Capacitive Sensing Accelerometer

Max Miller[1]
[1]University of New Hampshire

Tristan Anderson[2]
[2]University of New Hampshire

Abstract. For our final project in PHYS605, Tristan Anderson and I have built an electro-mechanical accelerometer utilizing technics outlined in at PDF released by Maxim Integrated [1]. The approach utilizes capacitance sensing and kinematics to translate arbitrary changes in the devices' sensed capacitance to the acceleration the device is experiencing at a given time. Our device is a one-dimensional accelerometer which utilizes two capacitance sensors, an Arduino Duemilanove, a known mass charged plate in an oscillating system made using non-conductive foam, and housing made of PVC. The creation is broken into two stages, with the first being the initial testing and prototyping of the circuit, code, and mechanical systems outline and the second being the creation of a prototype electro-mechanical, large-scale accelerometer implementing machined parts and materials. All code utilized can be found in the Code section 8.2. The primary package utilized for the capacitance sensing is CapacitiveSensor [2] , a standard library which allows for the measuring of charge-time between two GPIO pins on the Arduino. Any additional code was written by myself, including a web server written in Go which allows for us to actively monitor and log the accelerometers behavior. The source code is in the Code section under 8.2.1, as well as in the listed GitHub repository. The largest assumtion we took in the creation and testing of this device was that the foam provided behaved inline with Hooke's Law, having oscillations in the form:

$F = -kx$

This assumtion is needed to fomulate our equation for acceleration of the known mass.

1. Introduction:

Accelerometers have been in frequent use throughout modern physics. Within the past century, the electro-mechanical accelerometer has found favor in electronic monitoring systems as a dominant means of measuring the force/acceleration felt on a particular entity. If it has an onboard computer and moves, then it probably has an electromechanical accelerometer attached to it. As a result of these devices being so prevalent, a great deal of experimentation and engineering has gone into crafting accurate, space-efficient devices. Unlike most devices which are integrated into a circuit, an accelerometer requires both circuitry and kinematics to operate properly.

The circuitry is used to sense a change in the system, while kinematics provides what the circuit is measuring. Unlike most instruments, an accelerometer is both collecting the data and creating the data to collect at the same time, making it a unique challenge for us. For the purposes of this project, we will only analyze the popular two-capacitor accelerometer design.

2. Purpose

Our goal in this lab is to build a working and calibrated one-dimensional electro-mechanical accelerometer. As a result of this device being electro-mechanical and not purely electrical, our methods for constructing the device involved a more in-depth understanding of how the components we have used in previous labs, especially capacitors. This was a large motivation is us pursuing this project, as it allowed for a more interesting look at calibrating and constructing devices from raw materials and electrical components, rather than just electrical components.

3. Setup and Tools

We broke the setup of this lab into two stages: prototyping/testing and final device construction. For the prototyping stages, we used a standard medium-sized breadboard (5-pins per line, power and grounding rails), a metal filament resistor set, multilayer ceramic capacitors, and a standard wiring kit, including male-to-male jumper wires. In both stages, we chose our microcontroller to be an Arduino Duemilanove, running the Arduino Mega 328 processor (ATMEGA 328), with a clock speed of 20MHz [3]. The benefit of the Arduino over our other microcontroller option, the Raspberry Pi, comes from the Arduino not being a complete “desktop” computer, but rather a special purpose microprocessor for collecting data/running procedures in a loop. As a result, we need minimal setup to get the device to operate correctly, and we can output the data collected to a serial port on a desktop PC using certain software. For testing, we also utilized a number of non-circuitry components, including a non-static foam, thin copper plating, and thermal resistive tape, which we used to create a basic oscillating system and open capacitor plates. For the final product, our non-circuitry components included a brass “known mass”, which operates as our known, charged mass in the oscillating system, two cut copper plates for capacitor plates, and another set of non-static foam. For both sections, we used a standard soldering iron, which we shared with the other groups in lab.

3.1. Capacitance Sensing Library and Circuit

For this device, we utilized an RC circuit and the Arduino Capacitance Sensing Library to measure capacitance from the “C” value in the RC circuit.

The main driver of this device is the Arduino Capacitance Sensing Library, which basic RC circuit properties to read the capacitance. A DC signal is sent out through the charge pin and into the circuit, which is now an RC circuit with a variable capacitance resistor (the sensor) acting as the "C". The library then uses the RC time constant to measure the absolute capacitance from the sensor. The library uses arbitrary units to measure the capacitance, as gives no information on the units of the value measured. As a result, we can equate the arbitrary units to m/s^2 in the calibration phase, and all that matters in determining the configuration of our circuit are how many degrees of precision we have to work within the device. For example (as noted in section 5), the final circuit has a precision of 600 arbitrary units when at rest, meaning the absolute measured capacitance changes by 600 units between rest and full compression on each of the sensors. By this, when we equate a single arbitrary unit to a m/s^2 value, we can measure 600 points of m/s^2 on each compression of the oscillating system.

Because of the method the library chooses, resistor choice for the "R" of this circuit is key. The Arduino article on this library [2] includes a nice list of resistor values and their corresponding effects on the circuit. As expected, larger resistors allow for more precision when measuring , but have the trade-off of higher read time. Fortunately, the read time we require for this circuit is large enough (200ms-500ms) where a greater increase in resistor value will not have an adverse effect on device function. Additionally, too large a resistor value (10M - 40M) gives us precision at distances where the other components in our sensor, including the second capacitance sensor, could affect the capacitance sensor measurement.

4. Testing

A large portion of this project relied on testing the components in various configurations to ensure the final product would operate as we envisioned, as the system we needed to create would be much larger than the proposed small-scale system in the Maxim article. [1] The testing phases are broken down in the subsequent sections:

4.1. Capacitance Sensor Circuit

Once the Arduino was set up with the lab PC, the first step was to create the capacitance sensor circuit in Fig. 1, as well as write the code to drive the sensor. The first draft of the Arduino code can be found in section 8.2.2 and is derived from the code found in the documentation for the capacitance sensor library we used. The starting circuit utilized a 1M R1 and a 100nF CGND. In an adjustment from the proposed circuit, we made CGDN go into the discharge pin, rather than to ground. When the capacitor went to ground, the circuit was not properly sensing the capacitance and often overloaded. With the circuit built, we needed to create an open capacitor plate to properly test the capacitance sensing capabilities. We decided, given the resources present in the first lab period, to construct a capacitor plate from a single penny and short wire. Once soldered together, we placed our plate into the circuit and loaded the Arduino with the code.

Once in, we saw a reading of >100 units on each microsecond on the serial monitor. To properly test if the circuit could see a change in capacitance, Tristan lowered his hand closer and closer to the penny, creating a capacitance between the penny and his hand. As he did, we saw no noticeable change in the capacitance, until he got with ˜5 mm of the sensor, at which point we saw the numbers increase to ˜2000, with the sensor reading as high as 3500 when fully touched. During one of the tests, we collected 500 ms resolution data, shown in Fig. 2.

4.2. Increasing Sensitivity

Increasing the sensitivity of the capacitance sensor from the penny was essential to the proper function of the accelerometer. We decided to move to creating a pure copper plate out of the thin copper strips provided to us by Matt. We constructed the small plate using thermal tape to separate the two sides and soldered a wire to one side. Once connected to our circuit, the sensor was able to sense our hand up to 10 cm away with ˜800 units of precision. Touching this sensor, unlike the penny, caused the program to overload, which corresponds to a "-2" reading. A feature of these new plates was also that they held a charge for a lot longer than the penny plate capacitor, which was proven by each time we touched the plate, the "zero" reading of the sensor would be at a higher value than before. We also found that increasing the resistor R_1 from 5 M to 10 M gave us a more accurate and consistent reading.

4.3. Long-Term Runs

Once the sensor was properly built, and a more sensitive plate applied, we took a look at how the sensor operated for longer (<5 minutes) periods of time. To do this, we built a script using Go to monitor the sensor, record the output to a CSV file, and display/graph the capacitance overtime on a web server. The code for this can be found in section 8.2.1, which is contained in an attached file. The sensor showed a steady positive increase in sensed capacitance while left in open air for each of the tests run. As figures Fig. 3 and Fig. 4 show, the trend was consistent across multiple tests. Our hypothesis for why this occurred, which we cannot properly test, is that the capacitance plate, when exposed to open air in a non-controlled environment, will tend to gain free charges from the air, causing the increase in sensed capacitance. This would explain why the sensor, which is a measure of how long the circuit takes to reach a certain charge, would measure an upward trend. Interestingly, the sensor still operated as an accurate capacitance sensor with this observed anomaly.

4.4. FIR Averaging

A consistent issue with the Arduino capacitance sensor is data noise. After speaking with Professor Holtrop, he suggested we try finite impulse response (FIR) averaging. After some searching, we found the Arduino tutorial on smoothing analog input data to have an effective method for FIR averages. [4] This code is implemented in section 8.2.3 but was not used in the final product. The FIR averaging failed at higher capacitance values, giving negative readings due to the way the Arduino capacitance sensing library handled overloading. Due to time constraints, we opted to use the noisy data over the FIR averaged data. In a second look at this project, we would spend a considerably larger amount of time working through this method for averaging, as it would provide a much-needed increase in quality to the collected data.

5. Final Construction

After testing the parts individually, we moved onto creating the actual final electro-mechanical accelerometer. The final design is shown in Fig. 5, and includes only the physical schematics for the mechanical part of the device. The circuit we used can be found in Fig. 6, with noticeable adjustments from the circuit in Fig. 1. Tristan used the Demerrit Hall machine shop to mill our known mass out of brass, which at the time of writing has not been properly massed. By fixing it with a 1/4 28 hole, we were able to fix a loose wire to brass in with a screw and nut, allowing for the piece to be charged. In addition to the mass, Tristan also milled a 152 mm PVC pipe the same diameter as the brass mass, to allow for a sliding rail for the wire to come out of. While the machining was underway, I cut out eight pieces of our oscillating medium, non-static foam, of equal thickness and diameter, to be used in the accelerometer. Each piece measured 10.75 mm in radius, the same radius as the brass mass. Using an epoxy-based super glue, I glued two sets of the foam, each containing four pieces, to each other to create both sides of the oscillating medium. Once compressed to ˜50% of their total size, the foam blocks acted a good oscillating medium for the brass mass and restored back to their original size when left alone for a moment.

We created new plates for the capacitance sensing as well, using 1 mm thick solid copper plates rather than the thin copper strips we put together in testing. These new plates, once attached to a wire with solder, provided us an accuracy of 600 units on the capacitance sensor in the oscillating system. I've already provided an explanation of what these units mean in section 3.1. This allows for a greater deal of accuracy when measuring acceleration than we originally anticipated, allowing for a greater range of outputted acceleration values once calibrated.

Once these components were milled, we capped off the oscillating system with the two capacitance sensors and attached the brass mass to the foam with the epoxy-based super glue. Once together, the capacitance sensors can be attached to each end of the housing shown in Fig. 5. At the time of writing, this is as far as the project has gotten.

The next step is to calibrate the instrument, which I will describe in the section below.

In addition to the information provided, Tristan has put together a live render of how the accelerometer operates and is constructed. The full video showing this may be found here: `https://www.youtube.com/watch?v=qsVqTy0-z6M&feature=youtu.be`

6. Calibration

We did not have enough time during the project's timeframe to complete the calibration of the instrument. What follows is the procedure we would have used given more time for work. As a result of this, no data is given. I will note that the device up until this point functions as it should to get to the calibration step.

To calibrate our completed accelerometer, we will use a known acceleration to equate a force felt by the known mass to the arbitrary units read on the capacitance sensor. Our initial testing was going to use gravity, which we assume to be 9.81 m/s^2, but upon further testing we found this acceleration to be too low to measure accurately on the sensor. With this, we propose two changes if we were to do this again: a higher resistor value (10M $<$ R $<$ 24M), or foam with a much lower K constant. The latter choice would be the first change, as the ossillating medium requires a high acceleration to fully compress. With this, we can place the accelerometer on a level surface measuring along the Y-axis (up). The known mass (brass mass) will then experience a force, which we know to be equal to:

$$F_{\exp} = m_{Known} \cdot g$$

We can then equate this force with the reading on the capacitance sensors, and solve for what one of our arbitrary units will be equal to in terms of newtons and/or m/s^2. We can now calibrate the instrument on both of the capacitance sensors to ensure our reading is accurate to both degrees. Additionally, we can use the data collected on the capacitance sensors passive increase in reading value, mentioned in section 4.3, to create an equation for the line of best fit which describes the increase, and adjust our values accordingly. This, along with adjustments throughout testing, will ensure the most accurate acceleration measurements our device can provide.

7. Conclusion

The final product of our accelerometer turned out different than we had originally set out but proved to be an effective device for measuring the acceleration of an object in one dimension. The circuit design worked with little hiccups and only required slight adjustments from the original design. Overall, the quality of the electronics helped out quite a deal, such as using multi-layer ceramic capacitors over electrolytic, and ensuring the wiring was reduced in the final product. An additional improvement could have been made by using a more precise capacitance measuring device than the Arduino. While great at measuring a change in capacitance, there are variations in its accuracy which make it difficult to get a constant reading which is required for any deal of precision. Given more time, we would have used a more accurate, special made sensor which was provided to us.

The construction of the outer housing and oscillating system exceeded our expectations. The end product was larger than we set out for it to be, but this was a necessary trade-off for the time given for the project, and for our resources/abilities. Using PVC and solid brass rather than miniaturizing the system allowed for us to experiment with how we wanted to set up the sensors, and with the starting compression of the system, as well as give us a macro-scale representation of how the micro-scale accelerometers used commonly in circuits function. The quality of our end sensors and known mass charged plate also exceeded expectations, as we were able to use pure copper and pure brass respectively.

If given the chance to work on the project from scratch, I would spend less time on testing and more time on building the system with the parts that will be in the final product. With this, we could spend less time making sure the circuit operated correctly, and more time gaining data on the final product, as well as having more time to perform calibrations, which we did not get to do before the writing of this paper. Additionally, many of the issues we had with the circuit from the beginning were alleviated when we starting using the higher quality materials in the final product. Even though the full accelerometer was not calibrated to work correctly, the functioning device that we did make, which acts as a capacitance sensor to determine when the system is under high accelerations (<10 m/s^2), the actual function of the device exceeded our expectations for the project. Given more time, the device could have been tested more thoroughly in its final configuration, and a more precise device could be produced as a result.

8. Appendix

8.1. Diagrams

8.1.1. Diagram 1: Capacitance Values Measured for 15 Seconds on Touch

8.2. Code

All code may be found at the GitHub repository listed here: `https://github.com/mam1101/phys605AccelerometerFinal`

8.2.1. Web server for device monitoring (Go) (File Attached)

8.2.2. Arduino capacitance sensor code (C): (File Attached)

8.2.3. Arduino Capacitance Sensor Code With FIR Averaging (C) (File Attached)

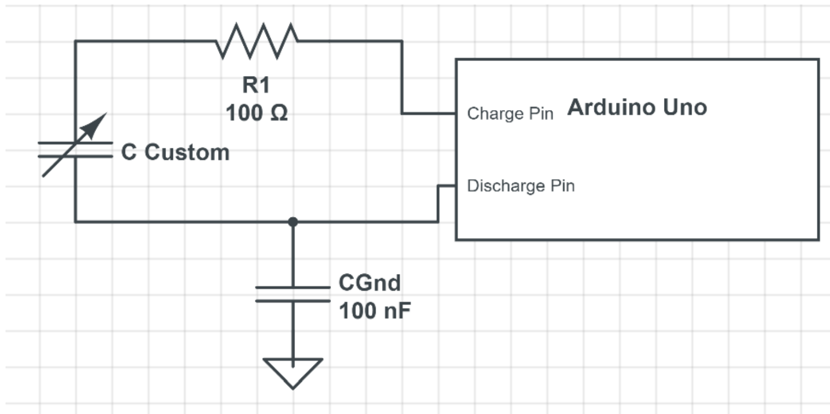

Figure 1. Original proposed circuit for capacitance sensor

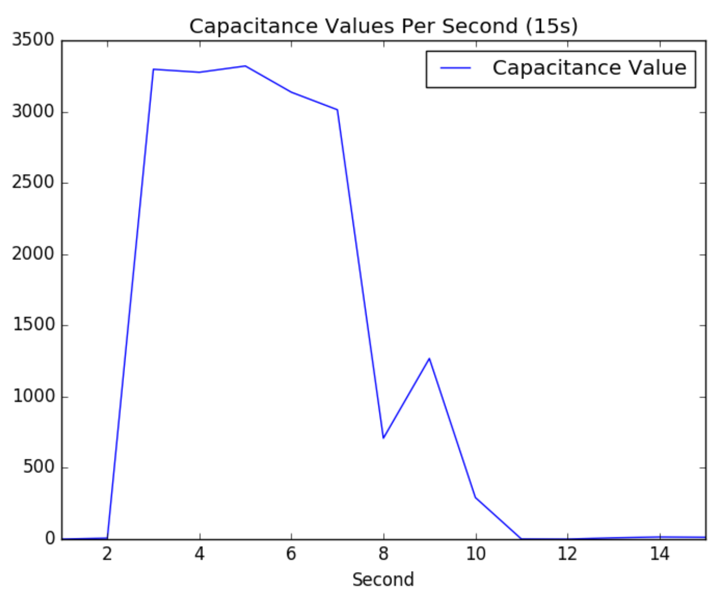

Figure 2. Capacitance values measured with Arduino code for 15 seconds after sensor was touched

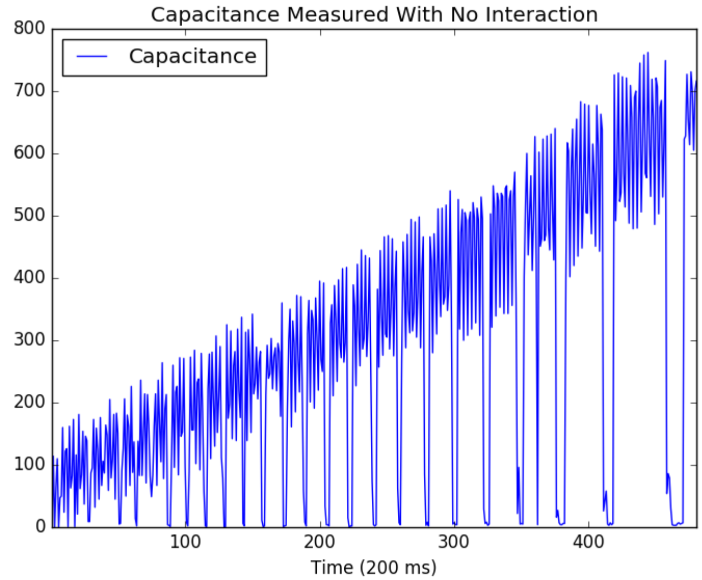

Figure 3. Passive capacitance sensor values (no interaction) with 200 ms sampling rate

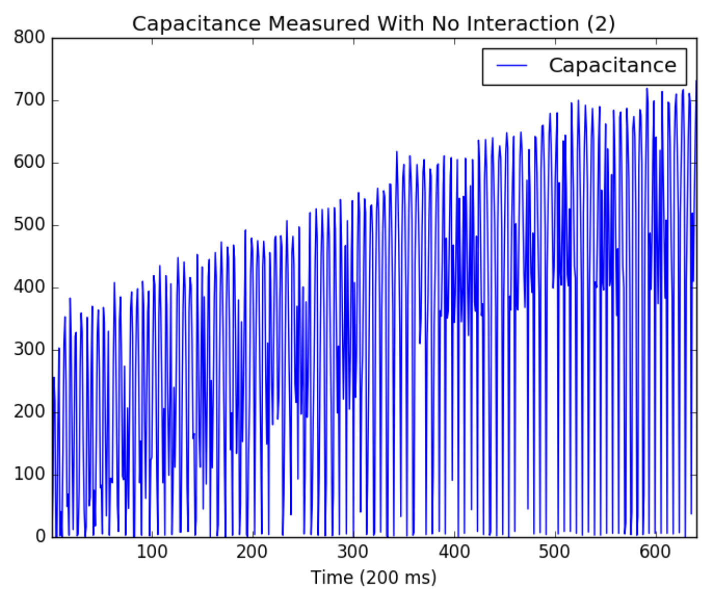

Figure 4. Passive capacitance sensor values (no interaction) with 200 ms sampling rate

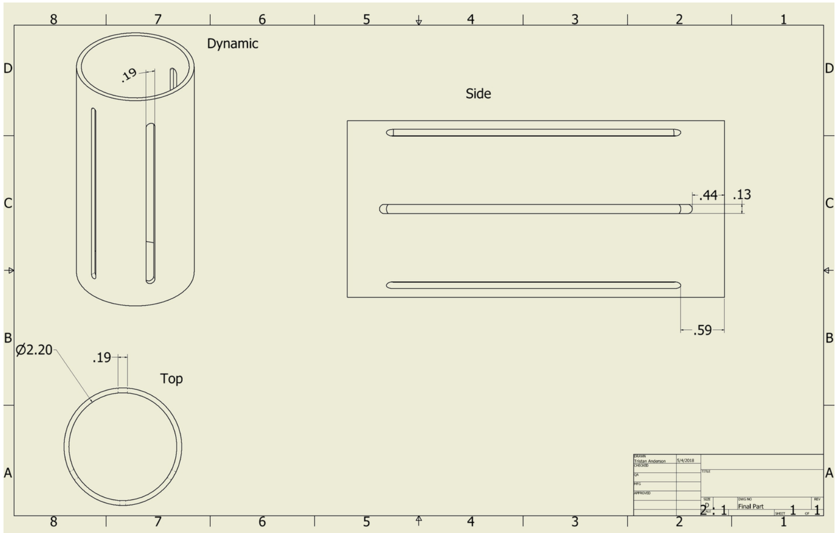

Figure 5. Final Acclerometer Housing

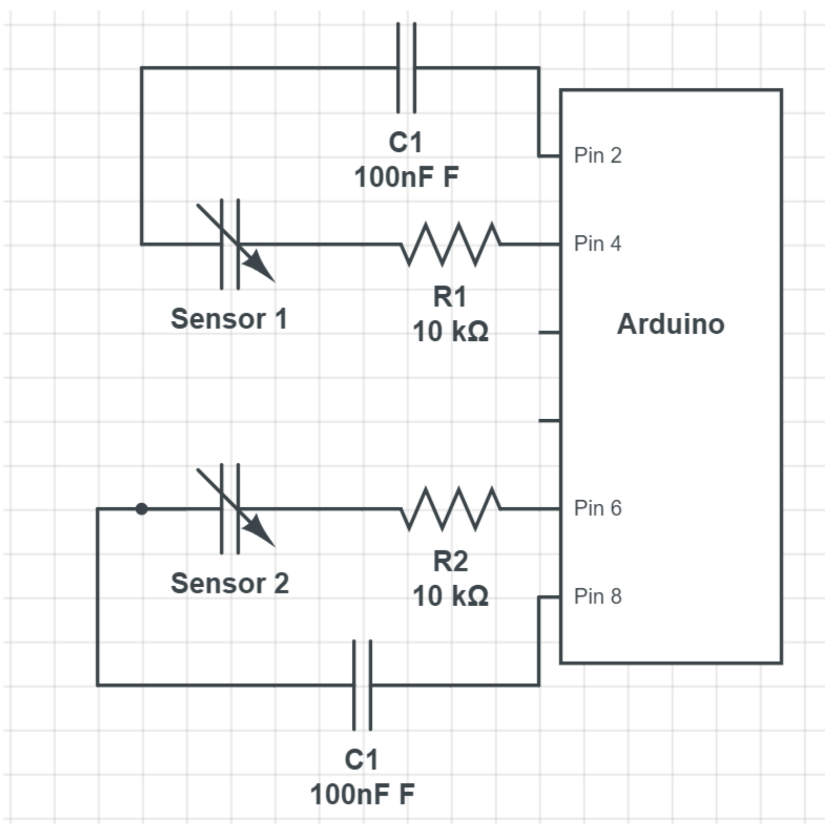

Figure 6. Final accelerometer circuit design (includes two capacitance sensors).

[1] Dadafshar M 2014 Accessed on Sun, May 06, 2018 URL https://pdfserv.maximintegrated.com/en/an/AN5830.pdf

[2] Arduino Playground - CapacitiveSensor https://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense accessed on Sun, May 06, 2018 URL https://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense

[3] ATmega328 - Wikipedia https://en.wikipedia.org/wiki/ATmega328 accessed on Sat, May 12, 2018 URL https://en.wikipedia.org/wiki/ATmega328

[4] Arduino - Smoothing https://www.arduino.cc/en/Tutorial/Smoothing accessed on Sat, May 12, 2018 URL https://www.arduino.cc/en/Tutorial/Smoothing