

**Problem Set 3: Hybrid Insect Robots Neuroelectric Stimulation**  
**ENGN/BIOL 267—Winter 2020**  
**Due: Thurs, Feb 13, 2020, 5pm**



Figure 1: Giant flower beetle *M. torquata* equipped with neural-electric interface climbs a tree. Image credit: Nanyang Technology University (accessed at: telegraph.co.uk)

### 1. Bug Zapper: Solving the Circuit Model

We have discussed in class the circuit model that underlies neural-electric stimulation of hybrid insect biobots. The main idea is to send a series of pulses at just the right amplitude, frequency, and duration to spoof the neural impulses that would normally result from natural sensory inputs (eyes, antenna, cerci) guiding the animals locomotion response. Figure 2 should look familiar to you: the classic RC model of a tissue-electrode interface, for neural-electric stimulation in hybrid insect biobots. Solving and understanding the circuit model is critically important to understanding just how voltage pulses can generate neuronal action potentials, thereby “hijacking” the nervous system.

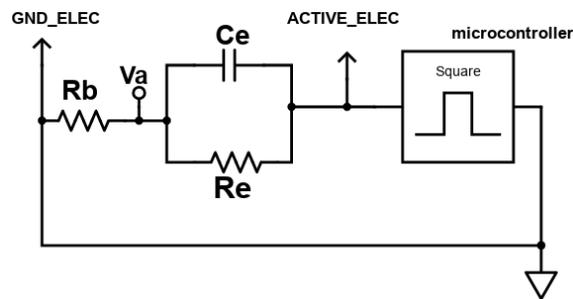


Figure 2: Neuro-electric stimulation model.  $R_e$  and  $C_e$  represent resistance and capacitance of the electrode-tissue interface; the resistor  $R_b$  represents the internal resistance of the insect’s abdomen.

Quite a lot of interesting behavior happens with just 3 circuit elements, and the math gets a bit hairy pretty quickly. Our solution strategy was to think of the total voltage response  $v_a(t)$  as the sum of contributions due to current through the capacitor  $C_e$ , and through the resistor  $R_e$ , both of which flow into  $R_b$  (representing the insect body. Of course, the insect only “known” about the voltage across  $R_b$ , which is  $v_a(t)$ .

We argued that  $v_a(t)$  responds to a step-function input with amplitude  $V_{pulse}$  as follows:

$$v_a(t) = V_{pulse} \left[ \exp\left(\frac{-t}{R_b C_e}\right) + \frac{R_b}{R_b + R_e} \right] \quad (1)$$

In Eqn 1, the first term represents the current through  $C_e$ , an exponential decay that is often called a “spike” of capacitive current. The second term represents the contribution due to current through  $R_e$ , fundamentally a voltage divider relation. We assume that the capacitor is initially uncharged ( $v_c(0) = 0$ ), equivalently that all voltage drops across  $R_b$  at time 0, that is to say  $v_a(0) = V_{pulse}$ .

- (a) This was (hopefully!) a good way to come to grips with how the circuit behaves. But truth be told, this solution is not exactly right. It neglects the interaction in the parallel combo  $C_e$  and  $R_e$ . Show that Eqn. 1 does not satisfy the initial condition criteria.
- (b) So what’s the fix, how do we obtain the exact solution for this circuit model? Glad you asked! The answer is we have to go back to the first principles, where we originally wrote down Eqn 2 using Kirchoff’s Current Law, Ohm’s Law, and the fundamental relation for a capacitor:

$$\underbrace{\frac{v_a}{R_b}}_{i_b} = \underbrace{\frac{V_{pulse} - v_a}{R_e}}_{i_{R_e}} + \underbrace{C_e \left(-\frac{dv_a}{dt}\right)}_{i_{C_e}} \quad (2)$$

Solving this requires a quick refresher in ODEs: write the transient solution, then the forced solution, then add them up. We’ll save the full process for another time. For now, imagine a little birdie lands on your shoulder and whispers the following (very good) guess of a solution into your ear, which you dutifully transcribe:

$$v_a(t) = V_{pulse} \left[ (1 - f) \exp\left(\frac{-t}{\tau}\right) + f \right] \quad (3)$$

where:

$$f = \frac{R_b}{R_b + R_e} \quad (\text{voltage divider})$$

$$\tau = R_p C_e \quad (\text{time constant})$$

$$R_p = \left( \frac{1}{R_b} + \frac{1}{R_e} \right)^{-1} \quad (\text{parallel equivalent resistance})$$

Carefully compare Eqns 3 and 1. What is the same? What is different? Hint: there are 2 subtle but important differences.

- (c) Show that Eqn 3 is compatible with the initial condition of an uncharged capacitor with a constant voltage  $V_{pulse}$  applied (to stimulate neurons, of course).
- (d) Show the proposed solution given in Eqn 3 does in fact solve the ODE in Eqn 2. Just plug in! If both sides end up equal, you’ve got a winner!
- (e) Lastly, make a good estimate of the all important time constant  $\tau$ . Assume you have 0.005-in Teflon coated silver wire as an electrode. The electrode is typically inserted a few mm deep. State any other assumptions you make in developing your estimate for you make and briefly justify them. Is the time constant long enough to plausibly stimulate a neuron? Briefly justify.

## 2. Practical Implementation

The goal here is to build some real-world electronics that could be used in principle to stimulate a roach. Instead of handling insects, we'll build the equivalent circuit out of discrete resistors and capacitors for now. This is a great opportunity to get hands-on with programming microcontrollers—a super practical, transferable skill.

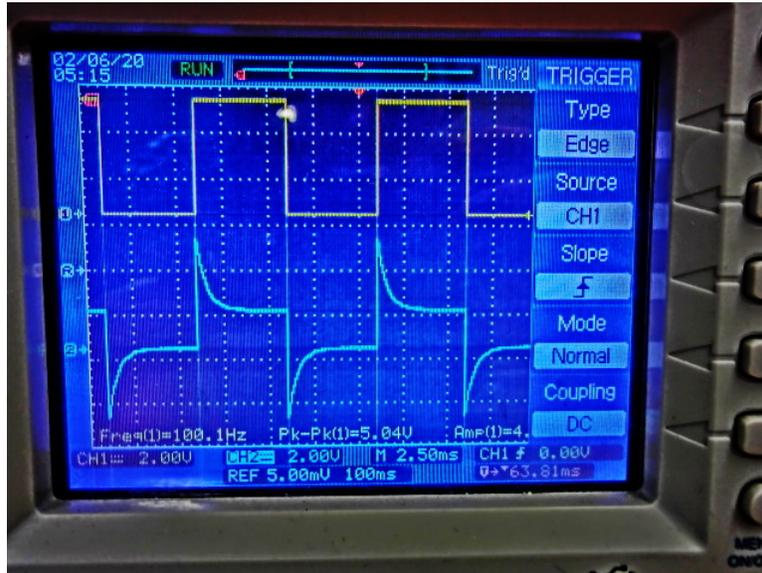


Figure 3: Example oscilloscope readings. If all goes well, you should get a similar picture. Yellow trace = square wave input, 5V amplitude; Blue trace shows  $v_a(t)$  with characteristic exponential “spike” decaying to a non-zero baseline value. The time constant  $\tau \approx 1$  ms.

- Build the circuit model of a biobot insect on a breadboard, as illustrated in Fig. 2. Choose R and C values that make a realistic circuit model, based on calculations we did in class. Measure and report the actual values for resistors and capacitor you use.
- Use an Arduino, Teensy, or Feather microcontroller to generate square waves at a desired frequency and duty cycle.
- Use the Arduino Serial Plotter (see coding hints/tips below) to measure and simultaneously plot the square wave input and the resulting voltage  $v_a(t)$  across the “insect body” (resistor  $R_b$ ). Care must be taken to appropriately measure and display these voltages using an Arduino Uno analog input pins (e.g. A0 and A1). Be aware that an Arduino Uno can only make a measurement at  $\approx 1$  ms intervals. That’s a long time between samples with time constants for exponential decay that are on the order of  $\tau \approx 1$  ms. A faster microcontroller board (Teensy 3.6, Feather M0) can sample voltages much faster than this. The other solution would be to build a circuit with a longer time-scale, e.g.  $\tau \approx 20$  ms,  $20\times$  longer time scale than an actual hybrid insect, but the same general behavior of current spike that exponentially decays. For instance, use a  $2.2 \mu\text{F}$  capacitor and resistors  $\approx 10 \text{ k}\Omega$  in this build. In order to see a decay, we have wait long enough for the decay to happen. That is, we have to space square waves far enough apart—with a slow enough frequency. Given a decay time of about 20 ms, set the frequency of the square wave input to be  $\geq 5\tau = 100$  ms. This corresponds to a 10 Hz period input. *Be sure to appropriately set the input wave frequency in your code so that you can clearly see the capacitive spikes followed by decay to non-zero baseline level.*

- (d) Starting with the example code provided in the BioE course box folder, write/modify code that does the following (also see next page for some coding resource/hints/tips):
- i. Outputs a square wave at a desired frequency and with desired duty cycle (see note above about appropriately setting the square wave frequency relative to time constant  $\tau$ .)
  - ii. Implements at least one or more of these practical user friendly features where the user can:
    - A. Manually set stimulation frequency and duty cycle of the square wave via serial monitor
    - B. Toggle the square wave on/off by inputting a character into the serial monitor
    - C. Set duration of square wave via serial monitor: one push of a keyboard button triggers pulses for desired duration

If all goes well, you should get a picture like the one in Figure 3

- (e) Now the punchline: Time to compare the actual circuit behavior to the circuit + math model. To do this:
- i. Capture a good screen shot of the oscilloscope or Arduino serial plotter a la Figure 3
  - ii. Carefully plot the expected behavior of the circuit according to the math model (Eqn 3). You will probably want to use a fairly sophisticated software package to help with this operation—Matlab or Python, anyone?!
  - iii. Compare and contrast the real world result to the circuit model. Do you see the capacitive spikes? The constant baseline voltage (not decaying fully to zero volts)? What is the measured time constant of your circuit? What did you expect it to be? How do these numbers compare—quantify.

**Coding tips and hint**—functions and example code you might use:

- Reading and using integers sent through serial monitor here: <https://www.arduino.cc/en/Tutorial/ReadASCIIString>
- Measuring voltages using the on-board analog-to-digital-converter (ADC): <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>
- Plotting voltages in real-time (tutorial on Adafruit): <https://learn.adafruit.com/experimenters-guide-for-metro/circ08-using%20the%20arduino%20serial%20plotter>