**Lab Assignment 2**

**183DA (Winter 2018)**

Design of Robotic Systems 1

Prof. Ankur Mahta

Date: February 8th, 2018

Jon Chang, UID: 104 853 981

Michael Warren, UID: 604 420 829

Nickolas Bruce, UID: 504 675 519



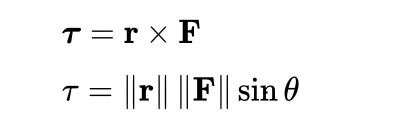
1. Introduction

In this lab we set out to create a robot that played instruments using designs which

are common in the field of engineering. The goal of this lab was to familiarize ourselves with sensors and actuators as well as to adapt to the dynamics of the lab group. In designing the instrument-playing robot, we sought to create both an autonomous mode as well as a graphical user interface (GUI) via ESP8266 WiFi-connection to control the robots’ actions.

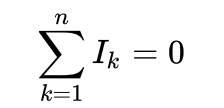
1. Theory

This lab incorporates the fundamentals of microcontroller design, breadboard prototyping, web-server/wireless communication, and electromechanical interfaces. Theory of classical mechanics was necessary to create the rack and pinion gearing as well as the moment arm which was used to create sufficient force to pluck the *Lap Harp.* The following torque equations were considered when attempting to press the keys of our keyboard instrument:



where ***τ*** (upper equation) is the torque vector and τ (lower equation) is the magnitude of the torque, **r** is the position vector (a vector from the origin of the coordinate system defined to the point where the force is applied. In this case, the origin of our coordinate system was at the center of our servomechanism). **F** is the force vector, *X* denotes the cross product, and *θ* is the angle between the force vector and the lever arm vector.

In building the circuit on the breadboard, the principles of KCL were used. Namely, the algebraic sum of all currents entering and exiting a node must equal zero.



Electromechanical induction is the force that drives the motor. When current flows in a given direction, this gives rise to an induced magnetic field which drives the motor. This property is summed up by the Lorentz force:

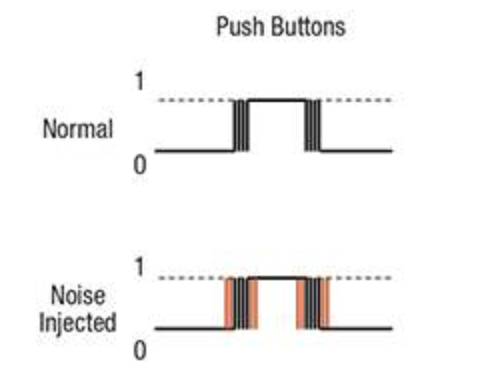
/Users/nickbruce/Desktop/Screen Shot 2018-02-08 at 12.06.00 AM.png

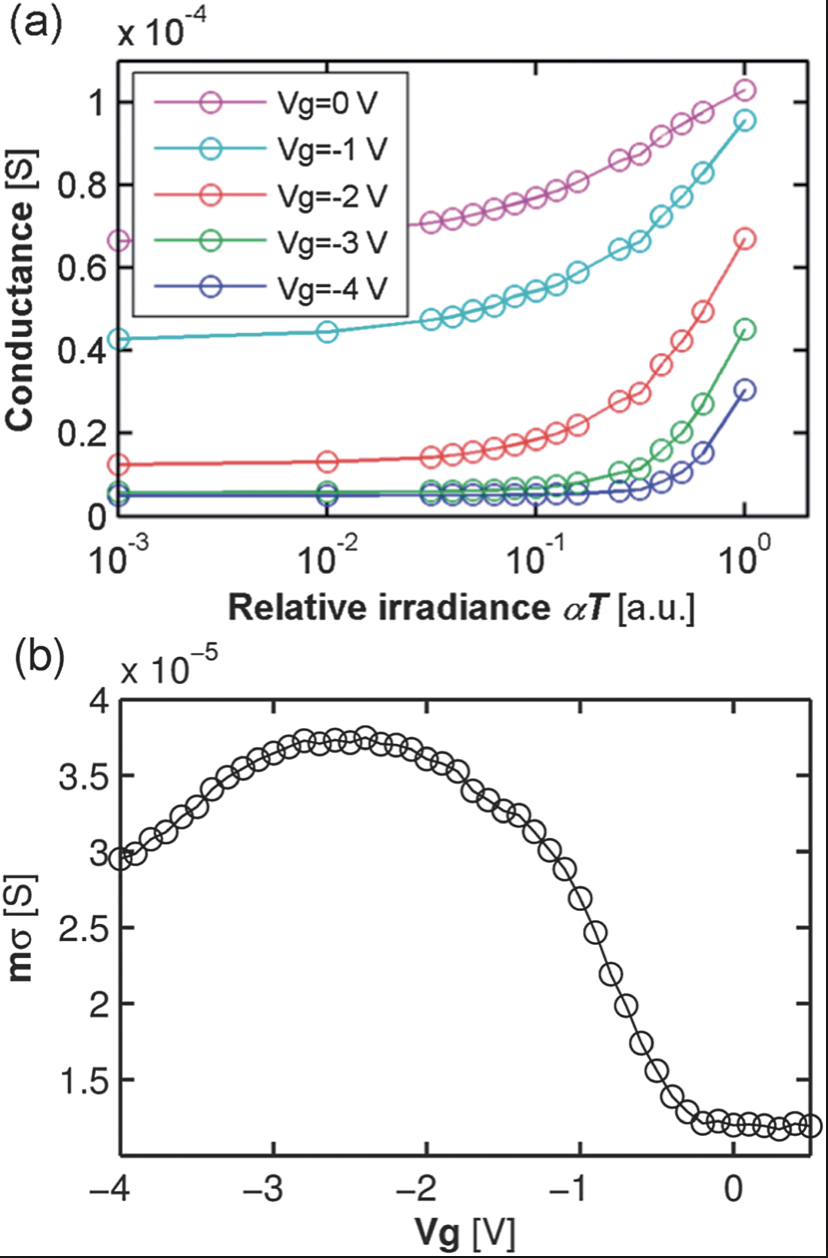
where the force vector **F**, **B** is the magnetic field vector, *q* is the charge of an electron, **E** is the electric field, and **v** is the voltage.

In selecting sensors, some theory about electromagnetics and signal processing was important. In our sensors, pushbutton switch and photoresistor, there are significantly less error than in other types of transducers. Shown in the figure below is a representation of the intended signal as well as the noisy signal. In using a photoresistor, it was useful to know properties of semiconductors which may cause a signal to become noisy. Shown in *figure 1* is a graph of phonon scattering and irradiance with variations in temperature. This relation is best exemplified in the relationship between resistivity and mobility:



where ρ is resistivity (defined as the resistance per distance or Ohms/Meters) and σ is the conductance of a given semiconuctor. These are related through the mobility, which varies proportional to temperature [K].



*Fig. 1. (above) noise on the signal of a push button (below, top) Causes for noise in Silicon photo-resistor (below, bottom) Changes in Voltage versus changes in mobility*

1. Materials

See attached bill of materials.

1. Procedure

Installing the driver software for the Arduino IDE to be able to communicate with the Node MCU was the first step. To do this, we followed the “ESP8266 I a guide” posted on the CCLE page as follows:

https://ccle.ucla.edu/mod/resource/view.php?id=1843161

To communicate the necessary information to the microcontroller,

commands are read from a source. To play music, there are two settings: autonomous

control and user-input control. Information about the keys/notes to be played are from

the microcontroller’s memory in the autonomous control case and from the user in the

instance of the graphical user-interface.

used USB-to-UART to connect to our microcontrollers to our programming IDE (we

opted to use the Arduino programming environment). After uploading our code to the

ESP8266, the ESP8266 is the receptive to receiving code via Wi-Fi in real time.

Upon completing the installation of the appropriate drivers, we then began to write the code to move the servo motors to pluck the appropriate strings of the harp and to strike the proper keys of the synth. After encountering difficulty in pressing the synth keys, we then began to try different design approaches. The problem was that the keys of the synth had too much spring resistance and would not depress even under the full load of the servo motor. Our attempt to design a more capable servomechanism arm began with testing a longer moment arm, a stronger servo motor, boost-regulators to supply more voltage to the servo. Finally, we decided to abandon the idea of using a synth. We then opted for a buzzer which would vibrate at the pitch-specific frequency when given a controlled voltage (CV) from the microprocessor. This CV is a square wave with a %50 duty-cycle, which comes standard as part of the *tone()* function, a function included in the typical Arduino IDE with no library necessary to include.

Upon realizing that the *lap harp* had similar difficulties with the related servo motor, namely that the inertia of the moment arm would cause the string which was behind the intending string would also be plucked, which resulted in a poor quality of sound, we decided to revamp our design for that as well. This lead to our idea of a beat-making machine. This is basically a servo motor with an extended arm which taps at a certain BPM, which is set either by the user or using the BPM which is hard-coded onto the board (BPM = 160).

Next, was the setting up of the web server. For this, we made a graphical user interface which was created using a Python framework known as Django. This interface sends requests from URL library to the ESP8266 MCU where the MCU then executes the commands given to create music.

As we neared an end to our project, we decided to disband and pick-up the next morning. Later that night, without unanimous approval of the group, Jon Chang, one of the team members, made an executive decision to ramp-up the difficulty of the project without first getting approval of the rest of the team. At the time of submission, our project wasn’t ready and we began to panic as our project, which was nearly completed, was now back to its early stages due to the ambition of one of the team’s members. Upon time of submission, our group decided we weren’t ready and opted to extend the deadline for a %10 grade reduction. Even still, we found that as 4pm on Friday, February 9th neared, we were still not ready.

1. Conclusion

Robots are difficult and prone to unexpected errors. In creating this lab, the difficulties in building our robots were plentiful and multivariate, ranging from serial communication errors, servo motors with insufficient mechanical advantage, and even time mismanagement spent attempting flawed design ideas. These errors were unavoidable and ultimately led to experience gained and lessons learned. Overall, this lab was instrumental in working out the kinks of hardware/software troubleshooting, scheduling conflicts, and design group personalities. This lab was challenging and worthwhile as a first consideration in the world of robotics. This shortcoming in communication and a lack of preparation highlight the difficulties in working as part of a team as well as the importance of being prompt and organized.