Controls System Lab 6 Colin Roskos

Summary:

The objective of lab was to learn commands in MATLAB to reduce the system block diagram using series, parallel and feedback configuration.

Conclusion:

**Exercise 1:**

In exercise one, we can see that all of the poles are less than zero in its real component, this means that the given system is stable.

Poles at :

-10.1174 + 0.0000i

-2.4403 + 0.0000i

-2.3493 + 0.0000i

-0.5882 + 0.8228i

-0.5882 - 0.8228i

-1.0000 + 0.0000i

Zeros at :

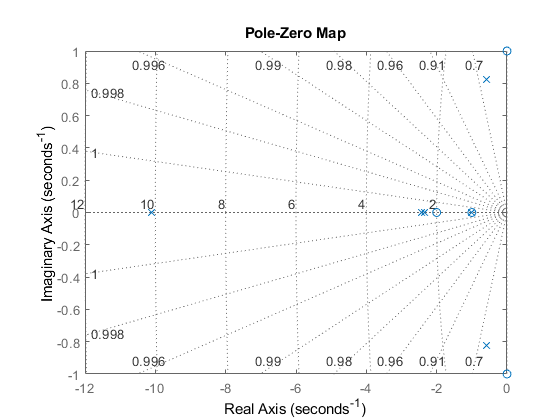
-2.0000 + 0.0000i

-0.0000 + 1.0000i

-0.0000 - 1.0000i

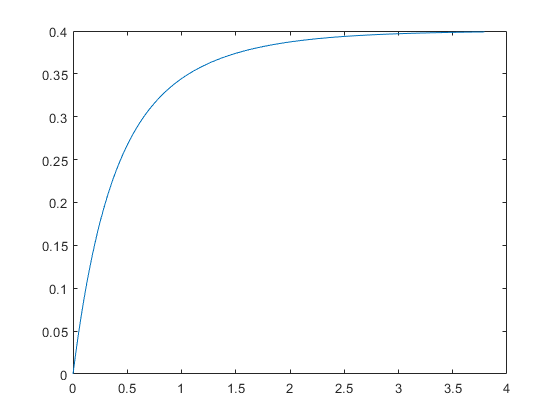
-1.0000 + 0.0000i

-1.0000 - 0.0000i



**Exercise 2:**

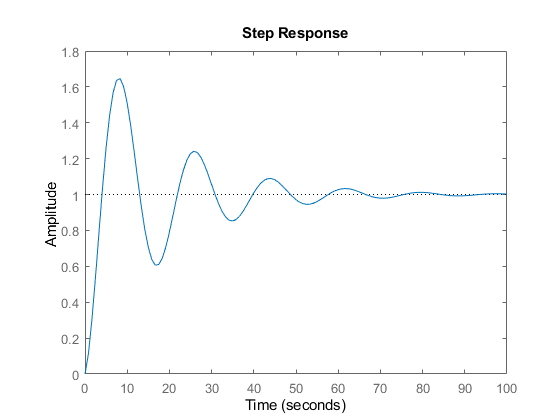
The expected result for the series then feedback combination created the system we were expecting, which did in fact result in convergence to 2/5.



Exercise 3:

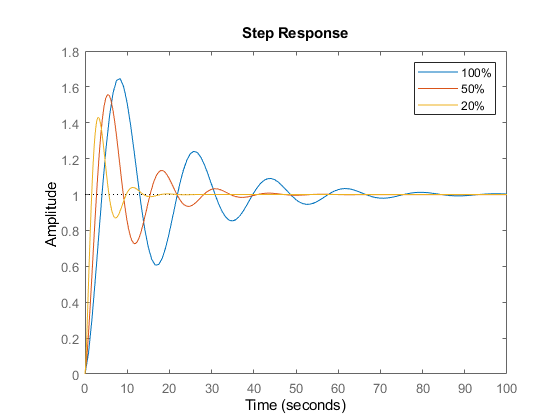
The TF for the spacecraft is :

With the step function \* 10^0 or 1(t):



With the varying spacecraft inertia:

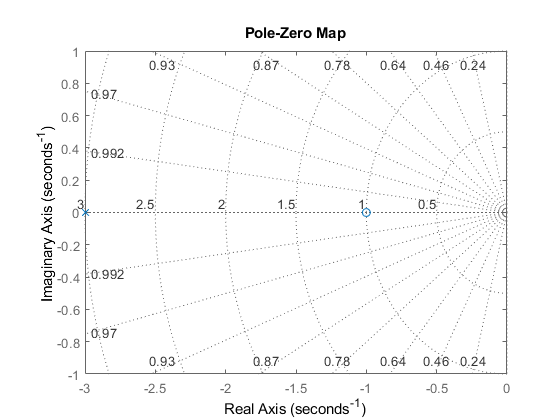
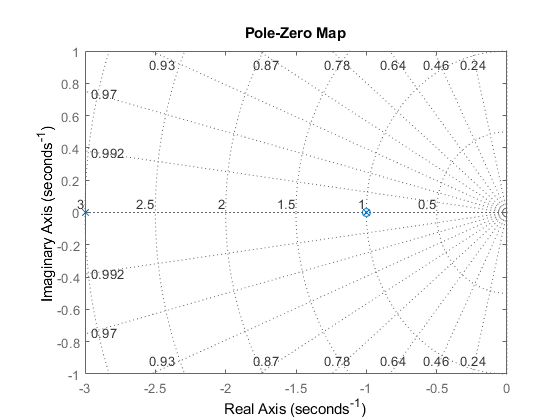
As the spacecraft inertia becomes reduced, there is a shorter response time, and less overshoot as seen in the graph below.



Exercise 4:

Before reduction we have poles at -3, -1 and a double zero at -1, so there is a cancellation. After cancelling we have a lone pole at -3. The importance of removing these poles is that they do cancel each other entirely when it comes to the system response. This reduces complexity of the problem.

The graphs of the poles and zeros before and after reduction can be seen below.



Code and Results:

%% Exercise 1

% For the following multi-loop feedback system, get closed loop transfer function

% and the corresponding pole-zero map of the system.

G1 = tf([1], [1 10]);

G2 = tf([1], [1 1]);

G3 = tf([1 0 1], [1 4 4]);

G4 = tf([1 1], [1 6]);

H1 = tf([1 1], [1 2]);

H2 = tf([2], [1]);

H3 = tf([1], [1]);

A = series(H2, 1/G4);

B = series(H1, 1/G2);

C = parallel(-A, B);

D = series(G2, G3);

E = series(Db, G4);

G = feedback(E, C, +1);

H = series(Gb, G1);

sys = feedback(H, H3)

%

% sys =

%

% s^5 + 4 s^4 + 6 s^3 + 6 s^2 + 5 s + 2

% ----------------------------------------------------------------

% 12 s^6 + 205 s^5 + 1066 s^4 + 2517 s^3 + 3128 s^2 + 2196 s + 712

%

p = pole(sys)

z = zero(sys)

%

% p =

%

% -10.1174 + 0.0000i

% -2.4403 + 0.0000i

% -2.3493 + 0.0000i

% -0.5882 + 0.8228i

% -0.5882 - 0.8228i

% -1.0000 + 0.0000i

%

%

% z =

%

% -2.0000 + 0.0000i

% -0.0000 + 1.0000i

% -0.0000 - 1.0000i

% -1.0000 + 0.0000i

% -1.0000 - 0.0000i

%

pzmap(sys); grid on;

Exercise 2:

%% Exercise 2

%

controller = tf([1], [1 1]);

plant = tf([1 2], [1 3]);

A = series(controller, plant);

sys = feedback(A, [1])

[y, t] = step(sys);

plot(t, y);

y(end) % 0.3989 -> .4 == 2/5

%

% sys =

%

% s + 2

% -------------

% s^2 + 5 s + 5

%

Exercise 3:

%% Exercise 3

% Satelight Altitude control system

% control parameters

k = 10.8\*10^8;

a = 1;

b = 8;

% spacecraft moment of inertia

J = 10.8\*10^8;

% System Description

controller = tf([k a\*k], [1 b]);

spacecraft = tf([1], [J 0 0]);

A = series(controller, spacecraft);

sys = feedback(A, [1])

%

% sys =

%

% 1.08e09 s + 1.08e09

% -----------------------------------------------

% 1.08e09 s^3 + 8.64e09 s^2 + 1.08e09 s + 1.08e09

%

p = pole(sys)

z = zero(sys)

%

% p =

%

% -7.8893 + 0.0000i

% -0.0553 + 0.3517i

% -0.0553 - 0.3517i

%

%

% z =

%

% -1

%

step(sys)

iner = [1 .5 .2];

printouts\_s = ["100%" "50%" "20%"];

% System Description

for i=1:3

controller = tf([k a\*k], [1 b]);

spacecraft = tf([1], [J\*iner(i) 0 0]);

A = series(controller, spacecraft);

sys = feedback(A, [1])

step(sys); hold on;

end

legend(printouts\_s);

hold off

Exercise 4:

%% Exercise 4

% feedback gain

G = tf([1 1], [1 2]);

H = tf([1], [1 1]);

sys = feedback(G, H)

pzmap(sys); grid on;

p = pole(sys)

z = zero(sys)

%

% sys =

%

% s^2 + 2 s + 1

% -------------

% s^2 + 4 s + 3

%

% p =

%

% -3

% -1

%

%

% z =

%

% -1

% -1

%

sys = minreal(sys)

pzmap(sys); grid on;

%

% sys =

%

% s + 1

% -----

% s + 3

%