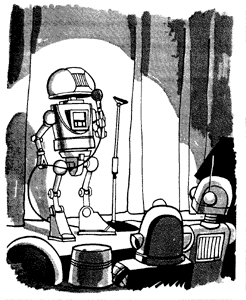
Introduction to Control Systems

Laboratory Section Manual



*“Welcome to GE 320…..”*

# Introduction

This section summarizes the course content and outlines the general procedure and reporting requirements.

## Goals of the Labs 1 through 5

* Gain a familiarity with standard laboratory instruments including the multi-meter, DC power supply, and Raspberry Pi
* Learn MATLAB and Simulink – the PC software we use in the lab.
* Gain working experience with DC motors, the direct measurement of their physical properties for use in deriving a transfer function, and the indirect measurement of the motor transfer function using frequency response methods.
* Design and implement the control for an actuator-positioning device.

## Introduction to Control Systems

Nowadays, more machines are controlled without any human assistance. This is not an easy task; therefore many theories have been proposed in order to achieve this goal. One of the most reliable theories is Feedback Control. The basic idea behind Feedback Control is to design and construct a piece of equipment capable of forcing the system to behave the way one wants it to. This piece of equipment is called a controller (or compensator). Depending on the system one intends to control, the controller will vary from a gearbox, to an analog circuit, to a digital computer. Controllers for the most part, use sensors to measure the actual behavior of the system (output), and compare it to the desired behavior (reference). Using this information, controllers react accordingly to manipulate the system in such a way the system behaves as desired.

Plant

Control Effort

error

Output

Reference

Controller

Sensor

Controllers are built from mechanical and electrical building blocks. Mechanical controllers are usually implemented with shafts, gears, pulleys, lever arms etc. Electrical controllers, on the other hand, are subdivided into digital and analog controllers. Digital controllers are implemented with computers and digital circuitry, while analog controllers are implemented with summers, integrators and differentiators. One of the objectives of this lab is to teach the student how to use op-amps, capacitors and resistors to create these summers, integrators and differentiators. Furthermore, the student will be introduced to two basic sensors: the potentiometer and the tachometer. These sensors can be used to determine a DC-motor’s position and speed.

## Equipment Introduction

This introduction section to the lab is intended to give you a feel for some of the instruments you will be using in the next few weeks. Let us start with identifying the equipment:

* Locate the following apparatus on you bench and refer to the equipment information sheet for a brief description:
  + - * Raspberry Pi
      * Interface Circuitry
      * Multimeter
      * HP power supply
      * Motor, potentiometer, tachometer

## Measurable Events and Electric Signals

Any measurable event can be mapped to an electric signal. This mapping is usually done with sensors. Sensors are electronic devices, which measure a physical event and output a proportional electrical signal.

## Electronic Devices and Parts

There are several circuits required to sensors and DC motor to the Raspberry Pi.

* H-Bridge: Converts the digital outputs of the Raspberry Pi to a voltage to drive the DC motor.
* Analog-to-Digital Convertor (ADC): Converts the analog voltage signal to a digital signal that can be read by the Raspberry Pi.
* Infrared LED and Receiver: When the encoder wheel blocks the light from the receiver the Raspberry Pi receives a low signal, otherwise it receives a high signal.

## Control Systems Building Blocks

Software controllers implement differential equations by using mathematical operations. The most used operations to solve these equations are: Summers, Differentiators, and Integrators.

## Electrical Sensors in the Lab

Electrical sensors are devices, which convert physical phenomenon into electrical signals.

* Rotational Potentiometers: Voltage varies as a function of rotational position (angle). (Figure 1.9.1).
* Tachometer: outputs a square wave where the frequency varies as a function of the motor’s speed (Figure 1.9.2).

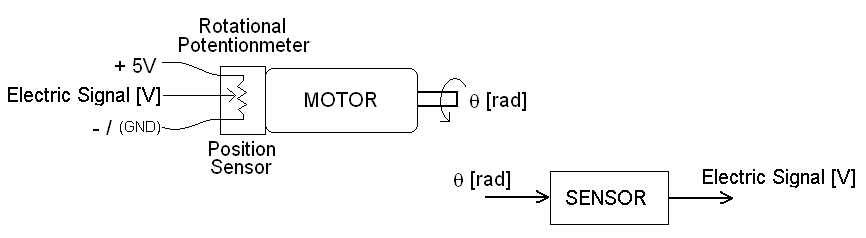
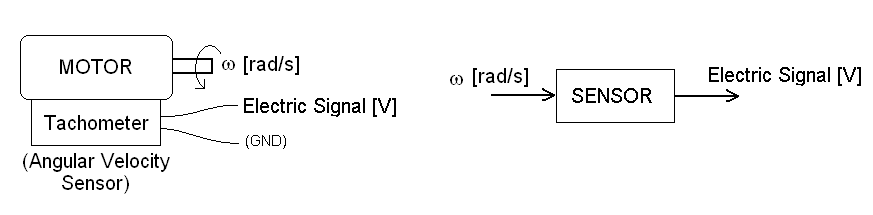


Figure 1.9.1



**Digital Signal**

Figure 1.9.2

# Lab 1: Tachometer, Potentiometer, Motor Dead zone

## Introduction

The student will be introduced to two basic sensors: the potentiometer and the tachometer. These sensors can be used to determine a DC-motor’s position and speed. Finally, the students will be introduced to the DC-motor’s dead zone and how to deal with it.

## Lab 1: Pre-Lab

1. With your own words, how would you define and describe a sensor.
2. What does “sensor calibration” mean?
3. What is a potentiometer?
   1. Find at least 2 different kinds of potentiometers and explain how they work.
   2. The potentiometer in the lab has 330 degrees of electrical travel and 3.3V will be applied across it. What is the potentiometer’s gain in V/rad
   3. What do you expect the output voltage from a circular potentiometer (spinning at constant angular velocity) to look like?
   4. Could a potentiometer be a sensor? Why?
4. What is a Tachometer?
   1. What are the units of input and output of a tachometer?
   2. If the period of one revolution of a motor is 0.3sec and the encoder wheel has 20 evenly spaced holes, what is the period of the square wave produced?
   3. Could a tachometer be a sensor? Why?

## Calibration and Dead Zone

This section of the lab focuses on taking measurements to establish the dead zone of the motor, calibrate the potentiometer and the tachometer.

### Motor Dead Zone Estimation

The motor requires a certain amount of power before it can rotate freely. We will find this value by varying the amount of current and voltage we apply to the motor.

* In HP VEE, open the file: “[***n:/labs/ge320/exp1/deadzone.vxe***](file:///C:\f:\labs\ge224\exp1\deadzone.vee)”
* Connect the HP variable power supply to the motor, from the patch panel.
* Turn the HP power supply on.
* Click the “Reset” button and set the “output enable” to “off”. This controls the HP power supply that is output on the patch panel, which you must now connect to the motor. Where is the HP power supply?
* Set the maximum current to 500 milliamps and turn “output enable” on.
* Gradually increase the voltage from 0 in 0.2 V increments. At each step, look at the motor and check for smooth, continuous unimpeded rotation. Show this to your TA and record the voltage and current values on the data sheet.
* Turn off the output enable.
* This is the motor dead zone in Volts and was measured with a limited current of 0.5 amps. We now consider the dead zone measured in amps:
* Set the voltage to 5 times the voltage dead zone value, but not more than 12 V.
* Set the current to 0 and increase it in steps of 0.01 amps. (Turn the power on first!) Continue until the motor rotates in a smooth, continuous and unimpeded fashion. Record these values.

### Calibration of Potentiometer

As you discovered in the prelab, the potentiometer is a variable resistor (usually carbon deposited) or variable voltage divider. The “slider” of the potentiometer has a friction contact to the potentiometer’s base resistor. Movement of the slider allows for a variance in the resistance from the slider to the end points of the potentiometer. Due to several factors a potentiometer may not be linear with angular displacement of the potentiometer shaft. We will be examining this non-linearity.

* Open MATLAB by double clicking on the desktop icon.
* You will need to enable the SPI interface on the Raspberry Pi. Type the following commands at the MATLAB command line:
  + mypi = raspi(‘ipaddress’,’pi’,’raspberry’)
    - Where ‘ipaddress’ is the IP address on the label affixed to your kit.
  + mypi.enableSPI
  + mypi.AvailableSPIChannels

If the above commands have executed correctly you should see:

‘CE0’ ‘CE1’ printed in the command window.

* Save a copy of the Simulink file: “[***n:/labs/ge320/kit/exp1/***](file:///C:\f:\labs\ge224\exp1\pot_cali.vee)***Lab0.slx***” on the hard drive of the computer (c:/matlab/ge320). You should create a folder within that directory to save all of your Simulink models throughout the semester.
* Change the working directory of MATLAB to the folder you just created. To change the directory click on the folder icon next to the path at the top of MATLAB window and browse to the folder.
* Open the Simulink file you saved on the hard drive.
* Next you will need to generate the code for the SPI interface. Double click on the box labels MCP3002Interface. After the dialog box opens, click Build in the upper right corner. Once the Compilation diagnostics window indicates the build was successful you can close the dialog box.
* Go back to the HP VEE program you used for the dead zone part of the experiment. Set the voltage to 6 V and the Current to 0.5 A, but do not turn it on yet.
* Press the green play button in Simulink. This compiles your code and deploys it to your Raspberry Pi.
* While Simulink is compiling and sending the code, turn on the power supply in the HP VEE program.
* Once Simulink has finished running the model, turn off the power supply. Then double click on the scope block to view the output. The first subplot will show the value of the potentiometer. It has already been scaled to radians for you.
* Sketch the shape of the waveform on your data sheet.

### Calibration of Tachometer

As you discovered in the pre-lab, a tachometer produces a square wave whose frequency is directly proportional to the angular velocity of its shaft. In this lab we will measure the proportionality constant or gain in units of voltage / unit angular velocity.

* Leave the equipment as setup above.
* Now look at the bottom two subplots of the scope. The plot in the middle is the square wave produced by the encoder wheel, Infrared LED, and receiver. There should be 20 periods of the square wave for each rotation of the motor. The bottom plot is a conversion to the speed in rad/sec.
* Make sure there are 20 periods of the encoder signal in every period of the potentiometer signal.
* Calculate the speed in radians per second based on one period of the potentiometer output and compare to the value in the bottom plot.
* Now change the voltage to 7V in the HP VEE program and turn the power supply on. Then click the play button in Simulink.
* Repeat the above steps in 1V increments up to 10 V. Show the TA.

Disconnect all the wires, and put them back where you found them, log off the computer, turn off all the equipment, turn off the bench power supply, put your chair at the desk and go home.

## Lab 1: Post Lab

Include the answers to the following questions in your lab report

1. What is the difference between the functionality of a tachometer and a motor?
2. What causes the motor to have a dead zone?
3. Why is it that when the voltage was set to five times the dead zone, but the current was limited, the motor did not have unimpeded, continuous rotation?
4. Based on this experiment, what do you think causes the motor to spin?
5. When you observe the potentiometer output, what is the shape of the output? Include a sketch of the potentiometer output.
6. Is the potentiometer linear? Why is there a small flat section at the bottom of the waveform? How does the motor speed affect the waveform?

# Lab 1: Data Sheet

| **Motor Dead zone** | | | |  |
| --- | --- | --- | --- | --- |
| Voltage Dead Zone | | Current Dead Zone | |  |
| Voltage |  | Voltage |  |  |
| Current |  | Current |  |  |

| **Tachometer Calibration** | | | |
| --- | --- | --- | --- |
| Voltage | Period | Angular Velocity | Tachometer Output |
| 6 V |  |  |  |
| 7 V |  |  |  |
| 8 V |  |  |  |
| 9 V |  |  |  |
| 10 V |  |  |  |

\*\*Include a sketch of the potentiometer output