

**NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY**

**Linear Control Systems (EE-371)**

**Assignment # 2**

(CLO-2)

**Submission Details**

|  |  |
| --- | --- |
| Submitted to: | Ms. Neelma Naz |
| Submitted by: | Muhammad Umer |
| Class: | BEE-12C |
| Semester: | 6th |
| Dated: | 9/3/2023 |
| CMS ID: | 345834 |

**In this case study, we investigate the time response of the vehicle dynamics that relate the pitch angle output to the elevator deflection input.**

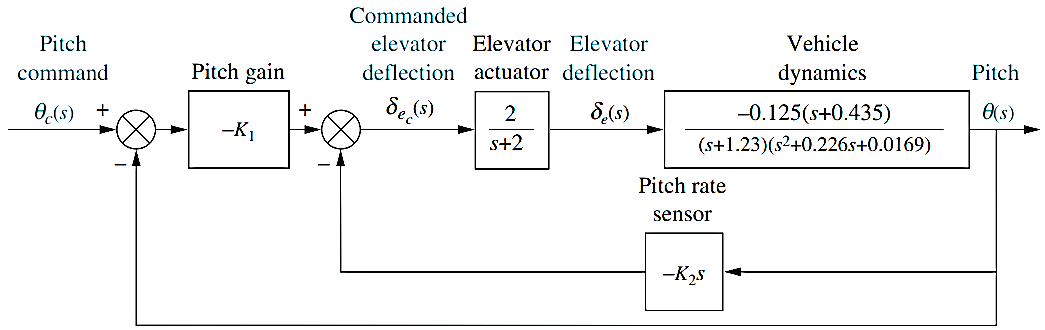
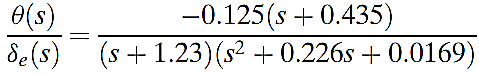


Figure : Pitch control loop for the UFSS vehicle

**The transfer function relating pitch angle, *θ(s),* to elevator surface angle, *δe(s),* for the UFSS vehicle is:**



**Requirements:**

1. Using only the second-order poles shown in the transfer function, predict percent overshoot, rise time, peak time, and settling time? [do it by hand]

**In MATLAB: [submit proper MATLAB codes and response plots]**

1. Using Laplace transforms, find the analytical expression for the response of the pitch angle to a step input in elevator surface deflection. [Use MATLAB to find the time domain expression by taking inverse Laplace transform].
2. Plot the step response of the system having just two poles [Let’s say there is pole zero cancellation. Maintain the steady state gain of system].
3. Plot the step response of the system having just two poles and a zero [Let’s say additional pole is cancelled by 5 times rule of thumb but maintain the steady state gain of your system]
4. Plot the step response of the complete system [having 3 poles and a zero].
5. Compare the responses in just 2 to 3 lines.

# Rise Time

%% Rise Time

syms wn\_t zeta;

zeta\_fit = linspace(0.1, 0.9, 5000);

c\_t = 1 - (1/sqrt(1-zeta^2))\*exp(-zeta\*wn\_t)\*cos(wn\_t\*sqrt(1-zeta^2)- ...

atan(zeta/sqrt(1-zeta^2)));

c\_fit = subs(c\_t, zeta, zeta\_fit);

norm\_time = zeros(1, 5000);

for i = 1:5000

t1 = vpasolve(c\_fit(i) == 0.9, wn\_t, [0 5]);

t2 = vpasolve(c\_fit(i) == 0.1, wn\_t, [0 5]);

norm\_time(i) = t1 - t2;

end

plot(zeta\_fit, norm\_time)

grid

title('Normalized time vs. Damping ratio')

xlabel('Damping ratio \zeta')

ylabel('Normalized time \omega\_{n}t')

% polyfit returns coefficients of nth order polynomial

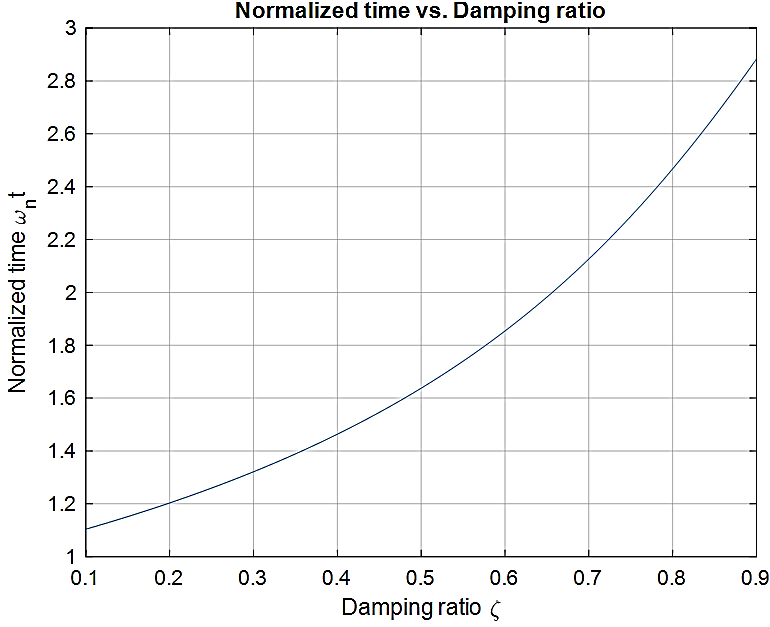
P = polyfit(zeta\_fit, norm\_time, 4);

poly2sym(P, zeta)

**Output**

P =

0.5089 0.7837 0.2144 0.8848 1.0114



Resultant polynomial can be written in terms of zeta as,

, which is used to find the rise time by substituting .

# Definitions

z1 = -0.435; p1 = [-1.23 -0.113-1j\*0.0642 -0.113+1j\*0.0642]; k1 = -0.125;

z2 = -0.435; p2 = [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; k2 = -0.1016;

z3 = []; p3 = [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; k3 = -0.0442;

G1 = zpk(z1, p1, k1); % Represents original tf

G2 = zpk(z2, p2, k2); % Represents G1 + 5 times rule reduction

G3 = zpk(z3, p3, k3); % Represents G1 + pole-zero cancellation reduction

G1 =

-0.125 (s+0.435)

---------------------------------

(s+1.23) (s^2 + 0.226s + 0.01689)

Continuous-time zero/pole/gain model.

G2 =

-0.1016 (s+0.435)

------------------------

(s^2 + 0.226s + 0.01689)

Continuous-time zero/pole/gain model.

G3 =

-0.0442

------------------------

(s^2 + 0.226s + 0.01689)

Continuous-time zero/pole/gain model.

# Time Domain Expression

syms s;

G1\_s = (-0.125\*(s+0.435))/((s+1.23)\*(s^2 + 0.226\*s + 0.0169))

g1\_t = ilaplace(1/s\*G1\_s)

display(g1\_t)

**Output**

g1\_t =

(28352500\*exp(-(113\*t)/1000)\*(cos((9\*51^(1/2)\*t)/1000) + (1891291\*51^(1/2)\*sin((9\*51^(1/2)\*t)/1000))/10411038))/10577879 - (165625\*exp(-(123\*t)/100))/2566231 - 18125/6929

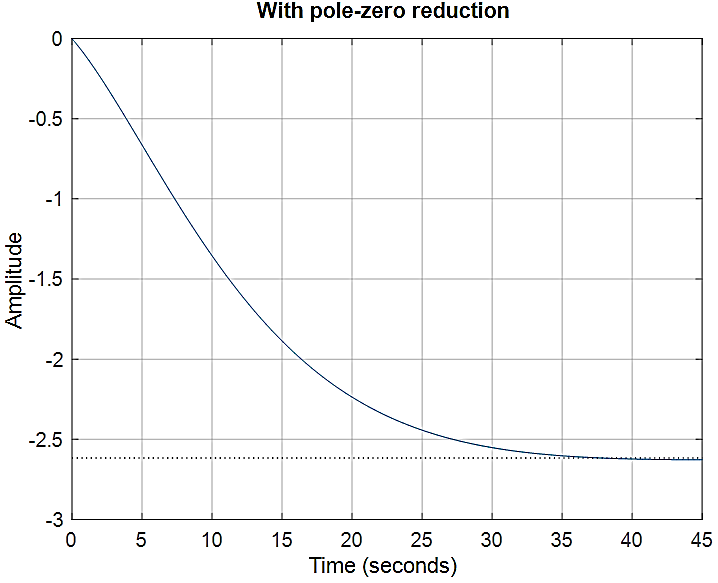
# Step Response (With Pole-Zero Reduction)

figure

step(G2)

grid

title('With pole-zero reduction')



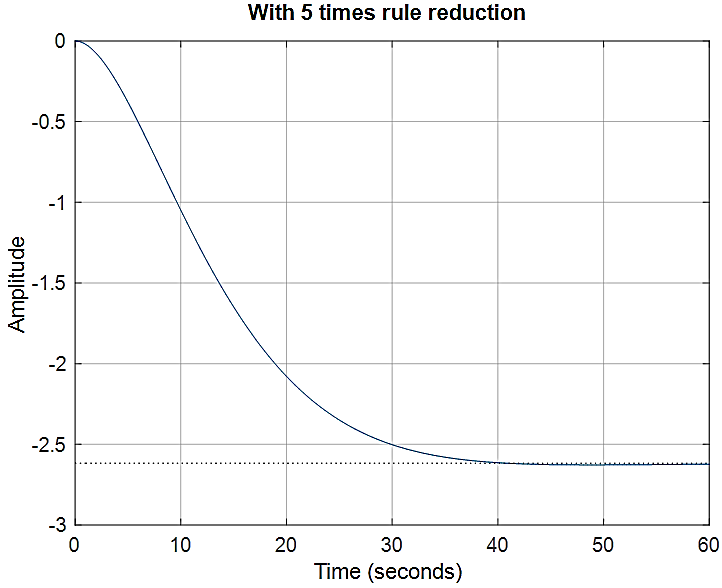
# Step Response (With 5-Times Rule Reduction)

figure

step(G3)

grid

title('With 5 times rule reduction')



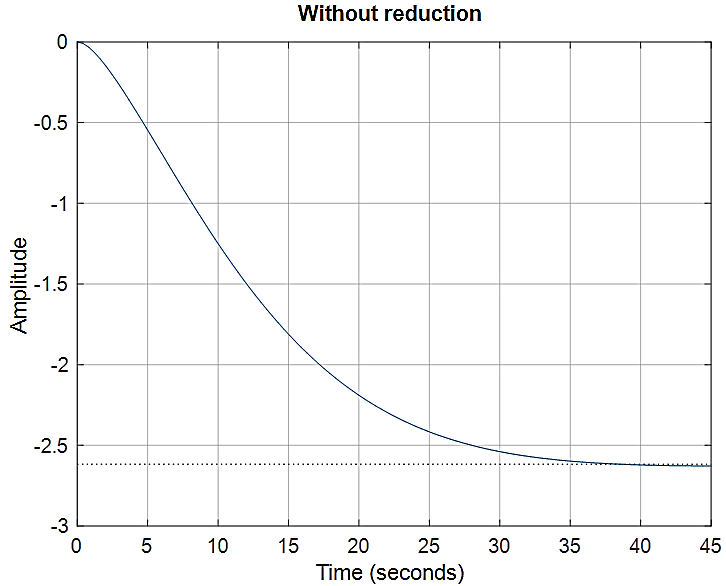
# Step Response (Without Reduction)

figure

step(G1)

grid

title('Without reduction')



# Comparison

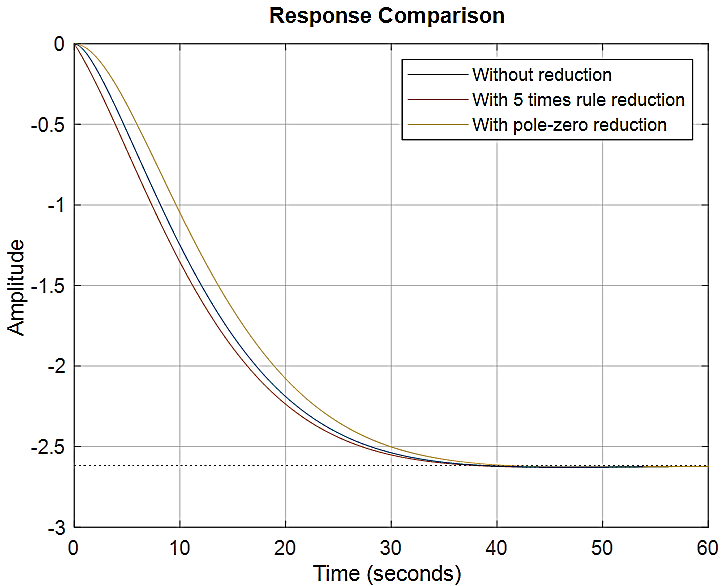
hold on

step(G1); step(G2); step(G3);

legend('Without reduction', 'With pole-zero reduction', 'With 5 times rule reduction')

grid on

title('Response Comparison')



**Comments:** The transfer function reduced using 5-times rule has the fastest response as it can be inferred as a second order system with an additional zero, and zeros decrease the response time albeit increasing the overshoot. The complete system also has a faster response than the system reduced to a simple second order system using pole-zero cancellation as the additional zero at -0.435, which is closer to the imaginary axis than the pole at -1.23, has a greater impact on the transient response and thus the system’s overall speed.