Surface Electromyogram Simulator for Myoelectric Prosthesis Testing

Sponsored by Liberating Technologies Inc, Holliston, MA

A Major Qualifying Project proposal to be submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science

By:

____________________________________
Stephen Jung

____________________________________
John Meklenburg

____________________________________
Sean Patrick

Submitted On: 8 March, 2010

Submitted To:

____________________________________
Professor Edward A. Clancy, Advisor, Electrical & Computer Engineering

____________________________________
Professor Yitzhak Mendelson, Advisor, Biomedical Engineering

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see http://www.wpi.edu/Academics/Projects.
Abstract

Myoelectric prostheses use the naturally occurring surface electromyogram (EMG) produced by extant muscle tissue to provide amputees control of artificial limbs. Design and testing of these devices is currently performed using function generators or the healthy EMG signal of the tester. However, these methods of testing either do not provide data representative of the intended usage or are inconvenient to the tester, respectively. In this paper, we present a simple and portable prototype device which simulates the surface EMG signal with a correlated Gaussian random process in order to test myoelectric prostheses with a currently unavailable level of precision.
Statement of Authorship

Abstract: Sean Patrick

Executive Summary: Sean Patrick

Ch. 1: Introduction: Stephen Jung, Sean Patrick

Ch. 2: Background: John Meklenburg, Sean Patrick, Stephen Jung

Ch. 3: Project Strategy: John Meklenburg, Stephen Jung

Ch. 4: Critical Paths: John Meklenburg, Stephen Jung, Sean Patrick

Ch. 5: Final Design: Sean Patrick, John Meklenburg

Ch. 6: Design Verification: John Meklenburg

Ch. 7: Discussion: Stephen Jung

Ch. 8: Conclusions and Recommendations: Sean Patrick, Stephen Jung

Appendices: John Meklenburg, Stephen Jung, Sean Patrick
Acknowledgements

The authors would like to thank:

**Liberating Technologies, Inc.**, for making this project possible, and in particular **Todd Farrell**, **Bill Hanson**, and **Bob Quinzani** for providing valuable feedback throughout the design process;

**Professors Ted Clancy** and **Yitzhak Mendelson**, for their continual guidance and advice in their respective roles as the ECE and BME department advisors for this project;

**Professors John McNeill, Susan Jarvis**, and **Gene Bogdanov**, for their expert advice regarding various design issues over the course of this project;

**James O'Rourke** and **Tom Angelotti** for their help with practical issues relating to the project.
Executive Summary

Myoelectric prostheses are prosthetic devices which use the electromyogram (EMG) signal produced naturally by muscle tissue as a mechanism of control. Currently, there are two methods available for testing myoelectric prostheses: the use of a human tester’s own EMG signal, or the use of commercially available sine-wave function generators. However, neither of these solutions is optimal, as use of human EMG is inconvenient to the tester and use of a sine-wave function generator does not provide a realistic test signal. The EMG simulator described in this paper is an attempt to improve myoelectric prosthesis testing by providing a specialized function generator which outputs a realistic EMG signal without inconveniencing the tester.

EMG signals can be modeled as random in nature, following a probability distribution which is approximately Gaussian. They are characterized in the frequency spectrum as being bandlimited with most signal power between approximately 20 and 200 Hz. During contraction of muscles, EMG signals experience an increase in standard deviation (amplitude) with increased muscle force. The EMG signal also often rides upon a sinusoidal voltage of relatively large amplitude due to power line interference at 50 or 60 Hz, depending on the regional standard. Based on these characteristics, the EMG simulator was designed to output bandlimited Gaussian signals and additive sinusoidal power line interference, each with adjustable amplitude.

In order to aid testers, three distinct modes of operation were specified for the EMG simulator. The default mode of operation is "manual," in which users control EMG amplitudes by hand using knobs on the face of the device. Users may also select the "ramp" mode in order to test graded contraction scenarios, where EMG amplitude is modulated by a triangle wave, causing amplitudes to rise and fall periodically. The third mode selectable by the user is "pulse"
mode, in which EMG is modulated by a square-wave with a duty cycle selectable by the user. Pictures of these operating modes are shown below in Figure 1:

![Figure 1 – EMG simulator operating in manual mode (left), ramp mode (middle), and pulse mode (right).](image)

The EMG simulator was required to interface with a variety of bipolar prosthesis electrodes. Bipolar electrodes are characterized by having two differential inputs, across which a gain is typically applied, and a third input to provide a voltage reference. Since the EMG simulator must interface with this type of electrode, the output stage was specified to provide three outputs per electrode: two signal outputs, across which a common-mode power line interference signal and a differential EMG signal were provided; and a reference output to minimize DC bias at the electrode. The simulator was required to interface with two electrodes simultaneously, for a total of six output voltages (2 differential outputs and a voltage reference per electrode).

During the design process, both analog and digital implementations of the EMG simulator were considered. Due to the flexibility and ease of digital implementation, a digital solution was pursued in the final design. Thus, the final prototype consisted of a digital microcontroller unit (MCU) on a prefabricated development board, supplemented by custom
auxiliary hardware to convert the digital output of the MCU into an analog format useful to the electrode.

The MCU was used in this application to handle user input, generate an EMG signal based on that input, and output the EMG to a digital to analog converter (DAC). User input was received via five buttons and two knobs on the face of the device. Using the buttons, users navigate through a menu system to adjust various parameters of the output signal. Each button had a context-specific function based on the current menu. In addition to the buttons, two knobs on the face of the device handle user input during manual operation of the device.

The EMG signal was simulated digitally by filtering uniformly distributed random numbers produced by a linear congruential random number generator. As expected based on the central limit theorem (which states that the addition of independent and identically distributed random variables of any distribution will result in a Gaussian distributed output), the filter produced a bandlimited signal with a first-order probability distribution that was approximately Gaussian.

The Gaussian signal, which approximates the surface EMG, was then multiplied by a mode-appropriate number (an ADC input in manual mode, a triangle wave in ramp mode, and a square wave in pulse mode), added to 50/60 Hz simulated power line interference generated using a sine table, and formatted to the serial peripheral interface protocol (SPI) before being output to the DAC. This process was repeated four times in order to produce two differential outputs for each of two bipolar electrodes, with the second output for each electrode having an additional step: the inversion of the EMG signal. This inversion is performed in order to present the electrode with a differential signal at both inputs (instead of providing the entire EMG at one input), so that faults in the prosthesis electrode may be diagnosed more easily.
The DAC used in this application, the TI DAC8564, communicated with the MCU via SPI, an industry standard protocol. Its function was to convert the digital output of the MCU into an analog signal varying between 0 and 2.5 V (with an offset of 1.25 V). After this conversion occurred, the analog signal was stepped down in amplitude and level-shifted for delivery to the electrode. The conversion was accomplished using operational amplifiers in single-supply (0 and 5 V power rails), inverting gain configuration (gain of 0.016, R1 = 1.6k, R2 = 100k), with 1.25 V at the positive input. The single-supply configuration allowed for the output signal to be produced with a 1.25 V offset, the same as the offset introduced by the DAC, but with an output amplitude of only 40 mVpp. In order to produce an output signal without a DC offset, 1.25 V was presented to the bipolar electrode at its reference input, so that the differential input signal was taken with respect to 1.25 V.

The EMG simulator had, in total, six outputs to the target prosthesis. These correspond to the three inputs on each of two electrodes: two differential inputs and one reference node per electrode. The reference presented to the prosthesis is the same for each electrode (1.25 V DC). The two remaining outputs for each electrode carried a common-mode 50/60 Hz sine wave signal (simulating power line interference), as well as a differential EMG signal (one EMG being inverted with respect to the other to produce a differential signal).

The EMG simulator is a fully functional prototype capable of outputting EMG signals accurately and precisely over a range from 10 µVpp to 20 mVpp in amplitude. Three modes of operation (manual, ramp, and pulse) are available to the user. However, there are several areas in which the design might be improved by future efforts.

One such possible improvement is code organization. The software code used in the current EMG simulator has many opportunities for improvement in structure. For example, there
are several instances of code reuse which might be averted by creating C functions for commonly used operations. Future versions of the device should organize the code so that modification of the software component is simple and intuitive.

Future work on the EMG simulator might include additional features, such as the implementation of data storage on the MCU, so that users may save settings that they use often. Another possible feature is the inclusion of a “phase” setting for each channel, so that users may have the input signal to one electrode be out of phase with another electrode’s input signal.

A final area for improvement is the mechanical interface between the EMG simulator and the target prosthesis electrodes. The current EMG simulator does not have any mechanical interface beyond the output wires, and as a result it is difficult to attach to the electrodes securely. Future versions of the device should incorporate a mechanical interface solution that allows safe and clean fixation to the prosthesis electrodes.

The EMG simulator prototype meets almost all of the design requirements. It produces a signal within the desired output range and with the desired characteristics, is battery powered, and is easy to use. A picture of the final prototype is shown in Figure 2. Note that the device will be much smaller (and handheld) if a printed circuit board (with surface mount components) to house the output stage electronics and the DAC. A concept (at this smaller size) of a potential final product is shown in Figure 3.
Figure 2 – Prototype of EMG simulator

Figure 3 - Concept of potential final product design of the EMG simulator
# Table of Contents

Abstract ........................................................................................................................................... 2
Statement of Authorship ................................................................................................................. 3
Acknowledgements ......................................................................................................................... 4
Executive Summary ........................................................................................................................ 5
Table of Contents .......................................................................................................................... 11
Table of Figures ............................................................................................................................ 13
Table of Tables ............................................................................................................................. 17
Table of Equations ........................................................................................................................ 18
Chapter 1: Introduction ................................................................................................................ 19
  2.1 Basic Neuromuscular Physiology ...................................................................................... 21
  2.2 Generation of the EMG ...................................................................................................... 23
  2.3 Properties of the EMG Signal ............................................................................................ 24
  2.4 Probabilistic Description of the EMG ................................................................................ 26
    2.4.1 The MUAP as a Random Variable ............................................................................. 29
    2.4.2 Gaussian Signals ......................................................................................................... 30
  2.5 Prosthetic Devices .............................................................................................................. 33
Chapter 2: Background ................................................................................................................ 21
  3.1 Initial Client Statement ...................................................................................................... 36
  3.2 Design Parameters: Objectives, Constraints, and Functions ............................................ 37
  3.3 Revised Client Statement ................................................................................................... 38
Chapter 4: Critical Paths .............................................................................................................. 40
  4.1 Signal Generation Methodology ........................................................................................ 40
    4.1.1 Design Alternatives ..................................................................................................... 40
    4.1.2 Feasibility Studies ....................................................................................................... 41
    4.1.3 Conclusion .................................................................................................................. 45
  4.2 Hardware Platform ............................................................................................................. 46
    4.2.1 Design Alternatives ..................................................................................................... 46
    4.2.2 Feasibility Studies ....................................................................................................... 50
    4.2.3 Conclusion .................................................................................................................. 65
  4.3 Signal Delivery System ...................................................................................................... 66
    4.3.1 Design Alternatives ..................................................................................................... 66
    4.3.2 Feasibility Studies ....................................................................................................... 67
    4.3.3 Conclusion .................................................................................................................. 73
  4.4 Design Decisions ............................................................................................................... 74
    4.4.1 Signal Generation Methodology ................................................................................. 74
    4.4.2 Signal Generation Platform ......................................................................................... 75
    4.4.3 Signal Delivery System ............................................................................................... 79
Chapter 5: Final Design ............................................................................................................... 80
5.1 System Architecture........................................................................................................... 83
  5.1.1 Device Functionality.................................................................................................... 83
  5.1.2 Software Architecture................................................................................................. 84
  5.1.3 Auxiliary Hardware Architecture ............................................................................... 86
5.2 User Interface ..................................................................................................................... 89
5.3 Signal Generation ............................................................................................................... 94
5.4 Signal Delivery .................................................................................................................. 98
5.5 Conclusion ........................................................................................................................ 101
Chapter 6: Design Verification ............................................................................................... 104
  6.1 DAC Output ..................................................................................................................... 105
    6.1.1 EMG Amplitude Verification ................................................................................... 106
    6.1.2 Noise Amplitude Verification ............................................................................... 108
  6.2 Output Stage ..................................................................................................................... 111
  6.3 Noise Frequency and Phase ............................................................................................. 114
  6.4 Overall EMG and Noise Amplitude Control ................................................................... 115
  6.4 Realism of EMG signal .................................................................................................... 116
  6.5 Modes of Operation ......................................................................................................... 119
Chapter 7: Discussion ................................................................................................................ 124
Chapter 8: Conclusions and Recommendations ................................................................. 131
References ................................................................................................................................... 134
Appendix A: Objectives Tree .................................................................................................... 136
Appendix B: MATLAB Code for Signal Generation Simulations ............................................ 137
Appendix C: Analog Signal Generation System Test Schematic .............................................. 140
Appendix D: Commonly Used Circuit Topologies and their Governing Equations ................. 141
Appendix E: Analog Signal Generation Test Results ............................................................... 144
Appendix F: MATLAB Code for Comparing Box-Muller Transform and FIR Filtering .......... 148
Appendix G: C-Code for MSP430 Tests ................................................................................... 149
Appendix H: Analog Signal Delivery System Test Schematic ................................................ 152
Appendix I: Analog Signal Delivery System Photographs ....................................................... 153
Appendix J: Analog Signal Delivery System Test Data ............................................................ 155
Appendix K: Cost Analysis ....................................................................................................... 160
Appendix L: Olimex LPC-MT-2138 Schematic ........................................................................ 161
Appendix M: Texas Instruments DAC8564EVM Schematic .................................................... 162
Appendix N: Auxiliary Electronics Schematic .......................................................................... 163
Appendix O: Function List ........................................................................................................ 164
Appendix P: Derivation of Output Gain Stage ......................................................................... 167
Appendix Q: DAC Output Tests for EMG and Noise Amplitude Verification ......................... 168
Appendix R: Output Stage Tests for Gain and Frequency Response Verification .................... 171
Appendix S: MATLAB Simulation for Testing Linear Congruential Generator ...................... 173
## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EMG simulator operating in manual mode (left), ramp mode (middle), and pulse mode (right).</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Prototype of EMG simulator</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Concept of potential final product design of the EMG simulator</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>The MUAP and EMG signal. Left: Individual MUAP as part of an MUAP train. Right: Schematic representation of EMG as a sum of MUAP trains (De Luca &amp; Van Dyk, 1975).</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>Causative, intermediate, and deterministic factors that affect surface EMG signals (De Luca, 1997).</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>PDF of a uniformly distributed random variable</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>CDF of a uniformly distributed random variable</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>Gaussian distributions with different parameters</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Normalized experimental PDF of EMG. Probability is plotted with respect to normalized value. Gaussian in dashed line, EMG in solid (Clancy &amp; Hogan, 1999).</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>Typical EMG power spectrum. EMG data source: WPI ECE443X course website.</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>Examples of prosthesis products from Liberating Technologies Inc</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>Variations in the time and frequency domain surface EMG signal based on electrode placement (De Luca, 1997).</td>
<td>35</td>
</tr>
<tr>
<td>13</td>
<td>Time domain representation (top), power spectral density (middle), and distribution (bottom) of surface EMG data simulated with a physiological algorithm.</td>
<td>43</td>
</tr>
<tr>
<td>14</td>
<td>Time domain representation (top), power spectral density (middle), and distribution (bottom) of EMG data simulated by a correlated Gaussian random process.</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>Time domain representation (top), power spectral density (middle), and distribution (bottom) of real EMG data. EMG data source: WPI ECE443X course website.</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>Thevenin equivalent of thermal noise across a resistor, consisting of a resistor and a noise source.</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>Olimex MSP430-449STK2 Development Board (<a href="http://www.olimex.com/dev/msp-449stk2.html">http://www.olimex.com/dev/msp-449stk2.html</a>)</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>Olimex LPC-MT-2138 Development Board (<a href="http://www.olimex.com/dev/lpc-mt-2138.html">http://www.olimex.com/dev/lpc-mt-2138.html</a>)</td>
<td>49</td>
</tr>
<tr>
<td>19</td>
<td>AD620B with internal ground reference. R is the resistor that generates noise, RG is set to 49 Ω for a gain of 1000.</td>
<td>53</td>
</tr>
<tr>
<td>20</td>
<td>Two stage gain and filtering process</td>
<td>55</td>
</tr>
<tr>
<td>21</td>
<td>Distributions of data generated with varying resistor values</td>
<td>58</td>
</tr>
<tr>
<td>22</td>
<td>Power Spectral Densities of data generated with varying resistor values and after passing through a 20 Hz to 420 Hz band pass filter</td>
<td>59</td>
</tr>
<tr>
<td>23</td>
<td>The Box-Muller transform. Gaussian distribution overlaid in red.</td>
<td>62</td>
</tr>
<tr>
<td>24</td>
<td>Linear filtering. Gaussian distribution overlaid in red.</td>
<td>62</td>
</tr>
<tr>
<td>25</td>
<td>Design alternatives for signal delivery system</td>
<td>67</td>
</tr>
</tbody>
</table>
Figure 26 - Experimental setup of signal delivery system test with node voltages ............................... 69
Figure 27 - Transmission of 20 µVpp signal at 150 Hz ........................................................................ 72
Figure 28 - Transmission of 20 µVpp signal at 400 Hz ....................................................................... 72
Figure 29 - Transmission of 5 µVpp signal at 150 Hz ......................................................................... 73
Figure 30 - Transmission of 5 µVpp signal at 400 Hz ......................................................................... 73
Figure 31 – Block diagram of final design of EMG simulator ............................................................... 82
Figure 32 – Interfacing between software in the MCU and the auxiliary hardware ............................... 84
Figure 33 - Software flowchart ............................................................................................................ 85
Figure 34 - Knob potentiometer and battery meter circuit ...................................................................... 87
Figure 35 - Hardware flowchart .......................................................................................................... 88
Figure 36 - Menu navigation diagram. Arrows on the left indicate that the menu is the final option in the sequence; arrows on the top indicate that there are further submenus. ...................... 90
Figure 37 - Example navigation of menu system ................................................................................... 91
Figure 38 - The start screen. A: Battery meter. B: Run status indicator. C: Channel label. D: Noise indicator. E: Mode indicator. F: Amplitude indicator ................................................................. 92
Figure 39 - Magnitude response of integer filter ................................................................................... 96
Figure 40 - Output stage electronics for a single channel ..................................................................... 100
Figure 41 - Output stage equivalent circuit .......................................................................................... 101
Figure 42 - Reference voltage circuit ................................................................................................... 102
Figure 43 - Prototype of EMG simulator ............................................................................................... 102
Figure 44 - Concept of potential final product design of the EMG simulator ...................................... 106
Figure 45 - Test apparatus for DAC output test to verify EMG amplitude control ............................ 106
Figure 46 - Test results of DAC output test for EMG amplitude control 10 µVpp input, approx 576 mVpp output, test apparatus gain of 1000 .................................................................................. 107
Figure 47 - Test results of DAC output test for EMG amplitude control 20 mVpp input, approx 1.24 Vpp output, test apparatus gain of 1 ......................................................................................... 108
Figure 48 - Test apparatus for DAC output test to verify noise amplitude control ............................ 109
Figure 49 - Test results of DAC output test for noise amplitude control 10 µVpp input, test apparatus gain of 100 .................................................................................................................... 110
Figure 50 - Test results of DAC output test for noise amplitude control 20 mVpp input, test apparatus gain of 1 .................................................................................................................................. 111
Figure 51 - Test apparatus for output stage verification ........................................................................ 112
Figure 52 – Output stage test. Input of 2.5 Vpp at 20 Hz ....................................................................... 113
Figure 53 – Output stage test. Input of 2.5 Vpp at 220 Hz ..................................................................... 113
Figure 54 - Simulated 50 Hz common mode line noise. Yellow signal is from the positive output terminal, blue signal is from the negative output terminal ................................. 114
Figure 55 – Simulated 60 Hz common mode line noise. Yellow signal is from the positive output terminal, blue signal is from the negative output terminal ........................................... 115
Figure 56 - Test apparatus for EMG realism test ................................................................................... 116
Figure 57 - Time domain representation, power spectral density, and distribution of simulated EMG data taken from the device. Device settings: pulse mode, 10 mVpp EMG amplitude, 20 mVpp 60 Hz noise amplitude, 1000 ms duration, 100% duty cycle. Note that the 60 Hz simulated line noise did not appear in the data because it was rejected by the differential amplifier. ....... 117
Figure 58 - Manual mode, full range ................................................................. 120
Figure 59 - Pulse mode, 10 mVpp amplitude, 500 ms duration, 25% duty cycle .......... 121
Figure 60 - Pulse mode, 10 mVpp amplitude, 500 ms duration, 50% duty cycle .......... 121
Figure 61 - Pulse mode, 10 mVpp amplitude, 1 s duration, 50% duty cycle ............... 122
Figure 62 - Ramp mode, 10 mV peak amplitude, 2 s period .................................... 123
Figure 63 - Ramp mode, 10 mV peak amplitude, 4 s period ..................................... 123
Figure 64 - Inverting op-amp circuit ................................................................. 141
Figure 65 – Resistive voltage divider ............................................................... 142
Figure 66 – Unity gain second-order Sallen Key lowpass filter ......................... 143
Figure 67 – Unity gain second-order Sallen Key highpass filter ......................... 143
Figure 68 - Signal delivery system test, overall setup. The simulator (transmitter) circuit is on the protoboard to the left, and the prosthesis signal conditioning (receiver) circuit is on the protoboard to the right. The protoboards are connected with the shielded cable from Cooner Wire................................................................. 153
Figure 69 - Signal delivery system transmitter circuit ........................................ 154
Figure 70 - Signal delivery system receiver circuit ........................................... 154
Figure 71 – Analog signal delivery system test data with input signal 20 µVpp, 400 Hz...... 155
Figure 72 - Analog signal delivery system test data with input signal 20 µVpp, 150 Hz ...... 156
Figure 73 - Analog signal delivery system test data with input signal 15 µVpp, 400 Hz ...... 156
Figure 74 - Analog signal delivery system test data with input signal 15 µVpp, 150 Hz ...... 157
Figure 75 - Analog signal delivery system test data with input signal 10 µVpp, 400 Hz .... 157
Figure 76 - Analog signal delivery system test data with input signal 10 µVpp, 150 Hz .... 158
Figure 77 - Analog signal delivery system test data with input signal 5 µVpp, 400 Hz ...... 158
Figure 78 - Analog signal delivery system test data with input signal 5 µVpp, 150 Hz ...... 159
Figure 79 – Test results of DAC output test for EMG amplitude control. 100 µVpp input, approx 6.20 Vpp output, test apparatus gain of 1000. ................................................................. 168
Figure 80 – Test results of DAC output test for EMG amplitude control. 1000 µVpp input, approx 6.20 mVpp output, test apparatus gain of 100. ................................................................. 168
Figure 81 - Test results of DAC output test for EMG amplitude control. 10000 µVpp input, approx 6.20 mVpp output, test apparatus gain of 1. ................................................................. 169
Figure 82 – Test results of DAC output test for noise amplitude control. 100 µVpp input, test apparatus gain of 100. ................................................................. 169
Figure 83 – Test results of DAC output test for noise amplitude control. 1000 µVpp input, test apparatus gain of 100. ................................................................. 170
Figure 84 - Test results of DAC output test for noise amplitude control. 10000 µVpp input, test apparatus gain of 1. ................................................................. 170
Figure 85 - Output stage test. Input of 2.5 Vpp at 60 Hz.......................................................... 171
Figure 86 - Output stage test. Input of 2.5 Vpp at 100 Hz.......................................................... 171
Figure 87 - Output stage test. Input of 2.5 Vpp at 140 Hz.......................................................... 172
Figure 88 - Output stage test. Input of 2.5 Vpp at 180 Hz.......................................................... 172
Table of Tables

Table 1 - Design parameters: objectives, constraints, and functions ....................................................... 38
Table 2 - True resistor values and their corresponding expected RMS value after amplification 53
Table 3 - Theoretical resistor values and their corresponding expected RMS values .......................... 54
Table 4 - Data from reproducibility test of analog signal generator (see Appendix E for full data measurements) ...................................................................................................................................... 56
Table 5 - Data from stability test of analog signal generator (see Appendix E for full data measurements) ...................................................................................................................................... 57
Table 6 - Run speed of candidate algorithms on the MSP430F449 microcontroller running at 4 MHz .......................................................................................................................................................... 63
Table 7 - Run-times for various calculation types on the MSP430F449 microcontroller running at 8 MHz .......................................................................................................................................................... 64
Table 8 - Estimated run speeds of program components ........................................................................ 65
Table 9 - Mapping of function generator output to simulated DAC output ......................................... 70
Table 10 - Mapping of simulated DAC output to overall system (voltage divider) output ................ 71
Table 11 - Pairwise Objective Comparison chart for quantitative design analysis .......................... 76
Table 12 - Numerical Evaluation Matrix for quantitative design analysis .................................................. 77
Table 13 - Cost analysis of final product .................................................................................................. 103
Table 14 – Results of reproducibility test for 10 MΩ resistor ............................................................ 144
Table 15 - Results of reproducibility test for 1 MΩ resistor .............................................................. 144
Table 16 - Results of reproducibility test for 100 kΩ resistor ............................................................. 145
Table 17 - Results of reproducibility test for 10 kΩ resistor ............................................................... 145
Table 18 - Results of stability test for 10 MΩ resistor ........................................................................ 146
Table 19 - Results of stability test for 1 MΩ resistor ........................................................................... 146
Table 20 - Results of stability test for 100 kΩ resistor ....................................................................... 147
Table 21 - Results of stability test for 10 kΩ resistor ........................................................................... 147
Table of Equations

Equation 1 - Calculation of positive-going portion of simulated EMG signal (represented by MUAP1). The variable “distance” represents the randomly generated distance from the recording site to the muscle fiber, and “fiberlength” represents the total length of the fiber. 41

Equation 2 - Thermal noise power spectral density in resistor (v = RMS voltage, k_B = Boltzmann’s constant, T = temperature in Kelvin, R = resistance, and f is frequency) (Nyquist, 1928). 51

Equation 3 - V_{RMS} calculation after one gain stage requires consideration of input noise from the AD620B. V_{RMS1} is the V_{RMS} value after the first stage gain. V_{RMS2} is the V_{RMS} after the second stage gain, and V_{RMS\text{total}} is the V_{RMS} value at the output. 52

Equation 4 - The Box-Muller transform. 61

Equation 5 – The general form of an FIR filter. 61

Equation 6 – Calculation of minimum DAC output amplitude for signal delivery system feasibility study. 69

Equation 7 - Linear congruential random number generator. 95

Equation 8 – Integer filtering of random numbers to produce EMG. 95

Equation 9 – Relation of the amplitude setting of the device to expected amplitude as measured from the output of the test apparatus for the DAC output. 105

Equation 10 - Relation of the input amplitude to expected amplitude as measured from the output of the test apparatus for the output stage. 111

Equation 11 – Calculation of time before signal repetition, Fs = sampling rate. 118

Equation 12 - Inverting op-amp circuit gain equation. 141

Equation 13 - Resistive voltage divider equation. 142

Equation 14 - AD620B instrumentation amplifier gain equation. 142

Equation 15 - Equation for cutoff frequency of Sallen Key filters. 143
Chapter 1: Introduction

Liberating Technologies Inc. (LTI) is a leading supplier of upper-limb prosthetic devices and is focused on the development of state-of-the-art microprocessor based prosthetic controllers, such as the Boston Digital Arm System and the VariGrip programmable prosthetic controller. LTI has sponsored this Major Qualifying Project (MQP), the design of a surface electromyogram (EMG) simulator that will be used to troubleshoot myoelectrically controlled prostheses.

Prior to the design of this device, LTI technicians were forced to use either a commercially available function generator or his or her own surface EMG signal to test the myoelectrically controlled prostheses. Both of these were poor test methods. The standard function generator did not closely approximate the surface EMG signal, and it was inconvenient and difficult for the tester to use his or her own EMG signals to test the prosthesis. Thus, a realistic surface EMG simulator was desired to fulfill the need for an easy to use and effective test method.

One intended market of the EMG simulator is prosthetists who need to test or calibrate prosthetic devices quickly and easily. Because a prosthethist may need to perform tests with little preparation time, the device was required to run on commercially available, disposable batteries rather than a rechargeable battery pack. In addition, disposable batteries reduce the safety concerns inherent to electrical systems by eliminating the need for 120 V AC power provided by a wall socket. Battery power also allows for the device to be handheld, making it easier to use.

The EMG simulator was also required to be relatively inexpensive (less than $1000 per unit) to produce. Because the EMG simulator is not intended for mass-production, individual units were required to be manufactured using inexpensive, commercially available parts and
processes. The device was desired to be manufacturable on a small scale by LTI with a minimum amount of maintenance required, necessitating the production of a sustainable and maintainable design.

The signal generated by the EMG simulator was desired to be physiologically realistic. A range in EMG amplitude from 10 µV<sub>pp</sub> to 20 mV<sub>pp</sub>, with an additive 50/60 Hz sinusoidal signal having a maximum amplitude of 20 mV<sub>pp</sub> was desired to simulate power line interference. Three modes of operation were specified for the device: a manual operation mode, in which users control the amplitude of the EMG signal in real time; a ramp mode, in which the EMG periodically increases and decreases linearly in amplitude; and a pulse mode, in which the EMG alternates between a user-defined amplitude and zero amplitude. Two independent output channels were needed so that the EMG simulator could interface with multiple electrodes. A device encompassing all of these features would provide a useful and realistic alternative test method to LTI.
Chapter 2: Background

In order to understand the EMG signal, it is necessary to understand how it is formed at the cellular level. The physiology and behavior of muscle tissue contribute greatly to the shape of the EMG. Because a basic understanding of natural EMG generation is necessary for understanding artificial EMG generation, this section will review the germane information in physiological EMG generation, acquisition, and analysis.

2.1 Basic Neuromuscular Physiology

Any muscle found in the human body can be categorized as one of three types: skeletal muscle, smooth muscle, or cardiac muscle. Skeletal muscles are attached to bones and are primarily responsible for limb control. Cardiac muscles are responsible for the involuntary rhythmic contractions of the cardiac walls, forcing blood out of the heart and into the circulatory system. Smooth muscles are also controlled involuntarily, and are found in the lining of veins and arteries, as well as the digestive tract. Skeletal muscles are voluntarily controlled, and therefore are the only muscles involved with the function of myoelectrically controlled upper limb prostheses. (Fox, 2008)

The basic mechanical structure of skeletal muscles is relatively simple. The entire muscular unit is attached to the bone with strong connective tissue called tendons. Each muscle is composed of many muscle cells, also called muscle fibers. Each of these muscle fibers is further broken down into bundles of filaments called myofibrils. It is these overlapping filaments that are intimately involved with muscular contraction. (Fox, 2008)

Skeletal muscles are innervated by somatic motor neurons. The axon, the elongated body of the motor neuron, starts in the spinal cord and splits into many branches that terminate in the
motor end plate of the neuromuscular junction, the region of the sarcolemma (the plasma membrane surrounding the muscle fiber) that receives the input from the nervous system. Each one of these terminating branches stimulates exactly one muscle fiber. Together, a motor neuron and all of the muscle fibers innervated by that neuron make up a single motor unit. (Fox, 2008)

The process through which muscles are stimulated for contraction is known as excitation-contraction coupling. An action potential (AP), or electrical impulse with biochemical origins, begins in the spinal cord and propagates down the axon of the somatic motor neuron. When the action potential reaches the neuromuscular junction, the neurotransmitter acetylcholine crosses the post-synaptic cleft (the region between the motor neuron branch and the motor end plate) where it binds to receptors in the motor end plate. This process opens sodium ion channels, which create an action potential in the sarcolemma. (Fox, 2008)

It is this exchange of sodium and potassium ions that generates the electric field responsible for the myoelectric signal or electromyogram (Merletti & Parker, 2004). This action potential propagates through the transverse tubules, resulting in the opening of voltage gated calcium channels in the sarcoplasmic reticulum (the main vessel for calcium ion storage in muscle tissue), which releases calcium into the sarcoplasm of the muscle fiber (analogous to the cytoplasm of a regular cell), causing a contraction. The contraction is released when calcium pumps actively transport the ions out of the sarcoplasm back into the transverse tubules. In order to sustain a contraction, repeated APs are necessary; otherwise the calcium pumps will end the contraction within a few milliseconds. (Fox, 2008)

Graded contractions for entire muscles are made possible through two processes: motor unit recruitment and action potential firing rate adjustment. When stronger contractions are needed, more motor units are activated asynchronously to create a smooth contraction in the
process known as motor unit recruitment. Additionally, the firing rate in a single motor unit is proportional to the average strength of the corresponding muscle fiber’s contraction. Motor unit recruitment is the primary means of increasing overall contraction strength until a certain threshold is reached. According to De Luca, this threshold is approximately 30 percent of the maximal voluntary contraction. After this threshold is reached, an increased firing rate is the primary method of increasing contractile strength. (De Luca, 1979)

2.2 Generation of the EMG

The EMG signal measured by a conventional surface electrode is actually a combination of many distinct action potentials produced by the muscle tissue, called motor unit action potentials (MUAPs). In general, MUAPs have a distinct spike-like shape, which can be affected by several factors. The stochastic combination of individual MUAPs from multiple muscle fibers is what gives the surface EMG its characteristic noise-like appearance. Figure 4 shows the shape of the MUAP and a schematic representation of the EMG signal generated by the summation of individual MUAPs. (De Luca, 1979)

Figure 4 - The MUAP and EMG signal. Left: Individual MUAP as part of an MUAP train. Right: Schematic representation of EMG as a sum of MUAP trains (De Luca & Van Dyk, 1975).
Each MUAP has the basic shape of a voltage spike followed by a voltage dip. However, this shape may be altered in amplitude, duration, or frequency content. Amplitude can be affected by muscle fiber radius, temperature of the fiber, or characteristics of the electrode and tissue. Duration of the MUAP is inversely related to the propagation velocity of the fiber. Frequency content can be filtered by the low-pass transfer functions of the surrounding tissue and the electrode. (De Luca, 1979)

The two main contributors to changes in EMG amplitude (defined as the time-varying standard deviation of the signal) are motor unit firing rate and the number of motor units recruited for a contraction; these are also the two main contributors to contraction strength. Therefore, the amplitude of an EMG signal is proportional to the strength of the contraction at a surface recording site. However, this relationship is not always linear. Depending on the characteristics of the muscle group, the relationship may be more parabolic, which is most likely due to differences in the relationship of muscle rate coding and muscle recruitment in producing force. This direct relationship between EMG amplitude and force is the principle on which some myoelectric prostheses operate. (Merletti & Parker, 2004)

### 2.3 Properties of the EMG Signal

There are numerous factors that affect the properties of a surface EMG signal, making the signal very difficult to characterize completely. The characteristics of the recording electrode have a great deal of influence on the signal, as they may vary in size, shape, contact materials, configuration (monopolar or bipolar), filtering properties, and placement. The result will also differ with the characteristics of various recording systems. Physiological factors such as skin impedance, subcutaneous tissue thickness, muscle type and location, fiber orientation (with
respect to electrodes), and fatigue also affect the signal. Figure 5 illustrates how these and other properties can alter the characteristics of the EMG signal. (De Luca, 1997)

![Figure 5 - Causative, intermediate, and deterministic factors that affect surface EMG signals (De Luca, 1997).](image)

The EMG signal contains important diagnostic data in both the time and frequency domains. Two commonly used time-domain parameters are root-mean-square value and mean rectified value (De Luca & Van Dyk, 1975). Variance of the rectified signal is another useful metric (De Luca, 1979). Proportional control of myoelectric prostheses is typically dependent upon the root-mean-square value of the signal (Philipson, Childress, & Strysik, 1981).

In the frequency domain, one commonly used metric is the power density spectrum. From the power spectrum, measures such as median frequency, mean frequency, and the ratio of low frequencies to high frequencies can be extracted (Stulen & De Luca, 1981). De Luca has
shown that, on a physiological level, these metrics are dependent on motor unit firing rate, the number of MUAP trains (MUAPTs) included in the recording, the size and shape of the MUAP, and the level of synchronization of the MUAPs (De Luca & Van Dyk, 1975).

While each EMG recording device is different, Merletti has compiled a list of recommendations in order to record a diagnostically significant signal. He recommends a highpass filter with a break frequency of not higher than 20 Hz and a rolloff of 40 dB/decade to prevent motion artifacts and baseline wander. A lowpass filter with the same rolloff and a cutoff frequency of not lower than 450 Hz should be used to reduce noise and aliasing of the signal if it is sampled at the minimum recommended sampling frequency of 1 kHz. A notch filter to remove 50 and 60 Hz line noise is not recommended because the EMG contains a significant amount of power in that frequency range (Merletti & Bonalo, 2008). De Luca also recommends a common-mode rejection ratio of 80 dB or more at power-line frequency, less than 2 $\mu$V$_{\text{rms}}$ of input-referred noise, and an input impedance of greater than 100 M$\Omega$. The amplitude of the resulting signal typically falls in the range of 0 to 10 mV$_{\text{pp}}$ or 0 and 1.5 mV$_{\text{rms}}$ (De Luca, 1997).

### 2.4 Probabilistic Description of the EMG

Describing the EMG signal in either the time or frequency domains can sometimes be facilitated by using the language of probability. Because the EMG signal shares some characteristics with “random” signals, the techniques developed to deal with stochastic processes can aid in the understanding and analysis of the surface EMG. There are four central concepts in this section: the random variable, the distribution function, statistical independence, and the central limit theorem.

Random variables are numbers which may take on a range of values. They differ from algebraic variables in that they are not defined deterministically, but rather take on any of a set of
values based on chance. The chance of a random variable taking on a certain value is called the probability of that value, defined as a fraction between 0 and 1 (1 being absolute certainty that a certain value will be taken). For any set of events, the total probability will always be 1 when all events are considered. (Gubner, 2006)

Random variables are often defined in terms of a distribution function, which may be a cumulative distribution function (CDF) or a probability distribution function (PDF), the latter being defined as the slope of the former. The distribution function describes the probability of some value X as a function of X. In the case of the PDF, the value of PDF(X) is the probability of X. The CDF is the cumulative area under the PDF, meaning that CDF(X) represents the summed probability of X and all values less than X. Graphs of each type of function for a uniformly distributed random variable (all events equally likely over a certain range) are shown in Figure 6 and Figure 7 for comparison. (Gubner, 2006)

![Figure 6 - PDF of a uniformly distributed random variable](image)
Central to understanding probability is the concept of independence. Two events are said to be independent if, for all values of the two events, the outcome of one event does not affect the outcome of the other. An example of two independent events is the roll of a fair die twice in a row: the first die roll does not affect the second one and the probability of all outcomes is one in six, regardless of the first number rolled. Similarly, random variables are independent if they do not affect each other’s value. Independent variables exhibit several useful properties, not the least of which is described by the central limit theorem. (Gubner, 2006)

The central limit theorem states that, for independent and identically distributed (i.i.d.) random variables, the sum of an infinite number of those variables will result in a normal or Gaussian distribution function. Regardless of the distribution of individual events, as long as they are independent of one another they will produce a certain distribution function when added together. This distribution, called Gaussian, is characterized by a certain norm and deviation from the norm (hence its other name, the normal distribution). In practice, however, even small numbers of i.i.d. random variables can sum to a distribution that is approximately Gaussian. Gaussian distributions with different means and standard deviations are shown in Figure 8. (Gubner, 2006)
The EMG signal as seen at the surface of the skin is, to a rough approximation, a Gaussian signal. The Gaussian form is no accident, and fortunately it has a number of useful properties for analysis and applications. Ensuing sections in this chapter will investigate the reasons for the EMG to take on this form, the characteristics of the form that are useful in analysis, and reasons for deviation from the approximated form by the actual signal.

2.4.1 The MUAP as a Random Variable

The generation of individual MUAPs is essentially random in a healthy and unfatigued neuromuscular system. Although each MUAP is generated deterministically by a signal from the central nervous system and the motor neuron, each successive AP occurs with a different inter-pulse delay, which can be modeled as an independent random variable. This randomness desynchronizes it from the other MUAPs, resulting in a signal that is pseudorandom as a function of time. Despite their common inspiration, under normal conditions each MUAP is fired so asynchronously that it can be approximated as independent with respect to other MUAPs.
However, MUAPs are not entirely independent. Some studies have shown a weak correlation between MUAPs, although these results have been only sporadically reproducible. In addition, there is always at least a certain small delay between action potentials on the same fascicle, due to the physiological refractory period required by the muscle cells (Fox, 2008). Despite these non-idealities, the approximation of MUAPs as independent is a reasonable one under normal (healthy, unfatigued) conditions: it is difficult to show even a weak correlation between MUAPs under these conditions, and the average inter-spike interval (ISI) is often large enough that the refractory period is not an important consideration. In addition to being independent, the MUAP can also be approximated as being identically distributed with the MUAPs of neighboring motor units, although this approximation is less realistic (De Luca, 1979).

By approximating the MUAP as independent and identically distributed, it is clear why the EMG exhibits a pseudo-Gaussian distribution function: the central limit theorem. Between the muscle and the surface of the skin, the electric fields generated by each MUAP are added together. Although it is not an infinite number of signals, the number is generally large enough to produce a Gaussian-like signal if the product of motor units recruited and firing rate in Hz is greater than 1000.

### 2.4.2 Gaussian Signals

The Gaussian-like EMG can be analyzed by assessing its characteristics as though it were a Gaussian random signal. One characteristic commonly assessed in random signals is the mean. However, in the raw surface EMG signal, the mean is roughly equal to 0, because the EMG takes on positive values as well as negative values with equal probabilities, and also because hardware highpass filtering is used to remove DC offset. Because of this, two similar measures of the
signal are usually used instead of the mean: the mean absolute value (MAV) and the root mean square value (RMS), which are reflections of the standard deviation of the signal. The standard deviation (σ or SD) is a measure of the size of the signal, representing the “spread” of the signal over all its possible values. It has been used in many EMG analyses successfully (Hodges & Bui, 1996).

Although it is useful for analysis to consider the EMG as a Gaussian signal, in reality this is not always an accurate assumption. Depending on the measurement and analysis techniques used, the EMG may appear to have different characteristics or even a different distribution. Furthermore, the EMG signal exhibits propagation delays and attenuation through the skin and tissue of the subject, affecting the signal seen at the surface. Despite these differences, it is often reasonable to approximate the EMG signal as Gaussian, especially once it is filtered (Clancy & Hogan, 1999). Figure 9 is shown for comparison between the actual EMG signal and the Gaussian equivalent.

![Graph](image.png)

Figure 9 – Normalized experimental PDF of EMG. Probability is plotted with respect to normalized value. Gaussian in dashed line, EMG in solid (Clancy & Hogan, 1999).
The power spectrum of the ideal Gaussian signal is flat or “white”, meaning it has equal signal power at all frequencies. However, the EMG signal typically exhibits a frequency band between 20 and 400 Hz, with most of the signal power occurring between 20 and 200 Hz. Despite this difference, the approximation of the EMG as Gaussian is still fairly accurate. An EMG frequency spectrum is shown in Figure 10.

![Real EMG Power Spectral Density](image)

**Figure 10 – Typical EMG power spectrum. EMG data source: WPI ECE443X course website.**

The EMG is not the only naturally occurring Gaussian signal. Many forms of noise, such as thermal noise, are Gaussian and this form of noise has been used in the generation of Gaussian random numbers (Saito, 2001). In addition to naturally occurring Gaussian random numbers, algorithms exist which produce a Gaussian distribution of pseudorandom numbers.
2.5 Prosthetic Devices

A prosthetic is an artificial substitute or replacement of a part of the body, and can be designed for function, cosmetic reasons, or both. In most cases there are at least five different options from which patient or doctor can choose: cosmetic prosthesis, body action prosthesis, myoelectric prosthesis, hybrid prosthesis, or specific prosthesis. This project focuses solely on myoelectric prostheses. (Galiano, Montaner, & Flecha, 2007)

![Figure 11 - Examples of prosthesis products from Liberating Technologies Inc.](image)

A myoelectric prosthesis is actuated through small electrical motors, which in turn are controlled by EMG signals. EMG signals are generated prior to the generation of muscle force, so by detecting these signals on remaining muscle tissue of the limb, it is possible to command the myoelectric prosthetic to move. In this case, the muscle acts as a biological amplifier for neural signals, which are too small to accurately detect in vivo. The EMG is the summation of
multiple MUAPs and is usually detected with surface electrodes. Since these signals are so small, the recorded EMG signal is amplified and processed so that it can be used to control the prosthesis. The quality of the surface electrode will make a difference in the recorded EMG signal. (Galiano et al., 2007)

Surface electrodes are non-invasive, and are therefore the preferred method of detecting EMG signals for electrical prosthesis control. Unfortunately, because they do not record the signal directly at the point of origin, certain considerations must be made in order to avoid contamination of the signal. One of the most significant challenges with surface EMG recordings is the presence of crosstalk. Crosstalk occurs when signals from active nearby muscles are detected in addition to the signal from the muscle of interest. Crosstalk presents a problem with EMG-controlled prostheses because the crosstalk-contaminated signal may not accurately represent the strength of the contraction at a given recording site. (Merletti & Parker, 2004)

Figure 12 illustrates how drastically the signal can change in both the time domain and frequency domain based on placement of the recording electrodes. Locating an optimal recording site can be a major problem for prosthetists, as they must measure the signal at multiple recording sites to find the strongest signal to use for prosthesis control.
When a surface electrode is placed on the skin, the physiological properties of skin and the conductivity of the metal electrodes produces an interface between the two that is inherently noisy. The interface is complex and has a capacitive impedance whose resistive (R) and capacitive (C) components are current and frequency dependent. The metal surface in contact with the skin will force the area under it to become equipotential, thereby modifying the skin potential distribution in the neighboring area. This modification is too complicated to be described analytically. (Merletti & Parker, 2004)
Chapter 3: Project Strategy

The purpose of the initial stages of the design process described in this chapter was to add specificity and details to the initial project description to further define the problem. Information gaps in the initial client statement were filled through research and client interviews. The objectives, constraints, and functions of the design were listed in order to draft a revised client statement. Once these key design parameters and specifications were identified, a plan for completion of the project was drafted.

3.1 Initial Client Statement

The initial client statement for this project was to “...develop a handheld, portable device that can serve as a realistic substitute for the human body…” and “to research the problem, then design, develop, test, and deliver a device that meets the sponsor’s needs.”

A client interview was conducted on the first day of work on this project in order to gather more details to better define the ultimate project goal. This interview, along with further communication with the sponsor yielded the following overall “wish list” for the device: a small, hand-held, battery powered device that outputs a realistic, amplitude-adjustable EMG-like signal with additive 50/60 Hz simulated power line interference on two channels. A multiplicative ramp function for the output signal was also desired. The device also needed to have a practical electrode interface. This “wish list” was used to define the design parameters and to create a detailed revised client statement discussed in the sections that follow.
3.2 Design Parameters: Objectives, Constraints, and Functions

To further define the goals of any design problem, the key objectives, constraints, and functions must be defined. Objectives describe what the device must “be,” i.e. safe or reliable. Constraints describe the strict limits within which the design should fall, most commonly, budget and scheduling restrictions. Functions describe what the device should “do.”

Through communication with the client, the following design parameters were established. The device must be sustainable, user friendly, reliable, low-power, safe, easy to manufacture, and versatile. The device must output a realistic EMG signal with adjustable amplitude, additive simulated power line interference, and a multiplicative ramp function. The device must also interface with multiple electrode types easily. Status indicators should clearly display the output and power status of the device. The device must meet size restrictions; in this case, it must be handheld and portable. The device must be completed on budget and on schedule. In this case, the device must be manufacturable for less than $1000, although the development budget can exceed this amount. The completion date is March 2010. These parameters are organized in Table 1 below. An objectives tree can be found in Appendix A.
Table 1 - Design parameters: objectives, constraints, and functions

<table>
<thead>
<tr>
<th>Objectives</th>
<th>The device must be...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Sustainable</td>
</tr>
<tr>
<td>2.</td>
<td>User friendly</td>
</tr>
<tr>
<td>3.</td>
<td>Reliable</td>
</tr>
<tr>
<td>4.</td>
<td>Low-power</td>
</tr>
<tr>
<td>5.</td>
<td>Safe</td>
</tr>
<tr>
<td>6.</td>
<td>Easy to manufacture</td>
</tr>
<tr>
<td>7.</td>
<td>Versatile</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constraints</th>
<th>The device must meet these restrictions...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Size: must be handheld</td>
</tr>
<tr>
<td>2.</td>
<td>Budget: must cost under $1,000 to manufacture</td>
</tr>
<tr>
<td>3.</td>
<td>Schedule: must be completed by March 2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functions</th>
<th>The device must do the following...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Output a realistic EMG signal with adjustable amplitude</td>
</tr>
<tr>
<td>2.</td>
<td>Include additive noise in output signal</td>
</tr>
<tr>
<td>3.</td>
<td>Include a multiplicative ramp function in output signal</td>
</tr>
<tr>
<td>4.</td>
<td>Interface with multiple electrode types</td>
</tr>
<tr>
<td>5.</td>
<td>Include indicators for power and output status</td>
</tr>
<tr>
<td>6.</td>
<td>Battery powered</td>
</tr>
</tbody>
</table>

3.3 Revised Client Statement

Based on information gathered through communication with LTI and the design parameters defined in the previous section, the following revised client statement was developed: “To design, develop, test, and deliver a device that will serve as a realistic substitute for the electromyogram (EMG) signals of the upper limbs. The device must be portable, handheld, and powered by batteries that are readily available. The two-channel output signal must consist of a simulated EMG with amplitudes ranging from 10 \( \mu V_{pp} \) to 20 \( mV_{pp} \) in the 20 Hz to 200 Hz frequency range. The signal must also have additive and selectable 50/60 Hz simulated power line interference with amplitudes up to 20 \( mV_{pp} \), and a multiplicative ramp function to modulate the signal. The system must have status indicators to confirm that it is operational. The output
electrodes must interface with those commonly used in prosthesis design. The device must also be designed with manufacturability in mind and should cost under $1000 to build, including labor. The project must be completed before March, 2010.”
Chapter 4: Critical Paths

In order to streamline the design process of the EMG simulator, critical paths in the design were assessed individually. Three critical paths were identified: signal generation methodology, hardware platform, and signal delivery system. Design alternatives were proposed for each critical path and the feasibility of these proposed designs for use in the final product was then tested. The results of these feasibility studies and experiments allowed critical design decisions to be made. The design phase culminated in the final design of the prototype, which is discussed in “Chapter 5: Final Design.”

4.1 Signal Generation Methodology

The generation of an EMG-like signal was an important first step in the design process for the EMG simulator. There were many methods of generating such a signal, two of which are outlined and compared in this section.

4.1.1 Design Alternatives

Any simulation trades a certain amount of realism for the sake of implementation, and this device was no exception. Based on the literature review, the use of a correlated Gaussian random process was considered as a method of signal generation. The purpose of this phase of the design process was to determine if the approximation of the EMG as a correlated Gaussian random process was acceptable for this application; that is, whether a correlated Gaussian random process was sufficiently similar to a physiologically realistic signal. The alternative to using a correlated Gaussian random process is an algorithm that more closely resembles how the surface EMG signal is generated in muscle physiology, i.e., the sum of many randomly spaced action potentials.
4.1.2 Feasibility Studies

In order to compare the two signal generation methods (physiologically realistic versus Gaussian), a MATLAB script was created which produced two signals, one physiological and one purely Gaussian. The code for these scripts can be found in Appendix B.

The physiological signal was generated in a similar fashion to the generation of the EMG in muscle tissue, using the summation of (uniformly) randomly spaced simulated motor unit action potential spikes to generate a signal which was then bandpass-filtered. The MATLAB code created to implement this algorithm consisted of three custom functions: makeMUAP(), makeTrain(), and physiologicEMG().

The makeMUAP() function took as a parameter the distance from the recording site to the muscle fiber, which was randomly generated from a uniform distribution over the interval of 0.5 to 2 in the calling function. Units used in these functions were arbitrary. A muscle fiber was simulated by specifying a length (500 units in this case) down which the action potential would travel. The positive portion of the MUAP was created by looping from 1 to the length of the simulated muscle fiber. Assuming that the signal strength at the recording site was directly proportional to the distance from the electrode to the action potential, the following equation was used to simulate the signal:

\[
MUAP1 = \frac{1}{\sqrt{distance^2 + \left(\frac{fiberlength}{2} - 1\right)^2}}
\]

Equation 1 - Calculation of positive-going portion of simulated EMG signal (represented by MUAP1). The variable “distance” represents the randomly generated distance from the recording site to the muscle fiber, and “fiberlength” represents the total length of the fiber.
The signal (MUAP1) was then zero-padded on both sides (1000 zeros on each side). Next, the negative-going portion of the MUAP was simulated by inverting MUAP1, and shifting it to the right by 150 units to create MUAP2. The complete MUAP was created by adding MUAP1 and MUAP2 together.

The function makeTrain() took a complete MUAP (as created by makeMUAP()) as a parameter. The function then concatenated 10 of these MUAPS together at random intervals to create the motor unit action potential train.

The main function, physiologicEMG(), first created the overall surface EMG signal by summing 100 random MUAPTs. The signal was then conditioned with bandpass filtering (4th order butterworth, 20 Hz – 200 Hz passband), and plotted, showing its time domain representation, power spectrum, and distribution. Note that these data were normalized to three standard deviations from the mean before plotting so the plots could be more easily compared. These plots are shown in Figure 13.
The Gaussian signal was produced using only a Gaussian random number generator and a bandpass filter (2\textsuperscript{nd} order butterworth, 20 Hz – 200 Hz passband). The resulting plots are shown in Figure 14 for comparison. Like the physiologic algorithm, the data were also normalized to three standard deviations from the mean.
Since the above data were generated only through simulations, they were compared to real EMG data for further validation. The following plots shown in Figure 15 are real EMG data (constant force, constant posture contraction) processed in the same manner as the simulated data. These data were also normalized to three standard deviations from the mean.
4.1.3 Conclusion

Based on visual inspection of the two signals generated, it was concluded that the use of a correlated Gaussian random process to approximate the EMG signal was acceptable for this application. This conclusion was reached based on the similarity between all three signals in terms of their time domain and frequency domain representations, as well as their distributions. The major feature that stands out among the three sets of data is the comb-like appearance of the power spectral density of the physiological signal, which was the result of the perfect geometry of the simulation; that is, certain frequencies were canceled out due to the relative timing of
identical action potentials. The real EMG signal nor the Gaussian EMG signal do not exhibit this behavior.

4.2 Hardware Platform

Another consideration important for signal generation, beyond the algorithm used, was the platform on which it was implemented. The design could be done through either analog or digital electronics, and could be implemented by modifying pre-existing devices (such as an iPhone or netbook), or by designing a new, original device. These alternatives and their related test results are detailed in this section.

4.2.1 Design Alternatives

Many different approaches for the signal generation were considered on a conceptual level. There were many preexisting devices on the market that could easily accomplish the task of EMG signal generation, however, they have drawbacks, often related to size, complexity, cost, and power requirements that make many such devices less than ideal for this application. For this reason, custom platforms (both digital and analog) were also viable options. Overall, five different alternatives were considered: analog signal generation with custom electronics, digital signal generation with the MSP430 microprocessor, digital signal generation with an ARM microprocessor, an Apple iPod Touch, and a netbook.

4.2.1.1 Analog Signal Generation

The characteristics of an EMG signal are comparable to that of random noise (white noise) band limited so that the signal only has frequency content within 20 Hz to 420 Hz. Therefore, one possible design of the EMG simulator using only analog electronics was to create
Johnson – Nyquist noise (thermal noise) with a resistor, and then filtering this thermal noise to contain frequency content in the range of 20 Hz - 420 Hz. Note that, at the time of test, the bandwidth for the EMG signal was to be 20 Hz - 420 Hz, and is discussed as such in this section. By the end of the project, the EMG bandwidth had changed to 20 Hz – 200 Hz.

Thermal noise is an electronic noise signal generated by the thermal agitation of electrons within an electrical conductor at equilibrium (Nyquist, 1928). Theoretically, thermal noise signals are random, infinite-bandwidth signals that maintain a flat power spectral density; this is what is meant by “white” noise. If thermal noise could be successfully generated, then passing this thermal noise through a bandpass filter (20 Hz – 420 Hz) would theoretically create a realistic approximation to an EMG signal.

Thermal noise is an electronic noise signal generated by the thermal agitation of electrons within an electrical conductor at equilibrium (Nyquist, 1928). Theoretically, thermal noise signals are random, infinite-bandwidth signals that maintain a flat power spectral density; this is what is meant by “white” noise. If thermal noise could be successfully generated, then passing this thermal noise through a bandpass filter (20 Hz – 420 Hz) would theoretically create a realistic approximation to an EMG signal.

4.2.1.2 Digital Signal Generation with MSP430

The majority of the options considered for signal generation were digital, due to the low cost and high precision of digital devices. One such device considered for use in this project was the Texas Instruments MSP430F449 microcontroller and an accompanying Olimex development board (shown in Figure 17). The MSP430 offered a relatively low cost and low power implementation with respect to other digital implementations, but was also much slower (8MHz) and did not support native 32-bit or floating-point operations. An advantage unique to using the

---

Figure 16 - Thevenin equivalent of thermal noise across a resistor, consisting of a resistor and a noise source
MSP430 was its familiarity; everyone on the design team had been exposed to the use of the MSP430F449 in particular by taking a course in digital design at WPI.

Figure 17 - Olimex MSP430-449STK2 Development Board (http://www.olimex.com/dev/msp-449stk2.html)

4.2.1.3 Digital Signal Generation with ARM Processor

One possible alternative to using the MSP430 is to use a more powerful processor with ARM architecture. The NXP LPC2138 is one such processor that supports native 32-bit calculations at speeds up to 60 MHz. It does not include hardware support for floating-point calculations, but it can run at fast enough clock speeds that simulating floating point calculations in software is likely possible. The LPC2138, like the MSP430, is available on many full-featured development boards that could serve as the platform for the project, such as the version shown below in Figure 18, also from Olimex. This board has many useful features, including an LCD display, five buttons, a buzzer, and an LED.
4.2.1.4 iPod Touch

Early on in the project, the iPod Touch was considered as a possible platform for its flexibility and portability. There is no doubt that the device has ample processing power to generate the relatively simple signal required by this application. The iPod’s operating system has readily available development kits and is highly customizable. The iPod platform could potentially reduce manufacturing costs since the device can be purchased for around $200 and requires no additional assembly. However, the manufacturing involved with the output stage and electrode could counteract these savings.

4.2.1.5 Netbook

One of the early platform design alternatives considered was a netbook PC. Netbooks are small, lightweight laptops, typically with relatively long battery life and slow processor speeds. A netbook implementation would provide the advantage of easy availability and familiar user interface, as well as relatively low price. However, users would have to charge a netbook
regularly or plug it into a wall socket to use it, and custom electronics would be required for interfacing with the electrodes on a prosthesis.

4.2.2 Feasibility Studies

Of the five design alternatives (analog, MSP430, ARM, iPod Touch, and netbook) proposed for the signal generation mechanism, feasibility studies were performed on two. Both analog signal generation and digital signal generation with the MSP430 needed to be tested before moving forward due to uncertainties regarding the design. Analog electronics are easily influenced by noise (60 Hz line noise and other noise), so the analog signal generator needed to be tested to prove that it could provide a stable, reliable signal that was relatively free from interference. The MSP430 needed to be tested due to the fact that it is a very low power processor, and concerns were raised about whether or not it was powerful enough for this application.

4.2.2.1 Analog Signal Generation

The analog design for the project held the promise of maintainability, inexpensive components, and sustainability. By proving the analog circuit as feasible, the analog design would define itself as major contender for the final steps in the design. The following sections will follow the testing and procedures that were followed to prove the analog option as feasible and a significant design alternative.

4.2.2.1.1 Calculation of Expected Root-Mean-Square (RMS) Noise Level Based on Resistance

In order for thermal noise to be considered as a viable option for signal generation, it must satisfy certain criteria. Recorded RMS values of thermal noise must match theoretical values. Recorded thermal noise values must follow a Gaussian distribution. Frequency content
of the signal should remain relatively flat within the EMG bandwidth (20 Hz - 420 Hz). To determine if the analog signal generator would meet these requirements, a series of hardware tests were performed.

Expected RMS values for thermal noise were calculated using Equation 2.

\[ v_n = \sqrt{4 \times k_B T R \times \sqrt{\Delta f}} \]

An analog circuit (see Appendix C) was created that generated thermal noise across a resistor, amplified the resultant signal by a factor of 100,000, and finally filtered the signal to contain frequency content only between 20 Hz to 420 Hz. This circuit, when implemented correctly, would generate a noise signal with a known \( V_{RMS} \) across a set bandwidth. The thermal noise’s \( V_{RMS} \) was found to be directly proportionally to the resistor value, and frequency bandwidth was determined by a lowpass filter and highpass filter in series to define the low and high frequency cutoffs.

The design team sought to prove that thermal noise could be properly utilized with a set procedure. The first step was to verify the characteristics of the analog circuit. An analog test circuit was built following the specifications of the schematic (see Appendix C) and inputting a known signal from a function generator. An oscilloscope at the output of the circuit found that signal was correctly amplified by 100,000 and was filtered to contain frequency content only between 20 Hz to 420 Hz.

With the test circuit working as expected, the next step was to experiment with different resistors to see if thermal noise could be generated. Four different resistor values were used: 10 M\( \Omega \), 1 M\( \Omega \), 100 k\( \Omega \), and 10 k\( \Omega \). The 10 M\( \Omega \) resistors were choosen as the largest value because
it was the largest value readily available, and 10 kΩ resistors were chosen as the smallest value because anything smaller than 10 kΩ resistors generate thermal noise too small to overcome the inherent noise generated by the AD620B instrumentation amplifiers used in the circuit. Before testing the four different resistor value categories, expected $V_{\text{RMS}}$ values were calculated for each using Equation 3. It is important to remember that each of the two AD620Bs used in the circuit generate an input noise, and the resultant AD620B noise must be added to the generated thermal noise as $V_{\text{RMS}}$ before the input signal was amplified at each gain stage. Table 2 shows the calculated thermal noise $V_{\text{RMS}}$ values and their respective resistors. Why and how the analog test circuit was designed is discussed more in the following section.

Equation 3 - $V_{\text{RMS}}$ calculation after one gain stage requires consideration of input noise from the AD620B. $V_{\text{RMS}1}$ is the $V_{\text{RMS}}$ value after the first stage gain. $V_{\text{RMS}2}$ is the $V_{\text{RMS}}$ after the second stage gain, and $V_{\text{RMS}_{\text{total}}}$ is the $V_{\text{RMS}}$ value at the output.

\[
V_{\text{RMS}1} = 1000 \times \sigma_{\text{Total}} \\
\sigma_1 = \sqrt{4 \times K_B \times T \times R \times \Delta f} \\
\sigma_2 = \frac{9nV}{\sqrt{Hz}} \times \sqrt{\Delta f} \\
\sigma_{\text{Total1}} = \sqrt{\sigma_1 + \sigma_2} \\
V_{\text{RMS}2} = 100 \times \sigma_3 \\
\sigma_3 = \sqrt{V_{\text{RMS}1} + \sigma_2} \\
V_{\text{RMS}_{\text{total}}} = V_{\text{RMS}1} + V_{\text{RMS}2}
\]
Table 2 - True resistor values and their corresponding expected RMS value after amplification

<table>
<thead>
<tr>
<th>Measured Resistor Value (MΩ)</th>
<th>Expected RMS (mVrms) After gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.97</td>
<td>780.8</td>
</tr>
<tr>
<td>.982</td>
<td>257.2</td>
</tr>
<tr>
<td>.0997</td>
<td>93.7</td>
</tr>
<tr>
<td>.00979</td>
<td>41.2</td>
</tr>
</tbody>
</table>

4.2.2.1.2 Experimental Setup

Thermal noise was generated by the resistor connected across the positive and negative terminals on the AD620B instrumentation amplifier. With the resistor shorted across both terminals, most interference, such as 60 Hz line noise, was removed. This resistor configuration is only possible because the AD620B has an internal reference as seen below in Figure 19. Generation of the thermal noise was the only time that the negative reference of the AD620B was not pulled to ground.

Figure 19 - AD620B with internal ground reference. R is the resistor that generates noise, RG is set to 49 Ω for a gain of 1000.

53
Thermal noise generated without any gain stage was calculated with Equation 2, and is shown below in Table 3.

Table 3 - Theoretical resistor values and their corresponding expected RMS values

<table>
<thead>
<tr>
<th>Theoretical Resistor Value (MΩ)</th>
<th>Expected RMS (μV_{RMS}) Before gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>8.11</td>
</tr>
<tr>
<td>1</td>
<td>2.57</td>
</tr>
<tr>
<td>0.1</td>
<td>0.811</td>
</tr>
<tr>
<td>0.01</td>
<td>0.257</td>
</tr>
</tbody>
</table>

As seen from the table, the V_{RMS} values of thermal noise generated directly from these resistors have expected V_{RMS} values that are only in the μV_{RMS} to nV_{RMS} range. This range of values is too small to be detected by a conventional oscilloscope, and in order for all four resistor categories to be measured, it was determined that the thermal noise would be amplified by a factor of 100,000. Selected for its low input noise, AD620Bs were used to implement a total gain of 100,000 in two separate stages in series. The generated thermal noise signal was passed through the first gain set at a factor of 1000, and the resultant amplified signal was passed to the second stage with a gain of 100. This setup amplified the signal by a total of 100,000 (Refer to Figure 20 for a depiction of the gain process). Note that DC offset was removed from the instrumentation amplifiers with a passive highpass filter between the two gain stages.
Once the signal had been amplified to recordable levels, the final step was to ensure that there would only be frequency content between 20 Hz and 420 Hz. To do this, the amplified signal was passed through a second-order highpass filter (implemented with Sallen-Key topology), with a 20 Hz cutoff frequency, and a second-order lowpass filter (also implemented with Sallen-Key topology), with a 420 Hz cutoff frequency. The governing equations for these Sallen-Key filters, as well as other commonly used circuit topologies in this project can be found in Appendix D. Second order filters were selected as a first order filter did not have sharp enough frequency cutoffs.

To ensure that the signal generated satisfied the three criteria listed at the beginning of this section, two sets of data were collected and analyzed. The purpose of the first test (“Reproducibility Test”) was to determine the repeatability of thermal noise generation by testing multiple resistors of the same value. Testing was done with up to three different resistors for each of the four different resistor values. Once repeatability was verified by comparing RMS values across different resistor samples, a second test (“Stability Test”) was conducted to monitor stability of the circuit over time. For the stability test, data were collected from a single resistor at different points in time over the course of one day. Time between data recordings were determined randomly, and four to five sets of data were recorded for each of the four
resistor values. The final testing procedure was done for each of the four resistor values, again checking to see that the RMS value remained constant. It was from this second set of data that the probability distribution of the signal was checked by displaying recorded data as a histogram, and frequency bandwidth was verified with a power spectrum plot.

4.2.2.1.3 Results

The reproducibility test (see section 4.2.2.1.2 Experimental Setup), yielded data (see Table 4) that showed we could create a precise noise signal for each resistor value, but the accuracy of the readings were very low. In some instances, measured $V_{\text{RMS}}$ for the resistors yielded errors of near 100%. Fortunately, generated noise signals were precise, and therefore the accuracy of the device could be fixed with the addition of a gain stage to be calibrated during production. The most important result found from the data was that the analog test circuit was able to consistently generate a noise signal proportional to the resistance used to generate thermal noise.

<table>
<thead>
<tr>
<th>Resistor Value (MΩ)</th>
<th>Expected RMS ($V_{\text{rms}}$)</th>
<th>Average RMS (mV_{rms})</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.97</td>
<td>169.128</td>
<td>571.5</td>
<td>.5511</td>
</tr>
<tr>
<td>.982</td>
<td>128.31</td>
<td>179.7</td>
<td>.1683</td>
</tr>
<tr>
<td>.0997</td>
<td>99.501</td>
<td>87.5</td>
<td>.0879</td>
</tr>
<tr>
<td>.00979</td>
<td>81.121</td>
<td>29.3</td>
<td>.0282</td>
</tr>
</tbody>
</table>

The stability test (see section 4.2.2.1.2 Experimental Setup), once again yielded results that still held high error, but remained consistent. This implies that the signal generated
sufficiently behaves in an inaccurate, but more importantly predictable manner over time (see Table 5).

Table 5 - Data from stability test of analog signal generator (see Appendix E for full data measurements)

<table>
<thead>
<tr>
<th>Resistor Value (MΩ)</th>
<th>Expected RMS (V&lt;sub&gt;rms&lt;/sub&gt;)</th>
<th>Average RMS (mV&lt;sub&gt;rms&lt;/sub&gt;)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.97</td>
<td>780.8</td>
<td>571.5</td>
<td>99%</td>
</tr>
<tr>
<td>.982</td>
<td>257.2</td>
<td>179.7</td>
<td>99.9%</td>
</tr>
<tr>
<td>.0997</td>
<td>93.7</td>
<td>87.5</td>
<td>99.9%</td>
</tr>
<tr>
<td>.00979</td>
<td>41.2</td>
<td>29.3</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

Also with the following histogram (Figure 21) and power spectral plots (Figure 22) taken for each of the four resistor values, it can be asserted that the signal is fairly random and power stays relatively equal over the frequency range from 20 to 420 Hz. The recorded signals followed a relatively Gaussian distribution, and as seen below the 10 MΩ resistor yielded the highest amplitude and most defined power spectral density plot. As the resistance was lowered, there was less power within the desired frequency range explaining why it was more difficult to differentiate the thermal noise from the reference. These resultant observation were reasonable as the amplitude of the thermal noise power was proportional to resistance.
Figure 21 - Distributions of data generated with varying resistor values
Figure 22 - Power Spectral Densities of data generated with varying resistor values and after passing through a 20 Hz to 420 Hz band pass filter

The data recorded from these tests show that it is plausible to use the analog noise generation circuit to create a random signal with frequency content from 20 Hz to 420 Hz. Generation of thermal noise on a resistor can be used as an analog random signal generator.

4.2.2.2 Digital Signal Generation with MSP430

Several algorithms were considered for use in a digital implementation of Gaussian signal generation. Algorithms were compared using MATLAB and the MSP430F449, where appropriate. The purpose of this experiment was twofold: first, to decide on an algorithm for
use in a digital implementation of the product; and second, to determine whether the MSP430 was powerful enough to carry out the operations required in a reasonably efficient implementation.

4.2.2.2.1 Experimental Setup

All of the algorithms considered were tested either in MATLAB or on the MSP430F449 development board from Olimex. The Olimex board was used due to the familiarity and immediate availability of the platform at WPI. Due to the prefabricated nature of the Olimex board, no experimental preparation was needed at the hardware level. Software was developed using the demo.c file for the MSP430F449 Olimex board (available via WPI) as a template. Candidate algorithms were appropriated from *Numerical Recipes in C* (Press et al., 1992) or custom-developed.

4.2.2.2 Results

The digital implementation testing can be split into three phases: first, MATLAB verification of candidate algorithms; second, implementation on the MSP430 of the candidate algorithms; and finally, testing of the MSP430's capabilities as a processor for use in the project. The first phase was necessary to ensure that the candidate Gaussian random number generators were able to produce sufficiently Gaussian results for use in this application. The second phase implemented the verified algorithms on the MSP430 and checked the run speed of the random number generator on the MSP platform. Phase three checked whether it was feasible to use the MSP430 as the platform for the final product based on projected run times of various components of the final design.
The two candidate algorithms tested were the Box-Muller transform and simple bandpass filtering. The Box-Muller transform produces a Gaussian random variable from two uniformly distributed random numbers using an algebraic procedure detailed below, with random variables X and Y being of uniform distribution. The Box-Muller transform is shown in Equation 4 below:

$$Z = \sqrt{-2 \cdot \ln X \cdot \cos 2\pi Y}$$

In contrast, the finite impulse response bandpass filter is merely a sum of weighted uniform random numbers, X, like so:

$$Z = \sum_{i=0}^{N} a_i \cdot X_i$$

where the coefficients $a_i$ are determined by the characteristics of the filter and N is the order of the filter. The Box-Muller transform produces a "white" Gaussian, with equal energy at all frequencies, while the filter approach produces a signal with the characteristics of the filter. Notably, MATLAB simulations of the two approaches are shown in Figure 23 and Figure 24 for comparison. The code for creating these plots can be found in Appendix F.
Figure 23 - The Box-Muller transform. Gaussian distribution overlaid in red.

Figure 24 - Linear filtering. Gaussian distribution overlaid in red.
Visual inspection reveals that the Box-Muller transform produces a slightly more purely Gaussian signal. The linear filter has two advantages over the Box-Muller transform, however: first, it is computationally simple; and second, it shapes the output signal (advantageous since a requirement of the EMG signal in its final application is a bandpass characteristic, so the Box-Muller output would need to be filtered anyway). Both approaches were considered for application on the MSP430.

Having determined the two candidate algorithms to be sufficient for use in this application, the next step was to translate them into C code for use on the MSP430 (available in Appendix G). The code was then tested with a 4 MHz clock to determine its run speed. The results are tabulated below:

Table 6 - Run speed of candidate algorithms on the MSP430F449 microcontroller running at 4 MHz

<table>
<thead>
<tr>
<th>Operation</th>
<th>Run speed (in cycles per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd-order FIR filtering</td>
<td>1928</td>
</tr>
<tr>
<td>Box-Muller transform</td>
<td>831</td>
</tr>
</tbody>
</table>

Given the relatively slow speed of the Box-Muller transform (less than half as fast as FIR filtering) and the relatively low requirement for strict adherence to a Gaussian distribution, as well as the necessity of further filtering of the Box-Muller variable, it was clear that the FIR filter was the better of the two candidate algorithms in this application.

The final phase of testing on the MSP430 regarded whether the MSP would be sufficient for use in this application. These tests, unlike the algorithm tests, were performed using the 8 MHz clock, the maximum available on the Olimex board, and consisted of testing individual operations. An estimate of the number of operations required for each essential piece of the final
program was then used to decide whether the MSP430 would run well above the required Nyquist frequency of approximately 400 Hz and have room for further modification should it be required. The tabulated results for the operation speed test are shown in Table 7, using a clock speed of 8 MHz:

Table 7 - Run-times for various calculation types on the MSP430F449 microcontroller running at 8 MHz

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed (in cycles per second)</th>
<th>Time elapsed per operation (in microseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit addition</td>
<td>266,640</td>
<td>3.75</td>
</tr>
<tr>
<td>32-bit multiplication</td>
<td>98,757</td>
<td>10.1</td>
</tr>
<tr>
<td>32-bit division</td>
<td>15,967</td>
<td>62.7</td>
</tr>
<tr>
<td>16-bit addition</td>
<td>399,957</td>
<td>2.5</td>
</tr>
<tr>
<td>16-bit multiply</td>
<td>163,250</td>
<td>6.1</td>
</tr>
<tr>
<td>16-bit division</td>
<td>43,006</td>
<td>23.3</td>
</tr>
<tr>
<td>16-bit modulus</td>
<td>42,548</td>
<td>23.5</td>
</tr>
</tbody>
</table>

Based on the measured run time of each operation, an estimate was produced for the run time of the final program. The estimate was based on individual estimates for each block, which are shown in Table 8.
Table 8 - Estimated run speeds of program components.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Component operations</th>
<th>Run time (microseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMG generation</td>
<td>4 16-bit multiplies</td>
<td>69.05</td>
</tr>
<tr>
<td></td>
<td>1 32-bit multiply</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 16-bit adds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 32-bit add</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 16-bit division</td>
<td></td>
</tr>
<tr>
<td>Additive sine wave</td>
<td>4 32-bit additions</td>
<td>15</td>
</tr>
<tr>
<td>Ramp function</td>
<td>4 32-bit multiplies</td>
<td>40.4</td>
</tr>
</tbody>
</table>

The total estimated run time was 124.45 microseconds, which equates to a running frequency of 8,035 Hz, well above the Nyquist frequency of 400 Hz. Although this estimate does not include the DAC output or user input handling routines, it is likely that these routines (and any further modification required) could be built into the 1 millisecond of processing overhead. However, given the age of the MSP430, it was replaced in later iterations of the design by a newer, faster processor for sustainability reasons.

4.2.3 Conclusion

These tests indicated that both the analog signal generation and the digital signal generation on the MSP430 were feasible. Since no single platform for signal generation had yet to stand out among the rest, a quantitative design analysis was done. The results of this analysis are shown in section 4.4 Design Decisions of this chapter.
4.3 Signal Delivery System

A robust analog signal delivery system was a critical component of the EMG simulation device. The design challenge presented by this system was providing a reliable interface between the simulator electrodes and the prosthesis electrodes that minimizes noise. This task was made more difficult given the fact that the electrodes were approximately one meter away from the signal generator, during which the signal was corruptible by power line and other interference. The simulated EMG signal is especially prone to noise corruption since its amplitudes are very small: the smallest signal generated by the simulation device (as specified by the client) was approximately 10 µV_{pp}.

4.3.1 Design Alternatives

Two methods of signal delivery were considered. The first and simplest method was to step the signal amplitude down to the final level directly at the output of the device. The second method was to transmit a higher-amplitude signal (less corruptible by noise), and step it down to the final amplitude at the electrode interface. For example, if the output device was a standard DAC, the output amplitude may have been in the range of ± 2.5 V, when only 400 mV_{pp} (note that this value had changed to 40 mV_{pp} by the end of the project, but 400 mV_{pp} was estimated at the time of this test, and was the value used in this section) was desired for the final signal (simulated EMG + simulated noise, projected value at time of test). These two configurations are shown in more detail in Figure 25.
Configuration 1 was simpler from both a design and assembly perspective. It allowed for the output stage to be integrated into the device itself, which reduced the number of components that would reside in the prosthesis interface. The electronics at the output stage could include resistors, operational amplifiers, or other integrated circuits. All of these components take up space and some even require power. Housing powered components of the output stage in the prosthesis interface would then require running power through the cable along with the signal. Thus, if experimentation indicated that configuration 1 was feasible, it would be used in the final design.

4.3.2 Feasibility Studies

Two methods of accomplishing low-noise signal delivery were considered. The configurations differed only in the location of the output stage (i.e. the fractional gain circuit) that stepped the output amplitude down to EMG levels. Configuration 1 (see Figure 25) was
markedly simpler than the alternative, in that it required no additional electronics to be housed in
the electrode itself, but was potentially too corruptible by noise. This experiment was conducted
to determine the minimum signal amplitude that could be transmitted over a 1 meter shielded
cable without becoming overly corrupted by noise. If this minimum was at or lower than the
minimum amplitude that was to be generated by the simulation device (at the time of test, 20
\( \mu V_{pp} \)), then it could be asserted that configuration 1 was a feasible design.

### 4.3.2.1 Experimental Setup

An experimental circuit was designed and tested to determine the feasibility of
configuration 1. For the purposes of this experiment, a digitally generated signal was simulated.
The experiment modeled the system by simulating the digital-to-analog converter and
transmitting the simulated EMG signal (in this case, a sine wave at multiple frequency bands that
lie within the bandwidth of the EMG signal) over a one meter shielded cable. The signal was
received and amplified by an instrumentation amplifier and filtered to simulate the receiver in the
prosthesis. A flow-chart of the experimental setup is shown in Figure 26. Note that this
experimental setup was not meant to suggest design decisions made about the final design of the
signal generator itself; it was constructed for this experiment to represent one possible
configuration in order to represent the conditions of the final device as closely as possible. The
primary purpose of this experiment was to test the feasibility of transmitting the signal at true
EMG amplitudes through a shielded cable. The detailed schematic of the circuit is shown in
Appendix H.
The DAC modeled by this experiment was the Texas Instruments DAC7632, which provided a 16-bit ± 2.5V bipolar (again, this had changed to unipolar by the end of the project) output. The smallest amplitude that could be generated by the DAC was calculated with the following equation:

\[
\text{minimum DAC amplitude} = \frac{5 \text{Vpp}}{2^{16}} = 76 \text{μVpp}
\]

The EMG simulator was to output a signal as small as 20 μV pp (minimum EMG signal) and as large as 400 mV pp (maximum EMG + noise signal), so the output of the DAC must stepped down to these levels. Since the smallest signal that the Tektronix CFG-8219A function generator could output was 20 mV pp, a fractional-gain, inverting op-amp was used at the output of the function generator to step down the signal to proper DAC output levels. Thus, the function generator, in combination with the op-amp circuit, was used to represent the DAC for
this experiment. The mapping of function generator output to simulated DAC output is shown below in Table 9.

Table 9 - Mapping of function generator output to simulated DAC output.

<table>
<thead>
<tr>
<th>Function Generator Output</th>
<th>Simulated DAC Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Amplitude</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>3.04 V&lt;sub&gt;pp&lt;/sub&gt;</td>
</tr>
<tr>
<td>Minimum Amplitude</td>
<td>152 mV&lt;sub&gt;pp&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
<td>152 µV&lt;sub&gt;pp&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

* This value is irrelevant since we are only concerned with finding the minimum signal amplitude that can be transmitted.

Note that although the function generator was capable of producing amplitudes as low as 20 mV<sub>pp</sub>, the minimum amplitude used was 152 mV<sub>pp</sub>; this adjustment was made to simplify calculations and make the fractional gain an even 1/1000 since the signal was being reduced to 152 µV<sub>pp</sub>.

At the output of the simulated DAC, the signal was split in order to create the differential signal used by the prosthesis. This process will be discussed in further detail in “Chapter 5: Final Design.” One of the signals was routed through an inverter, which was a unity gain inverting op-amp circuit. Next, both signals were routed through a voltage divider (with 510 KΩ and 75 KΩ resistors) which served two purposes. First, it stepped down the DAC output to the final EMG levels. The gain required to step down the output of the simulated DAC was 1/7.6. Second, it provided an appropriate output impedance of 510 KΩ, so that if the prosthesis electrode malfunctioned and drew more current than usual, it would not operate due to a voltage drop across the output impedance. Since the impedance at the skin-ele...
just needed to be large enough to create a voltage drop if excessive current was drawn. The mapping of simulated DAC output to overall system output can be found in Table 10.

Table 10 - Mapping of simulated DAC output to overall system (voltage divider) output

<table>
<thead>
<tr>
<th>Simulated DAC Output</th>
<th>Overall System (Voltage Divider) Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Amplitude</td>
<td>3.04 V_{pp}</td>
</tr>
<tr>
<td>Minimum Amplitude</td>
<td>152 µV_{pp}</td>
</tr>
</tbody>
</table>

The differential signal was then transmitted over a distance of one meter through a shielded cable. The cable was obtained from Cooner Wire (part number NMUF4/30-4046 SJ), and had four conductors and a shield. The added mechanical flexibility of this wire makes it especially suitable for medical applications. This Cooner cable has been specified for use in the final product.

Experimental simulation of a prosthesis signal conditioning circuit was necessary for two reasons. First, the EMG level signals were not detectable by the oscilloscope due to their extremely small amplitudes, thus they were amplified before they were read by the scope. Second, since the noise which would likely corrupt the miniscule signals has an extremely wide bandwidth, a lowpass filter was necessary to limit the observed signal to the frequency ranges that are dominant in EMG signals.

The signal at the output of the mockup EMG simulator (see Figure 26) was then amplified by a factor of 10000 by an Analog Devices AD620 instrumentation amplifier after transmission. This signal was also lowpass filtered using a second-order Butterworth filter, implemented with the Sallen-Key topology with a cutoff frequency of approximately 400 Hz. The output of the filter was monitored with a Tektronix TDS210 oscilloscope. Oscilloscope
screenshots were taken to capture experimental data. Photographs of the experimental setup are shown in Appendix I.

4.3.2.2 Results

Once the circuit was constructed and tested, an experiment was conducted to determine if a sufficiently small signal could be transmitted without excessive noise corruption. Four different signal amplitudes (20 $\mu$V$_{pp}$, 15 $\mu$V$_{pp}$, 10 $\mu$V$_{pp}$, and 5 $\mu$V$_{pp}$) were each tested at two frequencies (400 Hz and 150 Hz). Note that these are the amplitudes and frequencies of the actual signal that was presented to the shielded cable. One-hundred fifty hertz was chosen because that is where most energy in an EMG signal lies, and 400 Hz was tested because this frequency is approaching the maximum frequency present in an EMG signal. The 20 $\mu$V$_{pp}$ signal appeared very clean at the output (of the simulated prosthesis signal conditioning circuit), and retained its sinusoidal shape at both frequencies. These signals can be seen below in Figure 27 and Figure 28.

Figure 27 - Transmission of 20 $\mu$Vpp signal at 150 Hz.
- Channel 1 (yellow): output of function generator
- Channel 2 (blue): transmitted signal
- Channel 3 (purple): inverted transmitted signal
- Channel 4 (green): Amplified differential signal

Figure 28 - Transmission of 20 $\mu$Vpp signal at 400 Hz.
- Channel 1 (yellow): output of function generator
- Channel 2 (blue): transmitted signal
- Channel 3 (purple): inverted transmitted signal
- Channel 4 (green): Amplified differential signal
The signal of interest in these figures is channel 4 (green), the received, amplified, and filtered signal. Channels 2 and 3 show the two components of the differential signal as they were transmitted; no signals are visible because they were too small to be detected by the oscilloscope without external amplification. The smallest signal tested, 5 µVpp also performed very well. As shown in Figure 29 and Figure 30, it was also very clean and sufficiently sinusoidal. Note that the amplitude of the received signal is twice the amplitude of the transmitted signal due to the fact that the two differential signals were subtracted from each other. Also notice the lower frequency oscillations present in the signals, which was most likely caused by slight 60 Hz interference. The rest of the data can be found in Appendix J.

4.3.3 Conclusion

This experiment indicated that the simplest method of signal delivery, stepping down the DAC output voltage at the output of the DAC (rather than at the electrode interface) was feasible. The smallest signal that was to be generated by the EMG simulator at the time of test
was a 20 µV_{pp} signal, which was within the range shown in the above figures. The fact that the 5 µV_{pp} signal was successfully transmitted indicated that there was a certain amount of margin in the device, and that it was not pushing the limits of what was physically possible. These results indicated that, while the second method (as described at the beginning of this section) may very well produce even better results, it is not necessary or worth the added cost and complexity since the first method produced such favorable results. Note that the repeatability of the results of this test is dependent upon the electromagnetic environment of the experimenter or user. The tests were performed in an electrical engineering teaching lab at WPI, likely a very electrically noisy environment.

4.4 Design Decisions

Before a preliminary design was developed, the results of the feasibility studies presented above were carefully examined to determine the best possible solution for each critical path. In some cases, the results of the test were clear cut, pointing to one obvious solution. In others, further quantitative design analysis was required to determine the best solution.

4.4.1 Signal Generation Methodology

The results of the signal generation methodology testing (physiologic generation vs. uncorrelated Gaussian random process generation) were conclusive. Both were very similar in every aspect (time domain, frequency domain, and distribution) to the real EMG data, indicating that both methods are feasible. Since using a correlated Gaussian random process was both conceptually and computationally simpler, it was the clear winner in this category and was used in the final design.
4.4.2 Signal Generation Platform

Five options were presented as possible platforms for signal generation: analog signal generation, digital signal generation with the MSP430, digital signal generation with an ARM processor, an Apple iPod Touch, and a netbook. The group was confident that these last three options, the ARM, iPod Touch, and netbook, all had plenty of processing power for this application, so they were not tested. However, it was unclear if analog signal generation was possible, and whether or not the MSP430 had enough computational power for the application. Thus, these two alternatives were tested.

Since both of these options proved feasible, all five of these alternatives were still viable options for the final design. Thus, a quantitative design analysis was done to determine which was the best for this application. First, a Pairwise Objective Comparison was done for each objective identified in Chapter 3: Project Strategy to determine which were the most important. Then, based on the scores given to each objective, a weight was assigned to each. Then a Numerical Evaluation Matrix was created, which lists each constraint and objective and quantitatively determines which design alternative best fits all objectives.

The objectives included in the Pairwise Comparison Chart (shown in Table 11) were taken from Table 1, and include sustainability, user friendliness, reliability, low power consumption, safety, ease of manufacture, and versatility. These objectives are shown across the top and down the side of the table. Starting with the first row, each objective to the left was compared with the objective across the top. If the objective to the left was more important than the objective above, a “1” was placed in the corresponding box. Otherwise, a “0” was placed in the box. Since objectives cannot be compared to themselves, an “X” was placed in any box comparing two of the same objectives. The total score of each row was tallied and placed in the
last column. The higher the value in this column, the more important the objective. Finally, each objective was given a weight by adding 1 to the total, and calculating its proportion out of all of the adjusted totals.

Table 11 - Pairwise Objective Comparison chart for quantitative design analysis

<table>
<thead>
<tr>
<th>Objective</th>
<th>Sustainable</th>
<th>User Friendly</th>
<th>Reliable</th>
<th>Low Power</th>
<th>Easy to Manufacture</th>
<th>Versatility</th>
<th>Total</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sustainable</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0.18</td>
</tr>
<tr>
<td>User Friendly</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.11</td>
</tr>
<tr>
<td>Reliable</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0.21</td>
</tr>
<tr>
<td>Low Power Consumption Safe</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0.14</td>
</tr>
<tr>
<td>Safe</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.25</td>
</tr>
<tr>
<td>Easy to Manufacture</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>0.07</td>
</tr>
<tr>
<td>Versatility</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Next, a Numerical Evaluation Matrix (shown in Table 12) was created to calculate a quantitative value that described how well each design alternative met the objectives overall. The leftmost column contained all of the objectives (used in the Pairwise Objective Comparison) as well as all design constraints (size, cost, and time). Across the top of the chart, each design alternative was listed. For each alternative, a ranking was given for how well it met each constraint and objective. This ranking was on a scale of 0 (meets objective poorly) to 1 (meets objective well) for objectives, and a simple “yes” or “no” for constraints. If an alternative did not meet one of the design constraints, it was automatically disqualified and not given a final score.
Table 12 - Numerical Evaluation Matrix for quantitative design analysis

<table>
<thead>
<tr>
<th>Numerical Evaluation Matrix</th>
<th>Criteria</th>
<th>Weights</th>
<th>Analog (MSP430)</th>
<th>Digital (ARM)</th>
<th>iPod Touch</th>
<th>Netbook</th>
</tr>
</thead>
<tbody>
<tr>
<td>C: Size</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>C: Cost</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>C: Time</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>O: Sustainable</td>
<td>0.18</td>
<td>0.9</td>
<td>0.4</td>
<td>0.8</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>O: User Friendly</td>
<td>0.11</td>
<td>0.8</td>
<td>0.4</td>
<td>0.7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>O: Reliable</td>
<td>0.21</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>O: Low Power Consumption</td>
<td>0.14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>O: Safe</td>
<td>0.25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>O: Easy to Manufacture</td>
<td>0.07</td>
<td>0.5</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>O: Versatility</td>
<td>0.04</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>1.00</td>
<td>0.86</td>
<td>0.76</td>
<td>0.89</td>
<td>0.79</td>
<td>X</td>
</tr>
</tbody>
</table>

The analog design met all constraints. Using surface mount technology, a small printed circuit board could be developed that contains all necessary signal generation and output components, and could easily be put into a handheld device. A preliminary cost analysis (see Appendix K) suggested that an analog device could be manufactured for approximately $500, well under the $1000 limit. The group felt that this design was feasible to complete before the March 2010 deadline. The design was given a high score for sustainability, since analog components are unlikely to change in the near future. Since the devices complexity and feature list was limited because the design was restricted to analog components, it was given a very low versatility score. This restriction would not have allowed the addition of the “pulse mode” with configurable timing parameters, a feature that the sponsor added later in the project. The limited feature list would make the device very simple to operate, a fact that is reflected in the high user friendliness score. There was no reason to think that an analog design would be unreliable (if
designed correctly), unsafe, or overly power-hungry, so it was ranked very highly in all of these categories. Finally, this design was given a low ease of manufacture score since it is likely to have more components than any other design, requiring more manual labor. The overall score for the analog design was 86%.

The next implementation that was scored was the digital design with the MSP430 microcontroller. This design easily met the size constraint due to the small size of the Olimex MSP430-449STK2 development board. The cost and time constraints were also met (see the preliminary cost analysis in Appendix K). This design received a low score for sustainability, since it relies on the ongoing availability of the Olimex development board. User friendliness was not a strong suit either, since the development board only contains four buttons and a one line, seven character LCD display. All digital designs were rated slightly below the analog design for reliability since software often contains bugs that take a great deal of time to surface. The MSP430 was designed specifically for ultra-low power applications, thus it was rated very highly in this category. All digital implementations were rated higher than the analog design in ease of manufacture since more of the computation is done in software, and thus requires less assembly. The MSP430 implementation was given a 50% in versatility, because it is possible to add more features in software, but there is a limit to the computational power available. The overall score for the digital design with the MSP430 was 76%.

The highest scoring design alternative was the digital implementation with the ARM microprocessor, with 89% overall. Not only did the design meet all constraints (the cost analysis was considered to be very similar to that of the MSP430 implementation, see Appendix K), but it received either the same or higher scores than the MSP430 in all categories. This design was considered to be more sustainable since it uses newer, more powerful technology. It is more user
friendly, because the Olimex LPC-MT-2138 development board has five buttons and a two line, 32 character display, making menus and commands much easier to read and navigate. The ARM was also considered to be a very low-power processor that was safe and reliable. This design also scored very highly in the versatility category due to its more powerful processor core and thus the ability to easily add more features in software.

The final two design alternatives, the iPod Touch and netbook, were less successful due to limitations of these existing devices. While the iPod Touch meets all constraints, it scored poorly in many important categories. It received a low score in the sustainability category, since consumer devices are updated often, potentially requiring a great deal of work to maintain compatibility. It also does not have an easily replaceable battery, which is an important design requirement. The mini netbook was not scored, because it did not meet the constraints (not handheld). For these reasons, the iPod Touch and netbook were not serious contenders for the final design.

In conclusion, the digital design using the ARM processor scored the highest because it most closely met the design requirements. It struck a good balance of power, ease of use, and a compact size that is not achieved by the other design alternatives. For these reasons, it was used in the final design.

### 4.4.3 Signal Delivery System

The results of the feasibility study of the signal delivery system were also clear cut. There was no evidence to suggest that the output signal needed to be stepped down at the electrode. Since doing so would complicate the electrode interface, configuration 1 (stepping down the voltage at the output of the DAC, see Figure 25) was used in the final design.
Chapter 5: Final Design

In this chapter we present the final design of the EMG simulator. The organization of this chapter follows the path of the signal through the EMG simulator, from its input by the user to its eventual delivery at the prosthesis electrodes. Section 1, System Architecture, discusses the architecture of both the software and hardware involved in the implementation of the final design of the EMG simulator, and provides an overview of the path taken by the signal. Section 2, User Interface, discusses the handling of user input by the EMG simulator. Section 3, Signal Generation, discusses the production of a digital signal with the specifications provided by the user. Section 4, Signal Delivery discusses the conversion of the digitally-generated signal to an analog format that is interpretable by the prosthesis electrodes, as well as the interface between the EMG simulator and the electrodes.

The Olimex LPC-MT-2138 served as the platform for the design, providing the user interface (LCD display and buttons), as well as the signal generation hardware. At the heart of the development board was the NXP LPC2138 ARM processor. This native 32-bit microprocessor can run at speeds up to 60 MHz, which provided more than enough power for this application, leaving plenty of headroom for addition of more features in the future. The schematic for the Olimex LPC-MT-2138 development board is shown in Appendix L.

The signal was generated in software on the ARM, but it was output through the Texas Instruments DAC8564, a 16-bit, 4 channel, 0 V – 2.5 V output DAC. The DAC used for the preliminary design came mounted on an evaluation board that included input, output, and power headers for easy connection to the Olimex board and the auxiliary electronics board. Data were transmitted from the Olimex board to the DAC through the serial peripheral interface (SPI) of the ARM processor. The schematic for the DAC evaluation board is shown in Appendix M.
The final functional block in the design was the auxiliary electronics board. This board consisted of custom electronics that accomplished four main tasks: stepping the voltage output of the DAC down to EMG levels, controlling the amplitude of the EMG (when in “manual” mode) via knob potentiometers connected to the analog-to-digital converter inputs on the Olimex board, monitoring the battery voltage (also through the ADC inputs on the Olimex board), and providing a stable output reference voltage to the prosthesis electrode. For this prototype design, the circuit was constructed on a solderless protoboard. The schematic for this auxiliary electronics board is shown in Appendix N.

The prototype was powered by a battery pack consisting of 8 AA batteries (required to provide 12 VDC to the Olimex board). In addition to providing 12 V to the Olimex board, it also powered a 5 V regulator on the auxiliary electronics board which provided power to the op-amps. The connections between the components are shown in the flow chart in Figure 31.
Figure 31 – Block diagram of final design of EMG simulator
5.1 System Architecture

The EMG simulator consists of two physically distinct but functionally intertwined components: the digital, software-driven microcontroller unit (MCU), and the auxiliary hardware used to translate the digital output of the MCU into an analog signal appropriate for the prosthesis electrodes. These two components communicate using either analog to digital or digital to analog conversion (ADC and DAC respectively), as shown in Figure 32. Each of these components has an underlying architecture used in its design. In this section, the overall function of the device is discussed, as are the software architecture of the MCU and the hardware architecture of the auxiliary electronics.

5.1.1 Device Functionality

The EMG simulator was originally specified to output a EMG signal with additive 50/60 Hz simulated power line interference. During the design process, three distinct modes of operation were requested: a manual adjustment mode, in which the user can select EMG amplitudes manually using knobs on the face of the device; a ramp mode, in which users set a maximum value and the EMG ramps up to and down from that value; and a pulse mode, in which the EMG is turned on and off periodically with period and duty cycle selectable by the user.

The final prototype of the EMG simulator produced an EMG-like signal (band-limited correlated Gaussian random process) as discussed in earlier chapters. It also implemented three modes of operation and sinusoidal additive simulated power line interference, which could be enabled by the user. Relevant parameters such as amplitude are were adjustable by the user.
5.1.2 Software Architecture

The software used to drive the MCU had two layers of complexity: a background loop which ran at the maximum speed allowed by the hardware and dealt with user interactions, and an interrupt-driven foreground loop which occurred every 900th of a second and dealt with the generation and output of the EMG signal. These two loops communicated with each other via the global “state” of the machine, as defined by a number of variables which were accessible by either loop but were changed by the background loop.

The background loop, which was implemented by the main() function of the C program on the MCU, had two purposes: first, to receive user input via the five face buttons of the device and alter the machine state accordingly; and second, to output to the device’s LCD screen a menu based on the current state. The menu architecture was linked to the functions of the buttons, so that buttons had different functions in different states (although there was enough similarity so that buttons were intuitive to use).

The foreground loop, implemented by an interrupt service routine, took priority over the background loop and was used to produce the EMG signal. During this loop, depending on the...
machine state set by the background loop, a particular EMG signal was generated. It was then output to the DAC (a hardware peripheral) using the serial peripheral interface (SPI) protocol.

In order to help visualize the software architecture, Figure 33 presents a flowchart of the program. A list of all code functions, along with a brief description, can be found in Appendix O.
5.1.3 Auxiliary Hardware Architecture

There were four distinct sections of the auxiliary hardware used in the EMG simulator: the manual input knob potentiometers, which provided users a method of analog input; the battery meter, which provided information about remaining battery life; the DAC, which was mounted on an evaluation board and interfaced with the MCU via SPI; and the output stage, which put the DAC output signal through a fractional gain and output it to the prosthesis electrodes. The schematic for these auxiliary electronics can be found in Appendix N, and the derivation for the gain stage can be found in Appendix P.

5.1.3.1 Manual adjustment knobs

Two knob potentiometers were present on the auxiliary hardware board. Each was used for user input when the system was in the “manual” operation mode; they control EMG amplitude on channels 1 and 2, respectively. These potentiometers were connected across $V_{DD}$ (3.3 V) and ground, with the variable center taps providing input to ADC pins on the MCU. The potentiometers allowed for a variable voltage between 0 and 3.3 V to be presented to the ADCs as the user rotated the knobs, with each voltage (and each corresponding deflection angle of the knob) corresponding to a certain number within the MCU.

5.1.3.2 Battery meter

Like the knob potentiometers, the battery meter provided input to the ADC pins on the MCU. However, it used fixed resistances in a voltage divider configuration, thus varying the ADC’s output value only as the supply voltage changed. The varying numerical output was used to provide the user with information about the voltage output of the battery, so that it was clear when a new battery was needed. The battery meter and knob potentiometers drew approximately
38 µA of current altogether. A circuit diagram of the knob potentiometers and battery meter is shown in Figure 34.

![Circuit Diagram](image)

**Figure 34 - Knob potentiometer and battery meter circuit**

### 5.1.3.3 Digital to Analog Converter

Whereas the knob potentiometers and battery meter served as inputs to the MCU, the DAC served as its output. The DAC’s function was to convert the digitally generated signal coming out of the MCU into an analog signal with a quasi-continuous range of voltages. The DAC selected for this application, the TI DAC8564 was interfaced with the MCU using the standard SPI architecture, and had an output range of 0 to 2.5 V, with a least-significant bit (LSB) corresponding to 38 µV (16-bits of resolution).
5.1.3.4 Output stage

Since the output of the DAC was between 0 and 2.5 V and the desired output range of the simulated signal with noise is between -20 mV and 20 mV, a fractional gain (attenuation) and level-shifting were required before the signal was delivered to the electrode. The attenuation was 0.016 V/V, corresponding to 40 mV/2.5 V. Attenuation is accomplished through use of op-amps in the inverting gain configuration. The level-shifting procedure used took advantage of the battery powered EMG simulator’s floating ground, and is described in more detail in section 5.3.

To better visualize the role of each component in the analog hardware, Figure 35 provides a flowchart depicting each component and its relation to the MCU and electrodes.

![Figure 35 - Hardware flowchart](image-url)
5.2 User Interface

The user interface of the EMG simulator consisted of both input and output, implemented in hardware by the two manual adjustment knobs and five face buttons (input), and the LCD screen (output). From a software standpoint, the user interface was represented by a set of state variables which held information about which menu was being displayed. Menus were navigated through use of the buttons, which changed the state variables accordingly.

Menu navigation was intended to be intuitive, with a hierarchical organization. From a “start” menu, users could select either channel 1 or channel 2 to enter a submenu for that channel. This submenu allowed the user to select the mode of operation for that channel, and further submenus could be selected to set the amplitude, etc. for the features of that mode of operation. A flowchart of these menus is shown in Figure 36.
Figure 36 - Menu navigation diagram. Arrows on the left indicate that the menu is the final option in the sequence; arrows on the top indicate that there are further submenus.
The menus were navigated using the five face buttons of the device. On the start menu, the face buttons functioned differently than in other modes. The top button cycled through noise enabling – it enabled channel 1, then channel 2, then channels 1 and 2 noise with each successive press. Pressing the top button a fourth time disabled noise. The left and right buttons allowed users to enter the channel 1 and channel 2 submenus, respectively. Finally, the center button was used to start or stop output from the device. The bottom button was reserved for potentially recalling user settings, should that feature ever be incorporated into the device. In all menus other than the start menu, the face buttons had the following functions: top returned to the previous screen; left and right adjusted parameters; down entered the next submenu; and center returned to the start screen. Example navigation is shown below in Figure 37. The “start” menu is shown at the top, from which the left button was pressed to access the “channel 1 settings” submenu. From the “channel 1 settings” submenu, the bottom button was pressed to access the next submenu, “channel 1 mode”.

Figure 37 - Example navigation of menu system.
A picture of the start screen is shown below in Figure 38. In the top left is the battery meter (A), which displays the current voltage of the battery as a percentage of the nominal voltage (12 V). Below the battery meter is the run status indicator (B), which displays either “RUN” if the device is outputting a signal, or “OFF” if the device is not active. To the right of the battery meter and run status indicator, the upper line corresponds to channel 1 and the lower line corresponds to channel 2, as indicated by the channel labels (C). Immediately to the right of the channel labels are the noise indicator (D) and mode indicator (E), which tell the user if 50/60 Hz simulated power line noise is enabled (N: enabled, blank: disabled) and what operating mode (M: manual, R: ramp, P: pulse) has been selected for each channel. To the far right of the screen is the amplitude indicator, which displays the current amplitude (in V_{pp}) of the output signal.

The software implementation of the menu system began with the one-time generation of the menu tree at the start of the program’s execution by the function \texttt{buildmenus}. This function created menu structures for each menu in Figure 36, each of which contained a state variable as well as pointers to five other menus – one corresponding to each input button. These button pointers could be manually set to point to other menus, or could be left in their default state (as null pointers) if the respective button would instead perform a function such as adjusting amplitude. This function returned the “start” menu node of a superstructure which looks like Figure 36 in its layout.

Once the “start” menu had been linked to the other menus in the navigation tree, the user could begin navigation. By pressing buttons, users changed the variable \texttt{hit\_button}, which had two purposes. If the current menu had a non-null button pointer corresponding to \texttt{hit\_button}, the current menu was changed to the menu corresponding to that pointer. If the corresponding button pointer was null, however, the function \texttt{buttonfunc} was called. Depending on the value of \texttt{hit\_button}, as well as the value of the current menu’s state variable, \texttt{buttonfunc} performed a function specific to the menu and button that were passed to it. For example, if the current menu was an amplitude adjustment menu, and the hit button was the rightmost button, \texttt{buttonfunc} would increase the global variable corresponding to amplitude.

After the pressed button has been handled, the screen was updated to reflect any changes made by the user. \texttt{Menufunc} did exactly that, and it was called at the beginning of the next iteration of the background loop (it came before the user pressed a button to display the initial “start” screen). \texttt{Menufunc} took as an argument the current menu state, and output to the LCD the appropriate menu using the \texttt{write\_lcd} function.
5.3 Signal Generation

After the EMG simulator was activated for output, once every 1/900\textsuperscript{th} of a second the software foreground loop would execute (via a hardware interrupt), producing a signal and outputting it to the DAC. The characteristics of the output signal were determined by the global variables set by the user via the menus discussed in the previous section. However, the signal was always composed of up to two components: an EMG-like signal and/or a 50/60 Hz sinewave to simulate power line interference. This section details how these signals were produced, discussing the EMG signal first, the power line interference second, and the output to the DAC third.

There was a complex process that took place before each new EMG value could be output to the DAC. Note that this process, as well as others carried out in software, can be streamlined to greatly improve efficiency in both execution time and code size. Due to the time constraints of this project, these processes were implemented in the most conceptually straightforward way at the time that they were implemented in order to expedite progress on the project.

The EMG signal used by the simulator was generated by filtering uniform random numbers through a bandpass filter, as discussed earlier in this document. These raw EMG values were stored in the signed 16-bit range (-32768 to 32768) in a global array. The values in this array were updated by a call to a single function. Since this function would filter the uniform random deviates, a global filter buffer was also necessary. Random numbers were produced by the linear congruential random number generator discussed in *Numerical Recipes in C*, which consists of the operation:
Equation 7 - Linear congruential random number generator

\[ X[n] = (1664525 \cdot X[n - 1] + 1013904223) \mod 2^{32} \]

In order for this random number generator to work, \( X \) must be an unsigned 32-bit integer, since this random number generator exploited the way that unsigned integer overflows are dealt with in C. Before the random number was fed to the filter, however, it was right bitshifted by 16 (to take the 16 high order bits to retain randomness, while bringing the value within the unsigned 16-bit range) and shifted down by \( 2^{15} \). This operation yielded a uniformly distributed value that was within the desired signed 16-bit range, but still had the randomness of a 32-bit random number generator. The level shifting discussed above was done to prevent integer overflow.

Operations were done on signed integers in the 16-bit range whenever possible, so that the result were within the range of the 16-bit DAC. Additionally, the reduction in resolution from 32 to 16 bits meant that certain values would repeat every \( 2^{16} \) iterations, but the pattern would not repeat until the \( 2^{32} \) iterations.

This newly generated and processed random number was then filtered by a 17th order integer FIR band-pass filter with cutoffs at 20 and 200 Hz designed by MATLAB’s \texttt{fir1} function, as per Equation 8, shown below.

Equation 8 – Integer filtering of random numbers to produce EMG

\[
\begin{align*}
&+ 608 \cdot X[n - 6] + 17290 \cdot X[n - 7] + 26625 \cdot X[n - 8] + 17290 \cdot X[n - 9] \\
&+ 608 \cdot X[n - 10] - 6436 \cdot X[n - 11] - 3354 \cdot X[n - 12] + 93 \cdot X[n - 14] \\
&- 408 \cdot X[n - 15] - 399 \cdot X[n - 16]
\end{align*}
\]
In this equation, $E$, the output EMG-like signal, was a 32-bit unsigned integer.

Since the lack of floating-point processing on the LPC2138 necessitated an integer filter, the floating-point filter taps were scaled up to increase precision rather than truncating them. This yielded a filter with a gain much greater than one, as is shown in the magnitude response in Figure 39.

![Figure 39 - Magnitude response of integer filter.](image)

This meant that, although the values fed into the filter were within the 16-bit range of the DAC, the values at the output of the filter were much larger (32-bit range). A few steps were taken to remedy this. First, an absolute maximum and minimum of 2140200000 and -2140200000 (three standard deviations from the mean, as determined by MATLAB), respectively, were placed at the output of the filter. Then, $2^{31}$ was added to the filter output (to make all values positive before bit shifting, to prevent errors resulting from numbers stored in two’s compliment format). Then, the 16-bit right bitshift occurs, then the value was shifted down by $2^{15}$. The filtered value was then within the signed 16-bit range. The final step taken on the new raw EMG value was a division by 2 via a one bit shift to the right. This brought the value within exactly one half of the
full scale range of the DAC, leaving the other half for 50/60 Hz simulated power line interference.

The raw EMG signal alone, however, was not sufficient for output to the DAC. The user could also elect to add 50 or 60 Hz noise to the output signal, in which case a number was taken from a single period sinewave table generated at program startup and added to the EMG signal to produce an output. Two sinewave tables were generated, one for 50 Hz (which was 900/50 = 18 samples in length) and one for 60 Hz (which was 900/60 = 15 samples in length). These tables contained sinewave values ranging from -32768 to 32768 (the 16-bit range). While the simulator was running, a counter variable that incremented every 1/900th (and resets to 0 when it reaches 899) tracked the position within one period of the sinewave. If 60 Hz noise was selected, the noise values were updated by looping through (using the modulo operator so the index never exceeded 15) the 60 Hz sine table (incrementing every 1/900th of a second). If 50 Hz noise was selected, the noise values were updated by looping through the 50 Hz sine table with a modulo of 18. The value from the sine table was then divided by 2 (via a one bit shift to the right) so that the sinewave utilized exactly half of the full scale range of the DAC (like the EMG values).

For each channel, two output signals were generated: one with an EMG and interference, and another with an inverted EMG but non-inverted interference. This inversion allowed us to produce a differential EMG signal riding on a common-mode line noise, so that a differential amplifier (such as the amplifier in the LTI prosthesis electrode) would remove the common mode interference but not the EMG.

Each channel had a user-selectable mode. These three modes were manual, in which the user controlled EMG amplitude directly; ramp, in which the user set a maximum value and the EMG ramped up to and down from that value; and pulse, in which the EMG was turned on and
off periodically with period and duty cycle selectable by the user. In software, each mode was implemented by a different function during the final output stage.

The final output signal was generated by a call to the function that implemented scaling for the mode currently in use (for each channel). This function modulated the amplitude of the raw EMG and noise signals to the level set by the user, and added them if necessary (if noise was activated). Note that the amplitude of the EMG signal must be divided again by 2 to properly scale the amplitude so that it was seen correctly by a differential amplifier (which subtracts the two signals, thus doubling the amplitude). Since the full scale range of the DAC was no longer being used by the halved EMG, one quarter of the dynamic range (1-bit) was wasted. This should be fixed in future work with this project.

Once the final output (containing both EMG and noise in the signed 16-bit range) had been generated, one final step was required before it could be written to the DAC. The signal was shifted up by $2^{15}$, centering the signal around 32768 (half of the full scale range of the DAC) so that it could be output as an unsigned integer. Once this was done, all four values (positive and negative for each channel) were written to the DAC through the SPI interface.

5.4 Signal Delivery

After the signal reached the DAC, it was converted to an analog voltage between 0 and 2.5 V, centered at 1.25 V. This signal was too large for the electrode to meaningfully interpret, and was therefore be scaled down and level-shifted so that the full range of the output signal was between -20 mV and 20 mV, centered at 0 V (with respect to electrode reference). This fractional gain was accomplished by the use of op-amps in single-supply configuration with a fractional gain of 0.016 (in order to convert 2.5 V into 40 mV) as seen in Figure 40. In order to output a signal meaningful to the electrodes, however, there must be little or no DC offset at the
output. Because the signal was centered at 1.25 V as it came out of the DAC, some electrical level-shifting was required to present the electrode with a small DC offset.

Because the EMG simulator was a floating-ground device (i.e., not connected to earth ground), it was possible to provide the prosthesis electrode with a ground that was a nonzero voltage relative to our device. In this design, the ground provided to the prosthesis was 1.25 V from the EMG simulator’s perspective, half of the output of the DAC and approximately the DC offset of the output signal. Level-shifting by having a nonzero ground presented to the electrode allowed for the use of single-supply analog electronics to output what would appear to the electrode as a dual-supply signal with a relatively small DC offset.

With the floating ground technique in mind, the electronics involved in the output of the signal are relatively simple to understand. For each of the two outputs per channel (one with a non-inverted EMG, the other with an inverted EMG) there was a single supply operational amplifier in inverting gain configuration, with resistors selected to provide a gain of 0.016. The positive terminal of the amplifier was set to 1.25 V using a buffered voltage divider of the DAC’s 2.5 V reference, so that all amplification occurred around that voltage. The output could be modeled as an equivalent voltage source with a smaller AC component, but a DC component of 1.25 V. The output circuit and its voltage source equivalent are shown in Figure 40 and Figure 41, respectively. A derivation of the function of this circuit is shown in Appendix P.
The output stage consisted of four of the circuits shown above (two for each electrode), seen at the bottom of the schematic in Appendix N. In addition, 10 kΩ resistors were placed between the output of the op amp and the electrode as current limiters in the case of electrical failure. 10 kΩ restricted current output at 40 mV levels to 4 µA in the event of a short in the electrode, rather than the large amperage provided by the resistance of the electrode leads alone.

Each electrode received three inputs from the auxiliary hardware board. At its positive and negative terminals it received outputs from two of the fractional gain circuits discussed above, labeled (for channel 1) as ch1pos and ch1neg in Appendix N. Each electrode also received a 1.25 V reference voltage from the output of the TLV2770 buffer. The TLV2770 buffer circuit used the voltage division principle, using two 1 MΩ resistors to divide 2.5 V by 2, and is shown in Figure 42.
5.5 Conclusion

The final design prototype consisted of three functional blocks: the microprocessor development board, DAC evaluation board, and auxiliary hardware board. Power was supplied by 8 AA batteries connected in series to provide a 12 V source. The system accomplished three primary tasks: signal generation, user interface, and signal delivery.

The design presented in this chapter was a prototype (see Figure 43), and did not meet the original design requirement of being handheld. Because of its size, it also did not include an enclosure. The finished product, however, should incorporate the auxiliary hardware and DAC on a single printed circuit board (PCB) to reduce size requirements. A SolidWorks model of the finished product (with enclosure) is shown in Figure 44 and a cost analysis is presented in Table 13.
Figure 43 - Prototype of EMG simulator

Figure 44 - Concept of potential final product design of the EMG simulator
Table 13 - Cost analysis of final product.

<table>
<thead>
<tr>
<th>Part</th>
<th>Qty</th>
<th>Unit Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>16</td>
<td>$0.06</td>
<td>$0.96</td>
</tr>
<tr>
<td>Capacitor</td>
<td>5</td>
<td>$0.15</td>
<td>$0.75</td>
</tr>
<tr>
<td>TLV2774 Quad Op-Amp</td>
<td>1</td>
<td>$3.19</td>
<td>$3.19</td>
</tr>
<tr>
<td>TLV2770 Op-amp</td>
<td>1</td>
<td>$2.08</td>
<td>$2.08</td>
</tr>
<tr>
<td>TI DAC8564</td>
<td>1</td>
<td>$14.17</td>
<td>$14.17</td>
</tr>
<tr>
<td>Olimex LPC-MT-2138 Dev. Board</td>
<td>1</td>
<td>$89.73</td>
<td>$89.73</td>
</tr>
<tr>
<td>Knob Potentiometers</td>
<td>2</td>
<td>$10.00</td>
<td>$20.00</td>
</tr>
<tr>
<td>Printed Circuit Board</td>
<td>1</td>
<td>$100</td>
<td>$100</td>
</tr>
<tr>
<td>AA Battery Case</td>
<td>1</td>
<td>$1.84</td>
<td>$1.84</td>
</tr>
<tr>
<td>AA Battery</td>
<td>8</td>
<td>$0.40</td>
<td>$3.20</td>
</tr>
<tr>
<td>Battery Connector</td>
<td>1</td>
<td>$1.99</td>
<td>$1.99</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td></td>
<td></td>
<td><strong>$237.91</strong></td>
</tr>
</tbody>
</table>
Chapter 6: Design Verification

This chapter details the verification of the final design outlined in the previous chapter. Design verification required testing of the device to assure that it satisfactorily met all of the requirements specified at the onset of the project. Data were taken from the device itself and compared to their theoretical counterparts for verification. The revised client statement from Chapter 3: Project Strategy was revisited, and all items were checked for proper function in the final device.

The verification process required multiple stages to thoroughly test all aspects of the device. First the DAC (and indirectly, the microprocessor) and the analog output stage were tested. Signals were checked for accurate amplitude control at the output of the DAC, and were validated based on calculations that follow the steps taken in software to provide the final output. The analog output stage was tested to assure proper gain settings and frequency response. The EMG signal was then analyzed in MATLAB to confirm that the signal’s time domain representation, power spectral density, and distribution were consistent with both the real EMG data and the simulations shown in Chapter 4: Critical Paths. Finally, the modes of operation were tested by varying parameters and verifying proper response. The remaining items from the revised client statement can be verified through visual inspection of the device (i.e. the presence of proper status indicators).

Note that all tests presented in this chapter were done on channel 1 of the device. Since the channels operated identically, this was a safe assumption. However, general operation of channel 2 was verified.
6.1 DAC Output

The first tests performed on the device and detailed in this section confirmed proper generation of both the EMG and noise signals as they were outputted from the DAC. First, an amplitude was set using the LCD display and buttons on the device. Then, the expected output of the DAC was calculated based on the steps taken in software to condition the signal for it’s final output. The output signal was viewed on the oscilloscope, and the amplitude was verified, taking into account the gain of the test apparatus.

Tests were done (of both EMG and noise) at amplitude settings (via the controls on the device) of 10 µV pp, 100 µV pp, 1 mV pp, and 20 mV pp. Test results for the 10 µV pp and 20 mV pp (minimum and maximum) amplitude tests are shown in this chapter; all remaining results can be found in Appendix Q. Since these settings specify the amplitude of the signal at the output of the device as a whole (and NOT at the output of the DAC), the expected amplitude at the output of the test apparatus was calculated based on the following Equation 9:

\[
\text{expected amplitude (V)} = \frac{\text{amplitude setting (µV)}}{20000} \times 1.25 \times \text{test apparatus gain}
\]

This relation assures that the amplitude at the output of the device will correspond to the amplitude that was set, once it is put through the analog output stage (which has a fractional gain of 0.016).
It is important to note that the amplitudes in the results did not always exactly match the expected amplitude, but were satisfactorily close. This mismatch was due to the nature of correlated Gaussian random processes, and the fact that the absolute maximum value will not appear in every test. Also note that the blank area on the oscilloscope screenshots were a result of the window within which new samples were being acquired.

6.1.1 **EMG Amplitude Verification**

The test apparatus used for verifying EMG amplitudes from the output of the DAC consisted of one Analog Devices AD620 instrumentation amplifier, powered by ± 15 V from a desktop power supply. The gain for this amplifier was set to either 1000, 100, or 1, depending on the amplitude of the signal under test. The ground reference of the amplifier was connected to the reference voltage for the device (1.25 V). The differential signal from the device was presented to the inputs of the amplifier, and the output was monitored on the oscilloscope. A flowchart of this test apparatus is shown in Figure 45. The gain of the test apparatus is specified in the description of each test.

![Flowchart of test apparatus](image)

**Figure 45 - Test apparatus for DAC output test to verify EMG amplitude control**
With the EMG amplitude set at 10 µV_{pp}, the expected amplitude at the output of the test apparatus was calculated (based on Equation 9) to be 625 mV_{pp}, since the gain of the test apparatus in this case was set to 1000. The results are shown in Figure 46. The output amplitude of approximately 576 mV_{pp} was sufficiently close to the expected value; thus, this EMG amplitude at the DAC output was verified.

![Figure 46 - Test results of DAC output test for EMG amplitude control. 10 µV_{pp} input, approx 576 mV_{pp} output, test apparatus gain of 1000.](image)

With the EMG amplitude set at 20 mV_{pp}, the expected amplitude at the output of the test apparatus was calculated (based on Equation 9) to be 1.25 V_{pp}, since the gain of the test apparatus in this case was set to 1. The results are shown in Figure 46. The output amplitude of approximately 1.24 V_{pp} was sufficiently close to the expected value; thus, this EMG amplitude at the DAC output was verified.
Figure 47 - Test results of DAC output test for EMG amplitude control. 20 mV_{pp} input, approx 1.24 V_{pp} output, test apparatus gain of 1.

6.1.2 Noise Amplitude Verification

Since the noise signal is common mode (and the instrumentation amplifiers reject common mode signals), two amplifiers were used in this test apparatus. The test apparatus used for verifying noise amplitudes from the output of the DAC consisted of two Analog Devices AD620 instrumentation amplifiers, powered by ±15 V from a desktop power supply. The gain for these amplifiers were set to either 1000, 100, or 1, depending on the amplitude of the signal under test. The ground references of the amplifiers were connected to the reference voltage for the device (1.25 V). The common mode signal from the positive output was connected to one of the instrumentation amplifiers (positive input), while the signal from the negative output was connected to the other. Both negative inputs of the instrumentation amplifiers were connected to ground. The output of both amplifiers was monitored on the oscilloscope. A flowchart of this test apparatus is shown in Figure 48. The gain of the test apparatus is specified in the description of each test.
With the noise amplitude set at 10 µV_{pp}, the expected amplitude at the output of the test apparatus was calculated (based on Equation 9) to be 62.5 mV_{pp}, since the gain of the test apparatus in this case was set to 100. The results are shown in Figure 48. Based on the oscilloscope setting of 100 mV per division, it is clear that the amplitude of this signal is sufficiently close to the expected value of 62.5 mV_{pp}. Thus, this noise amplitude at the DAC output was verified.
Figure 49 - Test results of DAC output test for noise amplitude control. 10 µVpp input, test apparatus gain of 100.

With the noise amplitude set at 20 mVpp, the expected amplitude at the output of the test apparatus was calculated (based on Equation 9) to be 1.25 Vpp, since the gain of the test apparatus in this case was set to 1. The results are shown in Figure 50. Based on the oscilloscope setting of 1 V per division, it is clear that the amplitude of this signal is sufficiently close to the expected value of 1.25 Vpp. Thus, this noise amplitude at the DAC output was verified.
6.2 Output Stage

To test the analog output stage of the device, a 2.5 V_{pp} signal was presented to the inputs of the output stage from a function generator. This value of 2.5 V_{pp} was chosen because it represents the largest voltage that will be outputted from the DAC. Since the op-amp used in the output stage has linear gain, only this value was used to confirm the fractional gain of 0.016. However, multiple frequency values within the EMG bandwidth were tested to confirm the flat frequency response of the output stage. The frequency values tested were 20 Hz to 220 Hz in increments of 40 Hz. The minimum and maximum frequency values are shown in the text; the remaining tests can be found in Appendix R. The expected output value based on the gain of the output stage was calculated with the relation in Equation 10.

\textit{Equation 10 - Relation of the input amplitude to expected amplitude as measured from the output of the test apparatus for the output stage}

\[ \text{expected amplitude} = \text{input amplitude} \times 0.016 \times \text{test apparatus gain} \]
The test apparatus for this functional block consisted of the output stage, a desktop function generator, and a final gain stage. This final gain stage was necessitated by the very small fractional gain of the output stage that produces a signal too small to view on the oscilloscope without sufficient re-amplification. The AD620 instrumentation amplifier circuit from the previous test was used with a gain of 100 to serve as the final gain stage for this test. A block diagram of this test apparatus is shown in Figure 51.

![Block Diagram of Test Apparatus](image)

Figure 51 - Test apparatus for output stage verification

The test results for 20 Hz and 220 Hz are shown in Figure 52 and Figure 53, respectively. Notice that the amplitude of the signal shown in Figure 52 is slightly below the expected value (calculated using Equation 10) of 4 Vpp. This indicates that 20 Hz is on the edge of the passband of the output circuit. This slight attenuation at 20 Hz is acceptable since all of the subsequent tests (shown in Appendix R) yield highly accurate results, indicating an otherwise flat frequency response. Overall, these tests confirm the fractional gain of 0.016 and flat frequency response of the output circuit, indicating that it is fully functional.
Figure 52 – Output stage test. Input of 2.5 V_{pp} at 20 Hz.

Figure 53 – Output stage test. Input of 2.5 V_{pp} at 220 Hz.
6.3 Noise Frequency and Phase

The simulated common mode line noise generated by the device was tested for correct frequency and phase. For this test, the oscilloscope was connected directly to the positive and negative terminals of channel 1 at the output of the DAC. It was acceptable to perform this test in this manner since the proper operation of all functional blocks had already been verified. The device was set to manual mode, with the EMG amplitude turned down to zero so the only visible signal was simulated line noise. The amplitude of the noise was set to 10 mV. The common mode, simulated noise signals can be seen in Figure 54 and Figure 55, shown below. Their respective frequencies were verified by the frequency measurement on the oscilloscope screen.

![Simulated 50 Hz common mode line noise. Yellow signal is from the positive output terminal, blue signal is from the negative output terminal.](image)

Figure 54 - Simulated 50 Hz common mode line noise. Yellow signal is from the positive output terminal, blue signal is from the negative output terminal.
It is clear from the figures that the noise signals from the positive and negative terminals are in phase. This means that they appear as common signals to the myoelectrode, and will be eliminated (due to high common mode rejection). Thus, the proper frequency and phase performance of the simulated noise was verified.

### 6.4 Overall EMG and Noise Amplitude Control

Since the amplitude of the EMG and noise signals as measured from the output of the DAC were shown to correctly correspond to their respective amplitude settings on the device, and that the gain of the output stage is correct over the frequency range of importance, no further tests were necessary to prove that the amplitude control (for both EMG and noise) of the device overall is functioning properly. A few simple calculations demonstrated this fact. For example, for an amplitude setting of 10 µV<sub>pp</sub>, an application of Equation 9 (disregarding the test apparatus gain, since this does not exist in the final device) yields a DAC output of 0.000625 V<sub>pp</sub>. Then, an
application of Equation 10 (again, disregarding the final gain) yields an overall device output of 10 \( \mu V_{pp} \). Thus, proper amplitude control for both EMG and noise amplitude was verified.

## 6.4 Realism of EMG signal

In order for the EMG signal generated by the device to be considered realistic, it should be very similar to a real EMG signal in terms of time domain representation, frequency domain representation, and distribution. The signal should also be as “random” as possible, like a real surface EMG signal. To test these conditions, data were captured on a USB flash drive from the oscilloscope and imported into MATLAB. The apparatus for this test is shown in Figure 56. An LTI BE328 Myoelectrode Module was used for this test to simulate the conditions of real world use as closely as possible.

![Figure 56 - Test apparatus for EMG realism test](image)

These data were normalized for easier comparison with the real EMG data shown in Figure 15. The data collected from the device itself are shown in Figure 57.
Figure 57 - Time domain representation, power spectral density, and distribution of simulated EMG data taken from the device. Device settings: pulse mode, 10 mV pp EMG amplitude, 20 mV pp 60 Hz noise amplitude, 1000 ms duration, 100% duty cycle. Note that the 60 Hz simulated line noise did not appear in the data because it was rejected by the differential amplifier.

The scaling of the axes on these plots was kept as similar as possible to the scaling in Figure 15 to make them easier to compare. The only difference between the two was that the real EMG data contained 40 seconds of data sampled at 1000 Hz, while the test data from the device only contained 2.25 seconds (also sampled at 1000 Hz). This dissimilarity was due to the method of data collection. In order to achieve equivalent sampling rates of 1000 Hz, the window on the oscilloscope had to be set to display 250 ms per division, resulting in only 2.25 seconds of data.
The frequency domain representations are very similar between the two data sets, as both contain the most power between 20 Hz and 200 Hz. Both also show approximately Gaussian distributions. Note that fewer bins were used on the simulated data because there were far fewer data points than the real data. By comparing these two data sets, it can be stated that the simulated EMG signal from the device is satisfactorily similar to a real EMG signal in terms of time domain representation, frequency domain representation, and distribution.

The final requirement that needed to be fulfilled in order for the EMG to be considered satisfactorily realistic was randomness. The random number generator implemented in software on the ARM chip was a linear congruential generator (see Chapter 5: Final Design for more details). This was a 32-bit generator, meaning that it produced all values between 0 and $2^{32}$ in a pseudorandom pattern before repeating. The proper function of this generator was verified by comparing the first few values it produced (using the debugger in the integrated development environment) with the “correct” values listed in “Numerical Recipes in C.” Given that the generator was functioning correctly, the time before the sequence repeated was calculated. Note that since the output of the random number generator was scaled to the 16-bit range (to match the range of the DAC), repeated values would be produced every $2^{16}$ iteration. However, the pattern will not begin to repeat until the $2^{32}$ iteration. The time before signal repetition (based on the device’s sampling rate of 900 Hz) was calculated, as is shown in Equation 11.

\[
\text{Equation 11 – Calculation of time before signal repetition, } F_s = \text{sampling rate} \\
\]

\[
time \text{ before repetition (seconds)} = \frac{2^{32}}{F_s} = \frac{2^{32}}{900 \text{ Hz}} = 4772186 \text{ seconds}
\]

This result means that the signal will not repeat until the device has been running for 55.23 days continuously. This period is significantly better than a signal repetition time that is
on the order of milliseconds if a traditional function generator is used for prosthesis testing. Note that, since channel 1 and channel 2 of the device are statistically independent AND use the same random number generator, the repetition time for each channel is cut in half. Thus, the signal on each channel will repeat after 27.61 days of continuous operation.

These repetition times were also verified with a MATLAB simulation. The linear congruential generator was replicated in MATLAB, however, modulo operations were required (unlike in C) because MATLAB handles overflow of unsigned 32-bit integers differently than the C compiler. These times could be verified relatively quickly in MATLAB since it was capable of iterating through the generator much faster than the 900 Hz sampling rate of the device. This simulation verified the value calculated in Equation 11. The code can be found in Appendix S.

Thus, since the three key representations of the signal were satisfactorily similar to the real EMG, and the randomness of the signal was confirmed, it can be stated with confidence that the device produced a realistic surface EMG signal.

6.5 Modes of Operation

The device was programmed with three different modes operation: manual mode, pulse mode, and ramp mode. Manual mode took its input from the ADCs on the ARM processor and modulated the amplitude of the signal based on the position of the knob potentiometers. The device operating in manual mode throughout the full amplitude range (by turning the adjustment knob all the way up, then all the way back down) is shown in Figure 58. Note that all tests in this section were done with 20 mV_{pp} 60 Hz noise activated. It cannot be seen in these figures due to the common mode rejection of the LTI BE328 Myoelectrode.
Pulse mode generated EMG signals in short bursts, based on timing parameters set by the user. The adjustable parameters for pulse mode included amplitude, duration of pulse, and duty cycle. The device operating in pulse mode with a 10 mV_{pp} amplitude, 500 ms duration, and 25% duty cycle can be seen in Figure 60. Next, the duty cycle was changed to 50% while all other parameters were held constant. This setup is shown in Figure 60. Notice the decrease in the overall period. Finally, the period was lengthened to 1 s, and is shown in Figure 61. The proper function of the device with the variation of these parameters confirms proper function of pulse mode.
Figure 59 - Pulse mode, 10 mVpp amplitude, 500 ms duration, 25% duty cycle

Figure 60 - Pulse mode, 10 mVpp amplitude, 500 ms duration, 50% duty cycle
The third and final mode of operation was ramp mode. This would modulate the EMG signal in a ramp pattern based on a peak amplitude and period set by the user. The device operating in ramp mode with a 10 mVpp amplitude and a 2 s period is shown in Figure 62. Next, the period was increased to 4 s; these results are shown in Figure 63. It was clear from this figure that ramp mode was functioning correctly.
Based on the results of these tests, it was deemed safe to conclude that the overall device was functioning correctly, and that the design had been successfully verified.
Chapter 7: Discussion

The design team worked with the clients to determine what the final design should be: an EMG signal generator that satisfied a pre-defined set of constraints and objectives. The constraints for the design consisted of size, cost, and time (the deadline for the project was March 5th, 2010), while the objectives consisted of sustainability, user friendliness, reliability, low power consumption, safety, ease of manufacturing, and versatility. The final prototype was required to generate an EMG signal while satisfying all the constraints and objectives in order to be considered successful.

Constraints can be likened to restrictions; if they are not satisfied the design will not serve its purpose. The three constraints for the design were size, cost, and time; size was important because the product needed to be handheld for testing purposes, cost was important as the design team was allotted $1000 for research funding, and time was important as the group was given a time limit of seven months to deliver a final prototype. The design team was successful in satisfying all three constraints as the final prototype was delivered within the set time period of seven months, stayed well within our allotted research budget of $1000, and was designed to be used as a portable hand held diagnostic device (although the prototype violated this constraint).

With the constraints satisfied, the next set of considerations were the objectives or goals, that the final prototype was to satisfy. The list of objectives is as follows: user friendly, sustainable, reliable, low power consumption, safe, easy to manufacture, and versatility. Each of the objectives holds important weight in how the final design was approached, as the design team attempted to satisfy each objective in the best way possible.
One of the first objectives that the design group explored was versatility: how many different functions was the final design able to accomplish? The design group decided that the final prototype should output an EMG signal in one of three different modes. The first mode was manual mode, where the user was given full control of the EMG signal amplitude. In this mode, the user was able to adjust the amplitude of the signal in real time with the use of physical knobs found on the outside of the device. The second mode was pulse mode, which allows the user to define the duty cycle, period, and signal amplitude parameters and then continuously generate the defined signal once started. The pulse mode output a periodic signal of known amplitude with a given duty cycle to test how the myoelectric prosthesis reacted. The third and final mode was the ramp function. The ramp function also asked the user to define a peak signal amplitude, and once defined would generate an EMG signal that began at an amplitude of zero, increased to the peak and then returned to zero amplitude. This mode tested the range of amplitudes that could be detected by the myoelectric prosthesis. By creating three different signal generation modes, the design group was able to make the final prototype more versatile in the different types of testing that the device was able to accomplish.

A user friendly device was important because a device that is difficult or unwieldy to use means that a user is less inclined to purchase or even use the device, regardless of how accurate or functional the device may be. The EMG simulator was meant to be used as a diagnostic device to test or debug myoelectric prostheses for quality testing or actual implementation onto a human body. A single test would on average take slightly longer than than 20 minutes, and a user friendly interface would make the testing easier and less of a chore. To satisfy this objective, the design team designed and coded a user interface that is shown in Figure 36. The architecture of the code (see Figure 33) allowed for the user to set channels one and two
independently, and had a number of sub-menus that allowed for ease of use; manual, pulse, ramp, and the noise definition mode. Under these major sub-menus there were sub-functions that would allow the user to set parameter values (i.e., duty cycle or noise amplitude). To ensure that the user interface was user friendly, the user interface had been set up so that the left and right buttons would always be to scroll left and right between sub-menus of the same level, to increment or decrement amplitudes or duty-cycles, or to make a selection between two different options (i.e., 50 and 60 Hz noise). The top button was reserved as a back button, the bottom button was set as an enter button, and the middle button was the run signal generation button on the home screen, and at all other menu screens would be the home call button (refer to Chapter 5: Final Design for a more detailed instruction set for the user interface).

Additional considerations that were made include a quick noise toggle to allow for a quick on/off selection for simulated power line interference, and coding selected amplitudes to be written to FLASH memory so that a user would not need to constantly redefine the signal parameters upon device startup. FLASH memory writing was not fully implemented, and is discussed in Chapter 8: Conclusions and Recommendations.

Reliability is an important objective for any design project as the user needs to be able to trust the output signals generated by the device. To be considered reliable, the device must prove itself both accurate and precise. Accuracy of the device determines how close the generated signal compares with true human EMG signals; the more accurate is a device, the better it functions as a pseudo EMG signal. Precision is the repeatability of the device; a precise device is able to consistently output the same values. It is optimal that the device be able to consistently generate an accurate pseudo EMG signal for each usage. If the device’s output signals vary in unpredictable manners, then the device is not functioning properly, thus making it unreliable. A
formal testing for reliability of the final prototype was not done, but in the verification tests of the final prototype, the generated signal would consistently follow user defined characteristics accurately. For more information and data representations of the final prototype, refer back to Chapter 5: Final Design.

The design team pursued a low power consumption approach in the design process of the device to ensure that the final prototype be fully portable. In the design team’s meetings with the client, it was determined that a fully portable device should be battery powered rather than relying on an A/C power cord. Also considering the previous objective of user friendliness, the final prototype was designed to be powered with convenient, off the shelf AA batteries.

Batteries have a limited supply of power, thus necessitating that the final prototype was a low power device. If the device required too much power then it would drain the batteries too quickly and require frequent battery replacement. In order to provide the user with a reasonable amount of run time between battery replacement, the ARM microprocessor, DAC, and design of the output stage, were all selected or designed with power consumption in mind. It was estimated that the user could expect approximately one week of device usage before replacement batteries were necessary.

An important objective in any design is safety. A device cannot be commercially sold unless it is proven that the device is safe for the user. In this case however, the design team needed to account not only for user safety, but had to assure that the EMG simulator would not damage the myoelectric prosthesis under test. To ensure user safety, the final prototype was designed such that there was no possible way that dangerous amounts of current would reach the user.
The final prototype consisted of three major stages; the Olimex development board (with ARM processor), the TI DAC on an evaluation board, and the design team’s output hardware. The final prototype was made to demonstrate the functionality of the design, thus a printed circuit board (PCB) was not designed. However as a final PCB with only the necessary parts could be created, it would not be difficult to manufacture additional models of the design. It is important to note that the product itself serves a specific purpose, thus making it a product that would not be sold in large quantities.

The final prototype was designed to best satisfy the client’s objectives and therefore their most critical needs. However it was also important consider the impact that the product will have on the world. Once introduced into the market, the design could have either a direct or indirect influence on the economy, environment, society, politics, ethics, health, and safety of the world.

If released into the market, the EMG simulator device will be the first of its kind. The clients are planning on selling the device for three to four thousand dollars, ensuring a profit three or four times the cost of manufacture. The main purpose of the device is to be used as a quality testing or debugging tool to be used with myoelectric prostheses. While the device is being sold for a profit of three times its cost, it is not a device meant to be sold in large quantities. The device’s primary purpose is to assist in the development and sales of myoelectric prostheses.

Upper limb prostheses are still a developing field, and thus creating the EMG simulator may assist in the field’s development. Improving the field of myoelectric prostheses could give hope to those requiring the use of such devices. The EMG simulator is able to make a societal
influence by giving people with lost limbs the chance to regain functionality with the use of a replacement limb.

Designing an EMG simulator may also hold political ramifications. An important aspect of politics is the care and maintenance of the United States Military. With the recent increase of active duty (due to the wars in Iraq and Afghanistan) there are a percentage of soldiers who have suffered from injuries resulting in limb amputation. In an effort to offer a better life for those brave soldiers, the government is providing funding of myoelectric prosthesis development.

Ethically, the EMG simulator is important as it will reduce the danger to a human subject since it allows quality testing and debugging to be done more accurately. Also, with an accurate emulation of human EMG signals, it is now possible to reduce human subject experimentation in the design and development of future myoelectric prostheses.

Maintainability is another major factor that must be considered with the final design. Even after the device has entered the market it is important to determine whether or not the important components of the device, such as the Olimex development board, will continue to be available ten years after its entry to the market. In the design process the design team made great efforts to ensure that major components of the design, such as the Olimex development board and TI DAC, would continue to be available to ideally ten years from now. Also the design group assured that the ARM microprocessor would have enough computational power and memory space so that it would not only be able to accomplish any function that the design team coded, but also any added functionality that future design groups may seek to add.

The primary purpose of the design was to help promote myoelectric prosthesis development. As a result the environmental impact was not considered as a major design objective. The device does not make a large impact upon the environment, as it is not
disposable, requires low energy, and does not emit any harmful waste. Additional steps could be
taken to make the device more environmental friendly, but this consideration was not a major
design objective.
Chapter 8: Conclusions and Recommendations

The design group was successfully able to create an EMG simulator for use in the quality testing and debugging of a myoelectric prosthesis. The output signal of the device has proven itself to be Gaussian in distribution and also to have the expected power spectral density. The signal can be manipulated by the user and the device successfully provides three distinct modes of operation (manual, ramp, pulse) for diagnostic testing. Based on these criteria, the output signal is able to accurately and precisely emulate a human EMG signal for use in its diagnostics testing. There are, however, several improvements that we recommend be made to the device.

The first recommendation is utilization of FLASH memory. It is possible to have the ARM microprocessor write variable parameters to FLASH memory, so that the user can recall them at a later date, even after the device has been powered down. We suggest that a method be implemented for users to save and recall settings using the non-volatile FLASH memory.

The second recommendation is for the implementation of a mechanical apparatus to secure the output leads to the electrodes. Currently, the EMG simulator must be interfaced with the prosthesis to be tested by taping the output wire to the electrode or using some other simple but potentially suboptimal solution. We suggest that an apparatus be designed to affix the output lead to the electrodes securely and without the need for human intervention.

A third recommendation is the redesign of the hardware and software to utilize the entire dynamic range of the DAC. Because the EMG signal is differential, a 20 mV<sub>pp</sub> differential signal may be produced by combining two 10 mV<sub>pp</sub> signals on separate DAC channels. Because of the doubling effect of the differential output method, the hardware is currently designed to potentially accommodate a 20 mV<sub>pp</sub> differential signal, but the software does not support this option (nor is it a requirement of the device), limiting the usefulness of the DAC. We therefore
recommend that the hardware be redesigned for an output of 30 mV_{pp} instead of 40 mV_{pp}, and that the software be updated to accommodate this change and utilize the full range of the DAC.

A fourth recommendation is that the output of the DAC be lowpass filtered to restrict frequency content to the EMG band (less than 200 Hz). Lowpass filtering will smooth the jagged output of the DAC, producing a more realistic signal. To this end, we suggest placing a capacitor of the appropriate value in the op-amp fractional gain stages for each channel to produce active filters.

A fifth recommendation is that a PCB be designed for the final device. The prototype used a DAC evaluation board which occupies considerably more space than the SOIP packaged DAC, as well as a breadboard whose size was far larger than its equivalent PCB. We suggest that a PCB be designed and tested to reduce the size of the final version of the EMG simulator.

A sixth recommendation is that a different (possibly custom-designed) development board be used for the ARM microprocessor. The board used in the final prototype, the Olimex LPC-MT-2138, requires a 12V DC supply, the equivalent of eight AA batteries. The ARM microprocessor, however, requires only 3.3 V to run and could easily be powered by a single 9 V battery (as could the auxiliary electronics board) with appropriate voltage regulator circuits. Therefore we suggest that other options for microprocessor development boards (including custom boards) be considered.

A seventh recommendation is that an option be added for phase offset between the two channels of the device. The final prototype currently can output different amplitudes independently on each channel, but cannot set the output on one channel to happen out of phase with the other. We suggest that a menu be added for phase offset, and that it be implemented in software.
The final recommendation made by the design team is that the microprocessor code be reorganized to improve maintainability. As written, the code has large amounts of redundancy and could be improved by (for example) creating functions that perform general operations and take as arguments the relevant parameters, such as channel, amplitude, etc. instead of the current design which has several specific functions for each operation.
References


Leunissen, M. *Biology for biological engineering.*

http://www.soe.uoguelph.ca/webfiles/mleuniss/Biomechanics/EMG.html


Saito, T. (2001). *Thermal noise random pulse generator and random number generator*


Wilson, A. W., & Lovely, D. F. Reducing the impedance of passive stainless steel surface electrodes.
%% Construct and Plot EMG with Physiologically Accurate Algorithm
% clear all;
% fs = 2000;
% emg_phys = zeros(1, 100000); % Initialize EMG vector
for i = 1:1000
    % Sum 100 random MUAPTs
    emg_phys = emg_phys + makeTrain(makeMUAP(.5 + (2-.5).*rand()));
end
% Lowpass filter EMG
[b, a] = butter(2, [20/(fs/2) 200/(fs/2)]); % Use sampling rate of 2kHz
emg_phys = filter(b, a, emg_phys);
emg_phys = emg_phys(4000:85000);
% Normalize
max = 3 * std(emg_phys);
emg_phys = emg_phys / max;
% Plot EMG and power spectrum
% Time domain plot
t = 0:.0005:40.5; % Time vector
subplot(3,1,1); plot(t, emg_phys);
xLabel('Time (s)'); xlim([0 40]);
yLabel('Amplitude'); ylim([-2 2]);
title('Physiologic EMG Signal Simulation');
% Power spectral density plot
subplot(3,1,2); pwelch(emg_phys, [], [], [], fs);
xLabel('Frequency (kHz)'); xlim([0 .5]);
ylim([-100 0]);
title('Physiologic EMG Power Density Spectrum');
% Distribution plot
subplot(3,1,3); hist(emg_phys, 100); xlim([-2 2]);
xLabel('Value');
yLabel('Samples');
title('Physiologic EMG Distribution');

%% Construct and Plot EMG from Gaussian distribution
% clear all;
% fs = 2000;
% emg_gauss = randn(1, 81001); % Length chosen to match physiologic
% Lowpass filter EMG
[b, a] = butter(2, [20/(fs/2) 200/(fs/2)]); % Use sampling rate of 2kHz
emg_gauss = filter(b, a, emg_gauss);
% Normalize
max = 3 * std(emg_gauss);
emg_gauss = emg_gauss / max;
% Plot EMG and power spectrum
% Time domain plot
t = 0:.0005:40.5; % Time vector
figure(2);
subplot(3,1,1), plot(t, emg_gauss);
xlabel('Time (s)'); xlim([0 40]);
ylabel('Amplitude'); ylim([-2 2]);
title('Gaussian EMG Signal Simulation');
% Power spectral density plot
subplot(3,1,2), pwelch(emg_gauss, [], [], [], fs);
xlabel('Frequency (kHz)'); xlim([0 .5]);
ylim([-100 0]);
title('Gaussian EMG Power Density Spectrum');
% Distribution plot
subplot(3,1,3), hist(emg_gauss, 100);
xlabel('Value'); xlim([-2 2]);
ylabel('Samples');
title('Gaussian EMG Distribution');

% Compare to Real EMG Data
clear all;
fs = 1000;
emg_real = tread_wfdb('ted_emg_data03.dat');
emg_real = emg_real(1:40001); % Take first 40 seconds of data
% Normalize
max = 3 * std(emg_real);
emg_real = emg_real / max;
% Plot EMG and power spectrum
% Time domain plot
t=0:.001:40; % Time vector
figure(3);
subplot(3, 1, 1), plot(t, emg_real);
xlabel('Time (s)'); xlim([0 40]);
ylabel('Amplitude'); ylim([-2 2]);
title('Real EMG Signal Simulation');
% Power spectral density plot
subplot(3,1,2), pwelch(emg_real, [], [], [], fs);
xlabel('Frequency (Hz)'); ylim([-100 0]);
title('Real EMG Power Density Spectrum');
% Distribution plot
subplot(3,1,3), hist(emg_real, 100);
xlabel('Value'); xlim([-2 2]);
ylabel('Samples');
title('Real EMG Distribution');

function [MUAP]=makeMUAP(distance)
% Note: distance parameter specifies the distance of the fiber of interest
% below the skin (and recording site)
% Construct vector to represent single muscle fiber
fiberLength=500;
MUAP1=zeros(1, fiberLength);
% Construct MUAP
% Simulate action potential propagating down muscle fiber
for i=1:fiberLength
    % Calculate signal strength at recording site by calculating distance
    % Assumes that signal strength is proportional to distance
    MUAP1(i)=1/sqrt(distance^2+(fiberLength/2-i)^2);
end
MUAP1=[zeros(1,1000) MUAP1 zeros(1,1000)]; % Pad with zeros
% Add inverted peak with specified offset
% Accounts for the effect of bipolar differential recording electrodes
MUAP2=-[zeros(1,150) MUAP1];
MUAP1=[MUAP1 zeros(1,150)];
MUAP=MUAP1+MUAP2;

function [MUAPT]=makeTrain(MUAP)
%% Initializes MUAPT with a random number of leading zeros
MUAPT=[zeros(1, randi([2000 7000])) MUAP];
for i=1:10
    % Add MUAPs to train at random intervals
    MUAPT=[MUAPT zeros(1, randi([2000 7000])) MUAP];
end
MUAPT=[MUAPT zeros(1, 100000-length(MUAPT))];
Appendix C: Analog Signal Generation System Test Schematic
Appendix D: Commonly Used Circuit Topologies and their Governing Equations

![Inverting op-amp circuit](image)

Figure 64 - Inverting op-amp circuit

Equation 12 - Inverting op-amp circuit gain equation

\[
V_{out} = -V_{in} \frac{R_f}{R_{in}}
\]
Equation 13 - Resistive voltage divider equation

\[ V_{out} = \frac{R2}{R1 + R2} V_{in} \]

Equation 14 - AD620B instrumentation amplifier gain equation

\[ Gain = \frac{49.9 \, k\Omega}{Rg} + 1 \]
Equation 15 - Equation for cutoff frequency of Sallen Key filters

$$f_c = \frac{1}{2\pi \sqrt{R_1 R_2 C_1 C_2}}$$
Appendix E: Analog Signal Generation Test Results

Results for Reproducibility Test

Table 14 – Results of reproducibility test for 10 MΩ resistor

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>543.2 mV</td>
<td>99.6%</td>
</tr>
<tr>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>583.0 mV</td>
<td>99.65%</td>
</tr>
<tr>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>588.4 mV</td>
<td>99.65%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average:</td>
<td>99.6%</td>
</tr>
</tbody>
</table>

Table 15 - Results of reproducibility test for 1 MΩ resistor

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.982 MΩ</td>
<td>128.31 mV</td>
<td>195.0 mV</td>
<td>99.8%</td>
</tr>
<tr>
<td>0.982 MΩ</td>
<td>128.31 mV</td>
<td>168.0 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>0.982 MΩ</td>
<td>128.31 mV</td>
<td>176.0 mV</td>
<td>99.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average:</td>
<td>99.8%</td>
</tr>
</tbody>
</table>
### Table 16 - Results of reproducibility test for 100 kΩ resistor

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>84.0 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>93.4 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>85.0 mV</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

Average: 99.9%

### Table 17 - Results of reproducibility test for 10 kΩ resistor

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>28.3 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>29.9 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>29.6 mV</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

Average: 99.9%
Results for Stability Test

Table 18 - Results of stability test for 10 MΩ resistor

<table>
<thead>
<tr>
<th>Resistor</th>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1175 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td>A</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1066 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td>A</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1029 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td>B</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1021 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td>B</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1036 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td>B</td>
<td>9.97 MΩ</td>
<td>169.128 V</td>
<td>1030 mV</td>
<td>99.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average:</td>
<td>99.3%</td>
</tr>
</tbody>
</table>

Table 19 - Results of stability test for 1 MΩ resistor

<table>
<thead>
<tr>
<th>Resistor</th>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>306.7 mV</td>
<td>99.8%</td>
</tr>
<tr>
<td>A</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>309.1 mV</td>
<td>99.8%</td>
</tr>
<tr>
<td>A</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>312.8 mV</td>
<td>99.8%</td>
</tr>
<tr>
<td>B</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>223.4 mV</td>
<td>99.7%</td>
</tr>
<tr>
<td>B</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>196.0 mV</td>
<td>99.7%</td>
</tr>
<tr>
<td>B</td>
<td>.982 MΩ</td>
<td>128.31 V</td>
<td>216.0 mV</td>
<td>99.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average:</td>
<td>99.8%</td>
</tr>
<tr>
<td>Resistor</td>
<td>Input</td>
<td>Expected Value</td>
<td>Measured Value</td>
<td>% Error</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>A</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>70.2 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>A</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>69.4 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>A</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>72.1 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>66.0 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>66.2 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>99.7 kΩ</td>
<td>99.5013 V</td>
<td>67.8 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Average:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resistor</th>
<th>Input</th>
<th>Expected Value</th>
<th>Measured Value</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>26.6 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>A</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>25.9 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>A</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>29.4 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>25.9 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>26.6 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td>B</td>
<td>9.79 kΩ</td>
<td>81.121 V</td>
<td>28.2 mV</td>
<td>99.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Average:</td>
</tr>
</tbody>
</table>
Appendix F: MATLAB Code for Comparing Box-Muller Transform and FIR Filtering

%Boxmuller.m
%Compare Box-Muller transform to filtering

%u1, u2: uniform RVs
u1 = rand(1,10000);
u2 = rand(1,10000);

%x: x-axis
x = -4:0.08:(4-0.08);

%g1: gaussian generated with box-muller transform
g1 = sqrt(-2*log(u1)).*cos(2*pi*u2);

%Plot Box-Muller
figure(1)
hold on
N = hist(g1,100);
Y = 1/(sqrt(2*pi))*exp(-0.5*x.^2);
plot(-4:0.08:(4-0.08),N)

%Overlay Gaussian distribution
plot(-4:0.08:(4-0.08),300*sqrt(2*pi)*Y,'r')
xlabel('Histogram of Box-Muller transform. n = 10,000 b = 100')
hold off

%Filter uniform RVs
[b,a] = butter(3,[20/1000 400/1000]);
g2 = filter(b,a,u2);

%Plot filtered uniform
figure(2)
hold on
N = hist(g2,100);
plot(-4:0.08:(4-0.08),N)

%Overlay Gaussian
plot(-4:0.08:(4-0.08),300*sqrt(2*pi)*Y,'r')
hold off
xlabel('Histogram of filtered uniform deviate. n = 10,000 b = 100 N = 3')
Appendix G: C-Code for MSP430 Tests

FIR filter test

```c
int k = 0;
long int i = 0;
int j[3] = {0,0,0}, y[3] = {0,0,0};
int j2[3] = {0,0,0}, y2[3] = {0,0,0};
int sine[500];
int out1,out2, out3, out4;
int main( void )
{
    //FLL_CTL1|=BIT4;
    //FLL_CTL1&=~(BIT5|BIT3);
    FLL_CTL1 = (SELS + SELM1) & (~XT2OFF);
    do{
        IFG1 &= ~OFIFG; // Clear OSCFault flag
        for (i = 0xFF; i > 0; i--); // Time for flag to set
    }while ((IFG1 & OFIFG)! = 0);
    //SCFQCTL=(BIT6|BIT5|BIT4|BIT3|BIT2|BIT1|BIT0);
    init_sys();
    // Stop watchdog timer to prevent time out reset

    WDTCTL = WDTPW + WDTHOLD;
    _BIS_SR(GIE); // Global Interrupt enable
    runtimerb();
    writeWord("TEST");
    clearLCD();
    //long* lp = 0;
    //float i1=2.718, i2=3.14;
    //char num[7];
    for(int x = 0;x<500;x++)
    {
        sine[x] = (int)((2^16-1)*sin(x/500*2*3.14))%(2^16-1);
    }
    while(1)
    {
        y[2] = y[1];
        y[1] = y[0];
        y[0] = y[0]/1000;
        j[2] = j[1];
        j[1] = j[0];
        j[0] = (int)ran1(j[0]);
        
        y2[2] = y2[1];
        y2[1] = y2[0];
        y2[0] = y2[0]/1000;
        j2[2] = j2[1];
        j2[1] = j2[0];
        j2[0] = (int)ran1(j2[0]);
        //j[0] = rand();
    }
}
```

149
Box-Muller Transform Test

int k = 0;
long int i = 0;
int j[3] = {0,0,0}, y[3] = {0,0,0};
int j2[3] = {0,0,0}, y2[3] = {0,0,0};
int sine[500];
inout1,out2, out3, out4;
int main( void )
{
    //FLL_CTL1|=BIT4;
    //FLL_CTL1&=~(BIT5|BIT3);
    FLL_CTL1 = (SELS + SELM1) & (~XT2OFF);
    do{
        IFG1 &= ~OFIFG; // Clear OSCFault flag
        for (i = 0xFF; i > 0; i--); // Time for flag to set
    }while ((IFG1 & OFIFG) != 0);

    //SCFQCTL|=(BIT6|BIT5|BIT4|BIT3|BIT2|BIT1|BIT0);
    init_sys();
    // Stop watchdog timer to prevent time out reset

    WDTCTL = WDTPW + WDTHOLD;
    _BIS_SR(GIE); // Global Interrupt enable
    runtimerb();
    writeWord("TEST");
    clearLCD();
    //long* lp = 0;
    //float i1=2.718, i2=3.14;
    //char num[7];
for(int x = 0; x<500; x++)
{
    sine[x] = (int)((2^16-1)*sin(x/500*2*3.14))%(2^16-1);
}
while(1)
{
    x[0] = ran1(x[0]);
    x[1] = ran1(x[1]);
    y[0] = (int) sqrt(-2*ln(x[0]))*cos(2*3.14*x[1]);
    i++;
}

float ran1(long idum)
{
    return 1664525L*idum + 1013904223L;
}
Appendix H: Analog Signal Delivery System Test Schematic
Appendix I: Analog Signal Delivery System Photographs

Figure 68 - Signal delivery system test, overall setup. The simulator (transmitter) circuit is on the protoboard to the left, and the prosthesis signal conditioning (receiver) circuit is on the protoboard to the right. The protoboards are connected with the shielded cable from Cooner Wire.
Figure 69 - Signal delivery system transmitter circuit

Figure 70 - Signal delivery system receiver circuit
Appendix J: Analog Signal Delivery System Test Data

Key:

Channel 1 (yellow): output of function generator

Channel 2 (blue): transmitted signal

Channel 3 (purple): inverted transmitted signal

Channel 4 (green): Amplified differential signal

Figure 71 – Analog signal delivery system test data with input signal 20 µVpp, 400 Hz
Figure 72 - Analog signal delivery system test data with input signal 20 µVpp, 150 Hz

Figure 73 - Analog signal delivery system test data with input signal 15 µVpp, 400 Hz
Figure 74 - Analog signal delivery system test data with input signal 15 µVpp, 150 Hz

Figure 75 - Analog signal delivery system test data with input signal 10 µVpp, 400 Hz
Figure 76 - Analog signal delivery system test data with input signal 10 µVpp, 150 Hz

Figure 77 - Analog signal delivery system test data with input signal 5 µVpp, 400 Hz
Figure 78 - Analog signal delivery system test data with input signal 5 µVpp, 150 Hz
# Appendix K: Cost Analysis

## Analog:

<table>
<thead>
<tr>
<th>Part</th>
<th>Qty</th>
<th>Unit Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>60</td>
<td>$0.064</td>
<td>$3.84</td>
</tr>
<tr>
<td>Capacitor</td>
<td>20</td>
<td>$0.045</td>
<td>$0.90</td>
</tr>
<tr>
<td>Knob Potentiometer</td>
<td>4</td>
<td>$10.24</td>
<td>$40.96</td>
</tr>
<tr>
<td>LM348 Quad Op-Amp</td>
<td>5</td>
<td>$1.39</td>
<td>$6.95</td>
</tr>
<tr>
<td>LM556 2 x 555 Timers</td>
<td>2</td>
<td>$1.55</td>
<td>$3.10</td>
</tr>
<tr>
<td>LM3914 LED Driver</td>
<td>2</td>
<td>$2.54</td>
<td>$5.08</td>
</tr>
<tr>
<td>AD633 Hardware Multiplier</td>
<td>2</td>
<td>$7.98</td>
<td>$15.96</td>
</tr>
<tr>
<td>LED Bar</td>
<td>2</td>
<td>$7.00</td>
<td>$14.00</td>
</tr>
<tr>
<td>Printed Circuit Board</td>
<td>1</td>
<td>$100</td>
<td>$100</td>
</tr>
<tr>
<td>Labor (1 hour)</td>
<td>6</td>
<td>$50</td>
<td>$300</td>
</tr>
<tr>
<td>AA Battery</td>
<td>2</td>
<td>$1.00</td>
<td>$2.00</td>
</tr>
<tr>
<td>AA Battery Holder</td>
<td>1</td>
<td>$1.69</td>
<td>$1.69</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td></td>
<td></td>
<td>$494.48</td>
</tr>
</tbody>
</table>

## Digital:

<table>
<thead>
<tr>
<th>Part</th>
<th>Qty</th>
<th>Unit Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>15</td>
<td>$0.064</td>
<td>$0.96</td>
</tr>
<tr>
<td>LEDs</td>
<td>3</td>
<td>$3.00</td>
<td>$9.00</td>
</tr>
<tr>
<td>LM348 Quad Op-Amp</td>
<td>1</td>
<td>$1.39</td>
<td>$1.39</td>
</tr>
<tr>
<td>TI DAC8734DAC</td>
<td>1</td>
<td>$41.93</td>
<td>$41.93</td>
</tr>
<tr>
<td>Olimex MSP430-449STK2 Dev. Board</td>
<td>1</td>
<td>$88.11</td>
<td>$88.11</td>
</tr>
<tr>
<td>MOSFET</td>
<td>2</td>
<td>$2.00</td>
<td>$4.00</td>
</tr>
<tr>
<td>Printed Circuit Board</td>
<td>1</td>
<td>$100</td>
<td>$100</td>
</tr>
<tr>
<td>Labor (1 hour)</td>
<td>3</td>
<td>$50</td>
<td>$150</td>
</tr>
<tr>
<td>9V Battery</td>
<td>2</td>
<td>$3.99</td>
<td>$7.98</td>
</tr>
<tr>
<td>9V Battery Connector</td>
<td>2</td>
<td>$1.99</td>
<td>$3.98</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td></td>
<td></td>
<td>$407.35</td>
</tr>
</tbody>
</table>
Appendix L: Olimex LPC-MT-2138 Schematic
Appendix M: Texas Instruments DAC8564EVM Schematic
Appendix N: Auxiliary Electronics Schematic

Voltage Supplies
- $V_{DD}$ Olimex is the 3.3V power supply produced by a regulator on the Olimex board.
- Battery is the 8 AA batteries (12 V).
- The LM7805CT is a voltage regulator IC which outputs 5V.

Op-amps
- TLV2770 is the single op-amp used for buffering the 1.25V reference signal. It runs on 5V and 0V supplies.
- TLV2774 A-D are the quad op-amp used for fractional gain output. They run on 5V and 0V supplies.
Appendix O: Function List

void init_sys() – initializes all peripherals by calling all other “init” functions, runs once at startup.
void init_buzzer() – initializes buzzer.
void init_PLL() – initializes phase-locked loop to run CPU clock and peripheral clock to run at 14.7456 MHz.
void init_adc() – initializes analog-to-digital converter.
void init_spi() – initializes serial peripheral interface for communication with the DAC.
void load_dac() – toggles the LDAC signal to simultaneously load all DAC outputs, unused in current DAC configuration.
void disable_int_ref() – disables the internal reference of the DAC, unused in current DAC configuration.
void dac_test() – tests the DAC by outputting its full range of values in a loop.
void spi_sync_low() – brings the SYNC signal for the DAC low to signal the start of a data transfer.
void spi_sync_high() – brings the SYNC signal for the DAC high to signal the start of a data transfer.
void beep() – outputs a short “beep” to the buzzer, used as audible feedback for button presses.
void error_beep() – outputs two short, low pitched “beeps” to buzzer to indicate error condition.
void start_beep() – outputs three “beeps” of increasing frequency to the buzzer, runs once to indicate startup.
void flash_led() – flashes the LED once (on/off), runs once to indicate startup.
void start_timer1() – starts Timer1, used to control EMG and noise signal output.
void stop_timer1() – stops Timer1 (turns output off).
void timer1_interrupt() – interrupt service routine for Timer1.
void update_ch1_emg_values() – updates emg[0] and emg[1] global array with new filtered random values, to be outputted on the positive and negative terminals for channel 1.
void update_ch1_noise_values() – updates noise[0] and noise[1] global array with new values from the sinewave lookup tables, to be added to emg[0] and emg[1] and outputted on the positive and negative terminals of channel 1.
void make_sine() – generates the lookup table for the sinusoidal simulated line noise, runs once at startup.
void run_ch1_manual() – does the calculations necessary for channel 1 to run in manual mode, including amplitude modulation and adding noise.
void run_ch2_manual() – does the calculations necessary for channel 2 to run in manual mode, including amplitude modulation and adding noise.
void run_ch1_pulse() – does the calculations necessary for channel 1 to run in pulse mode, including amplitude modulation and adding noise.
void run_ch2_pulse() – does the calculations necessary for channel 2 to run in pulse mode, including amplitude modulation and adding noise.
void run_ch1_ramp() – does the calculations necessary for channel 1 to run in ramp mode, including amplitude modulation and adding noise.
void run_ch2_ramp() – does the calculations necessary for channel 2 to run in ramp mode, including amplitude modulation and adding noise.
void output_to_dac() – level shifts and outputs the four values in the global output[] array to each of the four channels of the DAC simultaneously.

unsigned int get_rand(unsigned int idum) – returns a uniform random number with seed idum between 0 and $2^{32}$ using a linear congruential random number generator.
void delay_us(int n) – software delay of approximately n microseconds.
void delay_ms(int n) – software delay of approximately n milliseconds.
static void count_time() – interrupt service routine for Timer0.
void update_batt_level() – reads ADC0.0 to update the battery level indicator on the home screen.
unsigned int get_num_digits(unsigned int val) – returns the number of digits in val, used when printing values on the LCD display.
void get_buttons() – updates the hit_button global variable with the button that was pressed.
void init_lcd() – initializes the LCD display.
void clear_lcd() – clears the LCD display.
void clear_top_lcd() – clears the top line of the LCD display.
void clear_bottom_lcd() – clears the bottom line of the LCD display.
void clear_section_lcd(unsigned char start, unsigned char stop, unsigned char row) – clears one section of the LCD display, between start and stop (indexing starts from 0), in the specified row (0 or 1).
void write_lcd(const char *s, unsigned char x, unsigned char y) – writes to the LCD display, starting on the specified x and y coordinates (indexing starts from 0).
void scroll_LTI() – scrolls “Liberating Technologies Inc” across the LCD display, runs once at startup.
void write_amplitude(unsigned int amp, unsigned char channel) – writes the amplitude for the specified channel to the proper location on the LCD display.

int buttonfunc(int state, int num) – executes the function for a given button pressed when the system is in the given state.

int menufunc(int state) – updates the LCD display based on the given state and state variables.

struct menu* buildmenus() – builds the menu tree for each state based on the next state given a specific button press, runs once at startup.
Appendix P: Derivation of Output Gain Stage

Assumptions

*The Current Assumption:*

\[ I_{in} = 0 \]

It is assumed that no current flows into the operational amplifier (i.e., it has infinite input impedance.)

*The Negative Feedback Assumption:*

\[ V_{+} = V_{-} \]

Because of negative feedback, the two input terminals will always be approximately equal.

Derivation:

\[ I_{R2} = \frac{(V_{in} - V_{-})}{R2} \quad \text{(Ohm's Law)} \]

\[ I_{R1} = I_{R2} + I_{in} \quad \text{(Kirchoff's Current Law)} \]

\[ I_{R1} = I_{R2} \quad \text{(Current assumption)} \]

\[ V_{out} = V_{-} - I_{R1}R1 \quad \text{(Ohm's Law)} \]

\[ V_{out} = V_{-} - R1 (V_{in} - V_{-})/ R2 \quad \text{(Substitution)} \]

\[ V_{out} = V_{+} - R1/R2 * (V_{in} - V_{+}) \quad \text{(Negative Feedback Assumption)} \]

\[ V_{out} = 1.25 - 0.016 * (1.25 \pm 20mV - 1.25) \quad \text{(Substitution)} \]

\[ V_{out} = 1.25 - 0.016 * (\pm 20mV) \quad \text{(Cancellation)} \]
Appendix Q: DAC Output Tests for EMG and Noise Amplitude Verification

Figure 79 – Test results of DAC output test for EMG amplitude control. 100 µVpp input, approx 6.20 Vpp output, test apparatus gain of 1000.

Figure 80 – Test results of DAC output test for EMG amplitude control. 1000 µVpp input, approx 6.20 mVpp output, test apparatus gain of 100.
Figure 81 - Test results of DAC output test for EMG amplitude control. 10000 µVpp input, approx 6.20 mVpp output, test apparatus gain of 1.

Figure 82 – Test results of DAC output test for noise amplitude control. 100 µVpp input, test apparatus gain of 100.
Figure 83 – Test results of DAC output test for noise amplitude control. 1000 µVpp input, test apparatus gain of 100.

Figure 84 - Test results of DAC output test for noise amplitude control. 10000 µVpp input, test apparatus gain of 1.
Appendix R: Output Stage Tests for Gain and Frequency Response Verification

Figure 85 - Output stage test. Input of 2.5 Vpp at 60 Hz.

Figure 86 - Output stage test. Input of 2.5 Vpp at 100 Hz.
Figure 87 - Output stage test. Input of 2.5 Vpp at 140 Hz.

Figure 88 - Output stage test. Input of 2.5 Vpp at 180 Hz.
Appendix S: MATLAB Simulation for Testing Linear Congruential Generator

```matlab
function [seconds] = rand_test_3()

count = 0;
random = 3519870697;
x = 0;
y = 0;
while(~((x == -14512) & (y == -17298)))
    random = my_rand(random);
    y = x;
    x = bitshift(random, -16) - 32768;
    count = count + 1;
end

seconds = count / 900;
```