University of Colorado - Boulder

ASEN 3200 - Orbital Mechanics/Attitude Dynamics and Control

Section 011, Group 8

Lab A3: Spacecraft Pitch Axis Control

Author:
Tomaz Remec[a]

Author:
Hunter Daboll[b]

Author:
Connor O'Reilly[c]

[a]107061117
[b]108976705
[c]107054811

Professors:
Penina Axelrad
Bobby Hodgkinson

10/19/2020

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO BOULDER

The objective of this lab was to learn the methods and processes of controlling spacecraft attitude. This was done by design and analysis of a model that simulates the rotation of a spacecraft about one free angular motion axis. The model uses data to create a control system response simulation, which is then run and compared to real data obtained by lab instructors/TAs. This model was created in MATLAB, and takes in experimental data from the spacecraft mockup Virtual Instrument (VI) used for this lab. The most significant results obtained were the calculated proportional and derivative gain values used in the spacecraft's control system, K_1 and K_2 respectively. These values are as follows. $K_P = 0.0734$, $K_D = 0.0257$. This lab taught the group the value of a solid model, and accurate experimental data in successfully modelling a system as intricate and complex as that of a rotating spacecraft.

Contents

I Preliminary Questions **2**

II Experiment and Analysis **3**

Conclusions & Recommendations **6**

Team Member Participation **7**

References **7**

Acknowledgements **7**

Appendix A: Code **7**

I. Preliminary Questions

1. Given a spacecraft on a spin table, a specific torque available from the spin table, and a way to measure the angular rate of the spacecraft, you can calculate the spacecraft's moment of inertia about the spin axis by solving for I in the following equation (in which T is the applied torque, and α is the angular acceleration of the spacecraft).

$$T = I\alpha \tag{1}$$

The angular acceleration of the spacecraft can be measured by taking the derivative of a line of best fit to the angular rate data obtained by an angular velocity sensor.

2. Below, Figure 1 displays a block diagram for a "single degree of freedom system comprising a spacecraft, a reaction wheel actuator, a rate gyro sensor, a rate integrator, and a PD (proportional + derivative) control law" [1]. Note that this block diagram is not representative of the transfer function experimented with later in the lab, although the two transfer functions are similar.

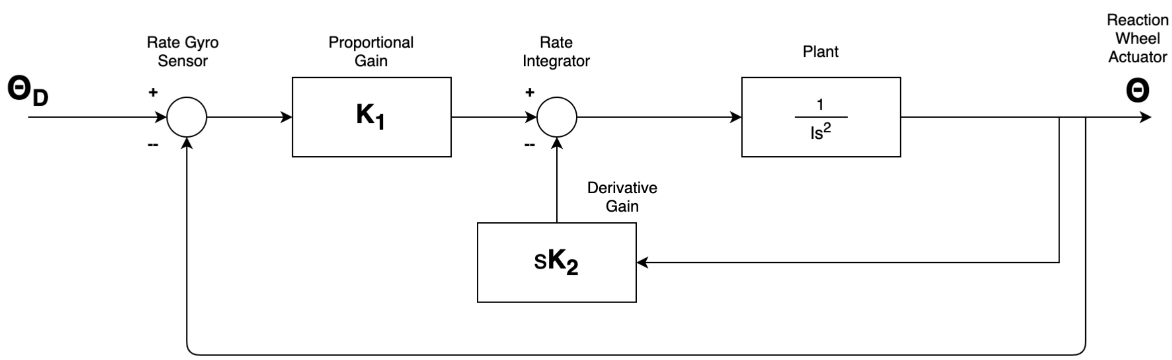

Fig. 1 Block Diagram For Question 2 System

3. Initially the equation for settling time was used to determine the product of the natural frequency $[w_n]$ and the damping ratio $[\zeta]$.

$$\zeta\omega_n = \frac{-ln(p)}{t_s} \tag{2}$$

Where p is the percentage of the unit step response and t_s is the settling time. With a 5% settling time of less than $1.5s$ the product was computed to be $\zeta\omega_n = 1.997155$ $[\frac{1}{s}]$. Following the dampening ratio was computed using the formula to determine the maximum overshoot of a system.

$$M_P = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi} \tag{3}$$

$$\zeta = \sqrt{\frac{\left(\frac{ln(M_p)}{\pi}\right)^2}{1+\left(\frac{ln(M_p)}{pi}\right)^2}} \tag{4}$$

For a maximum over shoot less than 10% $\left[M_p = 0.10\right]$ the dampening ratio was determined to be $\zeta = 0.59115$. Given our transfer function derived from the control block diagram in preliminary question 2 and comparing it to the transfer function for the forced response of a general second order system, values for $K1$ $[Nm/rad]$ and $K2$ $[Nm/rad/s]$ in terms of the moment of inertia about the spin axis were computed.

$$\frac{\theta(s)}{\theta_d(s)} = \frac{K_1/I}{s^2 + s\frac{K_2}{I} + \frac{K_1}{I}} = \frac{K_1/I}{s^2 + s(2\zeta\omega_n) + \omega_n^2} \tag{5}$$

$$K_1 = \omega_n^2 I \tag{6}$$

$$K_2 = 2\zeta\omega_n I \tag{7}$$

II. Experiment and Analysis

Given equations 6 and 7 above, the next step in computing values for K_p and K_d is to estimate I_{SC}, the Moment of Inertia of the spacecraft. However, the measured data only had direct values for the angular velocity, torque, and time: not angular acceleration. Therefore, angular acceleration was estimated for each data set provided in the MOI_Data folder on Canvas using MatLab's built in polyfit function. The moment of inertia could then be easily calculated using Equation 1. A value was calculated for each data file provided, and these MOI estimates were averaged to yield the final value:

$$I_{SC} = \quad 0.0064\frac{kg}{m^2}$$

With this value and equations 6 and 7, it was simple to find the desired values of K_p and K_d. These are shown below.

$$\begin{aligned} K_p &= \quad 0.0734\frac{N\cdot m}{rad} \\ K_d &= \quad 0.0257\frac{N\cdot m}{rad\cdot s} \end{aligned}$$

Unfortunately, the K_d value above was greater than 0.003 $\frac{N\cdot m}{rad\cdot s}$. This was thought to be beyond the acceptable bounds which could be submitted for K_d because it was believed that the bounds set in Lab Task 2 also applied to Lab Task 3 in [1]. In an effort to understand this issue, the team attended office hours multiple times - on 10/12/2020 and 10/15/2020 to try and "fix" the gain values which had been found. By the time it was decided that these values should simply be submitted and we'd just see what happened, it was 6:00 on Thursday the 15th. The team realized it was likely too late to submit a request for these gains to be tested, and emailed Professor Axelrad to confirm that it was indeed too late (which she did). However, Professor Axelrad noted that a few specific data sets had been posted in Canvas. One of those data sets used the following K_p and K_d values:

$$\begin{aligned} K_{p,Posted} &= \quad 0.0711\frac{N\cdot m}{rad} \\ K_{d,Posted} &= \quad 0.0249\frac{N\cdot m}{rad\cdot s} \end{aligned}$$

Below, Figures 2 and 3 show the expected response of the system given the gains which were calculated and the gains which were posted to Canvas. As should be expected from the nearly negligible difference between these two pairs of gains values, the behavior appears nearly identical.

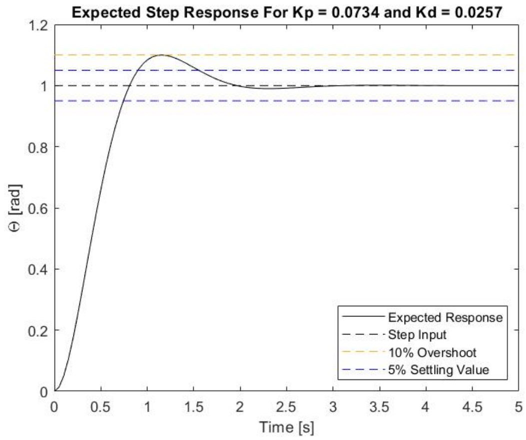

Fig. 2 Expected Behavior for Calculated Gains

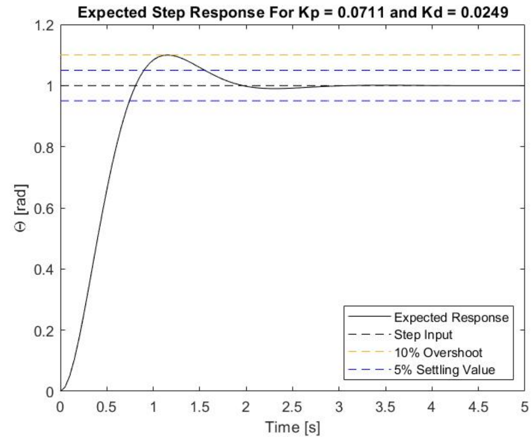

Fig. 3 Expected Behavior for Posted Gains

The true behavior of the posted gains is shown in Figure 4 below for a an alternating step input of 0.5 radians and 0 radians.

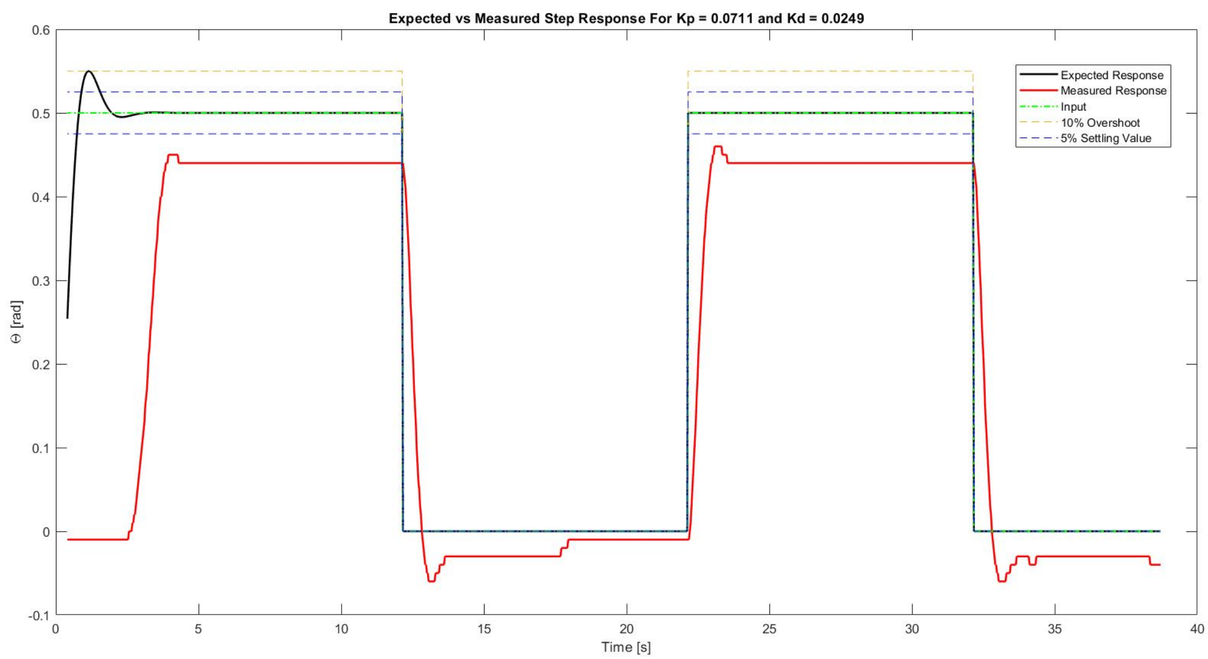

Fig. 4 Expected vs Measured Behavior at Varying Input

As expected from the modeled behavior (the black line in Figure 4, the true behavior (shown in red) increases towards the given value, overshoots its' final settling value, and then settles for each step input. However, there is significant error in the settling value, as the measured position is well outside the 5% settling requirement, never even reaching this threshold. This error is a by-product of the final value behavior of a PD control system, and can be eliminated through the use of a PID control law. However, the system (as it stands) does not meet the specified threshold for settling value.

The settling time also appears incorrect for the first step input. However, for sequential step inputs the system begins to react almost immediately, and settles to its final value quickly.

During these reactions, the reaction wheel spins in the opposite direction of the spacecraft. By the conservation of angular momentum, this causes the spacecraft to spin in the desired direction. When compared to the input current data, this behavior makes sense as the Actual Current data column in these data files is 1) consistently near zero when the spacecraft has reached it's desired angle and 2) is opposite of the angular acceleration of the spacecraft when the step input changes.

When given a disturbance input, i.e. moving the spacecraft mockup with your hand, the onboard reaction wheel tries to command the spacecraft back into its original "neutral" orientation. Say an initial displacement of $+1 rad$ counterclockwise was given to the spacecraft. The reaction wheel would first accelerate in the positive direction (counter-clockwise) in order to torque the spacecraft in the opposite direction (clockwise) causing a rotation towards the starting point. The spacecraft will eventually return to $\Theta = 0$, however it will overshoot its target angular position due to the imprecise nature of the control system being employed. The reaction wheel will then begin to accelerate opposite to its previous rotation to torque the spacecraft back the other way. This process will be repeated several times, with the spacecraft's overshoot of the "neutral" position becoming smaller every iteration. Eventually, in an ideal world, it will settle back in its original orientation, but due to imperfections (friction) in the real world, it will likely end up ever so slightly off due to errors in the measurement of its rotation, or the reaction wheel being shut off prematurely by the user.

To enact a control law where the system is merely instructed to stop moving, and ignore the current pointing direction, one simply has to set $K_p = 0$ and choose any nonzero K_d value, so long as K_d will cause an angular acceleration that is within the operating bounds of the reaction wheel/actuating mechanism employed on board the spacecraft. This behavior is shown below in Figure 5.

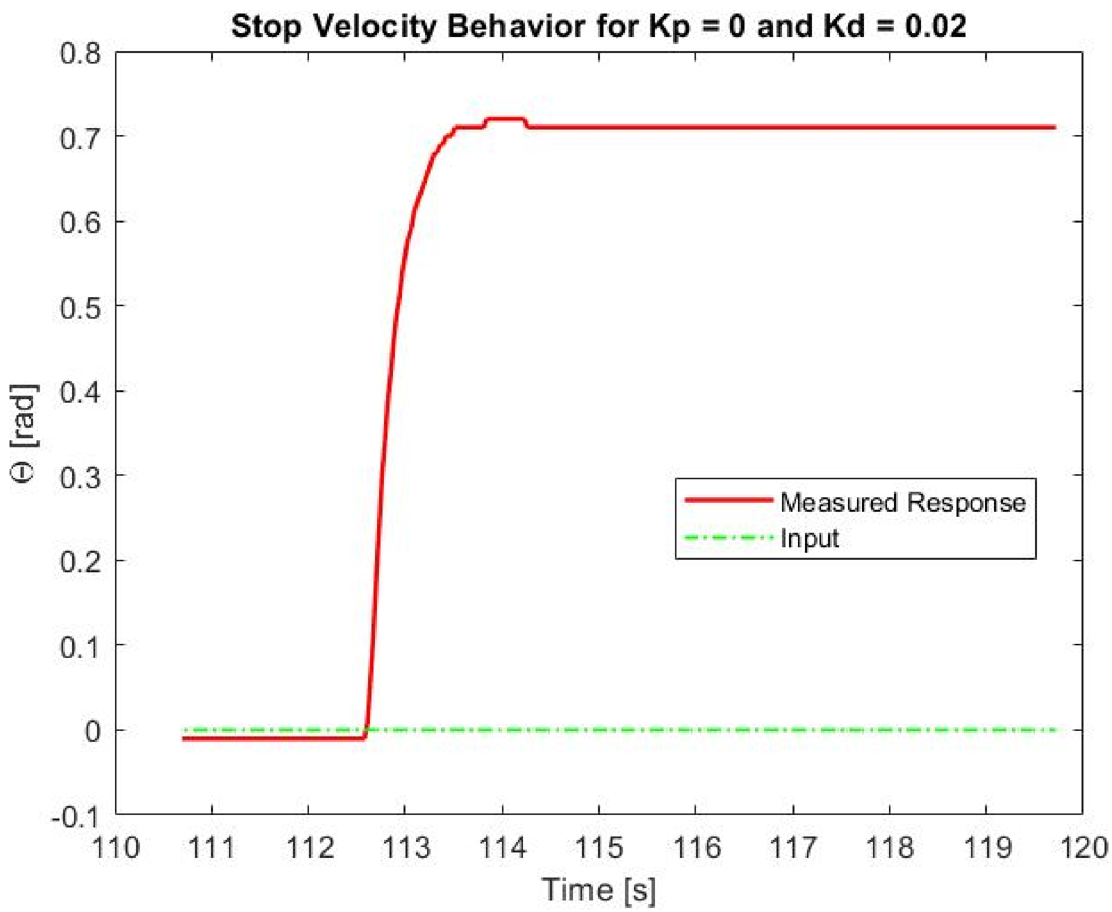

Fig. 5 Stop Velocity Behavior

As is clear in this figure, the system does not consider the difference between the input (desired) angle and the measured angle. It simply brings the change in measured angle (the angular velocity) to zero.

There was the option of adding an integral control gain component to the model of the reaction wheel-spacecraft system. While this was not fully explored by the group, it is still prudent to consider the effect an integral control aspect would have on a system such as this. Integral control takes into account the sign of the error between commanded and actual position values. If this difference is positive, the controller will command an action that will increase the position value, no matter how close to the desired value it is. This can be dangerous, as overshoot is quite easy in an integral control system, but it is well countered by the presence of the derivative control subsystem. An integral control gain, in conjunction with the proportional and derivative control already present, would make the system highly accurate and responsive. This combination of all three control methodologies creates what's known as a PID (Proportional-Integral-Derivative) controller.

Conclusions & Recommendations

This lab demonstrated the significant effect of friction and the consequences of neglecting it in modeling efforts. This one assumption is the root cause of significant error in the final behavior of a PD control system. The lab also demonstrated the power of control block diagrams and the ability to derive and use control laws and the general form control law for a second order system. Furthermore, this work brought to light the importance of clear communication and intentional time management - had similar gain responses not been posted to Canvas, much of the analysis crucial to this lab would have been impossible. In the future, lab members will be sure that any part of lab work which requires coordination with staff or students outside of the lab team is communicated with significant time to spare. This lab could be improved by including the analysis of a PID control law - perhaps even instead of a PD system. In ASEN 2003, the final lab already covered the use of a PD system and significant analysis therein. Because of that former experience, it might prove more valuable to the students to explore in greater depth with a more complicated (and significantly more accurate) system, better preparing them for any real-world scenarios where the consistent error inherent to a PD control law is at best unreliable, and at worst, unacceptable.

Team Member Participation

2 - Lead, 1 - Participated/Contributed, 0 - Did not Contribute

Name	Plan	Model	Experiment	Results	Code	Report
Tomaz Remec	1	2	1	2	0	1
Hunter Daboll	1	1	2	1	2	1
Connor O'Reilly	1	1	1	1	0	1

Group members have reviewed and approved this table.
Initials: WHD CTO TJR

References

[1] Axelrad, P., "ASEN 3200 Lab A-3: Spacecraft Pitch Axis Control" Sept. 15, 2020.

[2] Curtis, H., *Orbital Mechanics for Engineering Students*, 3rd ed., Elsevier, 2014.

Acknowledgements

We would like to thank Professor Axelrad, Professor Hodgkinson, and all of the TAs, TFs, and LAs of the ASEN 3200 instructional team for their support and assistance with the completion of this lab.

Appendix A: Code

```
%% Housekeeping
%{
Authors: Hunter Daboll
Last edited: 10/12/2020
%}
clear all; close all; clc;

%% Read in data
baseName = "MOI_Data/Unit7_Base_";
I_sc = zeros(1,5);
for i = 5:10
    filename = strcat(baseName,int2str(i));

    currentData = readtable(filename);
    time = table2array(currentData(100:end-100,1))*0.001; %ms to s
    rWheelTorque = mean(table2array(currentData(100:end-100,2))) * 0.001; %mNm to N*m
    angVel_SC = -1.*table2array(currentData(100:end-100,3))* 2*pi/60 ;
    % Convert RPM to rad/s, switch to positive spin direction

    %figure();
    %scatter(time, angVel_SC);
    %hold on;
    %xlabel("Time, [s]"); ylabel("Ang. Velocity, [rad/s]");
    P =  polyfit(time, angVel_SC, 1);
    %plot(time, P(2) + time.*P(1), 'LineWidth',2.5);
    angAccel = P(1); % rad/s^2

    I_sc(i-4) = rWheelTorque / angAccel; % kg*m^2
end

I_sc_avg = mean(I_sc)

%% Calculate proportional and dertivative gains
zeta = 0.591155;
wn = 3.37839;
```

```
Kp = I_sc_avg* (wn^2)

Kd = 2* zeta * wn * I_sc_avg

%% Plot expected step response to selected gains
t = linspace(0,5);
wd = wn*sqrt(1-zeta^2);
e = 2.71828;
x = 1 - (e.^(-zeta*wn.*t)).*(cos(wd.*t)+ (zeta*sin(wd.*t))/(sqrt(1-zeta^2)));

figure(); gold = [0.9290, 0.6940, 0.1250];
plot(t,x,'-k'); hold on; plot(t,ones(1,length(t)),'--k'); plot(t,1.1*ones(1,length(t)),'--','Color',gold)
plot(t,1.05*ones(1,length(t)),'--b'); plot(t,0.95*ones(1,length(t)),'--b','HandleVisibility','off');
xlabel("Time [s]"); ylabel("\Theta [rad]");
titleVar = "Expected Step Response For Kp = " + num2str(Kp,3) + " and Kd = " + num2str(Kd,3);
title(titleVar); legend("Expected Response","Step Input","10% Overshoot",
"5% Settling Value",'Location','southeast');


%% Plot expected step response to posted gains
t = linspace(0,5);
wd = wn*sqrt(1-zeta^2);
e = 2.71828;
Kp = 0.0711; Kd = 0.0249;
x = 1 - (e.^(-zeta*wn.*t)).*(cos(wd.*t)+ (zeta*sin(wd.*t))/(sqrt(1-zeta^2)));

figure();
plot(t,x,'-k'); hold on; plot(t,ones(1,length(t)),'--k'); plot(t,1.1*ones(1,length(t)),'--','Color',gold)
plot(t,1.05*ones(1,length(t)),'--b'); plot(t,0.95*ones(1,length(t)),'--b','HandleVisibility','off');
xlabel("Time [s]"); ylabel("\Theta [rad]");
titleVar = "Expected Step Response For Kp = " + num2str(Kp,3) + " and Kd = " + num2str(Kd,3);
title(titleVar); legend("Expected Response","Step Input","10% Overshoot",
"5% Settling Value",'Location','southeast');

%% Plot true response to posted gains on the same figure
currentData = readtable("spin_module_kp_71_kd_25");
time = (table2array(currentData(2:end,1))-1015000)*0.001; %ms to s
refPos = table2array(currentData(2:end,2)); %rad
angPos = table2array(currentData(2:end,3)); %rad

x = refPos.*(1 - (e.^(-zeta*wn.*time)).*(cos(wd.*time)+ (zeta*sin(wd.*time))/(sqrt(1-zeta^2))));
figure();
plot(time,x,'-k','LineWidth',1.5); hold on;
plot(time,angPos,'-r','LineWidth',1.5); hold on;

plot(time,refPos,'-.g','LineWidth',1.15); plot(time,0.1*refPos+refPos,'--','Color',gold)
plot(time,0.05*refPos+refPos,'--b'); plot(time,0.95*refPos,'--b','HandleVisibility','off');
xlabel("Time [s]"); ylabel("\Theta [rad]");
titleVar = "Expected vs Measured Step Response For Kp = " + num2str(Kp,3) + " and Kd = " + num2str(Kd,3);
title(titleVar); legend("Expected Response","Measured Response","Input","10% Overshoot",
"5% Settling Value",'Location','southeast');

%% Read in data and generate plot for stop velocity file
currentData = readtable("Stop_velocity");
time = (table2array(currentData(2:end,1))-1015000)*0.001; %ms to s
refPos = table2array(currentData(2:end,2)); %rad
angPos = table2array(currentData(2:end,3)); %rad

Kp = 0; Kd = 0.02;
figure();
plot(time,angPos,'-r','LineWidth',1.5); hold on; plot(time,refPos, '-.g',"LineWidth",1.15);
xlabel("Time [s]"); ylabel("\Theta [rad]");
titleVar = "Stop Velocity Behavior for Kp = " + num2str(Kp,3) + " and Kd = " + num2str(Kd,3);
title(titleVar); legend("Measured Response","Input",'Location','southeast');
```