Physics 238
Homework #2:
Torsional Oscillator Voltage Calibration

In[1]:=
```
Clear["Global`*"]; DateString[]
```
Out[1]=
```
Fri 28 Feb 2025 12:04:39
```

Problem 1: Linear Fits and Residuals

(a) Fit to full data set

In[3]:=
```
SetDirectory[NotebookDirectory[]] ;
```
In[4]:=
```
FilePrint["TO-static-20250212.csv", 5]
```
```
"Mass (g)", "Raw Angle", "Voltage"
0, 2.98, -0.0456
50, 2.79, -0.4083
100, 2.59, -0.7733
150, 2.38, -1.172
```

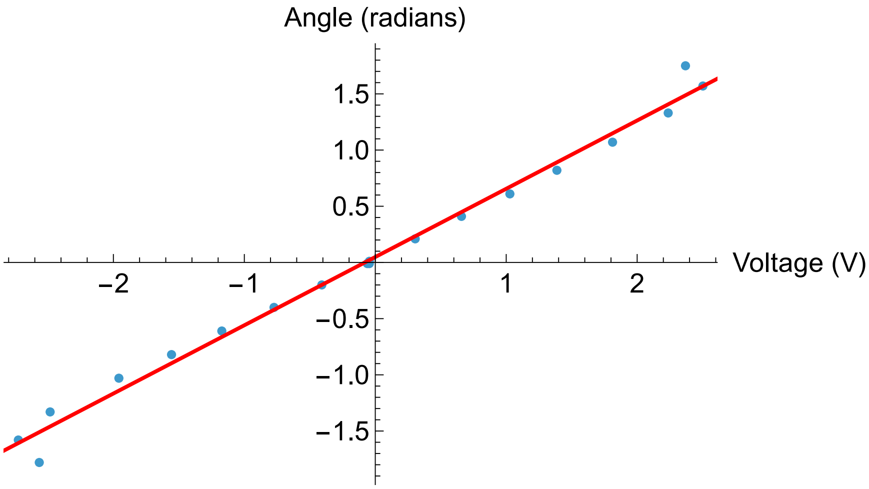
For this data, we want columns 2 and 3. Since Voltage is known more precisely, put it on the x axis. My value for the relaxed angle was 2.99 radians, so subtract that from all raw angles to get the angular displacement.

In[5]:=
```
rawdata = Import["TO-static-20250212.csv", "CSV"];
data = Select[rawdata, NumberQ[#[[1]]] && NumberQ[#[[2]]] &][[All, {3, 2}]];
data = Table[{data[[i, 1]], data[[i, 2]] - 2.99}, {i, 1, Length[data]}]
```
Out[7]=
```
{{-0.0456, -0.01}, {-0.4083, -0.2}, {-0.7733, -0.4}, {-1.172, -0.61},
 {-1.556, -0.82}, {-1.958, -1.03}, {-2.483, -1.33}, {-2.726, -1.58}, {-2.566, -1.78},
 {-0.0606, -0.01}, {0.3049, 0.21}, {0.6583, 0.41}, {1.028, 0.61}, {1.387, 0.82},
 {1.812, 1.07}, {2.237, 1.33}, {2.502, 1.57}, {2.369, 1.75}, {-0.04502, 0.01}}
```

In[11]:=
```
fit = LinearModelFit[data, x, x]
```
Out[11]=
```
FittedModel[ 0.0483 + 0.607 x ]
```

In[12]:=
```
dataPlot = ListPlot[data,
   LabelStyle → Larger,
   AxesLabel → { "Voltage (V)", "Angle (radians)" }, ImageSize → Scaled[0.7]];
```

In[14]:=
```
Show[{dataPlot, Plot[fit[x] , {x, -3, 3}, PlotStyle → Red]}]
```
Out[14]=



In[10]:=
```
fit["ParameterConfidenceIntervalTable"]
```
Out[10]=

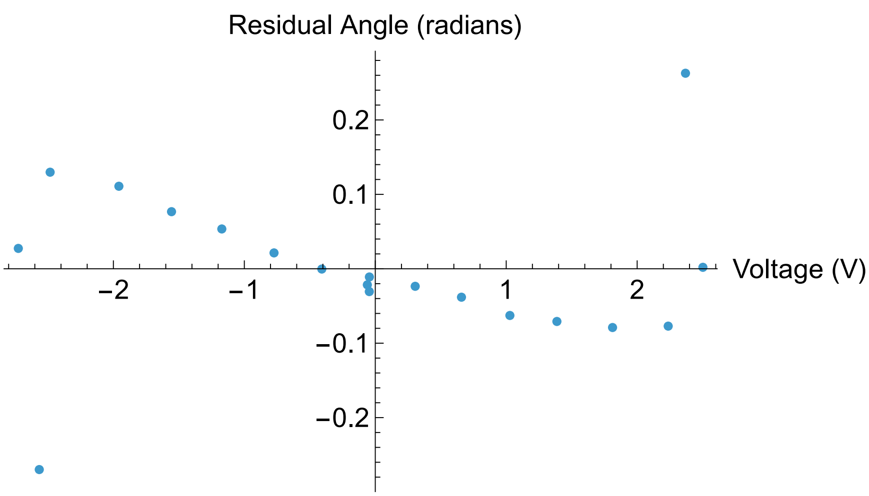
	Estimate	Standard Error	Confidence Interval
1	0.0483376	0.0252578	{−0.0049516, 0.101627 }
x	0.607383	0.0152313	{0.575248, 0.639518 }

The slope is 0.607 rad/Volt, and the intercept is 0.048 rad. We expect an intercept close to zero since we zeroed the voltage sensor when the oscillator was at equilibrium with zero added mass.

(b) Residuals

In[15]:=
```
residuals =
  Table[{data⟦i, 1⟧, data⟦i, 2⟧ - fit[data⟦i, 1⟧]}, {i, 1, Length[data]}];
```

In[16]:=
```
ListPlot[residuals, PlotRange → All, LabelStyle → Larger,
 AxesLabel → { "Voltage (V)", "Residual Angle (radians)"}, ImageSize → Scaled[0.7]]
```
Out[16]=



(c) Discuss quality of fit.

In[17]:=
```
rmse = Sqrt[fit["EstimatedVariance"]]
```
Out[17]=
```
0.109972
```

There are several basic observations here. The overall root mean square error is about 0.11 radians. This seems quite large. It is unlikely to be a measurement error --- it was easy to read the scale to at least a precision of 0.02 radians. The residuals plot also has two noteworthy features. First, at very large voltages (outside the ± 2 V range) the residuals get quite large.   Second, even at smaller voltages, there is a clear systematic (negative) trend with voltage. Clearly the best fit line for the whole data set is missing some important information.

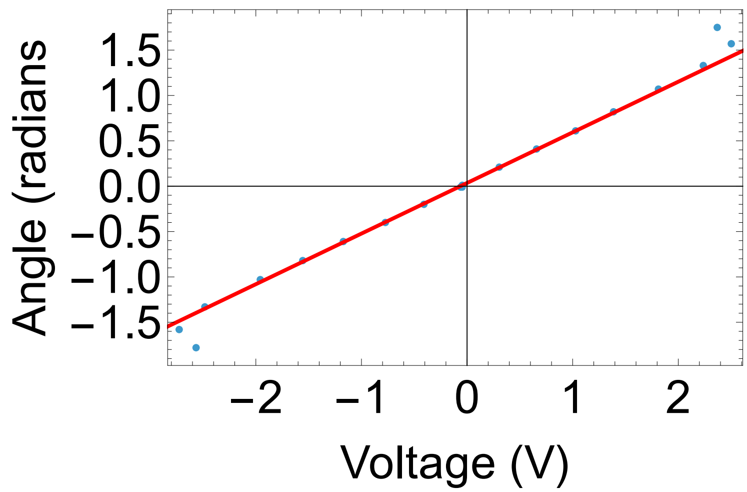
Problem 2: Voltage Calibration

(a) Limiting data to smaller angles

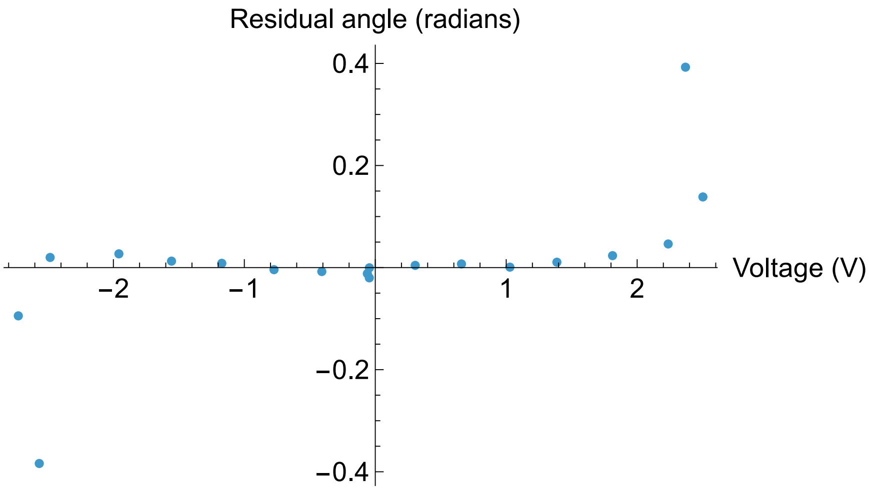
For convenience, bundle up all those commands above into a single cell to consider different limits on data. Select only a more narrow range of angles. Limiting the data to +- 1 radian from the equilibrium point of 2.99 seems to work well.

In[50]:=
```
fitdata = Select[data, -1 ≤ #⟦2⟧ ≤ 1 &];
(* Make a new restricted data set to use for fits. *)
fit = LinearModelFit[fitdata, x, x]
dataPlot = ListPlot[data,  ImageSize → Scaled[0.6], Frame → True,
   LabelStyle → Large, FrameLabel → {"Voltage (V)", "Angle (radians"}];
(* This is the same plot from above *)
Show[{dataPlot, Plot[fit[x] , {x, -3, 3}, PlotStyle → Red]}]
residuals =
  Table[{data⟦i, 1⟧, data⟦i, 2⟧ - fit[data⟦i, 1⟧]}, {i, 1, Length[data]}];
ListPlot[residuals, PlotRange → All, LabelStyle → Larger,
 AxesLabel → {"Voltage (V)", "Residual angle (radians)"}, ImageSize → Scaled[0.7]]
fit["ParameterConfidenceIntervalTable"]
rmse = Sqrt[fit["EstimatedVariance"]]
```
Out[51]=
```
FittedModel[ 0.0355 + 0.558 x ]
```

Out[53]=



Out[55]=



Out[56]=

	Estimate	Standard Error	Confidence Interval
1	0.0355343	0.00323501	{0.0282162, 0.0428524 }
x	0.557964	0.00377594	{0.549422 , 0.566505 }

Out[57]=

```
0.0107011
```

A typical scatter of 0.01 radians seems like a very good fit. The residuals are small and scatter without any obvious pattern for small voltages. Beyond ± 2V, the residuals rise rapidly. The conclusion is that the voltage calibration works very well for small angular displacements ≤ 1 radian. The exact range isn't critical. Using a voltage limit of ± 2 V instead of an angular limit of ± 1 radian gives nearly identical results (an rmse of 0.013 instead of 0.011, for example).

(b) Final Calibration Curve

Extract Calibration curve from fit. We want the slope, which is radians/Volt.

In[58]:=
```
{intercept, δintercept} =
  fit["ParameterConfidenceIntervalTableEntries"][[1, {1, 2}]]
```
Out[58]=
```
{0.0355343, 0.00323501}
```

In[59]:=
```
{dθdV, δdθdV} =
  {slope, δslope} = fit["ParameterConfidenceIntervalTableEntries"][[2, {1, 2}]]
```
Out[59]=
```
{0.557964, 0.00377594}
```

In[64]:=
```
Framed[
 StringForm["The slope is `` (radians/Volt) and the intercept is `` radians.",
  Around[dθdV, δdθdV], Around[intercept, δintercept]], Background → LightBrown]
```
Out[64]=

The slope is 0.558 ± 0.004 (radians/Volt)
and the intercept is 0.0355 ± 0.0032 radians.