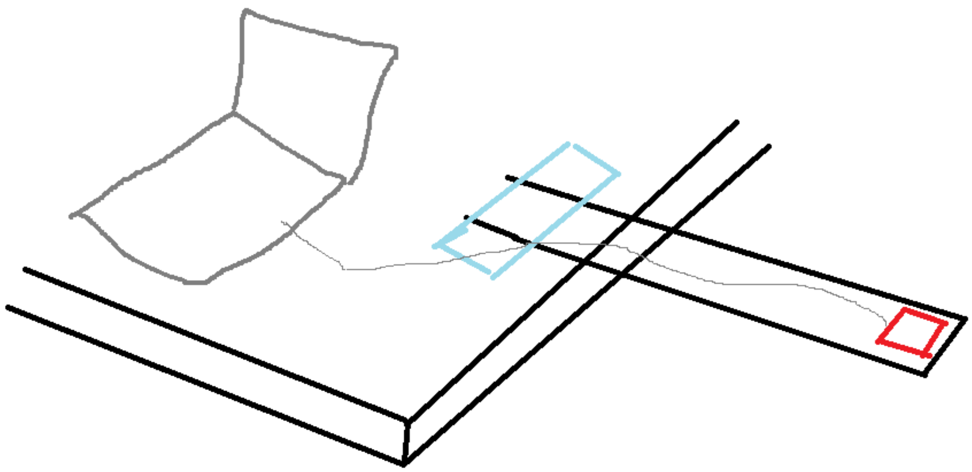
Lab 7

Kieran Cosgrove

1. Setup of experiment:



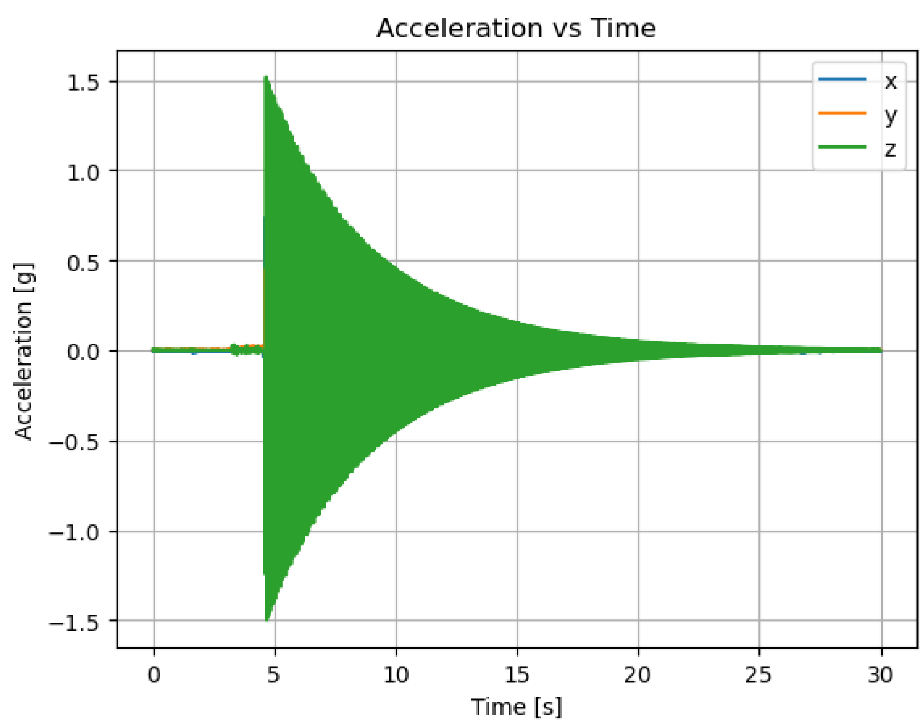
Measured values (E dependent on material – 6061 Aluminum):

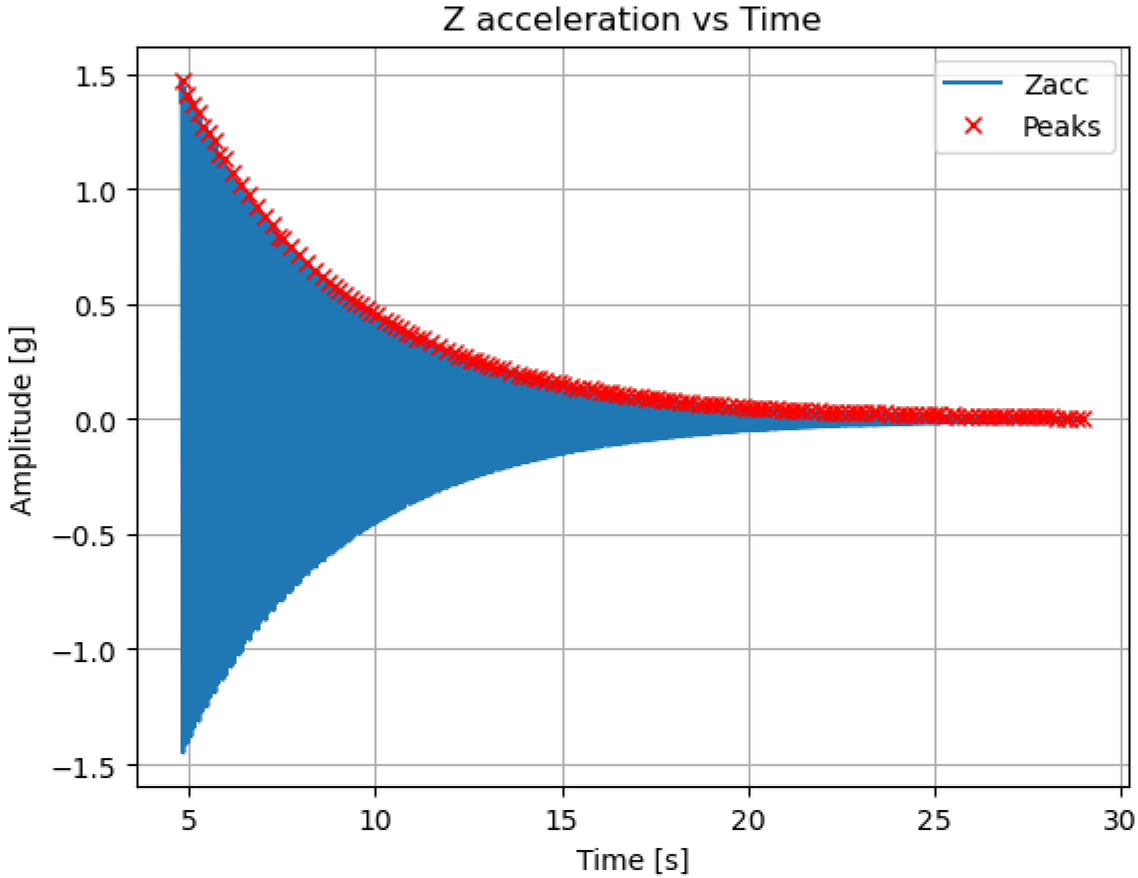
```
# L = 0.28 #25.5cm   # [m]
# h = 0.00317   # [m]
# w = .0255    # [m]
# E = 68.9e9        # [Pa]
# I = (1/12)*w*h**3 # [m^3]
# rho = 2700        # [kg/m^3]
# Mbeam = L*h*w*rho # [kg]
# Mlump = (20 + 4*5.6 + 5)/1000 # [kg], includes quarters/zip tie/duct tape
```

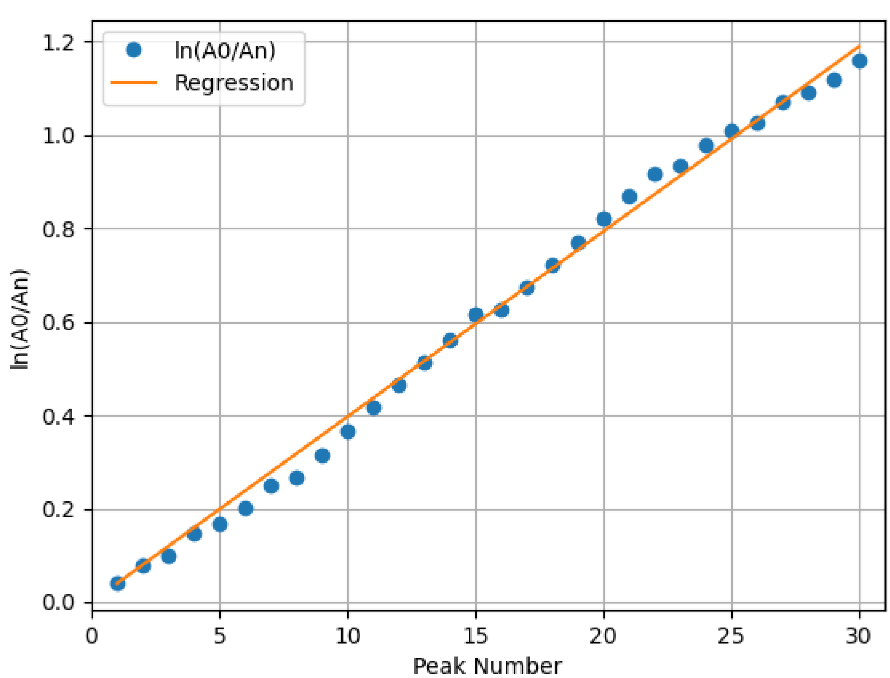
2.

```
---------------------------------------------------------
Theoretical Stiffness and Natural Frequency Bounds:
Theoretical Stiffness [N/m] = 637.39
Omega_n Min [rad/s] = 70.673574
Omega_n DH  [rad/s] = 88.951589
Omega_n Max [rad/s] = 97.901812
---------------------------------------------------------
---------------------------------------------------------
Experimental Signal Freq. [Hz] = 6.1672
Experimental period [s]        = 0.1621
Damped Natural Freq. [rad/s]   = 38.7498
---------------------------------------------------------
---------------------------------------------------------
Beta (slope) =  0.03962
R squared    =  0.99526
Zeta         =  0.00631
---------------------------------------------------------
---------------------------------------------------------
Experimental Damped and Natural Frequencies Results:
Omega_d Exp [rad/s] = 38.749771
Omega_n Exp [rad/s] = 38.750541
---------------------------------------------------------
---------------------------------------------------------
Effective Mass [kg] = 0.424471
Percentage of beam mass in Effective Mass [%] = 585.77
(Compare to DH Method value of 23%)
---------------------------------------------------------
---------------------------------------------------------
Damping Coefficient, b [N*s/m] = 0.20745
---------------------------------------------------------
```

3.




Z acceleration vs Time
Zacc
Peaks
Amplitude [g]
Time [s]
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
5
10
15
20
25
30


ln(A0/An)
Regression
ln(A0/An)
Peak Number
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30

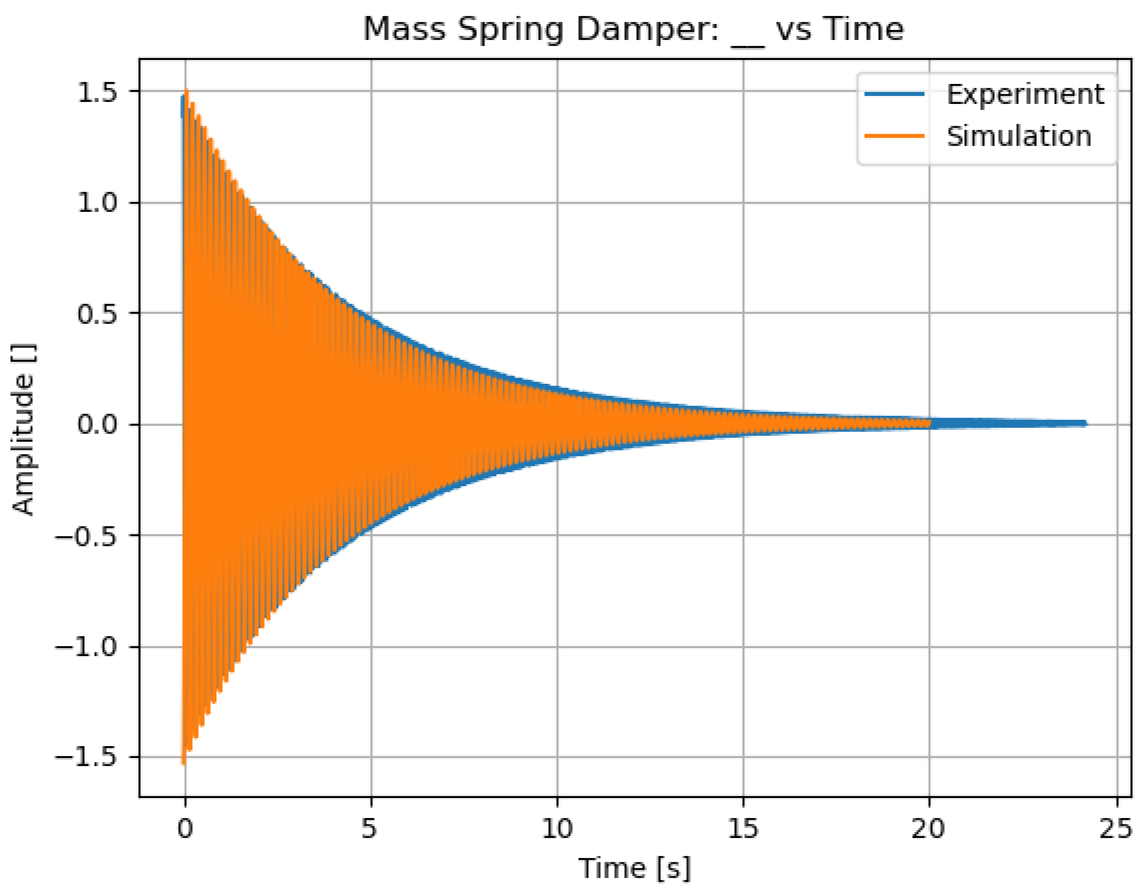
The log. Dec plot is fairly linear, it seems to be below the line towards the bottom and above it at the top though. This could be caused by non-linear damping. The non-linearity could be air induced, as the force created from the beam waving back and forth would change with velocity squared.

The frequency measured was not within the bounds expected. Probably due to errors in setup values.

There is a large difference of 586% between the theoretical and measured effective mass.

Given the large difference of 586% between the masses, it was very far off of the 23% expected.

4.



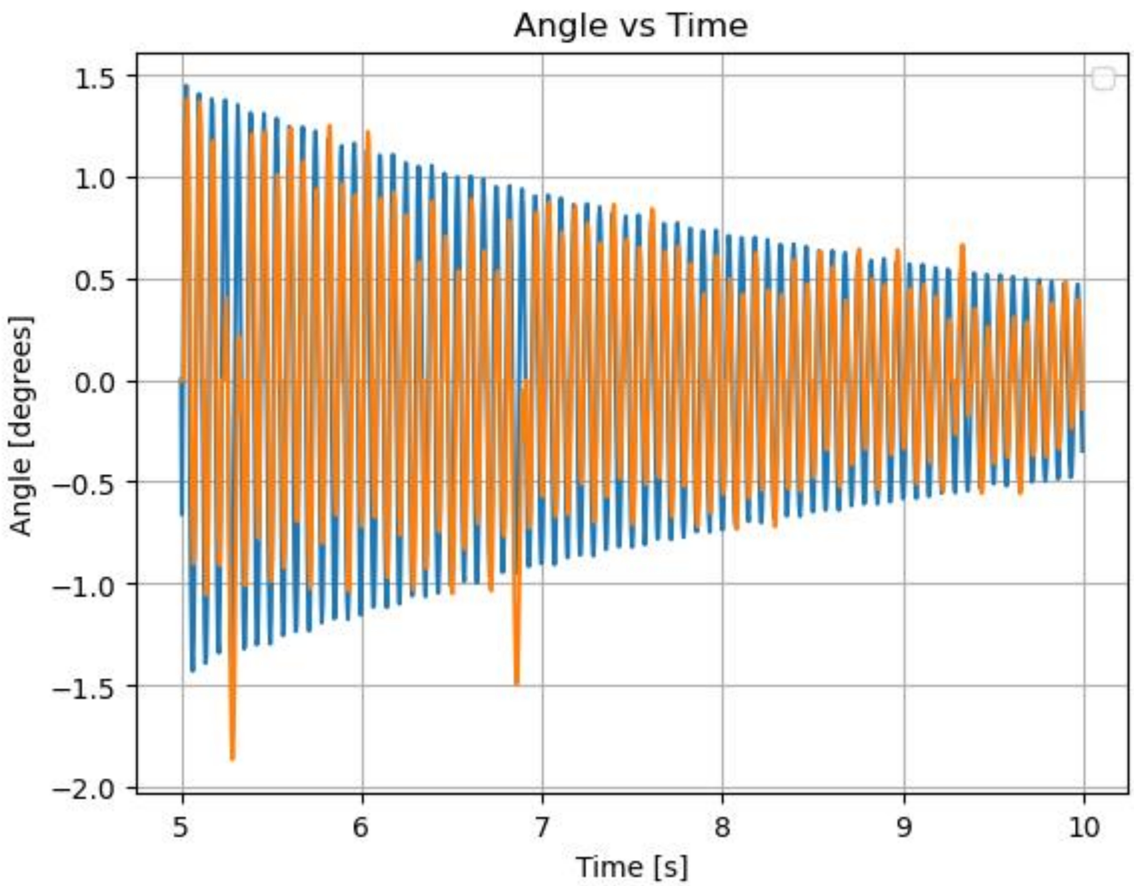
The periods don't match, the experiment appears to have higher damping at higher speeds and lower damping at lower speeds, meaning it isn't exactly linear damping.

The experimental peak-to-peak decay is more exponential than linear.

Sources of error include incorrect beam measurements; the length was very roughly measured. The other big factor is the position of the lump sum and its exact weight, which might have varied from setup to setup.

5.

I integrated angular rate at the beam tip to estimate beam tip slope. I calculated the theoretical slope by using the acceleration times the mass to get the force on the beam and using the stiffness to find the theoretical displacement. Blue below shows theoretical against orange measured values.




```
#CHALLENGE

wx = np.array(data.wx) # [counts]
wy = np.array(data.wy) # [counts]
wz = np.array(data.wz) # [counts]

conv = 1/1024  # conversion rate from raw data [counts] to units of [g]
x_g = x*conv   # [g]
y_g = y*conv   # [g]
z_g = z*conv   # [g]

theta = [0]
theta_alt = [0]
```

```
wx_u = wx[1000:2000]
time_u = time[1000:2000]
z_u = z_g[1000:2000]

dist = 20
peak_its, _ = find_peaks(abs(wx_u), height=0, distance = dist)
peaks  = wx_u[peak_its]

i_0 = 0
for i in range(0, len(wx_u)-1):
    theta_i = np.trapz(wx_u[i_0:i], x=time_u[i_0:i])
    if wx_u[i] in peaks:
        theta_i = 0
        i_0 = i
    theta.append(theta_i/180*3.314)

    theta_alt.append((m_eff*z_u[i])*L**2/2/E/I*60/3.314+.5)

theta = np.array(theta)



# Acceleration vs time
plt.figure()
#plt.scatter(time_u, wx_u)

plt.plot(time_u, theta_alt)
plt.plot(time_u, theta)
#plt.scatter(time[peak_its], peaks)
#plt.scatter(np.array(data.Time), wx)
plt.grid()
plt.legend()
plt.xlabel('Time [s]')
plt.ylabel("Angle [degrees]")
plt.title("Angle vs Time")
plt.show()
```