ME 3300 Lab-01: Potentiometer Data Acquisition and Calibration

Dr. Christopher Bitikofer

Contents

1 Learning objective

The objectives of this experiment are to provide the student with an opportunity to:

- Gain additional experience with Matlab.
- Create a simple program for data acquisition using toolbox and National Instrument's NI-MyDAQ.
- Become familiar with breadboards, digital power supply, wiring, digital multi-meters, and other laboratory skills.
- Perform calibration of a MEMS accelerometer and create the calibration plot.
- Compare measured normal acceleration to acceleration calculated from angular velocity during fixed axis rotation.

2 Required Equipment

1. USB Data Acquisition Device (NI-MyDAQ)
2. Breadboard (MyDigital protoboard)
3. Pendulum with Potentiometer (10kΩ Bourns 6639s) and end-mounted accelerometer (MMA7361LC).
4. Needle-nose pliers & wire strippers
5. Small flathead screwdriver
6. Digital multimeter (DMM)
7. Power supply and wire kit

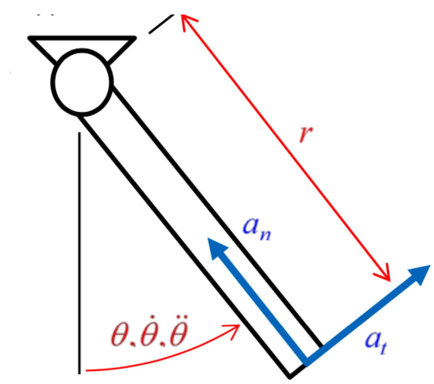


Figure 1: Schematic of the experimental setup.

3 Introduction

During rotation about a fixed axis, the normal acceleration of a point can be calculated from the angular velocity of the rotation motion according to the following formula:

$$a_n = -\dot{\theta}^2 r \tag{1}$$

4 Laboratory instruction

In this experiment, you will build on your work from the previous week. In addition to measuring the pendulum angle with a calibrated potentiometer, you will also measure normal acceleration of the end of the pendulum using a MEMS accelerometer. The schematic is shown in figure 1. Start by setting up the NI MyDAQ to acquire data from the potentiometer and accelerometer. Next, you will calibrate the y-axis of the accelerometer. Finally, we will compare the measured normal acceleration of the pendulum end to the normal acceleration calculated from the derivative of the pendulum angle.

4.1 Data Acquisition with MATLAB's Data Acquisition Toolbox

Utilize a similar MATLAB Data Acquisition Toolbox as experiment-02, but with the multichannel function activated. **MATLAB_Data_Acquisition.m**, is available on Canvas. The following instructions will guide you through preparing your own data acquisition programs.

5 Part 1: Prepare your MATLAB Program to Display and Record Data.

1. Launch MATLAB and prepare your code
   - A main acquisition script **MATLAB_Data_Acquisition.m** is posted on canvas along with two supporting functions **plotdata2.m** & **logData2.m**
   - Toggle "inputChannels" between single channel and multiple channel to initially calibrate the accelerometer and then collect the potentiometer data from "ai0" and accelerometer data from "ai1".
   - You need to set the parameters for data acquisition in the main program. They are set near the top of the file.
2. Set the value of Fs to set the sample rate (the number of samples to collect per second).
   - The MyDAQ can collect up to 200,000 samples per second, but this would produce a massive data file & we don't need that much resolution to observe the pendulum swing.
   - You will set this value differently for each part of the lab. Note what settings are best for different tasks.
   - For the final data acquisition, set the sample rate to 1000 samples/sec. We will be calculating a discrete derivative of the pendulum angle, and a higher sample rate will reduce the noise from the process.

   Q Why does a higher sample rate reduce the noise in the calculated derivative? Why does it increase the size of data files?
3. Set T to control the duration of your acquisition. This tells the DAQ how much time in seconds to collect data.
4. **inputChannels** controls the collected input signals. An example of collecting 2 channels is provided.
5. The two boolean variables **enableLogging** & **enableLogging** control if statements later in the code.
   - These turn on\off listener functions **logData.m** & **plotData.m**.
   - In other words, these are switches you can use to set your code to acquire data without saving it (useful for debugging your circuit) or acquire data without plotting it (can save CPU time, but is risky since you might not notice if something goes wrong with your recording).
6. **Filename** & **header** set the file location for writing data, and the text to use for header columns.
   - You need to make sure to create the **.csv** file to write data to in advance of running this code.
7. Please use the calibration equation from the previous laboratory experiment. If in doubt, please get in touch with the TAs for the correct calibration. The calibration should be able to convert the potentiometer output voltage to radians. Program this equation into the MATLAB acquisition program. Verify that the calibration is producing the correct output by moving the pendulum to several known angles.

Q: What is the difference in functions **plotdata2.m** & **logData2.m** different from **plotdata.m** & **logData.m**?

5.1 Part 2: Building the Accelerometer & DAQ Circuit

Connect the y-axis accelerometer output to the data acquisition using a breadboard.

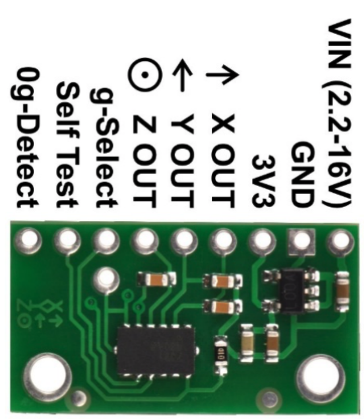
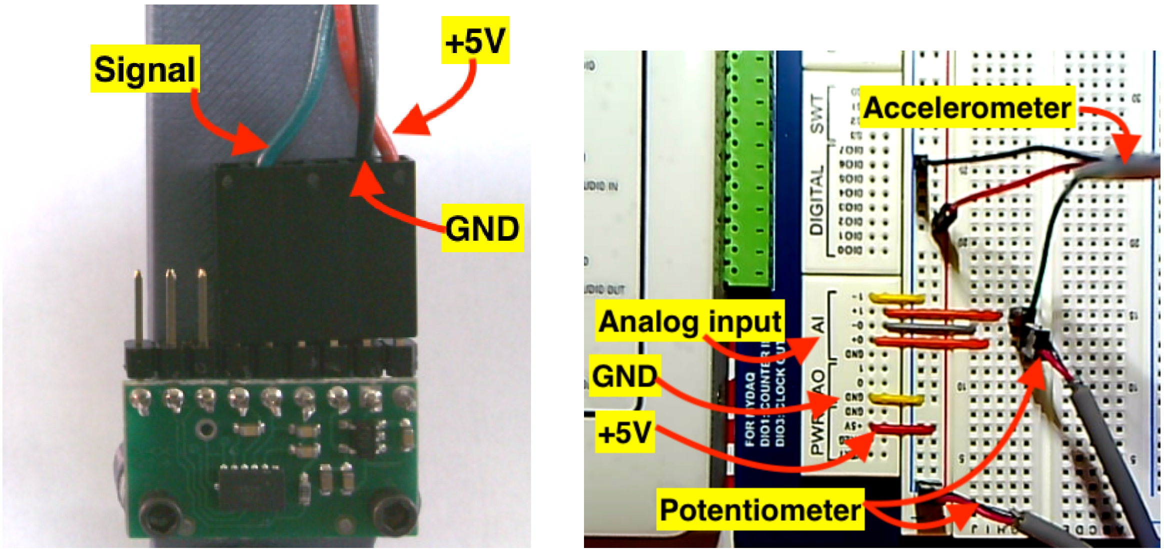


Figure 2: The MMA7361LC 3-axis accelerometer with voltage regulator.

1. For this experiment, a 3-axis accelerometer has been fixed to the end of the pendulum apparatus (see Figure 2). The accelerometer has been configured for ±6g operation. The y- axis of the pendulum points towards the pendulum rotation axis (see Figure 3). For more information about the accelerometer, see Pololu.
2. Connect the accelerometer to the breadboard using the provided 3-wire cable. See figure 3(a).
3. Connect the V_{in} and GND pins on the accelerometer to the breadboard 5 Volt supply power (already used to power the potentiometer). See figure 3(b) for reference.
4. Connect the y-axis accelerometer output to AI1 (Differential Analog Input 1). See figure 3(b).
5. Connect the DAQ GND, instead of the AI1-, to the ground on the breadboard. See figure 3(b).
6. With the tape measure, note in your logbook the length of the pendulum (in meters).



(a) Sample connection to the accelerometer.

(b) Sample connection to the DAQ system using Digilent board .

5.2 Use your MATLAB program to Test the Circuit and Data Acquisition

To verify that your circuit is implemented correctly, test it by rotating the potentiometer and confirming the voltage changes as expected at each position.

1. Adjust Duration
   - Change Duration to a value that fits your testing needs:
     - Use a short duration for quick checks.
     - Use a longer duration if you want data collection to run continuously. Note there isn't a built-in way to interrupt collection, so use good judgment!
2. Run Data Collection Program
   - Run your MATLAB program by clicking the *run* button arrow.
3. Observe Output While Rotating the Pendulum
   - While the program is running:
     - Rotate the angular potentiometer.
     - Observe the real-time output on the display figure window.
     - Keep in mind that the DAQ system takes a few seconds to initialize. You might notice a delay while the simulation starts.
4. Verify Smooth Voltage Response
   - Ensure the voltage changes smoothly over the 360° range.
   - Check to see if voltage for accelerometer at $-90°$ and $90°$ are the same.

- If you notice a sudden jump or drop in voltage:
  - The potentiometer may be misaligned.
  - To fix it, ask your TA for help, they can with loosen the grub screw, rotate the potentiometer until the voltage varies smoothly (jump should occur around +180°), then re-tighten it.
  - Re-test to confirm smooth output over the test range.

  Q The accelerometer has x-, y-, and z-axes, but in this lab, we only connect the y-axis output. When the pendulum swings, which component (normal or tangential) of acceleration is the y-axis actually sensing, and why is this the most useful axis for comparing to the calculated normal acceleration?

5.3 Part 3: Calibration

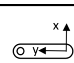
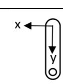
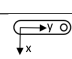
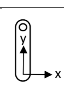
When the pendulum is held still, the only acceleration the accelerometer feels is gravity. The voltage read by the accelerometer can be linearly related to acceleration. In this step, you will perform a calibration using four positions corresponding to the y-axis accelerometer voltage outputs.

1. Create the table below in your logbook by rotating the pendulum to the four shown positions and recording the output voltages on the y-axis of the accelerometer.
2. Collect the single-channel voltage output from the accelerometer ("AI1").
   - Duration: 20 seconds
   - Sample rate: 20 samples per second
3. Record data at each angle. Each run:
   a. Run your program:
      - Save a data file using this naming convention: **Ang_(p/n)XXDeg.csv** where
        - `XX` indicates angle.
        - `p/n` indicates sign (positive or negative).
        - For example: for −90°, use **Ang_n45Deg.mat**.
      - Update the file name before each run to avoid overwriting data.

   b. Record the angular position (in degrees):
      - Reference experiment-02 to match angular position.

   c. Compute the Average Voltage:
      - Write a MATLAB script to calculate the mean voltage from each data file.
      - Record the result with at least two decimal places.
4. Plot the data points from the above table in order to create calibration equations for the y-axis acceleration measurements. Specifically, Plot ay_g vs. $ay_{v,avg}$ to find y-axis acceleration (in m/s^2) as a function of sensor voltage.

Table 1: Sample calibration data for experiment 2.

Pendulum orientation	Schematic	y-axis acceleration ay_g (m/s^2)	Average output voltage $ay_{v,avg}$ (v)
Right		0.00	
Up		9.81	
Left		0.00	
Down		-9.81	

5. Plot Your Calibration Data

- Use MATLAB to create a scatter plot:
  - X-axis: average voltage
  - Y-axis: angle in degrees
- Change $t_{\nu,p}$ value in the MATLAB script.
- Format the plot clearly and include appropriate axis labels, title, and units.
- Refer back to formatting guidelines from Lab 01 and the requirements below under *Post Laboratory Assignment*.

6. Fit a Linear Calibration Model
   - Use the **polyfit** command in MATLAB to compute a linear regression of the form:
   $$y = a_0 + a_1 x$$
   . where y is the pendulum angle (degrees) and x is the measured voltage (V).
7. Ensure Fit Quality
   - Record the **norm of the residuals** from **polyfit**.
   - Use this value to calculate the **standard error of the fit,** s_{yx}.
8. Create and Save Calibration Plot
   - Refer to Lab-02 to modify the calibration plot code you created previously.
   - Save your plot with **at least 600 dpi resolution.**
   - Reference figure 3 for proper figure creation.
   - **Q** When the pendulum is held still at each test angle, why should the accelerometer voltage be proportional to the component of gravity along the y-axis? What does this tell you about why a linear calibration equation makes sense? Would a linear fit make sense for a calibration of pendulum angle to voltage?

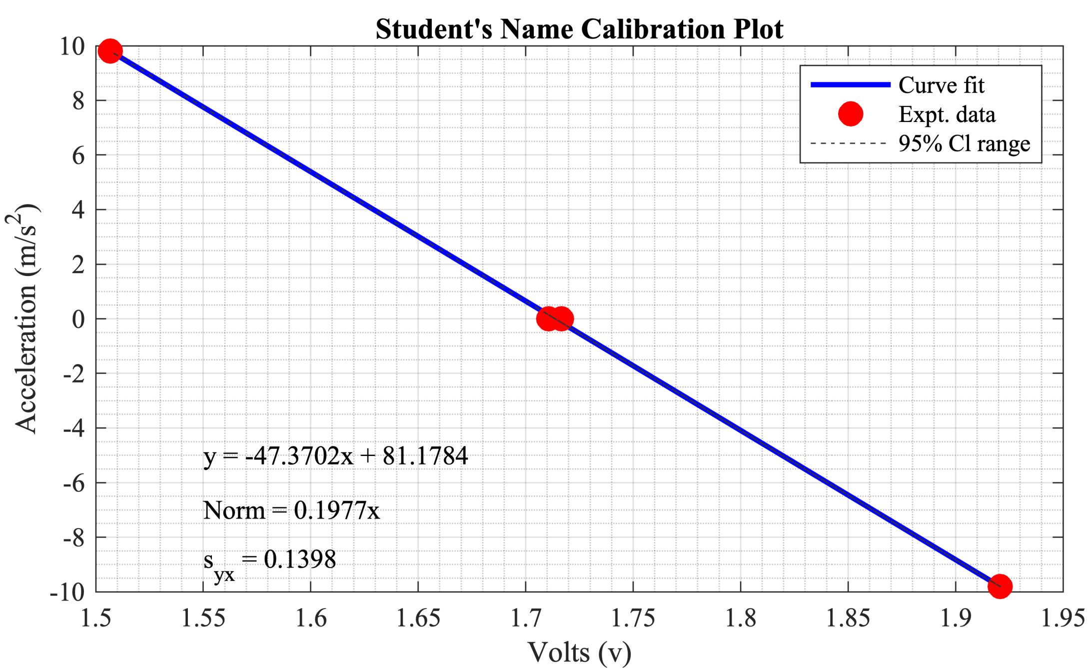


Figure 3: Calibration of the accelerometer.

5.4 Part 4: Modifying acquisition code

Now add your calibration equation to the acquisition program to display the pendulum angle in degrees during the data acquisition process and the acceleration values in m/s^2.

1. Toggle the multi-channel acquisition: potentiometer (“ai0”) and accelerometer (“ai1”).

- Duration: 20 seconds
- Sample rate: 1000 samples per second

2. Make sure the output file is named as "Expt_03_FirstName_Lastname.csv"
3. Change the slope and intercept values in **MATLAB_Data_Acquisition.m**.
4. Run the program and record a full pendulum swing.
   - Start your MATLAB program and release the pendulum from a horizontal position (90°).
   - Check to see if recorded data of the angle (in degrees, from your calibration equation) and acceleration (in m/s^2, from your calibration equation) versus time.
5. Show your program and results to the TAs to verify your program. **If the program is working properly, proceed to collect data.**

Q After adding your calibration equation to the MATLAB code, what's the advantage of seeing the pendulum angle in degrees and acceleration in m/s^2 instead of raw voltages during data collection?

5.5 Part 5: Normal acceleration experiment

Procedure for collecting data.

1. Move the pendulum to the horizontal position (90°).
2. Start the acquisition program.
3. Release the pendulum.
4. Ensure the pendulum comes to a complete stop before your acquisition program ends.
5. Make sure the plot shows pendulum angular location and pendulum normal acceleration.
6. Open your output file and check the data.
7. Output file should have three columns of data. The first column will have information about time, the second column will have the angular position of the pendulum in degrees, and the third column will have the acceleration value of the accelerometer in m/s^2.

5.6 Part 6: Plotting data

Post processing of the data.

1. Download **PostProcess.m** which has the following skeleton code prepared:
   - Load your recorded **.csv** file in MATLAB.
   - Convert the angle value in radians.
   - Remove the acceleration due to gravity from the accelerometer data.
2. Use Lab 01 and Lab 02 to plot the normal acceleration data from the accelerometer and normal acceleration using equation 1 in a single plot.
3. Plot the noisiest data first (this way the other data will remain visible).
4. Sample plot is shown in Figure 4.
   - Here you have to use free-body diagram to identify the contribution from acceleration due to gravity
   - Use the angular value for pendulum to accurately identify the position of the pendulum.
   - The result will be the measured normal acceleration of the pendulum due to angular velocity.

Q: When you plot normal acceleration from the accelerometer and from angular velocity, they won't match perfectly. List at least two reasons why differences might appear (for example, sensor alignment, noise, or assumptions in the equations). Why is the acceleration from the angular position noisier than the accelerometer data?

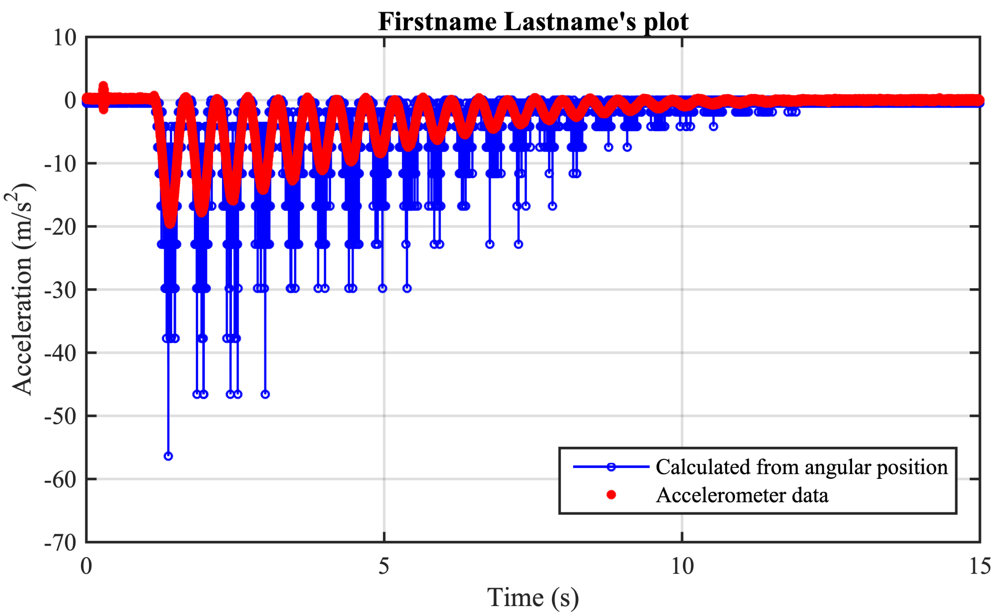


Figure 4: Sample post-process plot.

Part 7: Return Lab Space to Prior Condition

1. Remove all breadboard wires and place them back in the wire kit in an organized fashion.
2. Remove the pendulum wires from the breadboard and set them aside.
3. Confirm your data files are saved to OneDrive.
   a. Check that files are available online using your phone or personal laptop to confirm.
   b. Make sure that all team members have access to experiment files, including data and scripts.
4. Log off the computer.

6 Post-Lab Assignment

For this experiment, submit the following items as PDF files on Canvas:

1. **Accelerometer Calibration Plot**
   - The calibration plot should include:
     - Using "scatter" plot fuunction, raw data points as red circle markers with `MarkerSize = 90`.
     - A solid blue line showing the linear calibration curve with a 2 pt linewidth.
     - Dashed black lines showing the 95% confidence interval (compute using the standard error of the fit, s_{yx}).
   - The plot should be 5" tall and 6.5" wide.
   - Properly annotate your plot: axis labels, title, legend, etc.
     - Use title: "FirstName LastName's Calibration Plot".
   - Set axis grid lines on, box off, and figure background color to white.
   - Using the `text` command, place the following on the plot (use Greek symbols where appropriate):
     - The calibration equation with 4 decimal places for each number.
     - The norm of the residuals of your linear regression.
     - The standard error of the fit for your calibration equation.
     - Include units where appropriate.
2. **Normal Acceleration Plot**

- Submit a time series plot showing normal acceleration vs. time calculated in the two ways described in Part 6.
  (a) Normal acceleration measured from the accelerometer vs. time.
  (b) Normal acceleration calculated from the equation $a_n = \dot{\theta}^2 r$.
- The plot should be 5" tall and 6.5" wide.
- Properly annotate your plot: axis labels, title, legend, etc.
  - Use title: "FirstName LastName's Normal Acceleration Plot".

3. **Answer All Questions in the Post-Lab Canvas Assignment**
   - Compare your plots to the example plots shown in Figure 4 to confirm that you have completed the assignment correctly. Note that your values will differ, but your formatting and annotation should match.