More tips for Control Mode 0

This is most of the code you’ll need. You obviously need to fill in numbers and do some more stuff, but this loads in the values you’ll need for your Simulink simulation. More will need to be added later when you add the deadband. That part is commented out at the bottom.

% AstroEngr 445

% Dr Rob Brown; Fall 2011

% Final Project Control Mode 0

% clear all variables and the workspace

clear;

clc;

% Define Orbit(assume circular)

%I had some intermediate steps in here to find w\_c

w\_c = ; % Angular velocity of orbit (rad/s)

% Final Project - spacecraft pitch response to torque in stowed config

I1 = ; I2 = ; I3 = ;

Is = ;

% define simulation parameters

stepsize = 0.01; % Small size needed to see limit cycle

% set simulation time in seconds (You can pick your time, but 90 min seems

% like a good place to start because that's the time limit you have. If it % doesn’t converge in 90 minutes, you need to pick a different Kd or Kd.)

tfinal = 90\*60; % seconds

% define the time vector

t = [0:stepsize:tfinal]'; % This makes it a column matrix of all times

% Initial conditions

thd\_0 = 4\*0.1\*(pi/180); % in rad/sec; This is slightly more than 4 times the tumble rate

th\_0 = 0;

% thruster output

% The thrust given on the back page of the handout is

% for both thrusters in the pair. You do need to account for where they're

% mounted from the center of mass to get torque, but don't mutliply by two.

tmax = ; % max/min thruster torque (N-m).

% thruster time delay

td = .1; % time delay

delay = exp(-td\*s);

% define PD control gains. Pick one and guess the other, then adjust as we

% discussed in class.

K\_p = 1;

K\_d = ;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % Code relating to adding the deadband (we'll cover this later, but some

% % of it should look familiar). Most of this is to get the gain margin,

% % which you'll see relates to the deadband.

%

% % define the Laplace variable

% s = tf('s');

% % Define Linear plant for analysis purposes

% G = (1/(I2-Is))/(s^2 + 3\*w\_c^2\*(I1-I3)/(I2-Is));

% % Controler

% K = K\_p + K\_d\*s;

% % % combine G and K

% GK = minreal(G\*K);

% % predict the limit cycle response

% [gm,ph,wgm,wpm]=margin(GK\*delay);

% % calculate min deadzone size to eliminate limit cycles

% %We'll cover this more in class

beep

After you run this file, with your numbers, then you should be able to run Simulink and get answers. Remember, you can call your Simulink file directly from matlab. Here’s my Simulink without the deadband. This is almost identical to the handout from the class notes. You can add the deadband in later.



Then plot th2 and th2d ( plot(th2\*180/pi,th2d\*180/pi)) to see if it works.

There are other plots you need. That’s why I’ve added other outputs. They also help me diagnose cadet errors. However, this one plot of theta2 and theat2dot should let you know if you’ve got a controller that works.