AugerBot Calculations

In[122]:=
```
Quit
```

Trial 1: 9/19 - 9/26

Data

In[1]:=
```
(*Measured Drag Force: Auger Only*)
```

In[2]:=
```
g2N[g_] := g / 1000 * 9.81; (*Converts spring scale g to N*)
```

In[3]:=
```
avg2 = (237.5 + 260 + 240 + 250 + 275) / 5;
avg2N = g2N[avg2];
sd2N = StandardDeviation[Thread[g2N[{237.5, 260, 240, 250, 275}]]];
```

In[6]:=
```
avg3 = (275 + 280 + 310 + 300 + 262.5) / 5;
avg3N = g2N[avg3];
sd3N = StandardDeviation[Thread[g2N[{275, 280, 310, 300, 262.5}]]];
```

In[9]:=
```
avg4 = (325 + 320 + 300 + 350 + 310) / 5;
avg4N = g2N[avg4];
sd4N = StandardDeviation[Thread[g2N[{325, 320, 300, 350, 310}]]];
```

In[12]:=
```
(*Measured Drag Force: Motor Capsule Only*)
avgMotorF = (425 + 375 + 362.5 + 400 + 375) / 5; avgMotorN = g2N[avgMotorF];
sdMotorN = StandardDeviation[Thread[g2N[{425, 375, 362.5, 400, 375}]]];
```

In[14]:=
```
(*Measured Drag Force: AugerBot*)
```

In[15]:=
```
avg2bot = (362.5 + 375 + 325 + 362.5 + 315) / 5;
avg2botN = g2N[avg2bot];
sd2botN = StandardDeviation[Thread[g2N[{362.5, 375, 325, 362.5, 315}]]];
```

In[18]:=
```
avg3bot = (400 + 450 + 425 + 437.5 + 400) / 5;
avg3botN = g2N[avg3bot];
sd3botN = StandardDeviation[Thread[g2N[{400, 450, 425, 437.5, 400}]]];
```

In[21]:=
```
avg4bot = (525 + 512.5 + 537.5 + 500 + 475) / 5;
avg4botN = g2N[avg4bot];
sd4botN = StandardDeviation[Thread[g2N[{525, 512.5, 537.5, 500, 475}]]];
```

In[24]:=
```
(*Helix Angles in Radians*)
```

In[25]:=
```
a2 = 32.5398172 * (Pi / 180);
a3 = 23.1222041 * (Pi / 180);
a4 = 17.7979565 * (Pi / 180);
```

Calculating C_t and C_n : Modified (2) accounts for signficant wire size

In[26]:=
```
(*Constants*)
```

In[27]:=
```
Lx = 41.773 / 1000; (*In meters*)
Rw = 6.899 / 1000; (*In meters*)
```

In[28]:=
```
(*Measured Drag Force: Auger Only. Accounts for stem and significant helix radius*)
```

In[29]:=
```
eq[angle_, avgFd_] := (Rw) * (ct * Sin[angle] + cn * Cos[angle] / Tan[angle]) - avgFd / Lx
```

In[30]:=
```
newEq2 = eq[a2, avg2N]
```

Out[30]=
$$-59.2973 + 0.006899\,(1.32125\,\text{cn} + 0.537886\,\text{ct})$$

In[31]:=
```
newEq3 = eq[a3, avg3N]
```

Out[31]=
$$-67.047 + 0.006899\,(2.15382\,\text{cn} + 0.392694\,\text{ct})$$

In[32]:=
```
newEq4 = eq[a4, avg4N]
```

Out[32]=
$$-75.3839 + 0.006899\,(2.96593\,\text{cn} + 0.305661\,\text{ct})$$

In[33]:=
```
(*Solving for coefficients*)
```

In[34]:=
```
newSol23 = Solve[{newEq2 == 0, newEq3 == 0}, {ct, cn}][[1]]
```

Out[34]=
$$\{\text{ct} \to 8866.92,\ \text{cn} \to 2895.5\}$$

In[35]:=
```
newSol34 = Solve[{newEq3 == 0, newEq4 == 0}, {ct, cn}][[1]]
```

Out[35]=
$$\{\text{ct} \to 10\,446.4,\ \text{cn} \to 2607.51\}$$

In[36]:=
```
newSol24 = Solve[{newEq2 == 0, newEq4 == 0}, {ct, cn}][[1]]
```

Out[36]=
$$\{\text{ct} \to 9278.71,\ \text{cn} \to 2727.86\}$$

In[37]:=
```
newCtAvg = (newSol23[[1, 2]] + newSol34[[1, 2]] + newSol24[[1, 2]]) / 3
```

Out[37]=
$$9530.69$$

In[38]:=
```
newCnAvg = (newSol23[[2, 2]] + newSol34[[2, 2]] + newSol24[[2, 2]]) / 3
```

Out[38]=
$$2743.62$$

In[39]:=
```
(*Units for Ct and Cn are N/m*)
```

Plotting F_d vs N Loops

In[40]:=
```
(*Loading Error Bar Plots Package*)
Needs["ErrorBarPlots`"]
```

Formatting Data

Auger + Adapter & AugerBot

In[41]:=
```
(*Avg Fd*)
DataAuger = {{2, avg2N}, {3, avg3N}, {4, avg4N}}; (*Auger+Adapter*)
DataBot = {{2, avg2botN}, {3, avg3botN}, {4, avg4botN}}; (*Full Robot*)

(*Error*)
eAugerList = {{sd2N, sd3N, sd4N // N}}^T (*Auger+Adapter SD*)
eBotList = {{sd2botN, sd3botN, sd4botN}}^T; (*Full Robot SD*)

(*Reformatting Data for ErrorListPlot*)
DataAugerFull = Join[{DataAuger}^T, {Thread[ErrorBar[Flatten[eAugerList]]]}^T, 2];
DataBotFull = Join[{DataBot}^T, {Thread[ErrorBar[Flatten[eBotList]]]}^T, 2];
```

Out[43]=
$$\{\{0.151182\}, \{0.188699\}, \{0.184835\}\}$$

In[46]:=
```
(*Linear Fit*)
fitAuger = Fit[DataAuger, {1, x}, x]
fitBot = Fit[DataBot, {1, x}, x]
```

Out[46]=
$$1.80095 + 0.335992\, x$$

Out[47]=
$$1.80341 + 0.79461\, x$$

Auger Only

In[48]:=
```
(*Subtracting Flange Fd from
 Auger: Want expression which considers only the thrust generating components*)
FdIntercept = fitAuger /. x → 0;
DataAdjust =
  {{2, avg2N - FdIntercept}, {3, avg3N - FdIntercept}, {4, avg4N - FdIntercept}};
```

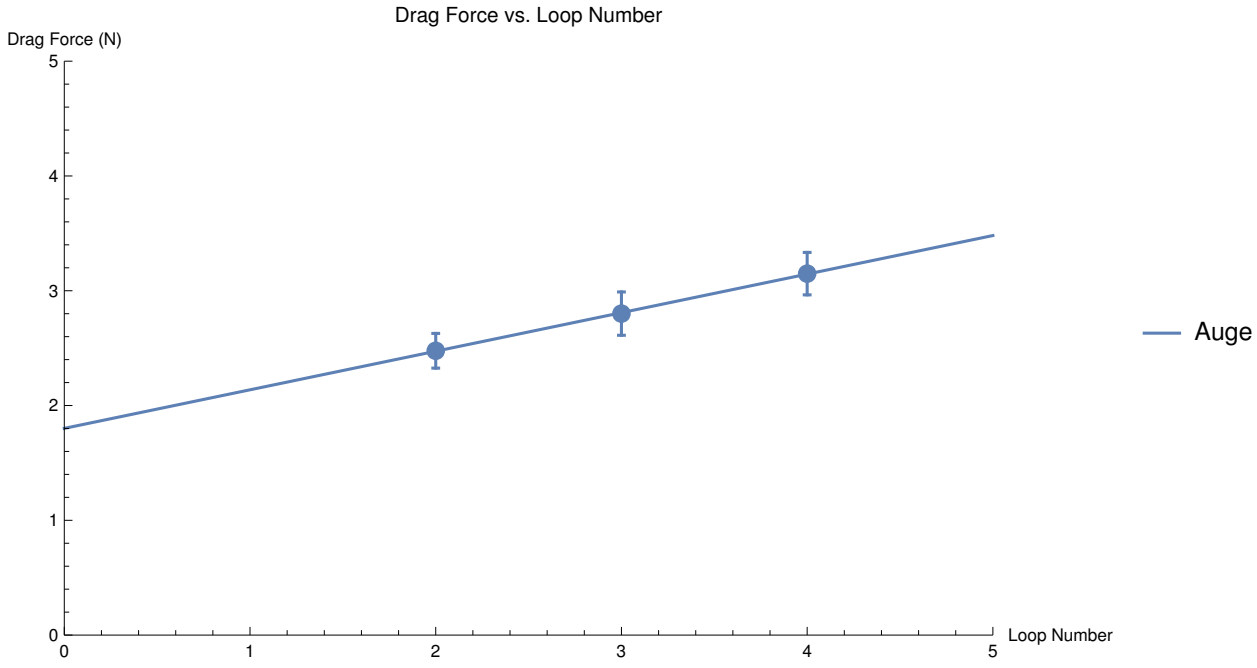
In[50]:=
```
(*Formatting Data for ErrorListPlot*)
DataAugerAdjust = Join[{DataAdjust}^T, {Thread[ErrorBar[Flatten[eAugerList]]]}^T, 2];
(*SD is the same as Auger+Adapter*)

(*Liner Fit*)
fitAdjust = Fit[DataAdjust, {1, x}, x]
```

Out[51]=
$$-1.98706 \times 10^{-15} + 0.335992\, x$$

Plotting Drag Force of Auger+Adapter and AugerBot vs. Loop Number

In[52]:=
```
Show[{ErrorListPlot[{DataAugerFull}, PlotLabel → "Drag Force vs. Loop Number",
    PlotStyle → PointSize[0.02], AxesLabel → {"Loop Number", "Drag Force (N)"},
    ImageSize → Large, PlotRange → {{0, 5}, {0, 5}}], Plot[{fitAuger}, {x, 0, 5},
    PlotRange → {{0, 5}, {-1, 6}}, PlotLegends → {"Auger+Adapter Fit", "AugerBot Fit"}]}]
```

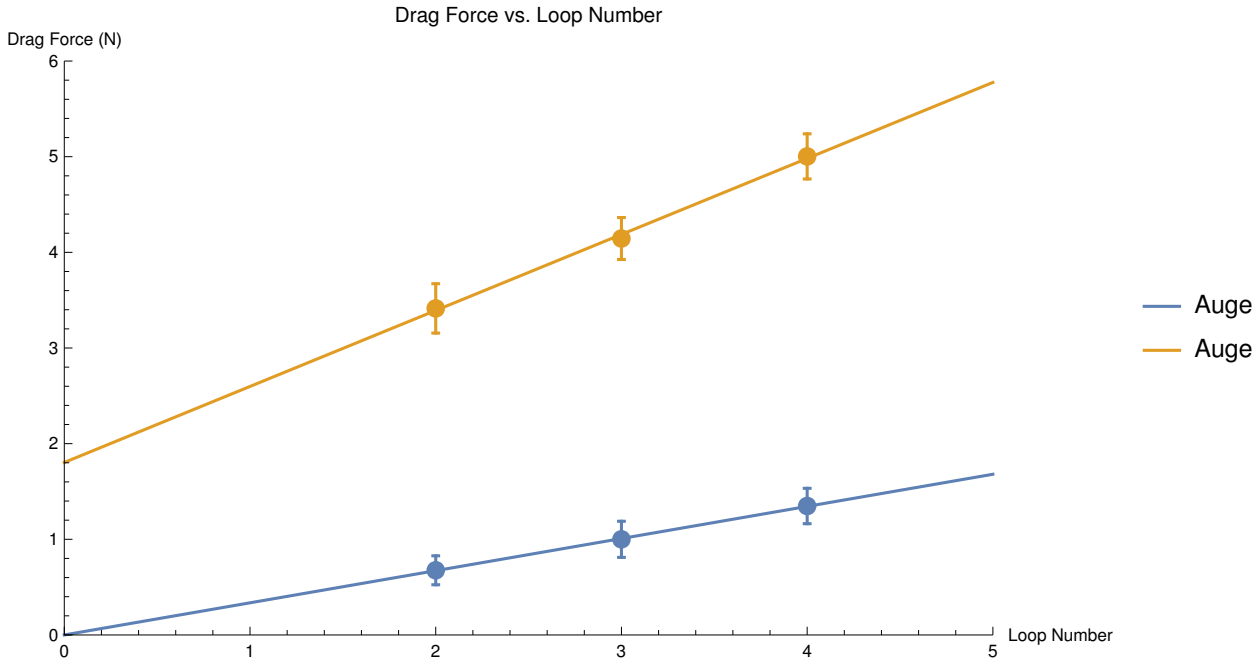
Out[52]=



Auger Only and AugerBot Force Trends

In[53]:=
```
Show[{ErrorListPlot[{DataAugerAdjust, DataBotFull},
   PlotLabel → "Drag Force vs. Loop Number", PlotStyle → PointSize[0.02],
   AxesLabel → {"Loop Number", "Drag Force (N)"}, ImageSize → Large,
   PlotRange → {{0, 5}, {0, 6}}], Plot[{fitAdjust, fitBot},
   {x, 0, 5}, PlotLegends → {"Auger Only Fit", "AugerBot Fit"}]}]
```

Out[53]=



In[54]:=
```
Print["Avg Motor Capsule Fd = ", avgMotorN, ", Sd = ", sdMotorN]
```

```
Avg Motor Capsule Fd = 3.80138, Sd = 0.24525
```

Feedback

Good that the plot is linear.

Expect F_d to be linearly dependent on depth & coil #

Concerns

y-intercept drag force | Paul: Seems bit big at 1.8N. (Not sure where Paul's intuition is coming into play here, since Fd varies with depth.)

From just poking the robot around in the seeds, obvious $F_d > F_x$

Questions

Why is the full robot trend not parallel? Attaching same body...so additional drag should be the same

Because can't see, possible I'm not burying it consistently at same depth. (Not sure how big of an affect this has.)

Also possible that the packing factor differs enough to make a difference.

(Measuring after 3 pulls approximately because I don't know the range. But this causes the front material to become more compact)

Had to use only the red adapter; green ones aren't a tight fit. Shouldn't be a concern because all made of PLA and the diameter and thickness is same.

Testing fixes: (Postponing future tests like this)

Put a piece of tape on side of tank to mark consistent surface height and device depth.

Use a "leveler" to make sure bottom of trench is flat. (So robot is actually pulled in +x and not diagonally.)

(*With Paul During 9/26 Meeting*)

Clipping the wings reduced F_d. Thrust actually tangible. But not strong. Also, more spinning.

Next

Try more loops over increasing length of helix to increase F_x

Make shorter wings - let's try rectangular for now

Add a cone at the base of the helix to reduce the blunt edge