PROBLEM SET #5

TO: PROF. DAVID MILLER, PROF. JOHN KEESEE, AND MS. MARILYN GOOD

FROM: NAMES WITHHELD

SUBJECT: PROBLEM SET #5 (STRUCTURE, ATTITUDE CONTROL, AND PROPULSION)

DATE: 12/11/2003

MOTIVATION

With greater demand for high-resolution images and the technological infeasibility of manufacturing and launching larger aperture optics, many missions are turning to interferometry as an alternative high-resolution imagery technique. One of the interesting aspects of structurally connected interferometry, as opposed to the use of formation-flying, is the trade off between acquiring higher-resolution images with a long spacecraft vs. shorter spacecraft that is less affected by the gravity gradient torque disturbance. While higher-resolution images may be desirable from the perspective of the science objective, a lighter spacecraft (i.e. shorter spacecraft with smaller effective aperture) may be more desirable from mass perspective. That is, a spacecraft with large effective aperture will be highly affected by the gravity gradient due to the long length of the spacecraft. With greater torque disturbance, larger reaction wheels are necessary to counter act the torque disturbance. In addition, the size of the reaction wheels as well as the size of the propulsion system used for momentum dumping will greatly depend on the frequency of momentum dumping. As such, a good balance among the design of the three subsystems (structure, attitude control, and propulsion) is necessary to produce high-resolution images with low mass spacecraft.

PROBLEM STATEMENT

Optimize the design of a satellite so that it has high-resolution imaging capability but with minimal mass spacecraft. Mass of the outer satellite structure, the reaction wheels, and the propulsion system should be considered. For a given satellite length, find the gravity gradient disturbance, the reaction wheels needed to compensate, and the propulsion system necessary for momentum dumping at a given rate. By performing trending analysis, find trade-offs between satellite size, and the reaction wheels and the propulsion system associated with the momentum dumping frequency. Note that the focus of this problem is not to design a high fidelity model of the subsystems, but rather to analyze the dependencies among the three aspects of the satellite system.

SOLUTION

A tool using MATLAB evaluates the three subsystems. Inputs to the tool are:

len: desired length of the satellite, in meters
rad: radius of orbit, in kilometers
life: lifetime of the orbit, in years
freq: frequency at which momentum dumping should occur, in number of orbits between each dump

The tool is called by the top-level code *angular_res.m*. To run the tool, at the MATLAB prompt, type:

angular_res(len, rad, life, freq)

with the inputs filled in accordingly.

The tool first evaluates the structure to determine the dimensions of the structure, the mass, and the moments of inertia. Next, the tool determines the attitude control subsystem, by finding the momentum storage needed in the reaction wheel, and the mass the wheel will be, to compensate for the gravity gradient torque from caused by the moments of inertia. Finally, the tool determines the mass of propellant needed perform the necessary momentum dumping, as well as the mass of the tank needed to hold the propellant. Propellant needed for other maneuvers is not considered, and neither is mass for other satellite systems.

Structure

Given the limitation on current space launching capability, a deployable truss system, such as those manufactured by AEC ABLE, are ideal for structurally connected interferometry. For the purpose of this trade study, the dimensions (length and with), mass moment of inertias, and total mass of the truss must be computed. To simplify the trade study and the design process, certain design parameters are fixed.

First, we assume that the truss members are cylindrical tubes made of graphite/epoxy with the outer radius of 2.5 cm and thickness of 2 mm. This assumption is representative of the designs used by AEC ABLE. Furthermore, change in the structural properties and dimension of the truss members will not affect the relative design results of varying truss lengths. As such, the overall trade analysis should not be affected by these design decisions.

We also assume that each bay of the truss is a cube with diagonal members on four sides of the cube along the longerons (see Figure 1(a)). The mass per unit length, ρ_l, of such truss is

$$\rho_l = \rho_v \pi \left(r^2 - (r-t)^2 \right) \left(8 + 4\sqrt{2} \right) \tag{1}$$

where ρ_v is the density, r is the outer radius, and t this the thickness of the truss member. The fundamental bending frequency, ω_n, of such truss system can be computed by approximating the truss as a beam with pinned ends:

$$\omega_n = A^2 \sqrt{\frac{EI}{\rho_l l^4}} \tag{2}$$

where A is a constant 4.73, E is the modulus of elasticity, I is the area moment of inertia, and l is the length of the beam. By restricting the fundamental bending frequency to be no more than 5 Hz (a typical value for space structures), the area moment of inertia can be determined. The area moment of inertia is given by Eq.(3), which assumes that the diagonal members have negligible affect on bending.

$$I = \pi \left(r^4 - (r-t)^4 + \left(r^2 - (r-t)^2 \right) w^2 \right) \tag{3}$$

From Eq.(3) the width of the truss system, w (i.e. the length of each bay), can be computed.

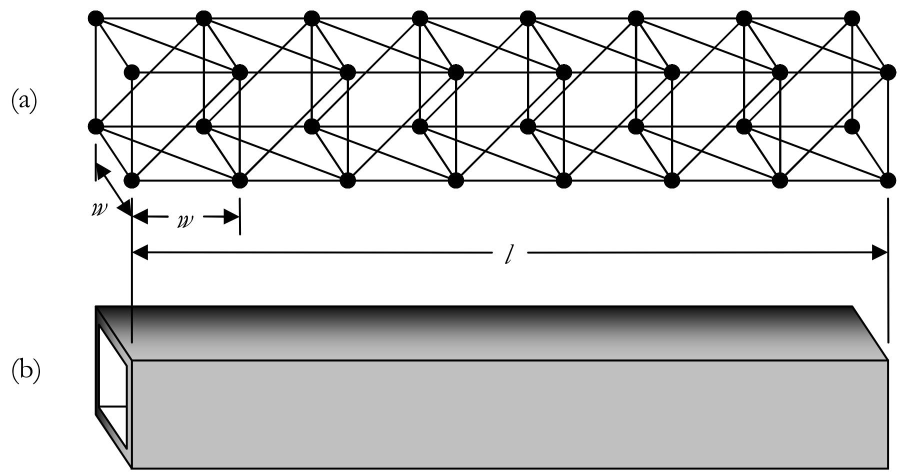

Figure 1. Truss structure configuration. (a) Illustrates how the truss members are distributed. (b) Illustrates a rectangular prism shell used to approximate the mass moment of inertia.

While the mass moment of inertia can be computed by taking each truss members into account, we take a simpler approach where we approximate the truss as a rectangular prism shell as shown in Figure 1(b). Recognizing that the shell is simply a smaller rectangular prism cutaway from a larger rectangular prism, the formula for calculating the mass moment of inertia of a rectangular prism is used.

Using the aforementioned methodology, the properties of the trusses with varying length can be determined. Figure 2(a) illustrates that the mass of the truss increases linearly with the length of the truss. The width of the truss, as shown in Figure 2(b), does not increase linearly with the length of the truss. This nonlinearity is due to the fact that the width is constrained such that the resulting fundamental bending frequency is no less than 5 Hz. Also, note that the line is not smooth. This is due to the design requirement that each bay of the truss is a cube. Thus, the allowable length of the truss is discrete based on the width of the truss. This effect will resurface when three subsystems are combined for a single trade study.

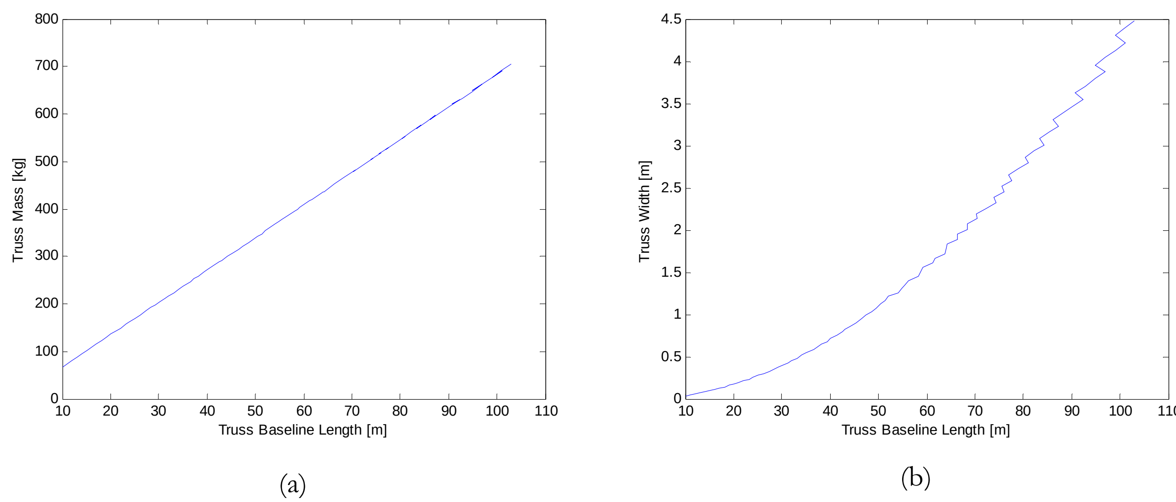

Figure 2. (a) Mass of a truss as a function of the length of the truss. (b) Width of the truss as a function of the length of the truss.

Finally, the mass moment of inertia is illustrated in Figure 3. This illustrates that the mass moment of inertial in the cross-sectional direction increase more rapidly than the mass moment of inertial in the longitudinal direction of the truss.

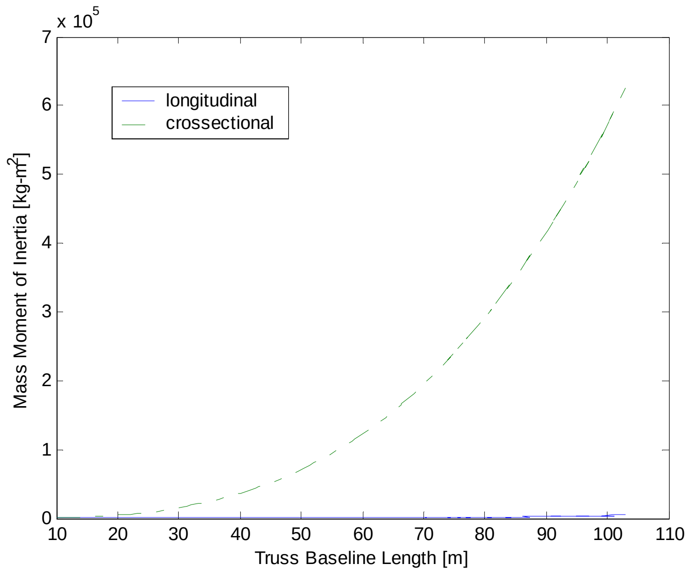

Figure 3. Mass moment of inertial as a function the truss length.

This analysis was done using the code *compute_truss_properties.m*.

Attitude Control

The attitude control system is determined by the code *reaction_wheels.m* and *get_wheel_data.m*. Reaction wheels can be used to compensate for the gravity gradient torque that occurs as a result of the shape of the satellite structure. The equation for finding the gravity gradient torque is:

$$T_g = \left(\frac{3\mu}{2R^3}\right) \cdot \left|I_{xx} - I_{yy}\right| \cdot \sin\left(2\theta\right) \quad (4)$$

The moments of inertia about x and y (I_{xx} and I_{yy}) were found in the previous module, by modeling the satellite structure as a truss. The earth's gravity constant, μ, is 3.986×10^{14} m^3/s^2, and R is the radius of the orbit in meters. Finally, the angle θ is the maximum deviation of the z-axis from local vertical. So that the reaction wheels can account for worst case conditions, this is set to 45 degrees.

To find the momentum storage needed in a reaction wheel to compensate for this torque, first the momentum storage for one orbit is found. The assumption is made that the satellite is inertially oriented, so that the gravity gradient torque will be cyclic, and accumulates in ¼ of an orbit. The momentum storage need for one orbit is then found by integrating over the orbit period P:

$$momentum = T_g P \frac{0.707}{4} \tag{5}$$

where 0.707 is the rms average of the sinusoidal function. The value is then scaled by the frequency at which momentum dumping occurs. The input *freq* represents the number of orbits that occur between each momentum dump, so the total momentum storage needed is:

$$T_{momentum} = momentum \cdot freq \tag{6}$$

The mass of a reaction wheel that can store this momentum was found by collecting data on a variety of real reaction wheels. The sources for this data is found in the *References* section. The real data can be plotted, and then a curve can be fit to it. Based on this curve, mass estimates for the system designed by the tool can be determined. The real data and the curve fit is show in Figure 1. While fitting with a polynomial curve gave better results for data in the range of 0 to 100 Nms, larger values can potentially yield negative masses. For that reason, a linear fit was done. Note that the curve fit does not give zero mass for zero momentum do to how MATLAB's fitting algorithm; this point must be taken into consideration when examining results.

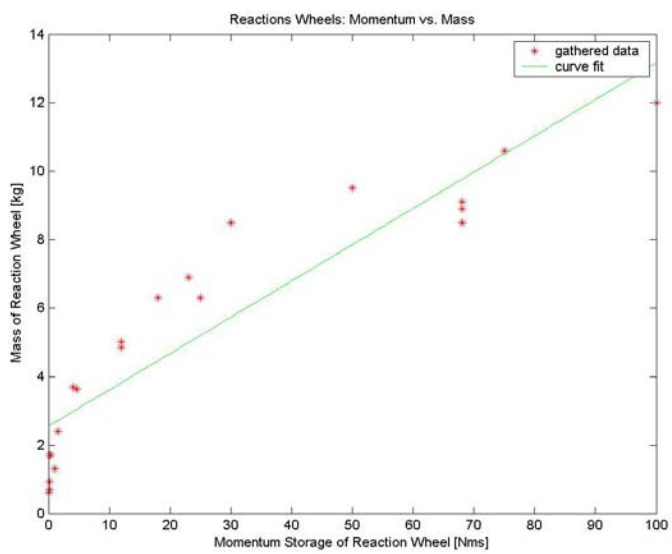

Figure 4. Momentum Storage of Reaction Wheels vs Mass

Lastly, the total number of momentum dumps that are done throughout the satellite lifetime is calculated in this section, to be used in determining the propulsion system. This value is based on the period of the orbit and frequency at which momentum dumps are done.

Trending was performed on this module to ensure that it was behaving correctly. Orbit size was always fixed to 6600 km radius. First, the moments of inertia were varied while dumping frequency was held constant at once per orbit. Next, the moments of inertia were held constant ($|I_{xx} - I_{yy}| = 500$ kg*m^2) while the frequency was varied from dumping once per orbit to once every 100 orbits. Based on these inputs, mass of the reaction wheel and stored momentum of the reaction wheel can be found. Because of the nature of the equations above, and the linear curve fit, the results were linear, as expected. They are shown in Figure 5, Figure 6, Figure 7, and Figure 8.

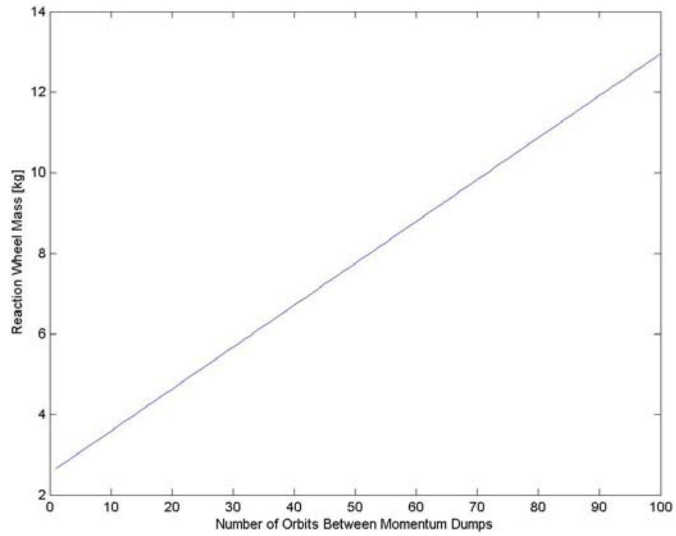

Figure 5. Frequency of Momentum Dumping vs Reaction Wheel Mass

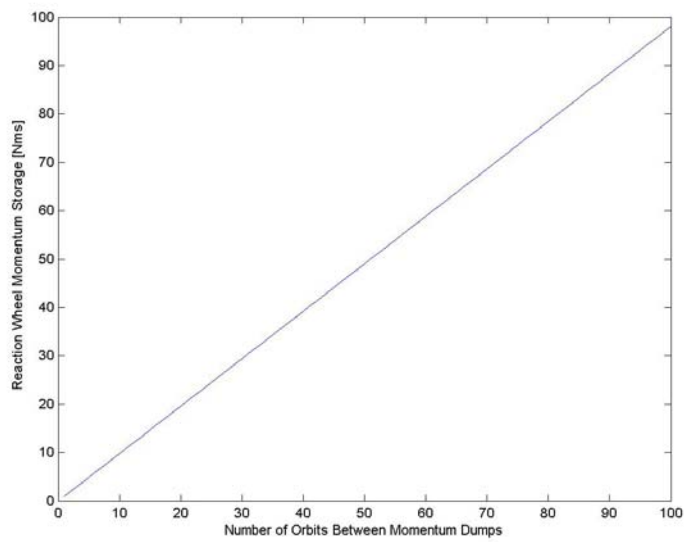

Figure 6. Frequency of Momentum Dumping vs Reaction Wheel Momentum

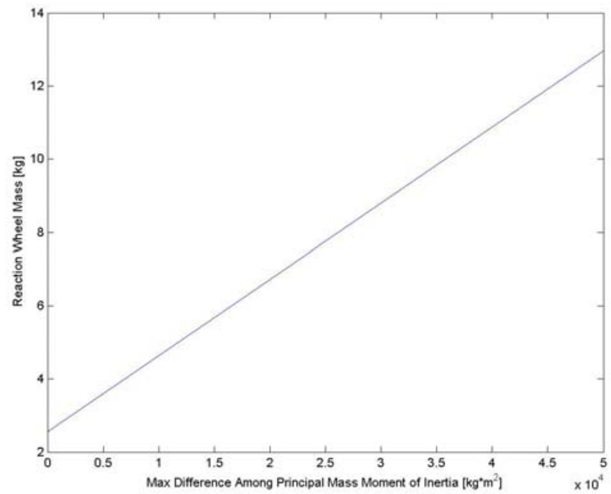

Figure 7. Moments of Inertia vs Reaction Wheel Mass

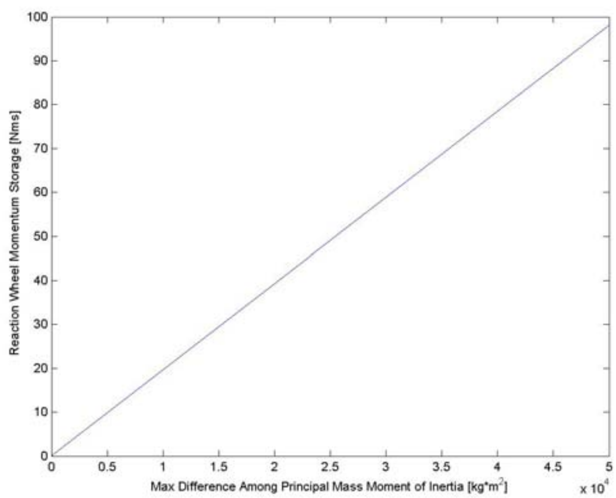

Figure 8. Moments of Inertia vs. Reaction Wheel Momentum

<u>Propulsion</u>

The propulsion system used for the attitude control is assumed to be a hydrazine monopropellant system that uses a simple blowdown pressurization system. The pressurant is nitrogen gas with an operating temperature of 293.15 K. As typically required by hydrazine thrusters, the tank is at 300 psi when it is full and the tank is at 100 psi when the tank is empty of all propellants. Note that these design and operational assumptions are common among hydrazine monopropellant systems. Then, given the total required propellant mass[1], m_{prop}, the necessary volume of a tank, V_{tank}, can be computed according to Eq (7):

[1] The mass of the propellant necessary for momentum dumping is determined from the number of momentum dumping, I_{sp} of the thruster, and the minimum moment arm of the thruster which is assumed to be 1 m. This computation is done using *compute_hydrazine_mass_for_momentum_dumping.m*.

$$V_{tank} = \frac{m_{prop}}{\rho_{prop}\left(1 - \frac{\rho_{pres,EOL}}{\rho_{pres,BOL}}\right)} \quad (7)$$

where ρ_{prop} is the density of the propellant, $\rho_{press,BOL}$ is the density of a pressurant when the tank is full, and $\rho_{press,EOL}$ is the density of a pressurant when the tank is empty. The density of a gas can be accurately estimated using the Beattie-Bridgeman equation. The pressurant mass, m_{press}, can also be computed, where

$$m_{press} = \frac{\rho_{press,EOL}}{V_{tank}} \quad (8)$$

Assuming a spherical titanium tank and a safety factor of 1.5, the tank thickness can be computed using a simple pressure vessel hoop stress equation. Finally, given the thickness and the volume of a spherical tank, the mass of the tank can be determined.

Using this exact methodology, *design_propulsion_system.m* computes the pressurant mass, tank mass, and tank dimension given the propellant mass. Figure 9 illustrates the linear relationship among tank, pressurant, and propellant mass. Notice that the pressurant mass is not negligible with respect to the tank mass.

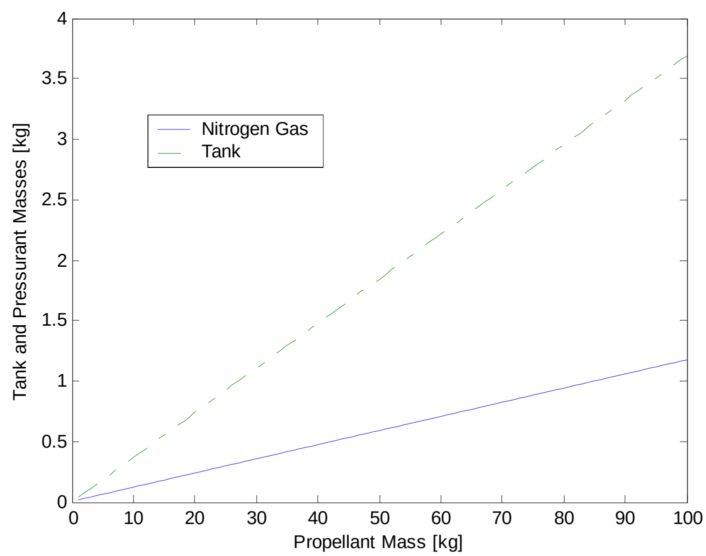

Figure 9. Tank and pressurant mass as a function of propellant mass.

Figure 10 illustrates the radius of a tank as a function of the propellant mass. This third order polynomial trend is as expected since the volume increase cubically with respect to the radius.

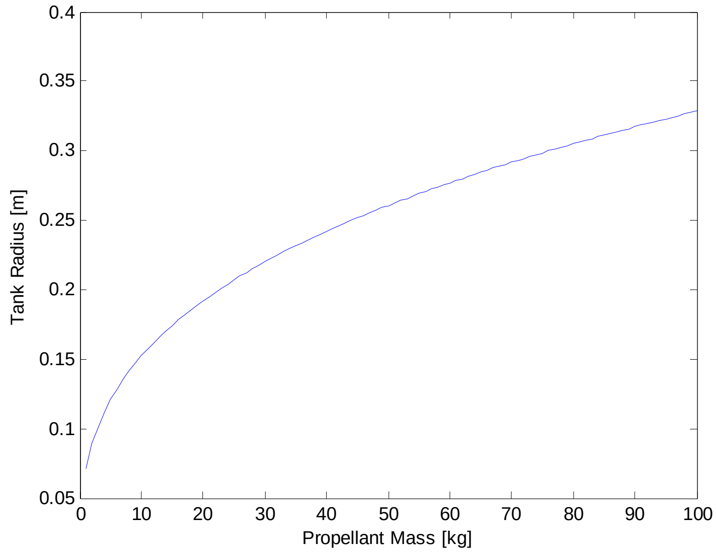

Figure 10. Tank radius as a function of propellant mass.

Finally, Figure 11 illustrates how the tank mass estimation method compares to the manufactured spherical titanium tanks. A typical way to compare tanks is by PV/W values, where P is the burst pressure of the tank, V is the volume, and W is the weight of the tank. In Figure 11, the dots represent the true PV/W of spherical titanium tanks manufactured by Pressure Systems, Inc. The line is what was computed. Note that the computed PV/W is on the higher end of the spectrum, i.e. the estimated tank mass is lighter than the manufactured tanks. This is expected since the manufactured tanks includes parts other than a true spherical pressure vessel, such as the boss (pipe mounting points, thicker than the pressure vessel), skirt (used for mounting the tank), pressure management device (device inside the tank that assures a possible pressure so that the liquid propellants flow out), and the liner inside the tank.

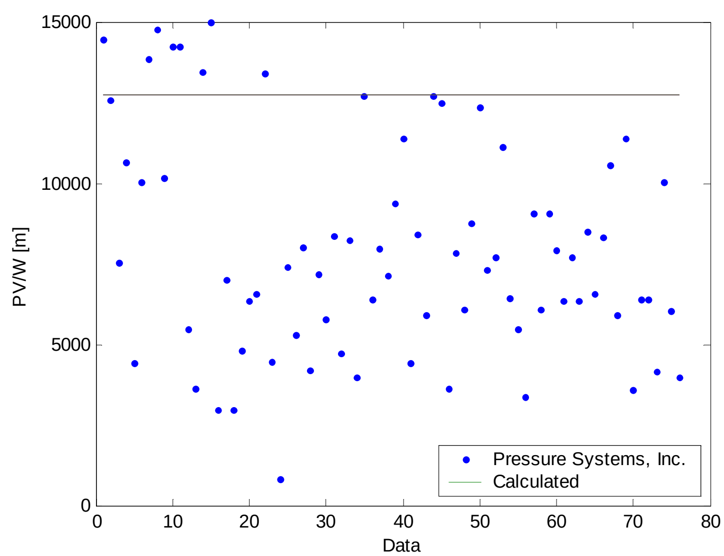

Figure 11. Calculated tank mass and manufactured tank mass comparison.

CONCLUSIONS

Tests were done on each module, and then on the full tool so that overall trades could be examined. For the sake of brevity, only the overall system trades will be shown here.

This tool allows trades to be shown between the length of the satellite structure, the frequency at which momentum dumping should occur, and the mass of the system. The length of the structure was varied from between 10 and 100 meters, and the frequency of momentum dumping was varied from once an orbit to once every 90 orbits (at intervals of 15). Mission lifetime was set at 2 years, and radius at 6600 km. The result is shown in Figure 6.

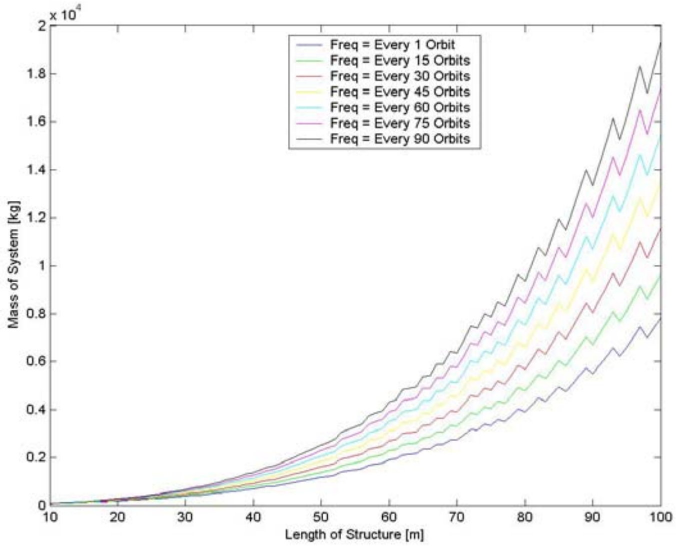

Figure 12. System mass as function of the structure length and the frequency of momentum dumping.

As seen in the figure, mass increases with increasing length of structure and less frequent momentum dumping. The shape of the graph follows the trends seen in the trends of the individual subsystems. In particular, the jagged curve is expected, as described in the *Structures* section. While decreasing the momentum dumping frequency decreases the propulsion system mass and increases the reaction wheel mass, this trade is not visible in the figure since the reaction wheel mass dominates over both the propulsion and structure mass (while others are within 100's of kg, the reaction wheel mass is in the order of 1000's of kg). This implies that, (1) the worst case disturbance cannot be rejected simply by a reaction wheel (to stay within reasonable reaction wheel mass), or (2) the mass moment of inertia estimate is not off beyond the order of magnitude.

Varying the inputs for both life and orbit radius would provide additional trending data. The farther the orbit is from earth, the less the satellite will be affected by the gravity gradient, and the smaller the reaction wheels will need to be. The longer the orbit lifetime, the more momentum dumps must be done, and this will increase the mass of the propulsion system.

The trends should be considered in the context of the original problem as well. While a greater structure length may result in a larger mass, it also provides more angular resolution. A satellite designer may be willing to build a satellite with a larger mass, as shown in these trade-offs, to get a certain angular resolution that is needed for a given mission.

FUTURE WORK

The first future task should be verification of the mass moment of inertia using a FEM code. This will verify the accuracy of the mass moment of inertia estimate as well as other properties of the truss (note that this work is beyond the scope of homework).

An addition to this tool would be a module that includes optics, so that the actual angular resolution for a given satellite length could be calculated. Then, trends could be seen between structure, optics, propulsion, and ACS. The tool would be able to show trends for how mass changes with different angular resolutions in addition to what it shows now.

REFERENCES

Wertz and Larson, Space Mission Analysis and Design, Third Edition, El Segundo: Microcosm Press, 1999

E. Ahronovich, M. Balling, "Reaction Wheel and Drive Electronics For LeoStar Class Space Vehicles", 12th Annual USU Conference on Small Satellites, 1998, www.sdl.usu.edu/conferences/smallsat/proceedings/12/ssc98/1/ssci5.pdf

Dynacon Enterprises Limited, "Dynacon MicroWheel 200", www.dynacon.ca/pdf/files/productpdf_6.pdf

Honeywell Aerospace Electronic Systems, "Constellation Series Reaction Wheels", http://content.honeywell.com/dses/assets/datasheets/constellation_series_reaction_wheels.pdf

Honeywell Aerospace Electronic Systems, "Miniature Reaction Wheels", http://content.honeywell.com/dses/assets/datasheets/mini-wheel_reaction_wheel.pdf

Honeywell Aerospace Electronic Systems, "Honeywell Model HR 0610 Reaction Wheel", http://content.honeywell.com/dses/assets/datasheets/hr0610_reaction_wheel.pdf

Teldix Space Product Group, "Momentum and Reaction Wheels 14-68 Nms with external Wheel Drive Electronics", http://www.teldix.de/P22/RDR23-68.pdf

Teldix Space Product Group, "Momentum and Reaction Wheels 14-68 Nms with integrated Wheel Drive Electronics", http://www.teldix.de/P22/RSI25-68.pdf

Teldix Space Product Group, "Momentum and Reaction Wheels 2-12 Nms with integrated Wheel Drive Electronics", http://www.teldix.de/P22/RSI4-12.pdf

Teldix Space Product Group, "High motor torque Momentum and Reaction Wheels 14-68 Nms with integrated Wheel Drive Electronics", http://www.teldix.de/P22/RSI18-68.pdf

Teldix Space Product Group, "Momentum and Reaction Wheels 0.04-0.12 Nms with integrated Wheel Drive Electronics", http://www.teldix.de/P22/RSI01.pdf

Teldix Space Product Group, "Momentum and Reaction Wheels 0.2-1.6 Nms with integrated Wheel Drive Electronics", http://www.teldix.de/P22/RSI02.pdf

MATLAB CODE

```
function [mass] = angular_res(len, rad, life, freq)

% INPUTS
% len: length of the satellite (m)
% rad: radius of orbit (km)
% life: mission lifetime (years)
% freq: frequency of momentum dumping (orbits/dump)

% OUTPUTS
% mass: mass of system including satellite body, reactions wheels, and
% propulsion system (kg)

% find mass and angular momentum of the structure
[structure_mass,w,l,I_xx,I_yy] = compute_truss_properties(len);

% find reaction wheels needed to counter-act gravity gradient torque
dI = abs(I_xx - I_yy);
[wheel_mass, momentum, tdumps] = reaction_wheels(rad*1000, dI, freq, life);

% find propulsion system
hydrazine_mass = compute_hydrazine_mass_for_momentum_dumping(momentum, tdumps);
[nitrogen_mass,tank_mass,tank_radius] = design_propulsion_system(hydrazine_mass);

mass = nitrogen_mass + tank_mass + hydrazine_mass + wheel_mass + structure_mass;
```

```
function [m,w,l,I_xx,I_yy] = compute_truss_properties(baseline)
% baseline

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Structure Requirement
minimum_fundamental_frequency = 5*2*pi; % [rad/s]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Material Properties
modulus_of_elesticity = 220e9;      % [N/m^2] Graphite/Epoxy
mass_density = 1640;                % [kg/m^3] Graphite/Epoxy

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Design Assumption
member_radius = 0.025;              % [m]
member_shell_thickness = 0.002;     % [m]
member_radius_inner = member_radius - member_shell_thickness;

% Assume the the mass per length, rho, is approximately equal to the mass of a
% single bin per bin length. A single bin consists of 4 longerons, 4
% diagnonal members, and 4 cross-sectiona members. A bin is assumed to be a
% cube.
mass_per_length = (8+4*sqrt(2))*mass_density*pi*...
        (member_radius^2 - member_radius_inner^2);   % [kg/m]

% Beam Bending: E*I*d^4(y)/(dx)4 - rho*omega^2*y = 0
%   where E is modulus of elesticity, I is the area moment of intertia, y
%   is the vertical displacement at the beam lenght x, rho is mass per unit
%   length, and omega is the vibartional frequency. The solution of the
%   equation is:
%
%       y = A*cosh(beta*x) + B*sinh(beta*x) + C*cos(beta*x) + D*sin(beta*x)
%
%   Assuming pin connections at each end of the beam, the boundary
%   condition is:
%
%       y(0) = 0, y'(0) = 0, y(l) = 0, y'(l) = 0
%
%   where l is the lenght of the beam. This results in the solution:
%
%       cosh(beta*l)*cos(beta*l) = 1
```

```
%
%   beta*l that corresponds to the fundamental mode is:
%
%       beta*l_fundamental = 4.73004074486271
%
%   The natural frequency is:
%
%       omega_n = (beta*l)^2*sqrt(E*I/rho/l^4)
%
%   Then, given the minimum fundamental vibration frequency, the minimum
%   area moment of inertia can be computed. This assumes that the tortional
%   vibration is negligible.
%
%       I = (omega_n/(beta*l)^2)^2*rho*l^4/E
fundamental_beta_l = 4.73004074486271;
minimum_area_moment_of_inertia = (minimum_fundamental_frequency/fundamental_beta_l^2)^2*...
                            mass_per_length*baseline.^4/modulus_of_elesticity;

% Assuming that the diagonal members do not contribute to beam bending, the
% area moment of inertia is:
%
%   I = pi*(r_o^4 - r_i^4 + (r_o^2 - r_i^2)*l^2)
%
% This assumption is valid since the bending deflection is assumed small
% and the diagonal members are in the middle whwere it goes under both
% compression as well as tension.
minimum_bay_length = sqrt((minimum_area_moment_of_inertia/pi - ...
                        member_radius^4 + ...
                        member_radius_inner^4)...
                       /(member_radius^2 - member_radius_inner^2));
minimum_bay_number = ceil(baseline./minimum_bay_length);
minimum_truss_length = minimum_bay_number.*minimum_bay_length;
minimum_truss_mass = mass_per_length.*minimum_truss_length + ...
                  4*mass_density*pi*(member_radius^2 - member_radius_inner^2).*minimum_bay_length;

% Assuming that the truss is much llike a retangular prism shell
retangular_prism_inner_volume = (minimum_bay_length - 4*member_radius).^2.*minimum_truss_length;
retangular_prism_outer_volume = minimum_bay_length.^2.*minimum_truss_length;
retangular_prism_shell_volume = retangular_prism_outer_volume - retangular_prism_inner_volume;
mass_density_of_retangular_prism = minimum_truss_mass./retangular_prism_shell_volume;
retangular_prism_inner_mass = mass_density_of_retangular_prism.*retangular_prism_inner_volume;
retangular_prism_outer_mass = mass_density_of_retangular_prism.*retangular_prism_outer_volume;
mass_moment_of_inertia_xx = 1/12*retangular_prism_outer_mass.*(2*minimum_bay_length.^2) - ...
                        1/12*retangular_prism_inner_mass.*(2*(minimum_bay_length - 4*member_radius).^2);
mass_moment_of_inertia_yy = 1/12*retangular_prism_outer_mass.*(minimum_bay_length.^2 + minimum_truss_length.^2) - ...
                        1/12*retangular_prism_inner_mass.*((minimum_bay_length - 4*member_radius).^2 + minimum_truss_length.^2);

m = minimum_truss_mass;
w = minimum_bay_length;
l = minimum_truss_length;
I_xx = mass_moment_of_inertia_xx;
I_yy = mass_moment_of_inertia_yy;
```

```
function [mass, momentum, tdumps] = reaction_wheels(rad, dI, freq, life)

% INPUTS
% rad: radius of orbit (m)
% dI: abs(I_xx - I_yy), difference in moment of inertias (kg*m^2)
% freq: frequency momentum is dumped (orbits/dump)
% life: mission lifetime (years)

% OUTPUTS
% mass: mass of one reaction wheel
% momentum: momentum storage of one reaction wheel

% the worst-case angle of the satellite from the locat vertical is pi/4
% sin (2*pi/4) = 1
```

```
% Earth's gravity constant
mu = 3.986e14; % m^3/s^2

% gravity gradient torque Tg [Nm]
Tg = ((3*mu)/(2*rad^3))*dI;

% orbit period [sec]
P = 2*pi*sqrt(rad^3/mu);

% momomentum storage needed in reaction wheels
% assume inertially oriented: torque is cyclic
% for one orbit
m_orbit = Tg*(P/4)*0.707;

% momentum that could build between each dump
momentum = m_orbit*freq;

% get the polynomials for a curve fit of the wheel data
p = get_wheel_data;

% find the mass for the needed momentum value
mass = polyval(p, momentum);

% total orbits in the lifetime
torbits = life*365*24*3600/P;
% total number of momentum dumps
tdumps = torbits./freq;

----------------------------------------------------------------------------------------------------------------------------------------------------------------
function p = get_wheel_data

% wheel data (mass and angular momentum) for different commercial reaction
% wheels.  this function fits a curve to this data (momentum vs. mnass)

wheel(1) = struct('name', 'Teldix RSI 01-5/15', 'ang_moment', 0.04, 'mass', 0.6);
wheel(2) = struct('name', 'Teldix RSI 01-5/28', 'ang_moment', 0.12, 'mass', 0.7);
wheel(3) = struct('name', 'LeoStar', 'ang_moment', 4.7, 'mass', 3.628);
wheel(4) = struct('name', 'Dyncon MicroWheel 200', 'ang_moment', 0.18, 'mass', 0.93);
wheel(5) = struct('name', 'Honeywell HR12', 'ang_moment', 50, 'mass', 9.5);
wheel(6) = struct('name', 'Honeywell HR14', 'ang_moment', 75, 'mass', 10.6);
wheel(7) = struct('name', 'Honeywell HR16', 'ang_moment', 100, 'mass', 12);
wheel(8) = struct('name', 'Honeywell Miniature Reaction Wheel', 'ang_moment', 1.0, 'mass', 1.3);
wheel(9) = struct('name', 'Honeywell HR0610', 'ang_moment', 12, 'mass', 5.0);
wheel(10) = struct('name', 'Teldix DR23-0', 'ang_moment', 23, 'mass', 6.9);
wheel(11) = struct('name', 'Teldix RDR68-6', 'ang_moment', 68, 'mass', 9.1);
wheel(12) = struct('name', 'Teldix RSI 25-75/60', 'ang_moment', 25, 'mass', 6.3);
wheel(13) = struct('name', 'Teldix RSI 68-75/60x', 'ang_moment', 68, 'mass', 8.5);
wheel(14) = struct('name', 'Teldix RSI 4-75/60', 'ang_moment', 4, 'mass', 3.7);
wheel(15) = struct('name', 'Teldix RSI 12-75/60x', 'ang_moment', 12, 'mass', 4.85);
wheel(16) = struct('name', 'Teldix RSI 18-220/45', 'ang_moment', 18, 'mass', 6.3);
wheel(17) = struct('name', 'Teldix RSI 30-280/30', 'ang_moment', 30, 'mass', 8.5);
wheel(18) = struct('name', 'Teldix RSI 68-170/60', 'ang_moment', 68, 'mass', 8.9);
wheel(19) = struct('name', 'Teldix RSI 02-25/30', 'ang_moment', 0.2, 'mass', 1.7);
wheel(20) = struct('name', 'Teldix RSI 04-25/60', 'ang_moment', 0.4, 'mass', 1.7);
wheel(21) = struct('name', 'Teldix RSI 1.6-25/60', 'ang_moment', 1.6, 'mass', 2.4);

for(i=1:length(wheel))
   ang(i) = wheel(i).ang_moment;
   mass(i) = wheel(i).mass;
end


[p,s] = polyfit(ang, mass, 1);
f = polyval(p, ang);
% used for creating plot
%ang_new = 0:1:100;
%plot(ang, mass, 'r*',ang_new, polyval(p, ang_new), 'g-');

%xlabel('Momentum Storage of Reaction Wheel [Nms]');
```

```
%ylabel('Mass of Reaction Wheel [kg]');
%legend('gathered data', 'curve fit');
%title('Reactions Wheels: Momentum vs. Mass');

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
function hydrazine_mass = compute_hydrazine_mass_for_momentum_dumping(stored_momentum, number_of_momentum_dumping)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
% Design Assumptions
%   o Hydrazine propellant
I_sp = 220;                 % [s]
minimum_moment_arm = 1;     % [m] minimum distance to a thruster

hydrazine_mass = number_of_momentum_dumping*stored_momentum/minimum_moment_arm/I_sp/9.81;

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
function [nitrogen_mass,tank_mass,tank_radius] = design_propulsion_system(hydrazine_mass)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants
hydrazine_density = 1008.153082;                    % [kg/m^3]
titanium_allowable_stress = 830e6;                  % [N/m^2]
titanium_density = 4430;                            % [kg/m^3]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Design Assumptions
%   o Blowdown monopropellant system
%   o Hydrazine propellant
%   o Operating at 293.15 K
%   o Density of nitrogen gas calcuated using Beattie-Bridgeman Equation
safety_factor = 1.5;
maximum_expected_operating_pressure = 300*4.44822/0.0254^2; % [N/m^2]
end_of_life_pressure = 100*4.44822/0.0254^2;                % [N/m^2]
beginning_of_life_nitrogen_density = 23.88991766;           % [kg/m^3]
end_of_life_nitrogen_density = 7.939967137;                 % [kg/m^3]

initial_propellant_volume = hydrazine_mass./hydrazine_density;

tank_volume = initial_propellant_volume/...
        (1 - end_of_life_nitrogen_density/beginning_of_life_nitrogen_density);

nitrogen_mass = end_of_life_nitrogen_density*tank_volume;

% Assume negligible thickness: V = 4/3*pi*r^3
tank_radius = (3*tank_volume/4/pi).^(1/3);

tank_thickness = safety_factor*maximum_expected_operating_pressure*tank_radius/...
          (2*titanium_allowable_stress);

% Assume thin wall: V = 4*pi*r^2*t
tank_mass = titanium_density*(4*pi*tank_radius.^2.*tank_thickness);
```