ASEN 2003 Lab 5:
Yo-Yo Despinner
Section 013
April 11, 2017

Kayla Gehring, * Junzhe He, † Nicholas Renninger, ‡
University of Colorado - Boulder

Yo-yo despinners are a very useful attitude control mechanism, as they allow spin-stabilized spacecraft to achieve high (stable) roll rates, while still being able to depsin for orbital maneuvers without expensive or complicated attitude controls. This report details the analytic modeling, and subsequent experimental testing, of the despinning of a model satellite. Analytic models were developed to determine the length of tether cord needed to successfully despin the satellite to zero angular velocity. Using the most accurate of these models, the length of cord needed to depsin the experimental satellite using the given despin masses of 0.125 kg (0.276 lbm) was 0.345 m (13.6in). Due to friction on the satellite, it slowed down on its own without the use of a despin mechanism, behavior that would not happen in the vacuum of space. Therefore, to improve the accuracy of the models, this friction (actually a frictional couple on the satellite) was measured to be -0.0419 N-m or -0.0309 ft-lb. Using high-speed video, the assumptions used in the models were shown to be somewhat incorrect, but the models still had reasonable agreement with experimental observation. This experiment proved the value of analytic models, their discrepancies with reality, and the methods that can be used to identify and improve them.

Nomenclature

$\alpha(t)$ = Angular Acceleration of Satellite $[rad/s^2]$
$\omega(t)$ = Angular Velocity of Satellite $[rad/s]$
ω_o = Initial Angular Velocity before Despin $[rad/s]$
I = Moment of Inertia about Satellite Rotational Axis $[kg \cdot m^2]$
$L(t)$ = Length of Tether Cords $[m]$
M = Frictional Couple Moment of Satellite due to Motor Assembly $[N \cdot m]$
m = Mass of Despin Weights $[kg]$
R = Outer radius of Uniform Satellite $[m]$
$T(t)$ = Tension in Tether Cords $[N]$

*105677160
†105925743
‡105492876

Contents

I. Theory

The most important analytical development needed to utilize a Yo-yo despinner is the length of cord needed to despin a spacecraft with a given moment of inertia, I, and mass of despin weights, m. To find what length of tether cord is necessary to deplete the ω of the satellite, an analytic model of the spacecraft's angular velocity versus time, $\omega(t)$, was developed using both conservation of energy and momentum methods. This process was first done assuming that the despin masses would detach when the tether cord was fully unwound and when still tangent to the surface of the satellite.

Then, after watching the high-speed video, as described in Section II, $\omega(t)$ was once again determined assuming the despin masses would detach when the tether cord was fully unwound and when the cord was parallel to the radius vector of the satellite (radius vector shown in Figure 8). The radial-release model was determined by changing the tangential-release model to include the motion of the despin masses past the tangential release point.

I.A. Tangential Release

I.A.1. Angular Velocity: $\omega(t)$, $\omega(L)$

To derive the angular velocity of the spacecraft, we need to use the conservation of energy and angular momentum, the derivation is shown as following:

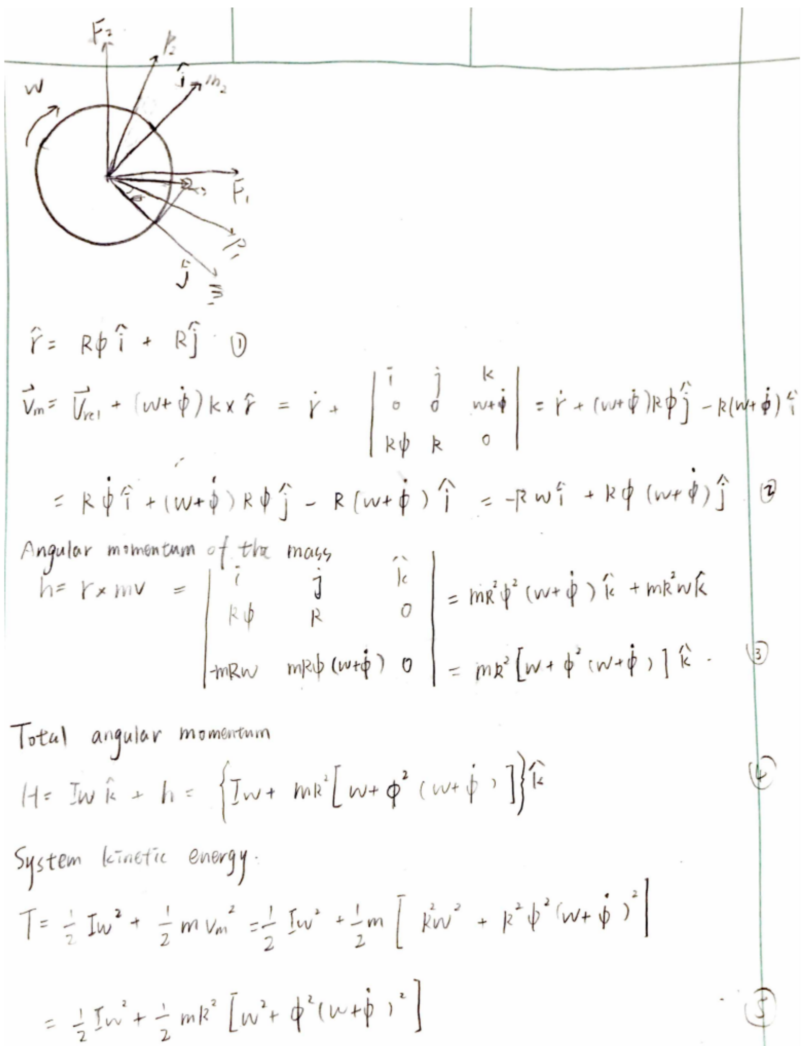


Figure 1: Derivations of angular velocity VS time and cord length

Using conservation of energy and momentum

$$H_0 = I\omega_0 = (I + mR^2)\omega_0 = I\omega + mR^2\left[\omega + \phi^2(\omega + \dot{\phi})\right] \quad (6)$$

$$\frac{I\omega_0 + mR^2\omega_0}{mR^2} = \frac{I\omega + mR^2\left[\omega + \phi^2(\omega + \dot{\phi})\right]}{mR^2}$$

$$\left(\frac{I}{mR^2} + 1\right)\omega_0 = \left(\frac{I}{mR^2} + 1\right)\omega + \phi^2(\omega + \dot{\phi})$$

let $C = \frac{I}{mR^2} + 1$

$$C\omega_0 = C\omega + \phi^2(\omega + \dot{\phi}) \quad (7)$$

$$T_0 = \frac{1}{2}I\omega_0^2 = \frac{1}{2}(I + mR^2)\omega_0^2 = \frac{1}{2}I\omega^2 + \frac{1}{2}mR^2\left[\omega^2 + \phi^2(\omega + \dot{\phi})^2\right] \quad (8)$$

$$\Rightarrow \frac{\frac{1}{2}I\omega_0^2 + \frac{1}{2}mR^2\omega_0^2}{mR^2} = \frac{\frac{1}{2}I\omega^2 + \frac{1}{2}mR^2\left[\omega^2 + \phi^2(\omega + \dot{\phi})^2\right]}{mR^2}$$

$$\left(\frac{I}{mR^2} + 1\right)\omega_0^2 = \left(\frac{I}{mR^2} + 1\right)\omega^2 + \phi^2(\omega + \dot{\phi})^2$$

$$C\omega_0^2 = C\omega^2 + \phi^2(\omega + \dot{\phi})^2 \quad (9)$$

$$(7) \rightarrow C(\omega_0 - \omega) = \phi^2(\omega + \dot{\phi})$$

$$(9) \rightarrow C(\omega_0^2 - \omega^2) = \phi^2(\omega + \dot{\phi})^2$$

$$\rightarrow C(\omega_0 + \omega)(\omega_0 - \omega) = \phi^2(\omega + \dot{\phi})^2$$

$$\frac{(9)}{(7)} \rightarrow \omega_0 + \omega = \omega + \dot{\phi} \Rightarrow \omega_0 = \dot{\phi} \Rightarrow \omega_0 t = \phi \quad (10)$$

Figure 2: Derivations of angular velocity VS time and cord length

substitude (10) into (7)

$$C\omega_0 = C\omega + \omega_0^2 t^2(\omega + \omega_0) \Rightarrow (C + \omega_0^2 t^2)\omega = (C - \omega_0^2 t^2)\omega_0$$

$$\Rightarrow \boxed{\omega = \left(\frac{C - \omega_0^2 t^2}{C + \omega_0^2 t^2}\right)\omega_0}$$

$$L = \phi R = \omega_0 t R \Rightarrow \omega_0 t = \frac{L}{R}$$

$$\Rightarrow \omega = \omega_0\left[\frac{C - \left(\frac{L}{R}\right)^2}{C + \left(\frac{L}{R}\right)^2}\right] = \boxed{\omega_0\left(\frac{CR^2 - L^2}{CR^2 + L^2}\right) = \omega}$$

Figure 3: Derivations of angular velocity VS time and cord length

I.A.2. *Angular Acceleration: $\alpha(t)$, $\alpha(L)$*

Angular acceleration of the spacecraft is the derivative of its angular velocity (as shown in Figures 1 - 3), with respect to time:

(b) $$w = w_0 \left(\frac{c - w_0^2 t^2}{c + w_0^2 t^2}\right)$$

$$\alpha = \frac{dw}{dt} = w_0 \frac{(-2w_0^2 t)(c + w_0^2 t^2) - (2w_0^2 t)(c - w_0^2 t^2)}{(c + w_0^2 t^2)^2}$$

$$= w_0 \frac{(-2w_0^2 t)(c + w_0^2 t^2 + c - w_0^2 t^2)}{(c + w_0^2 t^2)^2}$$

$$\Rightarrow \boxed{\alpha = \frac{-4c w_0^3 t}{(c + w_0^2 t^2)^2}} \quad (1)$$

$$L = \phi R = w_0 t R$$

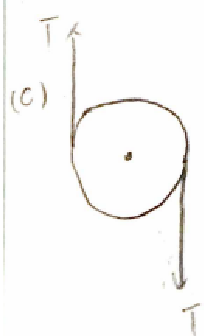
$$\Rightarrow \boxed{\alpha = \frac{-4c w_0^2 \frac{L}{R}}{\left[c + \left(\frac{L}{R}\right)^2\right]^2}} \quad (2)$$

Figure 4: Derivations of angular acceleration VS time and cord length

I.A.3. *Tension, $T(t)$*

Using the Newton's second law, the tension can be derived as following:

(c)



$$2T = \frac{I\alpha}{R} = \frac{I}{R} \cdot \frac{-4c w_0^2 \frac{L}{R}}{\left[c + \left(\frac{L}{R}\right)^2\right]^2}$$

$$\boxed{T = \frac{-2I c w_0^2 \frac{L}{R}}{R\left[c + \left(\frac{L}{R}\right)^2\right]^2}}$$

$$\boxed{= \frac{-2I c w_0^3 t}{R(c + w_0^2 t^2)^2}}$$

Figure 5: Derivations of tension VS time and cord length

I.A.4. Length and Time Required to Stop in Tangential Release, L

Using the equations of the angular velocity with respect to time and cord length and setting the angular velocity to be zero, we can easily derive that:

$$c - w_0^2 t^2 = 0 => t = \sqrt{\frac{c}{w_0^2}} = 0.4003s \tag{1}$$

$$cR^2 - L^2 = 0 => L = \sqrt{cR^2} = 0.4528m = 17.827in \tag{2}$$

I.B. Radial Release - Length, L

from the conservation of energy

$$\frac{1}{2} I W_0^2 + \frac{1}{2} m V_m^2 = \frac{1}{2} I W_f^2 + \frac{1}{2} m V_{mf}^2$$

$$=> \frac{1}{2} I W_0^2 + \frac{1}{2} m \cdot W_0^2 R^2 = \frac{1}{2} m V_{mf}^2$$

$$V_{mf} = \sqrt{\frac{I W_0^2}{m} + W_0^2 R^2} = R\sqrt{\left(\frac{I}{mR^2} + 1\right) W_0^2} = R W_0 \sqrt{c}$$

from the conservation of angular momentum

$$-I_{tot} W_0 = r \times m\vec{V}_{mf} + I W_f$$

$$-(I + mR^2) W_0 = -mRW_0\sqrt{c}\,(R+L) + I W_f$$

$$=> W_f = \frac{mRW_0\sqrt{c}\,(R+L) - (I + mR^2) W_0}{I}$$

$$W_f = 0 => mR\cancel{W_0}\sqrt{c}\,(R+L) = (I + mR^2)\cancel{W_0}$$

$$mR^2\sqrt{c} + mR\sqrt{c}\,L = I + mR^2$$

$$mR\sqrt{c}\,L = I + mR^2 - mR^2\sqrt{c}$$

$$L = \frac{I}{mR\sqrt{c}} + \frac{R}{\sqrt{c}} - R$$

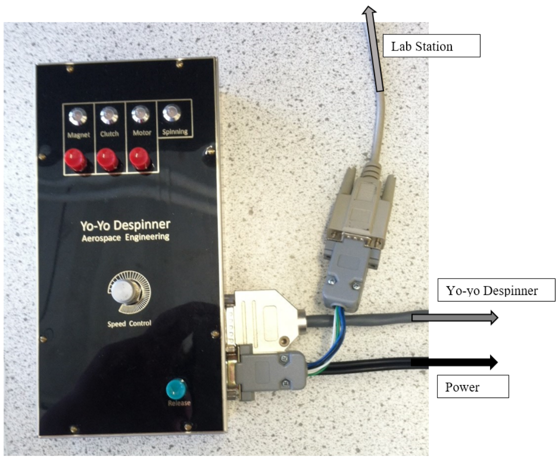
$$L = \frac{R}{\sqrt{c}}\left(\frac{I}{mR^2} + 1\right) - R$$

$$= \boxed{R(\sqrt{c} - 1)}$$

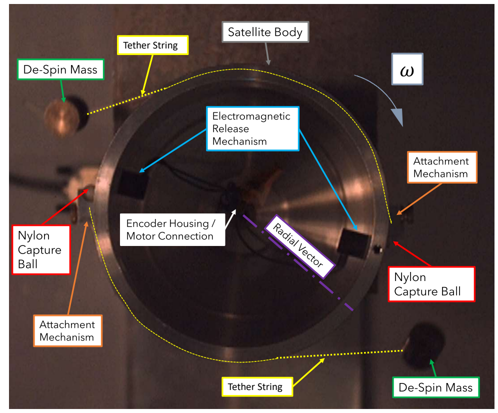
Figure 6: Derivations of required cord length in radial release

By plugging in the numbers, we found that L is required to be 0.34459 meters(13.57 inches)

II. Experiment



(a) The control box with interlocks and speed controls. *The encoder and motor are plugged into the control box, which is then connected to the computer and LabVIEW VI.*



(b) Experimental Setup of Satellite and Yo-yo Despinner. *The entire mechanical setup is shown for the de-spin mechanism. The motor and satellite stand are not shown, as they reside underneath the satellite.*

Figure 7: Experimental Setup of Satellite, Yo-yo Despin Mechanism, and Control Box

II.A. Experimental Setup

The Yo-Yo despinner analyzed in this report is attached to a test "satellite", spun up by a DC motor operated by a control box provided by the ASEN department (shown in Figure 7a). The control box allows the user to easily control the electromagnetic deployment mechanism, the clutch which connects the satellite to the DC motor for spin-up, the speed of the motor while connected to the satellite, and the precise release time of the yo-yo depsinner. For data acquisition, a LabVIEW VI was used to read in the data from a rotary encoder attached to the rotational axis of the satellite. The encoder was read directly using the micro-controller within the control box, which uses to ensure safe and controlled operation of the DC motors, and then this micro-controller communicated in turn with the LabVIEW VI to record the angular position and velocity data from the satellite.

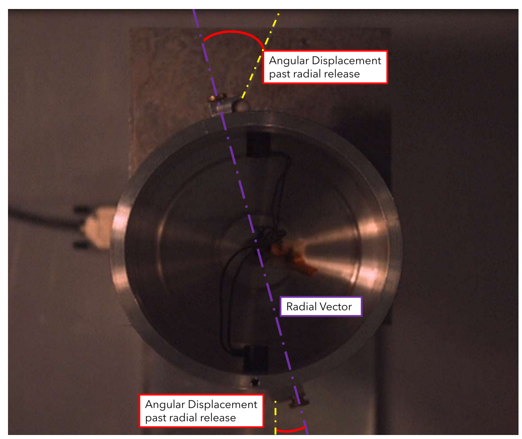


Figure 8: High-speed video showing the actual moment of release. *The masses actually swing past the point of radial release, motion which is not currently modeled. Also, note that while the top mass is about to be released, the bottom mass still has to swing at least another* $10°$ *before its release, an asymmetry not accounted for by the model.*

The simulated satellite is simply a hollow cylinder that is brought to approximately 100 rpm by the control box, which has a potentiometer-like speed control as shown in Figure 7a. Once the satellite has been spun up sufficiently, the user can push a button on the control box to disengage the electromagnets and release the weights. As the weights swing around the satellite, they unwind the tether strings which are wrapped around the satellite body. Once the tether strings are fully unwound, they continue to swing until the nylon capture balls are able to be freed from the attachment mechanism. The full configuration of the satellite testing apparatus can be found in Figure 7b.

While the satellite is being spun up and subsequently despun, the control box is relaying the positional data to the LabVIEW VI from the rotary encoder. This data is streamed to a file in the .CSV format, which is then processed using MATLAB.

II.B. Observations

In order to better understand the behavior of the yo-yo despinner, especially during release, a high-speed recording of the release process was used in the development of the analytic model for the despinner. The high-speed recording of the release proved to be very useful, and allowed for the development of a new model. Before the use of the high-speed video, the depsinner was assumed to be releasing tangentially, with the weights releasing at the instant the tether string was both fully unwound and still tangent to the surface of the satellite. The video showed the a radial release was more accurate, where the weights of the despinner actually release when the tether string is both fully unwound and pointing along the radial vector pointing from the center of rotation of the satellite, as shown in Figure 7b.

While the high-speed video did allow for a more accurate model to be developed, upon closer inspection there still exists a discrepancy between the radial release model and the actual behavior of the despinner. Due to the nature of the despinner attachment mechanisms, as shown in Figure 7b, the masses actually swing past the point of radial release. Also, the models we used assumed that both of the despinners would be released from symmetric angles relative to the radial vector of the satellite. This assumption proved not be true, as one of the masses is often release first. These two discrepancies are reported in Figure 8. While our model does not match the actual behavior perfectly, the small angle past the radial release will only contribute to a slight overestimation of the satellite's final ω.

Another aspect of the satellite not necessarily taken into account was the frictional losses of the satellite's angular velocity due to the DC motor used to drive the satellite to its ω_o. In order to calculate this loss, the frictional couple moment, M, being applied to the satellite was determined by running a test where the satellite was allowed to despin on its own (without the despin mechanism). This data, shown in Figure 9, showed that there was a linear trend between $\omega(t)$ and t, which confirmed the suspicion that the frictional couple acting on the satellite was constant. To calculate M, $\alpha(t) \equiv$ ¢ was computed for the data in Figure 9 by numerically differentiating the $\omega(t)$ data. Then, using Newton's Second Law, the total frictional couple M was computed using Eqn. 3:

$$\Sigma M = M = I\alpha \tag{3}$$

Through this analysis, the couple on the satellite was found to be, on average, M = -0.0419 N-m or -0.0309 ft-lb.

III. Results and Discussion

III.A. $\omega(\mathbf{t})$ - Not Using Yo-yo Despin Mechanism

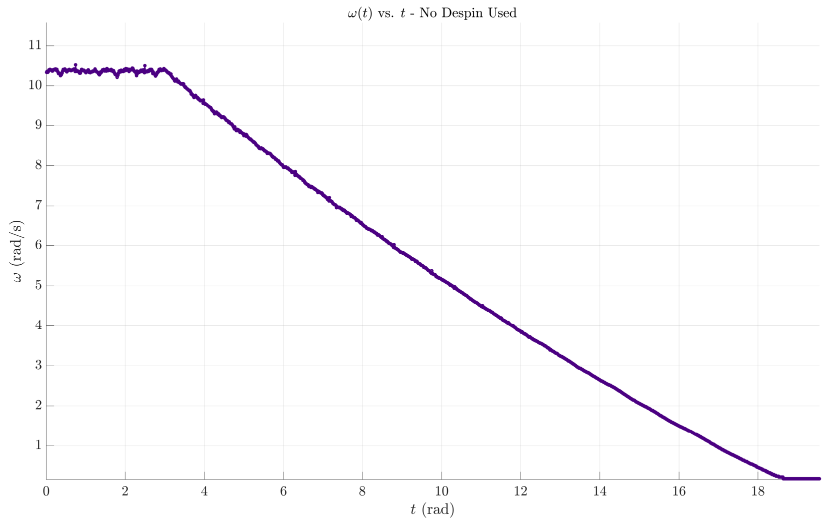


Figure 9: Despin of the satellite model due only to frictional forces, without deploying the despin mechanism. *This experiment allowed for the calculation of M = -0.0419 $N \cdot m$, by finding the associated α and multiplying by I to find M.*

III.B. $\omega(\mathbf{t})$ - Tangential Deployment Model

In this section, the model derived in above sections is compared to multiple string lengths.

In Figure 10, it's extremely apparent that without a long enough string, the satellite is unable to fully decelerate to $\omega = 0$. The point at which a tangential release would have occurred is marked, and the theoretical despin with a tangential release is shown.

In Figure 11, the model lines up reasonably well, but the satellite re-accelerates as the string pulls it around in the other direction.

In Figure 12, the model lines up pretty well, considering the type of release. The tangential release still takes much less time.

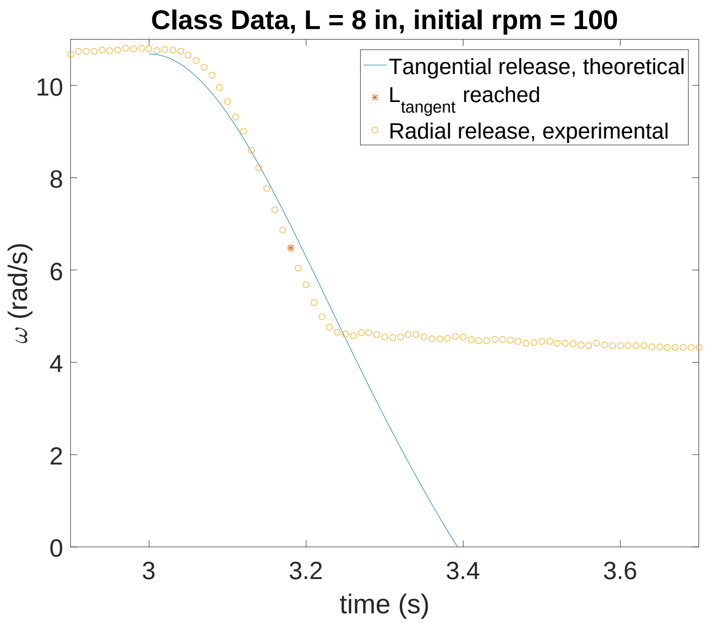


Figure 10: Experimental and Tangential Model $\omega(t)$ with L = 8in.

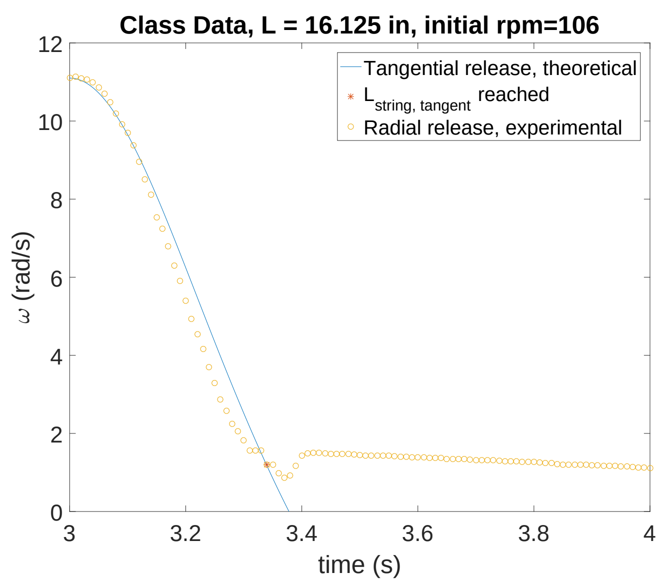


Figure 11: Experimental and Tangential Model $\omega(t)$ with L = 16.125in.

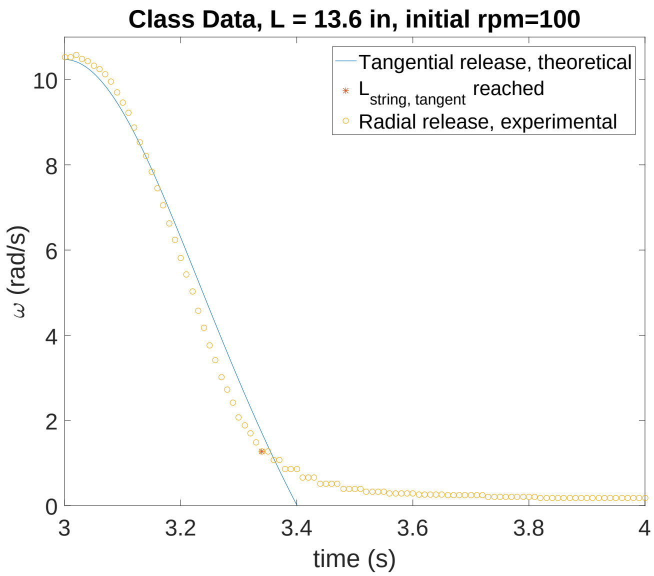


Figure 12: Experimental and Tangential Model $\omega(t)$ with L = 13.6in.

III.C. $\alpha(\mathbf{t})$ - Tangential Deployment Model

The tether cord length at which $\alpha(t)$ is at a maximum of 55.5rad/s is L = 17.83 m. The tether cord length at which $T(t)$ is at a maximum of 658.3N is L = 17.83 in.

The acceleration of the theoretical model versus the experiment is quite far off, despite using the best data possible. The model doesn't line up well enough with the actual experiment to show good data for acceleration.

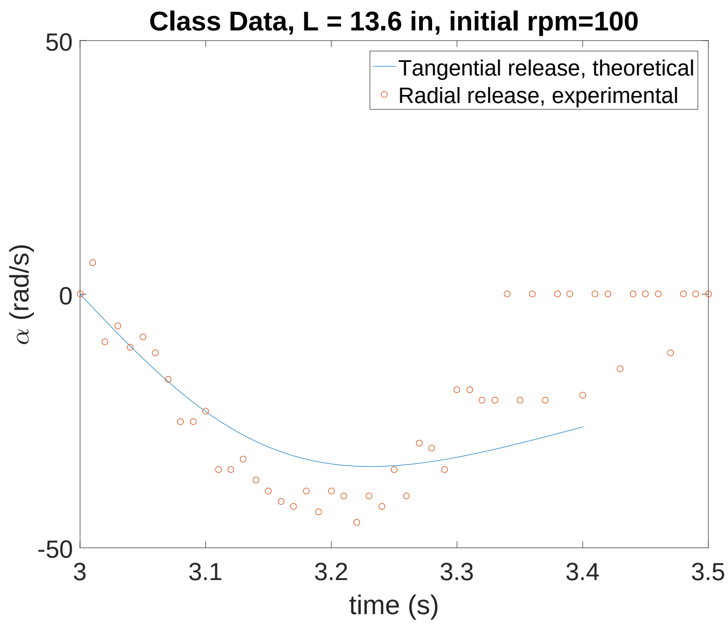


Figure 13: Experimental and Tangential Model $\alpha(t)$.

The models predict the despin of the satellite well initially, then less accurately towards the end of the model as the effects of friction and incorrect release assumptions add up and cause inaccuracy in the models. The string length predicted by the radial-release model, L = 13.57in, proved to stop the satellite as predicted, even with the errors in the assumptions of the model. Friction does play some role in the error in the models, as seen in Figures 10 - 12, with the model diverging from the experimental results as t increases.

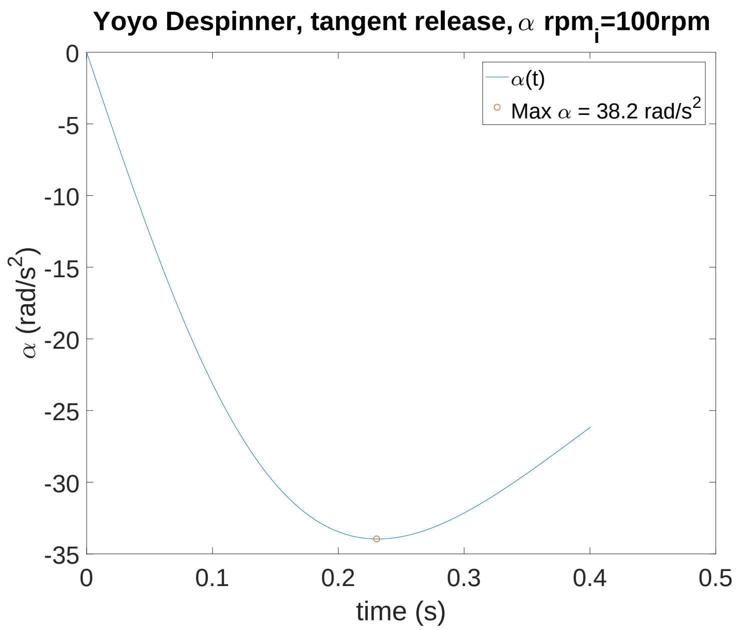


Figure 14: Tangential Model $\alpha(t)$ and maximum acceleration.

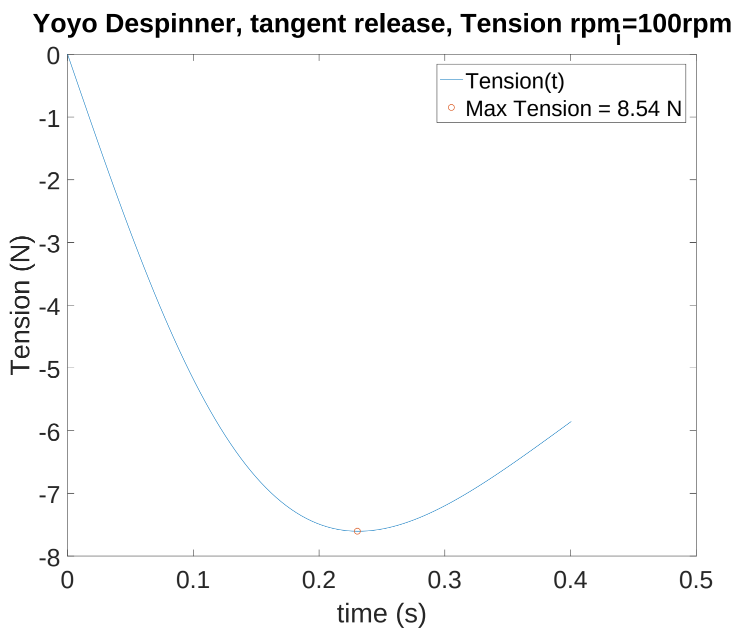


Figure 15: Tension in Tether Cord, from Model $\alpha(t)$.

IV. Applications

In modern sounding rockets, high spin rates are imparted on the rocket immediately following launch as a stabilization measure. During launch, the rocket is spun up by angling the fins at a slight cant angle relative to the vehicles direction of motion.[2] The rocket is then despun with a yo-yo despinner prior to the ejection of the primary stage of the rocket, and is necessary before the payload can begin its mission. The roll rate before despinning is much to high for the celestial ACS to handle, as the primary ACS system is only designed to handle small magnitude attitude adjustments. Therefore, a yo-yo despinner is used to rid the system of the majority of its angular velocity so it can begin making observations (where the ACS system needs extreme stability to track stars).

Figure 16: Sounding Rocket Despin. *Still image from launch footage, during the despin phase. Unwound weight attached to cable tether is highlighted in red. Immediately following the despin shown above, the primary stage of the rocket separates.*

Sounding rockets can achieve roll rate reductions of over 90% using a yo-yo despinner using despin masses weighing less than 1 kg each and with less than 1kg of total cable mass. Specifics of the yo-yo despin mechanism used on specific missions is proprietary information and thus cannot be included in this document. If more information about the specifics of the despinner is desired, please contact the P.I. of the CU Ultraviolet Astrophysics Rocket Group, Kevin France, at kevin.france@colorado.edu.

Ejection of the primary stage occurs when a pyrotechnic device is fired when the rocket reaches an altitude of approximately 80 km. This releases the spring loaded locking collar, located at the base of the gas spring in the primary stage, allowing it to rotate to the unlocked position and allowing the primary stage to separate from the rest of the payload.[3]

Spin stabilization keeps the rocket spinning during its ascent into the atmosphere, in order to reduce potential dispersion of the flight trajectory due to vehicle misalignments and manufacturing defects.[2] Spinning up the rocket also is done to create a gyroscopic precession effect ("coning"). This gyroscopic effect is induced in response to perturbations from the flight path due to the large angular momentum of the rocket when it is spinning. If a high enough roll rate is achieved, the rocket will remain relatively aligned with its original flight trajectory/path.

The yo-yo despinner is an extremely effective design for craft like a sounding rocket, but it does have some disadvantages. While the yo-yo despinner is a very simple, lightweight, and reliable design, it lacks the adjustability and modularity of other designs. A yo-yo despinner cannot be easily tailored to new conditions that the rocket experiences during liftoff, as caused by weather effects or variability in the launch conditions. Other solutions, such as reaction wheels or internal torque motors. Other methods can combine sensor feedback to more accurately control the current state of the rocket during launch. Another method for controlling the spin of the rocket could be orthogonally fired thrusters, which could provide the angular impulse necessary to slow the rocket properly. A solution involving thrusters would be more complex, and would require large quantities of propellant to provide the $\Delta\omega$ needed.

These solutions, which are normally already a part of the ACS system, can struggle to despin the rocket when it needs to be spun up as fast as is necessary in a sounding rocket, so they are generally not used. On account of the their efficacy, simplicity, and reliability, yo-yo despinners remain the most common method for despinning sounding rockets to this day.

V. Conclusions and Recommendations

This lab allowed for the modeling of an actual yo-yo despinner system, using multiple methods for the analysis. Derivations for the motion of the satellite were derived using principles of conservation of both energy and angular momentum, allowing for the development of an analytic models of the motion of the satellite during yo-yo despin. These models were compared with experimental data taken from an actual yo-yo despin experimental setup, and they were found to have a good match with experimental observation.

However, the experimental observations did not match the analytic models as well as they could. To help to better understand the discrepancy between the model and experimental results, a high speed video was taken of the experimental setup in action. This video revealed that the release regime of the yo-yo despinner was not properly accounted for in the models, as the despin masses actually swing past the point of radial release. The satellite also experiences a frictional couple by the motor once it is turned off, an effect that was not incorporated into the analytic model. The model, even when accounting for the error produced by the frictional couple, does not match the experimental observations perfectly. This discrepancy is likely a result of error in the measurements of key quantities, namely the moment of inertia of the satellite and the length of the despinner strings. Both of the uncertainties in the measurement of these quantities likely lead in part to the discrepancy seen between the model and experimental observations.

In the future, improved measurement of the important system-definition quantities would certainly help to improve the accuracy of the model. Incorporation of the frictional couple into the model would also greatly increase the accuracy of the model, especially later in the despin process as frictional losses begin to make the model and experimental observations diverge. Finally, correctly modeling the release of the despin weights would also increase the accuracy of the model's prediction of the final angular velocity.

References

[1] Axelrad, P. ASEN 2003 Lab 5: Yo-Yo Despinner. University of Colorado, Boulder. 2017.

[2] Sounding Rockets Program Office. NASA Sounding Rockets User Handbook. NASA Goddard Space Flight. July 2015. 181p. 810-HB-SRP. Guide to using NASA sounding rockets.

[3] Nelson M. W. Spin Stability of Sounding Rocket Secondary Payloads Following High Velocity Ejections. Utah State University. 2013. 77p.

Acknowledgments

We would like to thank Bobby Kane and Kevin France from the CU FUV Rocket Group for the information about sounding rockets.

Appendix A: MATLAB Code

```matlab
%{
The purpose of this code...

Written by:
Created: 3/21
Last modified:
%}
close all; clear all; clc;

%% Declare constants
%spacecraft constants
I = 165; %lb(m)*in^2, total moment of inertia without despin masses
R = 4.25; %in, outer radius
%convert to metric
I = I*0.000293; %kg*m^2
R = R*0.0254; %(in)*(m/in)=m

%other constants
m = 2*125/1000; %grams*kg/g = kg, 2 despin masses
w_106 = 106*2*pi/60; %rpm*rad/(s/min) = rad/s
w_102 = 102*2*pi/60; %rpm*rad/(s/min) = rad/s
w_100 = 100*2*pi/60; %rpm*rad/(s/min) = rad/s

%% Calculations for tangential release of despinners
% For comparison with w_0 = 106
[t_tan106, w_tan106, al_tan106, L_req106, t_req106] = tangent_theoretical_despin(I, R, m, w_106); %matrix with [t; alpha; omega; Tension]
% For comparison with w_0 = 102
[t_tan102, w_tan102, al_tan102, L_req102, t_req102] = tangent_theoretical_despin(I, R, m, w_102); %matrix with [t; alpha; omega; Tension]
% For comparison with w_0 = 100
[t_tan100, w_tan100, al_tan100, L_req100, t_req100] = tangent_theoretical_despin(I, R, m, w_100); %matrix with [t; alpha; omega; Tension]

%% Calculations for radial release required length
[L_req_rad105, w_rad106, t_rad106] = despinner_radial(I, R, m, w_106);
[L_req_rad102, w_rad102, t_rad102] = despinner_radial(I, R, m, w_102);
[L_req_rad100, w_rad100, t_rad100] = despinner_radial(I, R, m, w_100);

%% Experimental Data - No despin
[t_0in, w_0in, ~] = import_data('110_RPM_NoMass_Lubed.txt');

%% Experimental Data - Too short
[t_short, w_short, ~] = import_data('100_RPM_8_INCH.txt');

%time offset from data - doesn't despin until ~3s
t_tan102_short = t_tan102 +3;

%With a string length of 8in (0.2032 m), a tangential release would have released
```

```
%at:
t_full_length_short = 0.2032/w_102/R;
t_full_length_short = t_full_length_short +3;

%find index where t_13_5 is closest to t_full_length
[~, idx1] = min(abs(t_short-t_full_length_short));

%% Experimental Data - 13.6 in - Closest length
[t_13_6in, w_13_6in, al_13_6in] = import_data('100_RPM_13.6_INCH.txt');

%time offset from data - doesn't despin until ~3s
t_tan100_13_6 = t_tan100 +3;

%With a string length of 13.5in (0.34544 m), a tangential release would have
    released
%at:
t_full_length_13_6in = 0.34544/w_100/R;
t_full_length_13_6in = t_full_length_13_6in +3.03;

%find index where t_13_5 is closest to t_full_length
[~, idx2] = min(abs(t_13_6in-t_full_length_13_6in));

%% Experimental Data - Too Long
[t_long, w_long, ~] = import_data('106_RPM_16.125_INCH.txt');

%time offset from data - doesn't despin until ~3s
t_tan106_long = t_tan106 +3;

%With a string length of 16.125in (0.409575 m), a tangential release would
    have released
%at:
t_full_length_long = 0.409575/w_106/R;
t_full_length_long = t_full_length_long +3;

%find index where t_13_5 is closest to t_full_length
[~, idx3] = min(abs(t_long-t_full_length_long));

%% Plots
% experimental w as a function of time for the case without using the
% yo-yodespin mechanism
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
scatter(t_0in, w_0in,'.')
xlabel('time (s)')
ylabel('\omega (rad/s)')
title('Satellite Spin, no despin mechanism')
axis([3, 16.5, 0, 12])
set(gca,'fontsize',24);

% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
```

```
    [fid, errmsg] = fopen('nodespin.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'nodespin.pdf');
set(gca,'fontsize',24);

%Plot omega, experimental vs theoretical - Sample data, short string
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t_tan102+3, w_tan102)
hold on
plot(t_short(idx1), w_short(idx1),'*')
scatter(t_short, w_short)
xlabel('time (s)')
ylabel('\omega (rad/s)')
title('Class Data, L = 8 in, initial rpm = 100')
legend('Tangential release, theoretical','L_{tangent} reached','Radial release, experimental')
axis([2.9, 3.7, 0, 11])
set(gca,'fontsize',24);

% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('omega_short.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'omega_short.pdf');
set(gca,'fontsize',24);

%Plot omega, experimental vs theoretical - Sample data, 13.6in string
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t_tan100+3, w_tan100)
hold on
plot(t_13_6in(idx2), w_13_6in(idx2),'*')
scatter(t_13_6in, w_13_6in)
xlabel('time (s)')
ylabel('\omega (rad/s)')
title('Class Data, L = 13.6 in, initial rpm=100')
legend('Tangential release, theoretical','L_{string, tangent} reached','Radial release, experimental')
axis([3, 4, 0, 11])
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
```

```
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('omega_136.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'omega_136.pdf');

%Plot omega, experimental vs theoretical - Sample data, long string
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t_tan106_long, w_tan106)
hold on
plot(t_long(idx3), w_long(idx3),'*')
scatter(t_long, w_long)
xlabel('time (s)')
ylabel('\omega (rad/s)')
title('Class Data, L = 16.125 in, initial rpm=106')
legend('Tangential release, theoretical','L_{string, tangent} reached','Radial
     release, experimental')
axis([3, 4, 0, 12])
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('omega_long.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'omega_long.pdf');

%alpha plot
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t_tan100+3, al_tan100)
hold on
plot(t_13_6in, al_13_6in,'o')
xlabel('time (s)')
ylabel('\alpha (rad/s)')
legend('Tangential release, theoretical','Radial release, experimental')
title('Class Data, L = 13.6 in, initial rpm=100')
axis([3, 3.5, -50, 50])
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('alphacompare.pdf', 'w+');
```

```
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'alphacompare.pdf');
```

```
function [t, w, al] = import_data(fileID)
%% Get data from text file
%open file
data = load(fileID);

%% Set data for export
w = data(:, 2)*2*pi/60; %rpm->rad/s
t = data(:, 1);

%% Calculate angular acceleration al
dt = .01;
al = diff(w)/.01;
al = [al; 0];

end
```

```
function [L_req, w, t] = despinner_radial(I, R, m, w_0)
%UNTITLED6 Summary of this function goes here
%   Detailed explanation goes here

% will fill in more later: files would refer to any intake filenames needed

%constants
c = I/(m*R^2)+1; %coefficient c, describes the angular momentum at initial
    conditions

% Find required Length to despin satellite
syms w_f L
w_f = (m*R*w_0*sqrt(c)*(R+L)-(I+m*R^2)*w_0)/I == 0;
c = (I / (m *  R ^2)) + 1;
L_req = ( I / (m * R * sqrt(c)) ) + (R / sqrt(c)) - R;


%Find omega_f
%t_f = w_0*R/L_req;
t = linspace(0,.5);
L1 = linspace(0,L_req);
%L = w_0*R*t;
w = (m*R*w_0*sqrt(c)*(R+w_0*R*t)-(I+m*R^2)*w_0)/I;
w_f = (m*R*w_0*sqrt(c)*(R+L_req)-(I+m*R^2)*w_0)/I;
wL =(m*R*w_0*sqrt(c)*(R+L1)-(I+m*R^2)*w_0)/I;

%{
plot(t, w)
hold on
plot(L1,wL)
legend('w(t)','w(L)')
%}
end
```

```matlab
function [t, w_t, al_t, L_req, t_req] = tangent_theoretical_despin(I, R, m,
    w_0);
%ANALYZE Calculates the angular velocity and acceleration of a satellite,
%and tension of the despinner strings as a function of time
%satellite despinner for a tangential release.
%   More info....
%
% Inputs: -Total Moment of inertia of satellite (w/o despinners), I(kg*m^2).
%         -Outer radius of same satellite, R (m).
%         -Mass of each despinner, m (kg).
%         -Initial angular velocity before despinners released, w (rpm)
%         -Total time to compute, t_f
%         -Length of despinner strings, L
% Outputs: -none, but creates graphs
%
% Created by:
% Created on: 3/21
% Last editted:

%% Convert constants to consistent units, declare other necessary constants
 %coefficient c, describes the angular momentum at initial conditions
c = I/(m*R^2)+1;

%% Calculate required t and L
syms t
w = w_0*((c-w_0^2*t^2)/(c+w_0^2*t^2));
t_req = double(solve(w, t));
t_req = t_req(t_req>0); %t_req returns a negative and positive, extract pos

L_req = w_0*t_req*R;

%% Length with respect to time
t = linspace(0,t_req);
L = w_0*t*R;

%% Angular velocity vs. time, w(t)
w_t = w_0*((c-w_0^2*t.^2)./(c+w_0^2*t.^2));
%w_L = w_0*((c*R.^2-L.^2)/(c*R.^2+L.^2));

%% Angular acceleration, al(pha), vs time, al(t)
%al_L = -4*c*w_0^2.*(L/R)./(c+(L/R).^2).^2;

al_t = -4*c*w_0^3.*t./(c+w_0.^2*t.^2).^2;
al_max = min(al_t);

%find index
[~, idxal] = min(abs(al_t-al_max));

%% String Tension vs time, T(t)
%T_L = (-2*I*c*w_0.^2*L/R)/(R*(c+(L/R).^2).^2);

T_t = -2*I*c*w_0.^3*t./(R*(c+w_0.^2*t.^2).^2);
T_max = max(abs(T_t));

%find index
```

```
[~, idxT] = min(T_t-T_max);
%% Graphs
%al(t)
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t, al_t)
hold on
plot(t(idxal),al_t(idxal),'o')
xlabel('time (s)')
ylabel('\alpha (rad/s^2)')
title('Yoyo Despinner, tangent release, \alpha rpm_i=100rpm')
legend('\alpha(t)','Max \alpha = 38.2 rad/s^2')
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('al_max.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'al_max.pdf');

%w(t)
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t, w_t)
xlabel('time (s)')
ylabel('\omega (rad/s)')
title('Yoyo Despinner, tangent release, \omega rpm_i=100rpm')
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('omega_theor.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'omega_theor.pdf');

%T(t)
hFig = figure;
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)
plot(t,T_t)
hold on
plot(t(idxT),T_t(idxT),'o')
xlabel('time (s)')
```

```
ylabel('Tension (N)')
title('Yoyo Despinner, tangent release, Tension rpm_i=100rpm')
legend('Tension(t)','Max Tension = 8.54 N')
set(gca,'fontsize',24);
% setup and save figure as .pdf
    curr_fig = gcf;
    set(curr_fig, 'PaperOrientation', 'landscape');
    set(curr_fig, 'PaperUnits', 'normalized');
    set(curr_fig, 'PaperPosition', [0 0 1 1]);
    [fid, errmsg] = fopen('Tension_max.pdf', 'w+');
    if fid < 1 % check if file is already open.
       error('Error Opening File in fopen: \n%s', errmsg);
    end
    fclose(fid);
    print(gcf, '-dpdf', 'Tension_max.pdf');

end
```

```
function determineFricMoment(shouldSaveFigures)


    %% Setup

    %spacecraft constants
    I = 165; %lb(m)*in^2, total moment of inertia without despin masses
    R = 4.25; %in, outer radius
    %convert to metric
    I = I*0.000293; %kg*m^2
    R = R*0.0254; %(in)*(m/in)=m

    %other constants
    m = 2*125/1000; %grams*kg/g = kg, 2 despin masses
    w_0 = 107*2*pi/60; %rpm*rad/(s/min) = rad/s, initial angular velocity


    %%% Plotting
    set(0, 'defaulttextinterpreter', 'latex');
    titleString = 'omega_vs_t_no_despin';
    saveLocation = '../../Figures/';
    saveTitle = cat(2, saveLocation, titleString);
    LINEWIDTH = 2;
    MARKERSIZE = 3;
    FONTSIZE = 20;
    SCALE_FACTOR = 1;
    colorVecs = [0.294118 0 0.509804; % indigo
                 0.180392 0.545098 0.341176; % sea green
                 1 0.270588 0; % orange red
                 0 0.74902 1; % deep sky blue
                 0.858824 0.439216 0.576471; % forestgreen
                 0.133333 0.545098 0.133333; % palevioletred
                 0.803922 0.521569 0.247059; % peru
                 1 0.498039 0.313725]; % coral

    markers = {'+','o','*','.','x','s','d','^','v','>','<','p','h'};

    hFig = figure('name', titleString);
```

```
    scrz = get(groot, 'ScreenSize');
    set(hFig, 'Position', scrz)

    %% Calc Moment
    [~, ~, al] = import_data('110_RPM_NoMass_Lubed.txt');
    M = mean(al(500:1300)) * I;



    %% Plot

    % get data from when
    [t, w, ~] = import_data('100_RPM_0_INCH.txt');

    xmin = 0;
    xmax = inf;

    ymin = min(w)*0.9;
    ymax = max(w)*1.1;

    hold on
    grid on

%     leg_string = '';

    p1 = plot(t, w, ...
                [markers{2}, ':'], ...
               'linewidth', LINEWIDTH, 'markersize', MARKERSIZE, ...
               'Color', colorVecs(1, :));

    xlim([xmin, xmax])
    ylim([ymin, ymax])
    xlabel('$t$ (rad)')
    ylabel('$\omega$ (rad/s)')
%     leg = legend(p1, leg_string, ...
%                     'location', 'best', 'interpreter', 'latex');
    set(gca, 'FontSize', FONTSIZE)
    title(sprintf('$\\omega(t)$ vs. $t$ - No Despin Used'), 'fontsize', ...
          round(FONTSIZE * SCALE_FACTOR))
%     set(leg, 'FontSize', round(FONTSIZE * 0.7))
    set(gca, 'defaulttextinterpreter', 'latex')
    set(gca, 'TickLabelInterpreter', 'latex')

    %% setup and save figure as .pdf
    if shouldSaveFigures
        savefig(saveTitle, 'pdf', '-r500');
    end

    %% print the moment calculated
    fprintf(['The frictional moment on the spacecraft is: ', ...
               '%0.3g N-m or %0.3g ft-lb\n'], M, M * 0.737562149277)

end
```