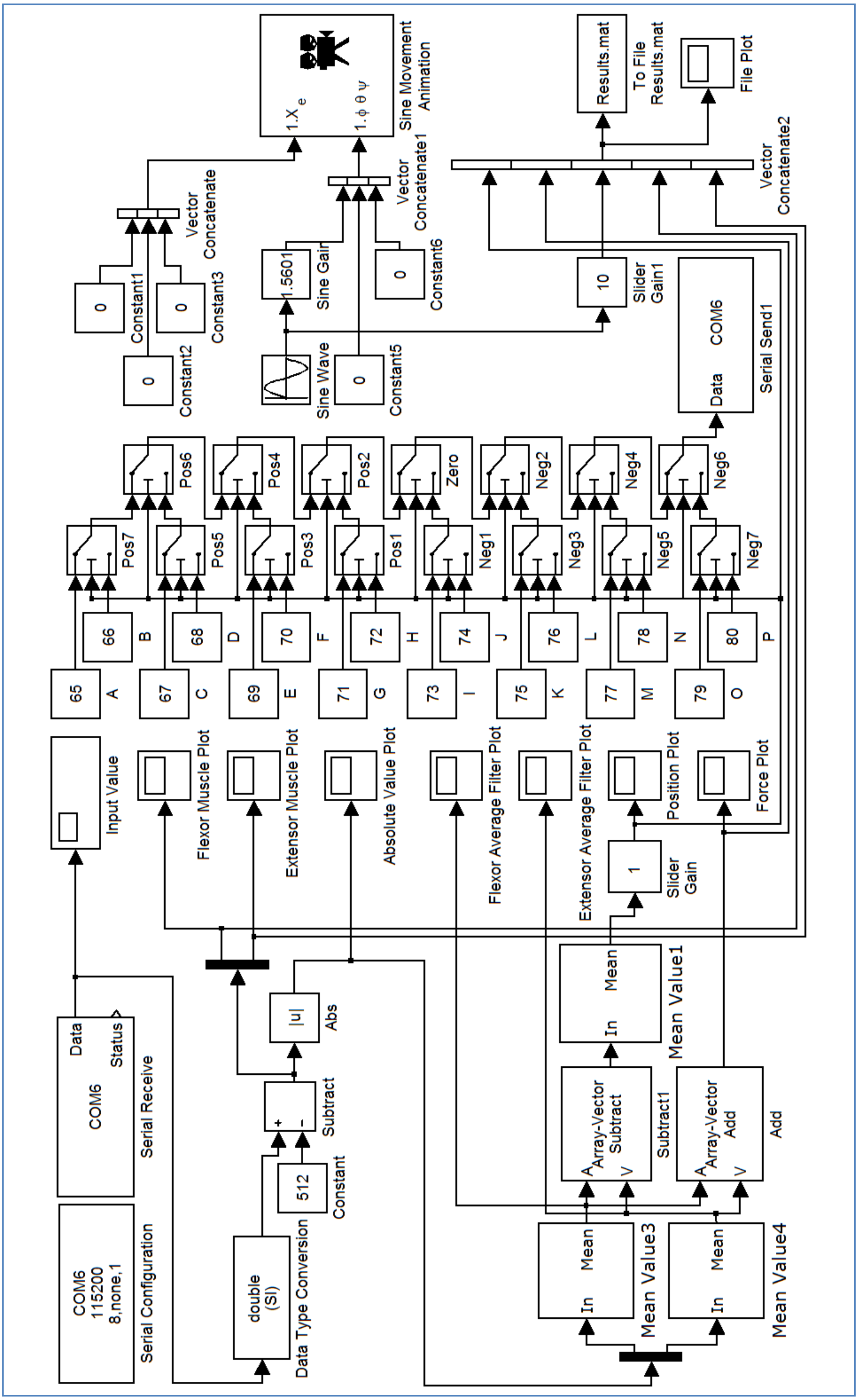

Figure 64: Simulink® model screenshot

5.3.6. ACQUISITION AND DECODING ALGORITHM

All the functional blocks used in Simulink® inherit their sampling period from the COM port. The COM port waits for incoming data samples, expected at 1 ksps (4 bytes representing 2 antagonist muscle sEMG samples in little-endian format).

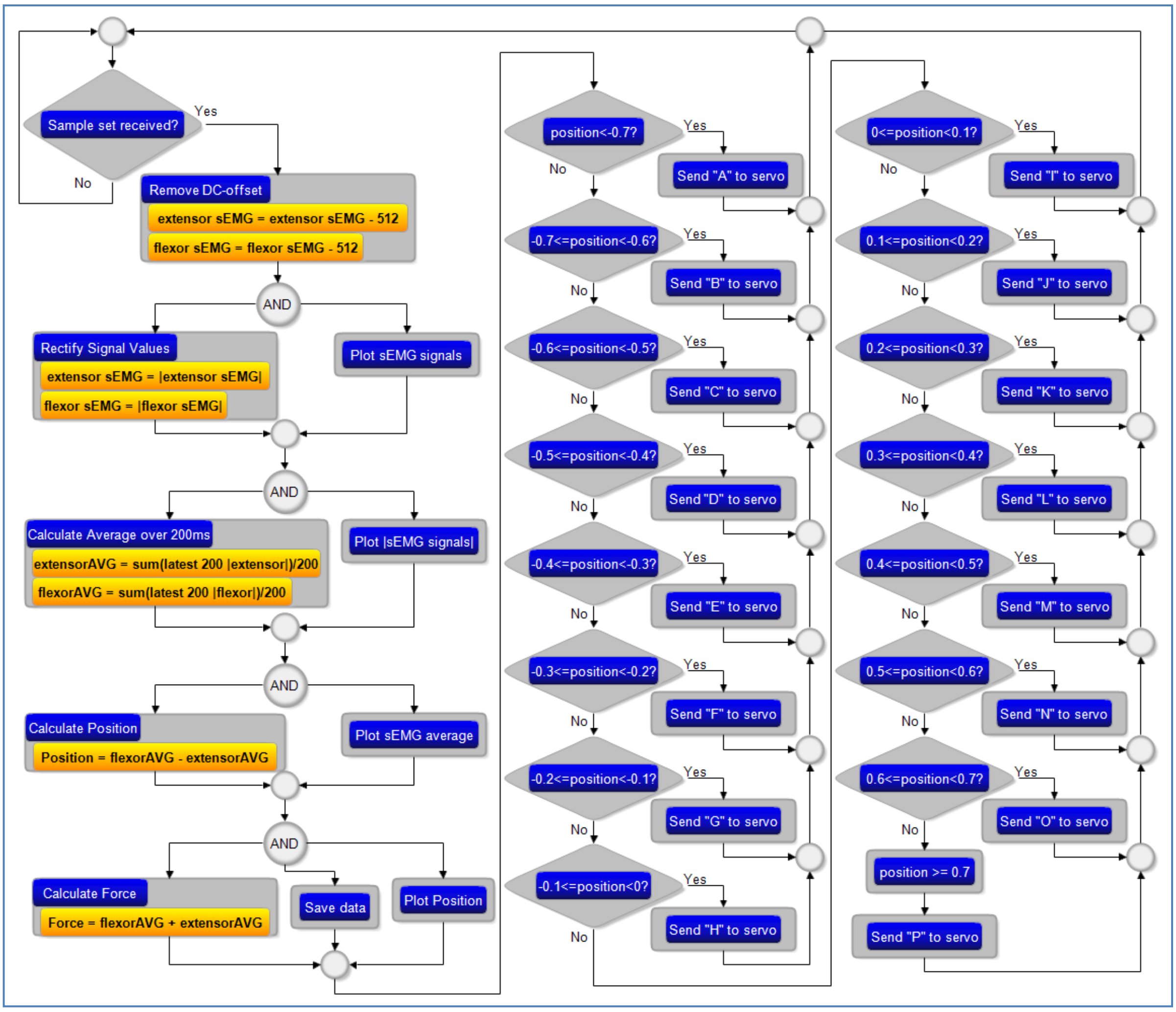

Figure 65: Simulink® algorithm

5.3.7. DATA CAPTURING PROCEDURE

The fist experiment involves patients who have no visual feedback of system output. Patients are asked to follow the open/close gestures reference animation while capturing sEMG signals in the .mat file. They are allowed to view their own hand, and not the actual output of the Simulink® model. A 100 second acquisition with animation is captured per experiment for each patient.

Next, patients are asked to repeat the experiment, but with visual feedback of system output. They are allowed to view the PDM servo's actual position thus receiving the actual output of the Simulink® model. The result section includes a discussion of the power of visual feedback, as the results show a significant improvement. Again, a 100 second acquisition with animation is captured per experiment for each patient.

The animation shown in figure 63 follows an oscillation pattern created by the sine wave generator. Figure 66 is an illustration of the intended gestures the patients has to mimic when viewing the animation during the simulation. Each gesture matches the intended gesture shown by the animation.

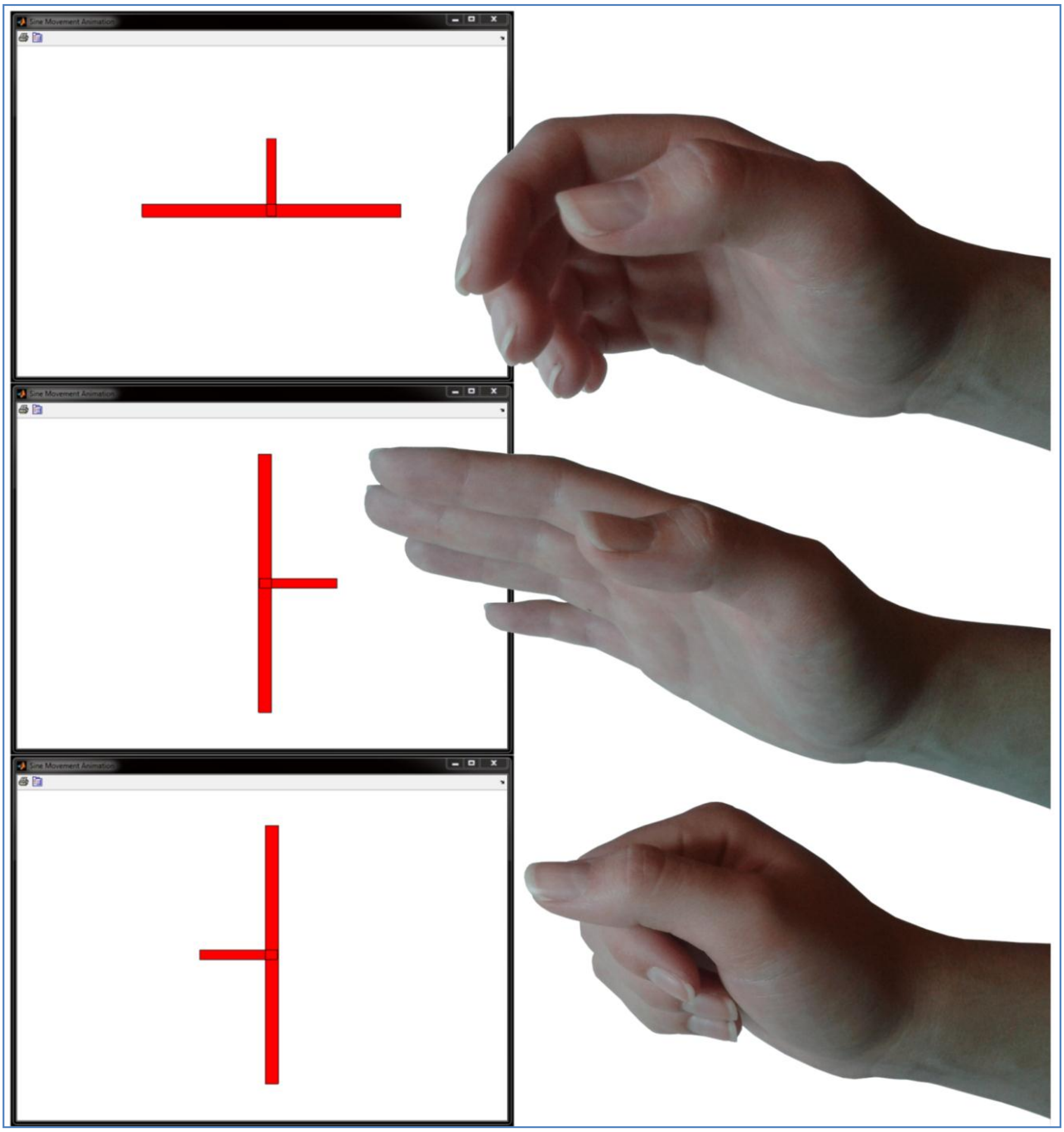

Figure 66: Basic gestures compared to the animation

5.4. HARDWARE DETAILED DESIGN

The PCB layout is only described in brief. For the complete schematic and PCB layout please refer to Appendix B and to Appendix C for project files. The description of the PCB layout discussed in this section refers to figure 67. This section covers the aspects focused on when the final PCB layout is done using Altium®.

5.4.1. ANALOG AND DIGITAL ISOLATION

Digital electronics could contaminate analog signals due to the high speed voltage changes in the digital tracks, and the large amplification stages used in the analog circuitry. The first component placement concept is the splitting of the analog and digital electronics as shown in figure 67 (the dotted blue line). The user interfaces (LEDs and pushbuttons) are regarded as DC circuitry and placed between the analog and digital circuitry. DC-sources generate insignificant noise, and create a separation between analog and digital circuitry. The gain control has both analog (potentiometer) and digital interfaces (I2C communication), therefore it is placed between the analog and digital circuitry.

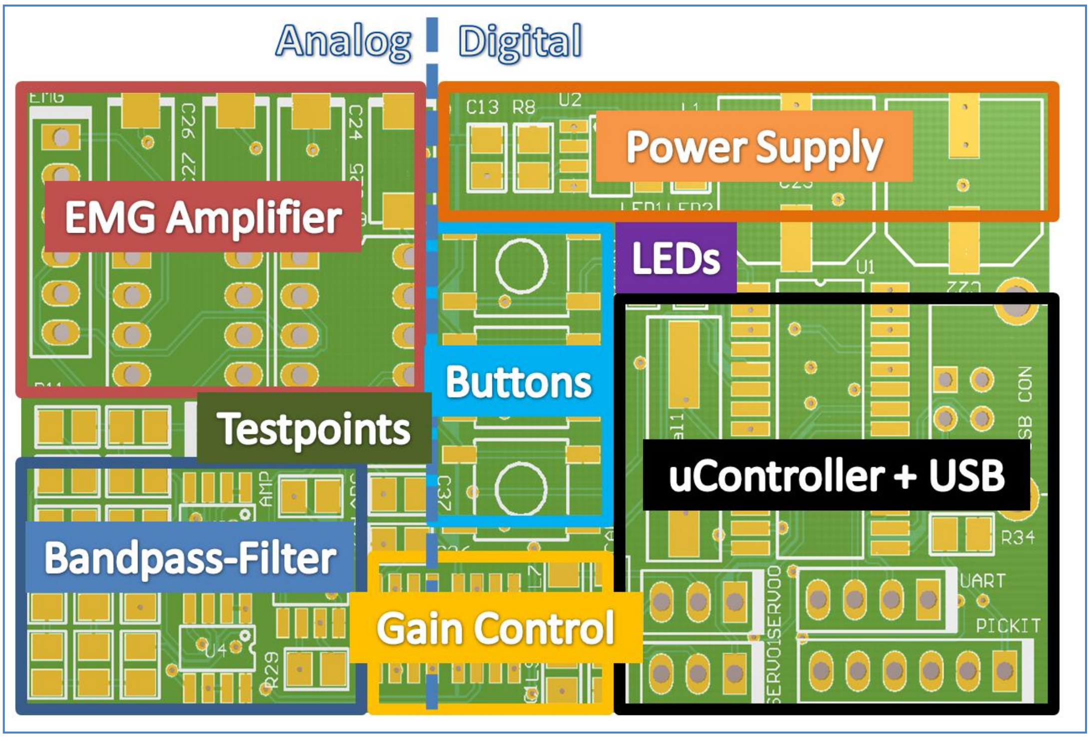

Figure 67: Modular component layout of the sEMG platform

5.4.2. POWER AND GROUND PLANES

The analog circuit receives power from the digital circuit (USB power). Figure 68 shows a diagram of the power plane connections with the analog power derived from the digital power. Component placement and the routing of tracks are done with care, not to divide each of the power planes into smaller planes, especially the reference (GND_{ref}) plane.

The decoding algorithm depends on the assumption that signals common to both inputs are canceled by the differential input comparison (performed in the proportional control model). In the case of poor ground/power plane layout any noise sensed by the reference plane, as an example, would be different at each of the analog differential amplifiers. This could affect the performance of the proportional control model, if noise, supposed to be common noise, would appear as differential information.

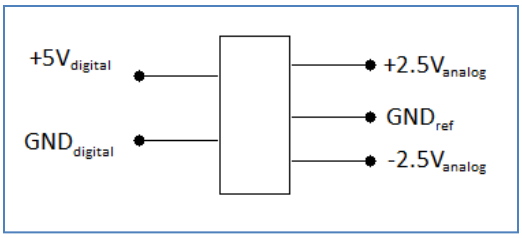

Figure 68: Power planes connections

Figure 66 illustrates how the power and ground reference planes are placed around the components they relate to. The +2.5V analog power plane supplies positive power to the EMG amplifiers and the band-pass filters (analog modules). The GND_{ref} is the analog ground reference of all the analog modules the analog circuit consists of. The +5V digital power plane supplies power to the microcontroller, the AGC and the PDM servo. The digital ground (referred to as GND) is the digital reference plane required by the digital circuit. Figure 69 illustrated the placement of components and power/ground planes on the PCB layout.

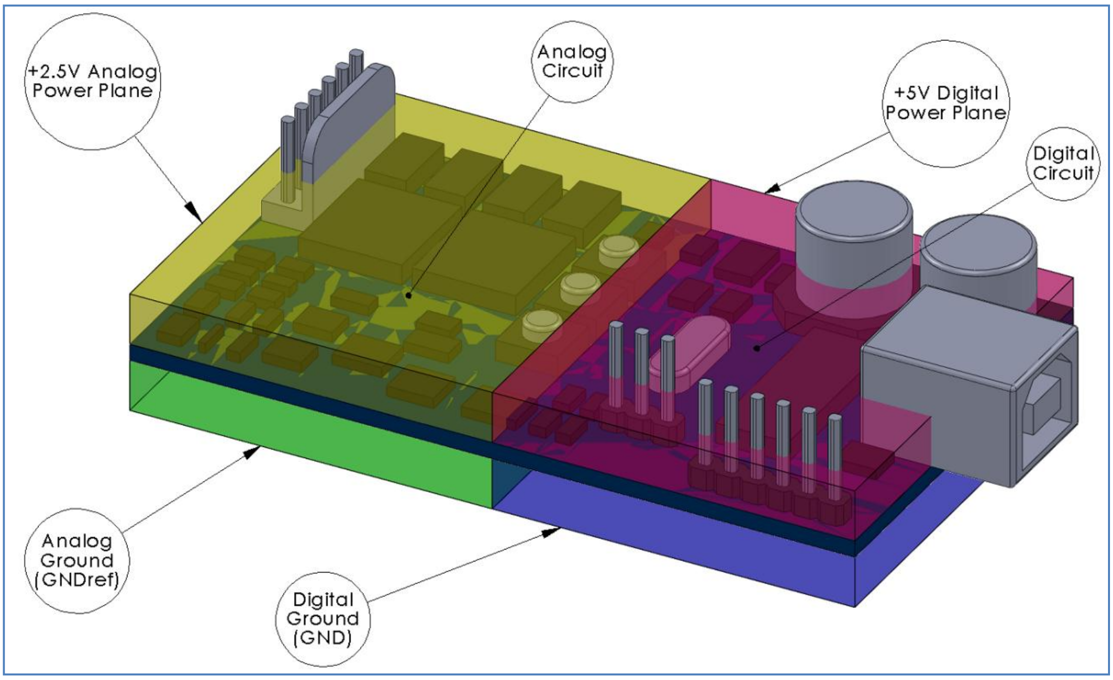

Figure 69: Power planes in platform

5.4.3. EMI

The digital circuit receives power from the USB port. The noisy digital USB-port, the power planes and analog power planes on the platform are isolated from one another using EMI-suppression beads (inductors) to prevent any digital noise from the USB-port and digital power planes to reach the analog power planes. Figure 70 contains all the parts of the sEMG platform schematic related to EMI suppression. All the components labeled “L” in the schematic (Appendix B) are EMI suppressor beads with an impedance of 50 Ω at 100 MHz.

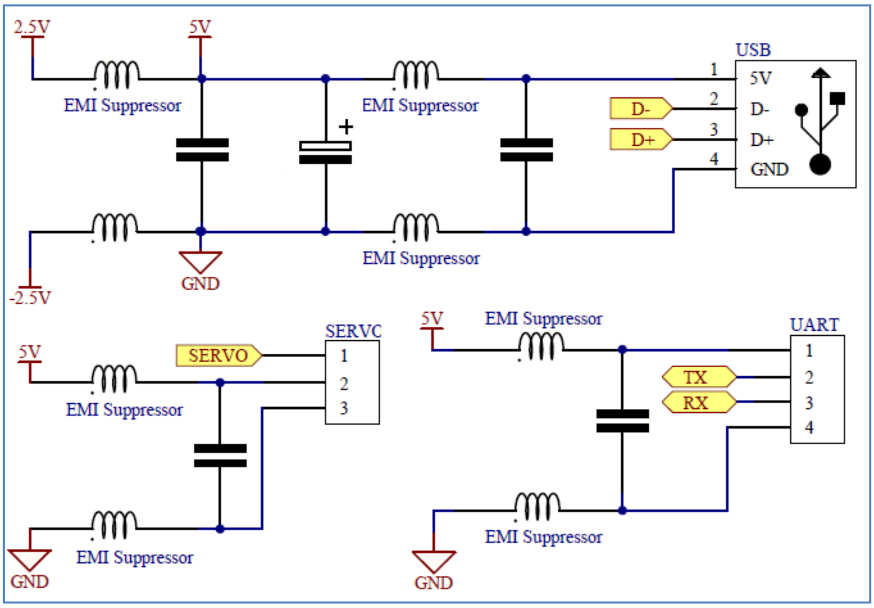

Figure 70: EMI provision in platform

5.4.4. DIMENSIONS

This platform is intended for research purposes, and no specification is given for the size of the platform. The size is important, as it could be used for a final prosthetic controller as the final PCB layout is 40mm x 70mm, which should fit easily into a prosthesis arm. Although larger in size, a type-B USB-connector is selected as it more robust than a smaller type mini-A or mini-B USB-connector. Figure 71 is a photograph of the prototype platform and the final PCB layout. Note the difference in size between the prototype platform (it consist of mainly thru-hole components soldered to two boards) and the final double-sided, single-layer PCB (mostly surface mount components on a tight-spaced layout).

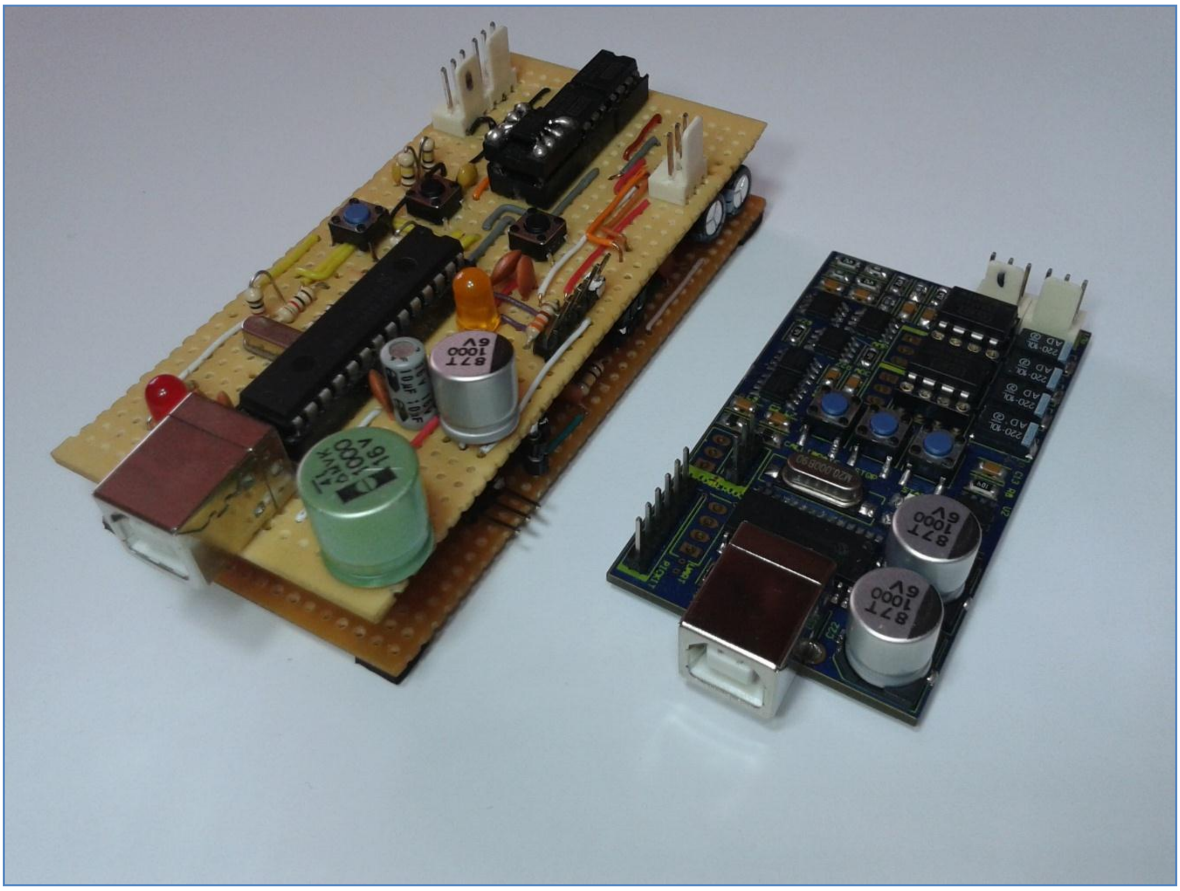

Figure 71: The prototype (left) and the final sEMG platform (right)

5.5. CONCLUSION

The detailed design includes two hardware versions. The first version proves the sEMG platform concept, and the second version is the final PCB design of the sEMG platform. The first achievement is to implement a sEMG sensor design and be able to send the information to Matlab®/Simulink® environment. The next achievement is to mix analog signal amplifiers and digital electronics in a sound functional design, and manage to achieve good results,

The performances of these two versions are compared to another, and to past studies to gain knowledge on the success of the hardware design. This will be discussed in the following chapter.

The decoding algorithm is implemented using Simulink®, and shows the ease of implementing DSP techniques, as the essence of this study is to create a sensing platform for future studies to make use of the Matlab®/Simulink® environment. The results obtained are also discussed in the following chapter.