Sam Wisnoski

Introduction to Sensors, Instrumentation, and Measurement

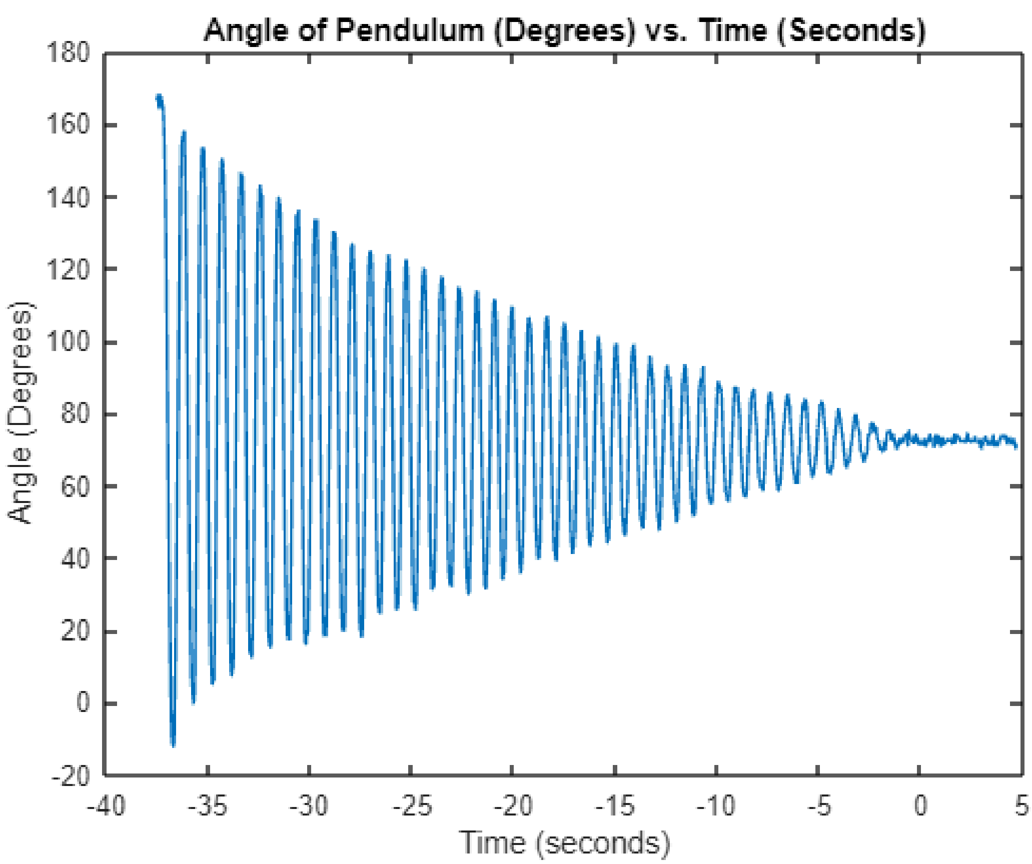
01/25/2024

Lab One: Pendulum

Purpose:
Measure the change in pendulum angle with time, θ v. t, using a potentiometer as a position sensor.

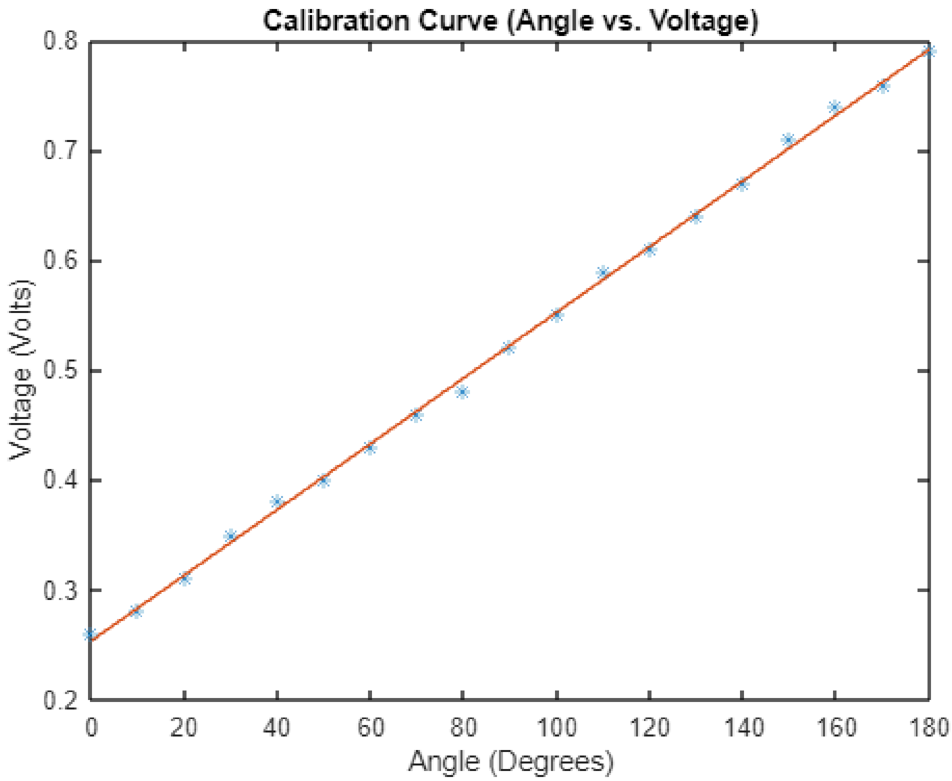
Results:

1.) Graph of the pendulum motion (θ v. t):



Time v. Angle for the pendulum. A 1 kΩ potentiometer was used as an angular position sensor in a voltage divider circuit with 1 V DC input. The potentiometer was connected to a pendulum, with 90 degrees as its resting state.

2.) Calibration curve for $V(\theta)$:



Calibration curve for voltage vs potentiometer angle. The angle readings come from a pendulum attached to 10 kΩ potentiometer, measured using a protractor. The voltage is the output of the voltage divider with a 1 V DC input that corresponds to each potentiometer angle.

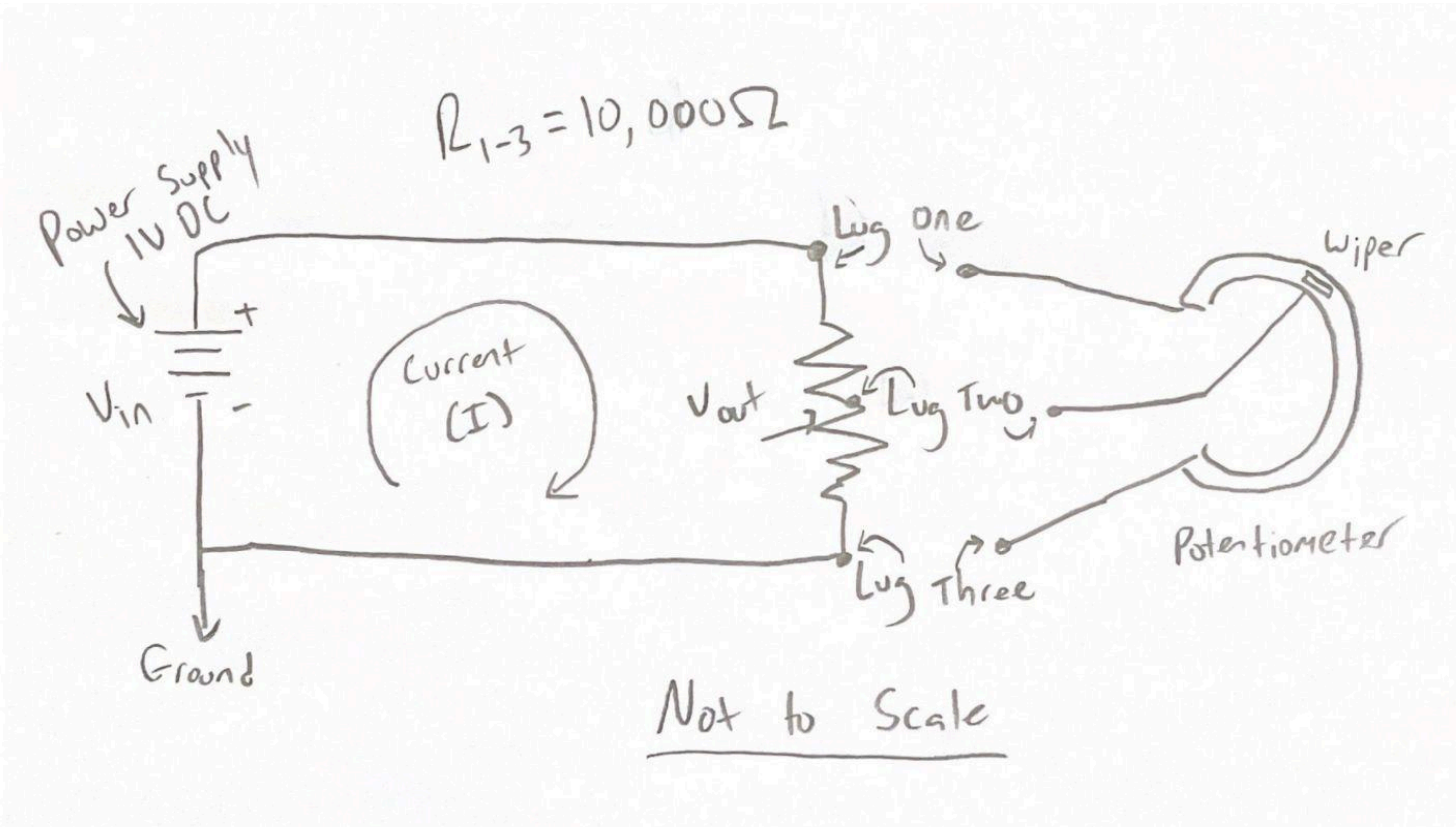
The calibration curve, created in MATLAB using polyfit, is:

$$f(x) = 0.003*x + 0.2533$$

In order to convert voltage to angle, use the transfer function:

$$\text{Angle} = (\text{Voltage} - 0.2533)/0.003$$

3.) Circuit diagram for the instrumentation circuit:



4.) Code used for producing the pendulum motion graph:

```
Unset
clear; clf; % clears memory - useful if you run many scripts in the same session

%take in data from csv file
tname='Pendulum_Data.csv.csv'; % <-input your data file's name! test2.csv was
exported from O-scope software with a line of headers
datatable = readtable(tname);%makes data table with headers

%interpret data as times and voltages
time1 = datatable.t1; % stores the t1 column of data in a variable called time1
V1  = datatable.ch1; % stores the ch1 column of data in a variable called V1

%plot (you can change this section to suit your plotting needs!)
plot(time1,V1) %plots first channel
xlabel('Time (seconds)'); % add x axis label
ylabel('Voltage (V)'); % add y axis label
hold off

Angle  = [0; 10; 20; 30; 40; 50; 60; 70; 80; 90; 100; 110; 120; 130; 140; 150; 160; 170;
180];
Voltage = [0.26; 0.28; 0.31; 0.35; 0.38; 0.4; 0.43; 0.46; 0.48; 0.52; 0.55; 0.59; 0.61;
0.64; 0.67; 0.71; 0.74; 0.76; 0.79];

plot(Angle,Voltage,'*'); % plot data as * points
hold on;    % hold the plot so that the next one will overlay

p = polyfit(Angle,Voltage,1);  % p returns 2 coefficients fitting r = a_1 * x + a_2
r = p(1) .* Angle + p(2); % compute a new vector r that has matching datapoints in x
% line of best fit:  r = 0.003 * Angle + 0.2533
% to convert voltage to angle: Angle = (Voltage - 0.2533)/0.003

% now plot both the points in y and the curve fit in r
plot(Angle, r, '-');
hold off;
title('Calibration Curve (Angle vs. Voltage)')
xlabel('Angle (Degrees)')
ylabel('Voltage (Volts)')

plot (time1,((V1 - 0.2533)/0.003))
xlabel('Time (seconds)'); % add x axis label
ylabel('Angle (Degrees)'); % add y axis label
title('Angle of Pendulum (Degrees) vs. Time (Seconds)')
hold off
```

Conceptual Ideas:

1.) Transfer function equation: Mathematical Relationship

As mentioned below the Calibration Curve graph, the transfer function is Angle = (Voltage - 0.2533)/0.003. This is a positive linear function. This means as the angle increases, the output voltage increases as well, portionally.

2.) Transfer function equation: Circuit Diagram Relationship

The transfer function equation makes sense. As seen in the circuit diagram, the wiper moves along the resistive track as the pendulum moves. As the wiper moves along the resistive track, the percentage of the resistance between Vin and Vout changes, since the wiper acts as a voltage divider. Therefore, as the wiper/pendulum moves, the voltage changes accordingly. Therefore, the relationship between voltage and angle should be linear.