Experiment-02: Potentiometer Data Acquisition and Pendulum Calibration

ME-330

Dr. Vibahv Durgesh

vdurgesh@uidaho.edu

Week-02

Contents

1 Learning objective

The objectives of this experiment are to provide the student with an opportunity to: 1) Gain additional experience with Matlab and Simulink. 2) Create a simple program for data acquisition using Simulink and National Instruments USB-6001. 3) Become familiar with breadboards, digital power supply, wiring, digital multi-meters, and other laboratory skills. 4) Calibrate a potentiometer-based angle measurement system and create a calibration plot. 5) Record and plot angle vs. time data for a swing motion of the pendulum system.

- Gain additional experience with Matlab data acquistion toolbox.
- Create a simple program for data acquisition using Matlab and National Instruments USB-6001.
- Become familiar with breadboards, digital power supply, wiring, digital multi-meters, and other laboratory skills.
- Calibrate a potentiometer based angle measurement system and create a calibration plot.
- Learn more about pendulum system and
- Plotting data plot angle vs. time data for a swing motion of the pendulum system.

2 Introduction

In this experiment, you will connect a simple angle sensor (angular potentiometer) to the NI USB-6001 data acquisition device and collect digital data using a Matlab data acquisition toolbox. First you will you will collect data to create a calibration equation, then use the information from the calibration to convert the output voltage from the angular potentiometer to the pendulum angle.

3 Part-A

Creating data acquisition program using Matlab data acquisition toolbox
The Matlab program is provided in the appendix section of the instruction. The main program sets the parameter for the acquisition and function file plots and record the data as data being acquired from the data acquisition device.

4 Part-B

Connect an angular potentiometer to the NI USB-6001 using a breadborad.

- With the power supply turned OFF, connect the power supply output to the Va and GND connections on the breadboard (see Figure). **Need figure** Use jumper wires to connect the power supply to the vertical power rails on the breadboard (see Figure). This will be used as the reference voltage for the angular potentiometer.
- Use a Digital Multi-Meter (DMM) to test the wire connections on the breadboard. Your lab TA will help explain the basic operations of the DMM and breadboard to you during lab.**Need figure**
- Connect the potentiometer leads labeled Power and Ground to the 5 Volt reference signal and the reference ground from the power supply.**Need figure**
- Connect the NI USB-6001 to the angular potentiometer leads labeled Signal and Ground. The red wire should connect to AI0+ and the black wire to AI0- as shown in figure .**Need figure**
  - The AI0+(Analog Input 0 positive) and AI0-(Analog Input 0 negative) screw terminals are differential inputs in that both the ground voltage and the measured voltage are input and then the DAQ determines the difference between them. The ground input in the DAQ should not be connected to ground on the breadboard.
  - When connecting wires to the screw terminals, put the wire above the metal tab and then turn the screw to the right. This will bring the metal tab up and tighten the wire in place.

5 Part-C

Run your acquisition program and testing your connections.

- Verify your circuit with the TAs
- Once your circuit is verified, run your acquisition program.
- Rotate the angular potentiometer and verify the results. The display should change with the movement of pendulum.

6 Part-D

Create a calibration equation for the potentiometer to output the angle of the pendulum.

- Determine a repeatable method for measuring each angle with the supplied protractor (or your phone).
- YOu will need to perform calibration for 10 or more angles.
- At each of 10 (or more) angles, ranging from ± 90 degrees, collect the voltage at a sampling rate of 20 Hz for 5 seconds. Save each of these files with the following name Ang_(p/m)XXDeg.dat where xx represent angle and p/m represent plus or minus sign. For instance, the calibration file for -45^o will be saved as “Ang_m45Deg.dat” and the calibration file for -60^o will be saved as “Ang_p60Deg.dat”.
- Repeat the above step for all the calibration angles.
  - Angle (in degrees). This should be zero when the pendulum is straight down.
- Plot the data collected in above step using Matlab. The x-axis should be the recorded average voltages and the y-axis should be the angle in degrees. Be sure to properly annotate your plot. **Use the instruction from lab-01**
- Use Matlabs polyfit tool to determine a linear fit to the calibration data. This is your calibration equation. **Make sure to save your data in case you need to repeat any analysis**.
- Record the norm of the residuals from the linear regression provided by Matlabs basic fitting tool. Use this to calculate the standard error of the fit.

Table 1: Sample calibration data for experiment 2.

Pendulum angle (degree)	Average output voltage (v)
-90°	
-70°	
-50°	
-30°	
-10°	
0°	
10°	
30°	
50°	
70°	
90°	

7 Part-E

Now add your calibration equation to acquisition program to display the pendulum angle in degrees during the data acquisition process.

- You will have to change the slope and intercept values in the provided Matlab program.
- Run your Matlab program without storing data and see if your program displays the correct angle.
- Show your program and results to the TAs to verify your program.

8 Part-F

Create a time series plot showing the pendulum swinging to rest from a horizontal starting position.

- Use the Malab program with the modified slope and intercept values.
- Change the acquisition time in the Matlab program to 15 seconds at a sampling rate of 200 Hz.
- You will need to save the data for this part. Change the output file name to "AngVsTime.dat"
- Make sure your program is modified to acquire and store the data file.
- Request TAs to validate your code prior to executing the code.
- Move the pendulum to the horizontal position,
- Execute the program and then release the pendulum from a horizontal position,
- If everything is done correctly, the program should display the angle (in degrees from your calibration equation) vs. time.
- Confirm the file is stored correctly.
- Open the stored data file and determine the period, T_d, of the response oscillations and the corresponding frequency, ω_d.

9 Part-G

Return lab space to prior condition

- Turn off the power supply.
- Remove wires connecting the power supply to the breadboard and return to the wall
- Remove all breadboard wires and place them back in the wire kit in an organized fashion.
- Remove the pendulum wires from the breadboard and set aside.
- Log off the computer.
- **Do this for every lab!**

10 Instruction

- Make sure to name the files your FirstName_ LastName_ QuestionNumber, use the format provided in the sample Matlab code. Make sure the submitted figure has the title. Include the apostrophe!
- Make sure to submit your post-lab assignment on the BBLearn website.
- Here are some details to help.
  - Plot the experimental data using red circles with MarkerSize 8. Experimental data points should always be plotted using markers without lines connecting the points
  - Plot curve from theory (or curve fit line) using a solid colored line with LineWidth 2.
  - Set the x and y-axis limits for each figures. Follow the instruction.
  - Make sure the major grid lines are visible
  - All plot text should be in Times font. You will need to specify this for the title, labels, and legends. The axis labels and numbers should be in 10-point font
  - The title should be in 10-point font
  - Make sure to add the legend. Keep in mind that legend should not hide the plotted data
  - For full credit make sure that your submitted files look similar to the sample figures shown in the document
  - If in doubt make sure to request the TAs for help

11 Post lab (50 points)

Postlab are due at Friday midnight. For this experiment submit the following items on BbLearn for your post-laboratory assessment.

1. Calibration plot. Submit a calibration plot showing the raw data points as markers and the calibration curve as a line. The plot should be 6.5 wide. Make sure to properly annotate your plots: axis labels, titles, legend, etc. Using the text command in Matlab, place the equations or numbers listed below on the plot. Use Greek symbols where appropriate. Make sure to include units.
   - The calibration equation on the plot with 4 decimals places for each number.
   - The norm of the residuals of your linear regression.
   - The standard error of the fit for your calibration equation.
2. Pendulum Swing Plot. Submit a time series plot showing the trajectory of your pendulum swinging from a horizontal starting position. The plot should be 6.5 wide. Make sure to properly annotate your plots: axis labels, titles, legend, etc. Using the text command in Matlab, place the equations or numbers listed below on the plot. Use Greek symbols where appropriate.
   - The period of the oscillations, T_d.
   - The frequency of the oscillations, ω_d.
3. Answer all question in the post-lab assessment on BbLearn

12 Appendix

Sample code for data acquisition using Matlab

```
% This is a genric program to acquire the data from NI- 6001
%% Sample data acquistion program
% Date: July 19th
% Dr. Vibhav Durgesh
% Rev 0.0
% This is basic program to acquire data from a data acqusition device.
% User has to provide apropriate information see - begining of the code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THIS CODE OVERWRITE THE DATA. PLEASE MOVE THE FILES OR REMANE PRIOR TO
% RERUNNING THE CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clc

%% ------- START USER REQUIRED INFORMATION --------%%%%%%%%%%%%%%%%%%%%%%%%
outputFileName = 'Test.dat'; % This program will overwrite the file
slope = 1; % Slope for converting voltage to appropriate unit
intercept = 0; % Intercept for converting voltage to appropriate unit
Fs = 30; %Sampling rate Data per sec
T  = 3; % Time to acquire data
choice = 0; %% Set to 1 to write data to a file otherwise 0
%% ----------END USER REQUIRED INFORMATION ----------%%%%%%%%%%%%%%%%%%%%%%%%
d = daq.getDevices;
s = daq.createSession('ni');
addAnalogInputChannel(s,'Dev2', 'ai0', 'Voltage');
s.Rate = Fs; % Sampling rate modify as required
s.DurationInSeconds = T; % Sampling time modify as required
if choice == 1
    fid1 = fopen(outputFileName,'w'); % File name to store your data
    fprintf(fid1,'%s \t %s \n','Time(s)','Volt (v)');
else
    fid1 = 0;
end
lh = addlistener(s,'DataAvailable', @(src, event)plotAndLogData(src, event,
    fid1,slope,intercept,choice));
errorListener = addlistener(s, 'ErrorOccurred', @(src,event) disp(getReport(
    event.Error)));
drawnow
startBackground(s);
pause
delete(s)
delete(lh)
if choice == 1
    fclose(fid1);
end
```

```
function plotAndLogData(src,event,fid,m,c,choice)
time= event.TimeStamps;
voltage=event.Data;
plot(time, voltage,'k.','markersize',6)
title('Press any key when data acquistion is done','fontsize',14)
hold on
xlabel('time (s)')
ylabel('voltage (v)')
if choice == 1
    for ii = 1:length(voltage)
        fprintf(fid,'%3.5f \t %3.5f \n',time(ii),voltage(ii)*m + c);
    end
end
end
```