Unsteady Behavior of a Pin-Pin Beam

Alexander Kessler
California Polytechnic State University, San Luis Obispo, CA, 93407, USA

This report encompasses analysis done to assist a structural designer in designing a beam for a spacecraft to conform to maximum deflection and vibration metrics. Using finite differencing methods, the vibration behavior is found to be problematic for conforming to the metrics. Further analysis with an improved method and more considerations to vibration behavior in the design of the beam are needed to correct this.

I. Introduction

This document outlines the analysis of a structural designer for a new long-range communication spacecraft. Because of the design specifications laid out by the customer, the spacecraft is too large to fit the launch vehicle. The spacecraft has instead been split into a main body and a deploy-able antenna and electronics system. This antenna is attached to a telescoping beam of exactly pi meters in length. We can model the initial positions of all scenarios of meteorite impacts with this equation, where m is the number of the mode:

$$y = sin(m * x) \tag{1}$$

The designer discerns from this equation that the maximum deflection of a first-mode collision is below the allowable and thus has no need to worry about that consideration, but still has no clue of the vibration behavior associated with each case. For this we will need to mode the displacement of the beam from its original position with respect to time. We will delve deeper into the nature of the method used in the next section. Once we can model the beam deflection with respect to its length and time, we can analyze the vibrational behavior.

II. Methods

We will need to model how the beams displacement from its original position changes with respect to its length along the beam and time. Finite differencing is a numerical method to approximate derivatives that can accomplish this task by evaluating a function at two or more points with a special step, h, in between and dividing the difference by a combination of coefficients or powers applied to h. Specifically, we will use two different finite differencing equations of first and second order accuracy in time and second order accuracy in space:

$$W_{i,j+1} = \frac{Q}{2} * (W_{i+2,j} - 4W_{i+1,j} + 6W_{i,j} - 4W_{i-1,j} + W_{i-2,j}) + W_{i,j} \tag{2}$$

$$W_{i,j+1} = Q * (W_{i+2,j} - 4W_{i+1,j} + 6W_{i,j} - 4W_{i-1,j} + W_{i-2,j}) + 2W_{i,j} - W_{i,j-1} \tag{3}$$

These methods have stability conditions associated with them that involve the value for Q being small and negative. For this we will assign a value of -0.2 to Q. This finite differencing method was then implemented as a MATLAB function with an input of the mode number, m, and the special step, h. First, an appropriate time step is determined according to the value h and Q. Once we have determined the displacement after one time step, we could use Eq. 3 with a higher order of accuracy in time and propagate the displacements for the rest of the desired time steps using for loops and if statements. The frequency was measured by finding the time step indices at which the signs of the displacements flipped. The lengths of time between these indices represent a half-cycle of the sine function used to model the mode of the collision. This information can then be converted to a value in Hertz. If statements were used to apply different equations to the appropriate parts

III. Results

A. First and Second Modes

Using our finite differencing methods, we can obtain a frequency of about 51.8 Hz for the first mode and 205.9 Hz for the second mode. The resulting vibrations over a time span of 0.1 seconds are shown in the below figures.

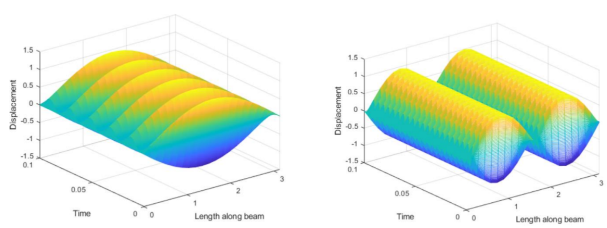

Fig. 1 Vibration over Time for First and Second Mode Beams

B. Forcing Q to be Positive

Manually forcing our scheme to have a positive Q value causes it to be unstable and thus is not useful for this analysis.

C. Second Mode over a Longer Period

Stretching the time span out to 1 second for the second mode yields a frequency measurement of 207.2 Hz with the resulting vibration over time. The figure below shows that the beam does not appear to have internal damping. This is not realistic as the beam would tend to vibrate with gradually less amplitude over time as kinetic energy is lost to stray heat transfer and other energy losses.

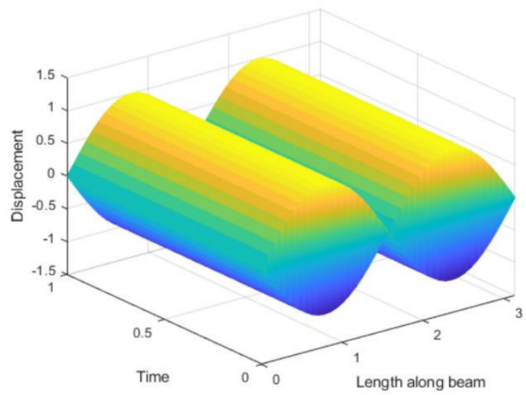

Fig. 2 Vibration over Time of Second Mode Beam over One Second

D. Maximum Allowable Frequency

As shown in Parts A and C, the vibration of the beam for a second mode collision is greater than the maximum allowable frequency of 180 Hz by about 25-30 Hz. For the beam to conform to this frequency, some structural redesign will be needed, or we will need to account for damping in our analysis. The beam's material properties are an important consideration for correcting the frequency of vibration. A material with a higher shear or elastic modulus or perhaps a different mass density could perform better. As our analysis shows us, the beam fails under a second-mode collision. This is an indicator that beam deflection and steady-state stresses are not sufficient to design a beam to succeed in these circumstances.

E. Frequency vs. Mode

Using our scheme, we can plot the frequency vs. mode for the first twenty modes:

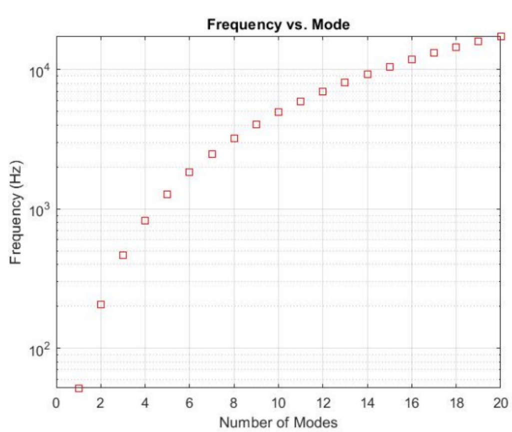

Fig. 3 Frequency vs. Mode

The frequency follows a hyperbolic path as the number of mode increases. As more micro-meteorites impact the beam simultaneously, the total net force will not increase dramatically, as it is always offset by micro-meteorites traveling in opposite directions, however the bending moment stress between these points of contact will increase dramatically as the mode increases, with greater total force and less distance between points of contact. These greater stresses will lead to greater speeds and acceleration leading to faster oscillations. Frequency will begin to increase slower as the mode increases past a certain point because the changes in length and moment are less dramatic.

IV. Conclusion

We can conclude that the structural designer of this spacecraft must make changes to both his analysis techniques and beam structure to meet certain performance metrics, such as the maximum allowable frequency of 180 Hz. Although the finite differencing methods used are a good basis, our model shows now damping as time progresses, which is unrealistic. Accounting for damping in an improved method will yield more accurate frequency values that can be used for a more thoughtful redesign of the beam.

V. Appendix

Here is the MATLAB code used for the function:

```
function [freq] = beamvibe(mode, n_pts, plots)
% Input mode to determine mode of sin function which is initial position,
% break up space type according to n_pts, use this to determine time step
% and use a three-point centered difference to find displacement for a
% length span of 0 to pi, and desired time span. Output cycles per second
% and plots if selected for.

% Desired time span
tspan = [0 1];

% Define steps
Q = -0.2; % constant
C = -9.3979e-6; % constant
h = pi/(n_pts - 1); % spacial step
k = sqrt(Q*C*(h^4)); % time step

% Discretize
% includes columns directly left and right of boundary conditions
M = n_pts;
n = (tspan(2) - tspan(1))/k + 2;
x = linspace(0, pi, M);
```

```matlab
t = linspace(tspan(1), tspan(2), n);
N = round(n);
W = zeros(N,M+2);

% Define boundary conditions
fun = @(x) sin(mode*x);
left = @(t) 0;
right = @(t) 0;
W(1,2:(M + 1)) = fun(x);
W(:,2) = left(t);
W(:,M + 1) = right(t);
W(1,1) = -1*W(1,3);
W(1,(M+2)) = -1*W(1,M);

% Solve at second time step
for j = 1
    for i = 3:M
        W(j+1,i) = 0.5*Q*(W(j,i+2) - 4*W(j,i+1) + 6*W(j,i) - 4*W(j,i-1)...
            + W(j,i-2)) + W(j,i);
    end
    W(j+1,1) = -1*W(j+1,3);
    W(j+1,(M+2)) = -1*W(j+1,M);
end

% Solve for rest of time steps
for j = 2:(N-1)
    for i = 3:M
        W(j+1,i) = Q*(W(j,i+2) - 4*W(j,i+1) + 6*W(j,i) - 4*W(j,i-1)...
            + W(j,i-2)) + 2*W(j,i) - W(j-1,i);
    end
    W(j+1,1) = -1*W(j+1,3);
    W(j+1,(M+2)) = -1*W(j+1,M);
end

% Redescretize without extra columns
w = W(1:length(t),2:(M + 1));

% Measure average period for half a cycle in terms of time steps
count1 = 0; % tracks number of times function has flipped signs
count2 = 0; % tracks number of times half-period has been recorded
for i = 2:(M - 1)
```

```
    if abs(w(1,i)) > 1e-13
        for j = 1:(length(t) - 1)
            if w(j,i)*w(j+1,i) < 0
                count1 = count1 + 1;
                flipTime = j + 1;
                flipTimes(i-1,count1) = flipTime;
            end
        end
        if count1 < 2
        % checks if not enough data recorded in flipTimes
            error('Not enough time has passed to give a frequency measurement');
        else
            count2 = count2 + 1;
            for g = 1:(length(flipTimes(i-1,:)) - 1)
                T_timestep_val = flipTimes(i-1,(g + 1)) - flipTimes(i-1,g);
                T_timestep_vec(i-1,g) = T_timestep_val;
                T_step_val = mean(T_timestep_vec(i-1,:));
            end
        end
    end
    T_step_vec(1,count2) = T_step_val;
    count1 = 0;
end
T_step = mean(T_step_vec);

% Convert time step period to frequency
T = T_step*2*k;
freq = 1/T;

% Plot if desired
tf = strcmp('yes',plots);
if tf > 0
    figure
    for i = 2:length(t)
        mesh(x,t(1:i),w(1:i,:))
        xlim([0 pi])
        ylim([tspan(1) tspan(2)])
        zlim([-1.5 1.5])
        grid on
        xlabel('Length along beam')
        ylabel('Time')
```

```
        zlabel('Displacement')
        drawnow
    end
end

end
```