ENGN/PHYS 207—Fall 2018
Assignment #6: EMG Bionic Arm

Phase I: must be completed by Noon Tues Nov 13, 2018
Phase II: scheduled completion by Noon Fri Nov 16, 2018
Final Report Due Date: Noon Tues Nov 27, 2018

Circuits You'll Build

1. Analog circuitry: Electromyogram (EMG) sensing circuit to measure electrical signals generated by muscles contractions
2. Digital electronics: Arduino interface to control robot arms through motor rotation

Lab Skills You'll Learn

1. Designing and integrating circuit systems from standard building blocks (instrumenation amp, active filter, buffer, rectifying and smoothing)
2. Arduino control of servo motors, in response to analog signals measured

EMG Robot Arm: Helping Hand

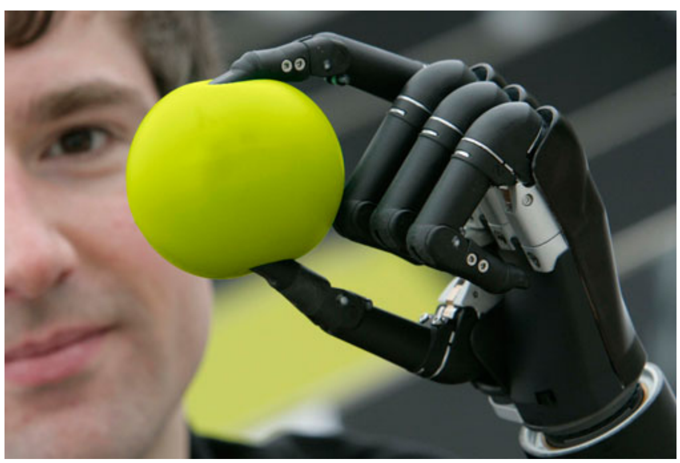

Figure 1: Touch Bionics iLimb Ultra. Dexterity is sufficient to squish a foam ball. The motion/grip of the prosthetic hand is driven by the EMG measured on the existing part of the amputee's forearm. Image credit: `http://www.touchbionics.com/products/active-prostheses/i-limb-ultra`

The prosthetics field has been undergoing somewhat of a revolution the past decade, thanks to advances in miniaturized electronics, high performance batteries, and machine learning techniques (for example, see Figure 1). As discussed in class, one common control paradigm is to use to measure the electromyogram (EMG)—electrical activity associated with muscular contractions—and actuate in the prosthetic hand according to the EMG analog signal measurements.

The prerequisite for such a prosthetic device is to non-invasively measure the electrical activity associated with muscle contractions. Electrodes placed on the skin can measure the electrical signal associated with muscular activity, if the signal is properly filtered and amplified. This is the Electromyogram (EMG). The EMG is processed and interpreted by a computer or microprocessor to convert a pattern of muscular activity measured with electrodes on the skin surface into commands that move motors in a desired manner. For instance, flexing the biceps twice in quick succession might be interpreted as: "Move robotic arm forward-right to grab green squishy ball." A good prosthetic hand can help perform functions like grasping a water bottle, picking up a pencil, and buttoning clothes, thus restoring a high-quality of life to an amputee. So wouldn't it be cool to.... build the EMG hardware that measures electrical signals generated by muscular activity and program a microcontroller to do something when a contraction is detected, such as actuate a motor to turn a robot arm? This assignment gives you the opportunity to do exact that!

1 Design Problem Statement

Phase I: Your task is to design, build, characterize, and perform proof of concept experiments for a circuit that is capable of cleanly measuring the EMG.

The net result of Phase I should be sufficient evidence to establish proof-of-concept of: 1) appropriate filtering; 2) appropriate amplification; 4) Arduino measures and displays EMG signal. This evidence needs to be presented by showing appropriate decibel gain vs. frequency data, and a quick live-demo with the Arduino serial plotter.

Phase II: Your system must incorporate a Arduino analog measurement which actuates a servo motors based on the strength of the muscle contraction (hence intensity of the EMG signal).

You should first establish proof-of-concept of working Arduino code that can rotate a motor to an angle proportional to the strength of an input signal.

The net result of this phase should be a live-demo of your muscle contractions appropriately controlling a robot arm!

Safety Considerations: For safety reasons, under no circumstances should you connect your EMG circuit to a human subject without first consulting the instructor! When you are ready the instructor will consult with you to properly connect EMG skin surface electrodes as well as help safely connect power supplies to the robot arm motors.

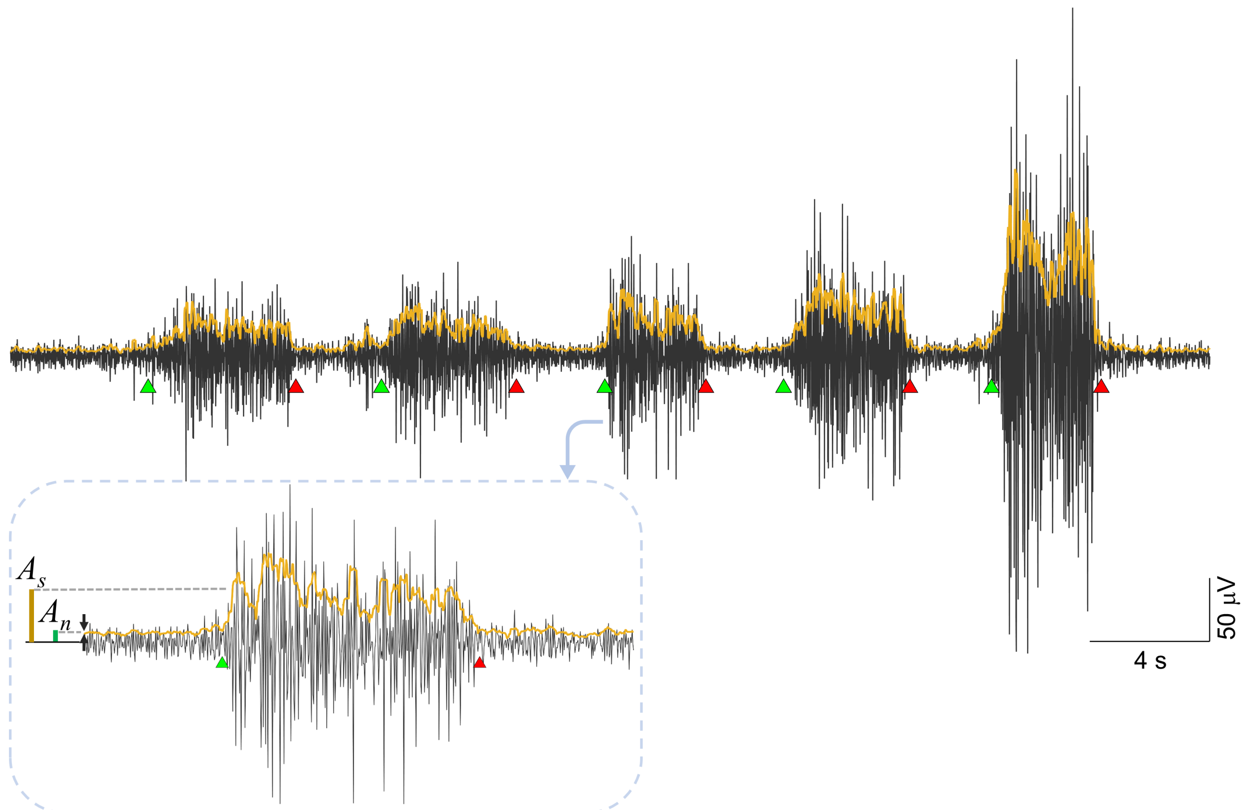

Figure 2: Example EMG (black trace) measured from the forearm over 5 contractions of the forearm muscle with increasing strength. The average signal intensity trace (orange) is shown overlaid. It is a rectified and smoothed version of the EMG trace. Green and red triangles indicate on and off times of each contraction. The inset at bottom left shows the fine temporal detail of the 3rd contraction. The labels A_s and A_n indicate the signal and the noise levels from which the SNR is computed.

2 EMG Signal Measurement

Electrical signals are generated by the ion flow of Na^+,K^+, Ca^{++} necessary to contract muscles. This current flow passes through very small mechanical channels of saline solution distributed throughout the muscle and surrounding layers. These mechanical channels have some resistance and capacitance, hence we get a voltage drop thanks to Ohm's law (V = IZ). It is this small, but detectable EMG signal we want to appropriately capture. From previous studies, we know the characteristic amplitude and frequency range of muscle contractions measured with skin surface electrodes[1] and how they need to be manipulated to control a robot arm.

1. EMG frequency range is $\approx 5 - 500$ Hz. The low frequency noise components ($f \leq 5$ Hz) often exhibit large amplitudes and need to be more strongly filtered out than the high frequency noise components ($f \geq 500$ Hz). So, the low-frequency components need to be be attenuated

[1] c.f. De Luca CJ, The Use of Surface Electromyography in Biomechanics, *J. Applied Biomechanics*, 1997: 13, 135,163. `http://delsys.com/decomp/078.pdf`

at 40 dB/dec. The high frequency noise can be attenuated at 20 dB/dec.

2. EMG signal magnitude measured on the skin surface is in the range of $\approx 100 - 1000\ \mu$V. In order to be accurately measured and usefully processed by an Arduino to command a motor to rotor, the amplified EMG signal magnitude should be about 1 V. Don't amplify too much—the Arduino saturates at 5 V.

3. Prosthetic hands are usually programmed to recognize the the ***time-averaged intensity*** of the EMG signal. Time averaging (aka ***rectifying + smoothing***) can be accomplished with rectifying (diode) and smoothing (RC circuit) or via a software (Arduino computes the RMS moving average). The orange colored trace in **Figure 2** is an example of a smoothed EMG signal with a smoothing time scale of ≈0.1 s.

3 Arduino: Analog Measurement and Digital Control of Servo Motors

In phase II of this project, you'll interface your EMG measurement circuitry to an Arduino microcontroller. The Arduino will 1) make an analog measurement the EMG signal and 2) actuate a motor to rotate according to the strength of the contraction. There are several considerations to be aware of when interfacing the analog and digital sides.

1. **Analog signal acquisition**Analog signals measured by the Arduino must be in the range of 0-5 V. Exceeding either of these limits can fry the Arduino in part or whole. Leave the bbq sauce at home; it doesn't go well on burnt plastic.

2. It is typical to center the EMG signal about a ≈ 2.5 V reference offset. That allows us to see positive and negative swings in the input signal. Doing so requires careful implementation of ***op-amp biasing*** **using** ***rail-splitting***.

3. You should be able to cleanly view the analog signal using the Serial Plotter feature in the Arduino IDE, so your Arduino code needs to be programmed to output that data.

4. Once you have ascertained you can see a clear EMG signal, we need to control the robot arm using a smoothed version of it. The smoothed version tells us the strength of the signal due to the muscle contraction. There is a design choice here: a software vs. hardware solution.

   Software Solution: We mentioned before the root of the mean squared (RMS) signal is often computed for this purpose:

$$RMS[k] = \frac{1}{L}\sum_{n=k}^{k+L-1} |v[n] - v_{offset}| \tag{1}$$

   where $RMS[k]$ is computed at the kth time point using L samples of $v[n]$ (the nth voltage sample acquired by the Arduino). Notice we are subtracting the constant offset voltage ($v_{offset} \approx V_{cc}/2$ in this case) then taking the absolute value, which guarantees a positive number. We are averaging this signal over L total samples. Assuming the Arduino acquires samples at intervals of $T_{sample} = 1$ ms (1 kHz sampling rate), then $L = 100$ is usually a good choice, as this averages over a 100 ms window of time. You can adjust to taste. Notice that if voltage signal $v[n]$ is very different than the offset voltage during contractions, then you'll get a large RMS value. One the other hand, if $v[n] \approx v_{offset}$ while the muscle is not contracting, the RMS output is nearly 0. If the contraction is fairly weak, then we'll get an intermediate value for the RMS, as desired.

An example showing how to compute a moving average (aka the ***smoothed*** signal) using the Arduino can be found here: `https://www.arduino.cc/en/Tutorial/Smoothing`. Note you'll need to modify your sketch to compute the running average of the absolute value, per Eqn 1. Carefully read through each line of code to understand how it works.

Hardware Solution: Alternatively, there's also a hardware solution to smoothing; see Figure 3. The diode allows the capacitor C_1 to charge up to the maximum of the input voltage V_{in}. The diode will stay forward biased, allowing current to flow, until the voltage across C_1 equals the maximum input voltage. The role of the op-amp is to make sure the capacitor charges to the actual max of the input voltage; Without the op-amp we'd only charge up to the max input voltage minus a diode drop, which may be a significant amount, e.g. 0.7 V out of $\approx$ 2 V dynamic range. The role of the resistor is allow the circuit to forget what's happened in the distant past; the peak detector doesn't "hold" or "remember" the peak value forever. Instead, the resistor R_1 allows the capacitor to discharge with a time constant $\tau = R_1C_1$. The time constant can be thought of as the smoothing window time, the mathematical doppelganger to the choice of L in the software solution. The hardware solution is really easy to wire, and really easy to set the smoothing time constant.

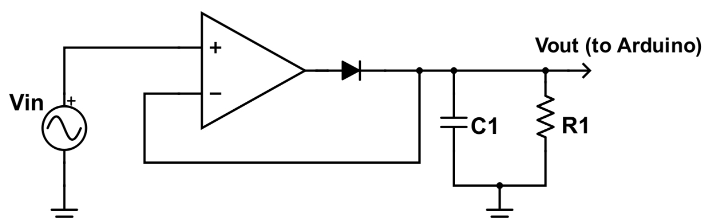

Figure 3: Smoothing Circuit based on diode and RC discharge. The time constant = the smoothing window: $\tau = R_1C_1$.

5. **Servo Motor Control:** The Arduino will be used to rotate a servo motor to a desired position based on the strength of the EMG signal. The servo motor, in turn, actuates a gripper and/or prosthetic joint. A great demo tutorial how to do this analog input-servo output operation can be found here: `https://www.arduino.cc/en/Tutorial/Knob`.

   Also, you should know what your Arduino code is doing under the hood. ***Pulse Width Modulation*** (PWM) is commonly used to control servo motors. Take 10 min to learn what this term means by reading this wonderful tutorial: `https://www.servocity.com/how-does-a-servo-work`

6. **Power Supply Considerations:** The Arduino power supply will power your analog circuitry only. The A separate high-current LiPo battery (or other power supply) will be used to power the servo motors. The motors draw $\approx$ 2 A, whereas Arduino's power supply maxes out at 500 mA).

4 Final Report

You may choose to do a written report OR a video lab report! For either medium, present the following core elements:

1. **Final circuit diagram** (1 page). Clearly delineate individual functional blocks and all component values shown by each component (a la Backyard Brains circuit diagram we have reviewed in class).

2. **Design rationale** (2 pages absolute max) For each individual functional block describe the what and why. For example:

   > Stage [B] is an active band pass filter. We set the gain to 20x because ..., and we set the band pass range to x to y Hz because ..."

   Also be sure to clearly describe how these functional blocks are integrated. Justify why you arranged the functional pieces this way, and emphasize the tradeoffs involved. For example:

   > We implemented an active band pass filter at each of the EMG electrode inputs (functional block [A]) for two reasons: i) This re-centers the output to a baseline of 2.5 V which is required for valid inputs of the INA 126 instrumentation amplifier configured for single-supply operation (0 to +5V operation) shown in stage [C]; and ii) This design offers an added benefit of an initial filter and amplification of EMG signals. This design requires careful matching of the active BPFs connected to both of the inputs to ensure we don't introduce any additional differences in output signals that will be amplified by the INA 126 in the next stage.

   Note that this section should also include some suggestions for an improved design, where appropriate.

3. **Proof of concept**. (1 page max) Preferably post a video of your circuit in action. Alternately, show a series of figures. Either way, this section should establish that your design allows you to clearly measure a signal directly related to the strength of muscle contraction.

4. **Appendix**.

   (a) I. Include input-output relations (transfer functions) for each functional block. Include sketches of decibel gain functions, where appropriate. For example, you might plot/sketch the decibel gain vs. log frequency for any filter (active or passive) that you implemented.

   (b) II. Arduino code