AAE364: Controls System Analysis

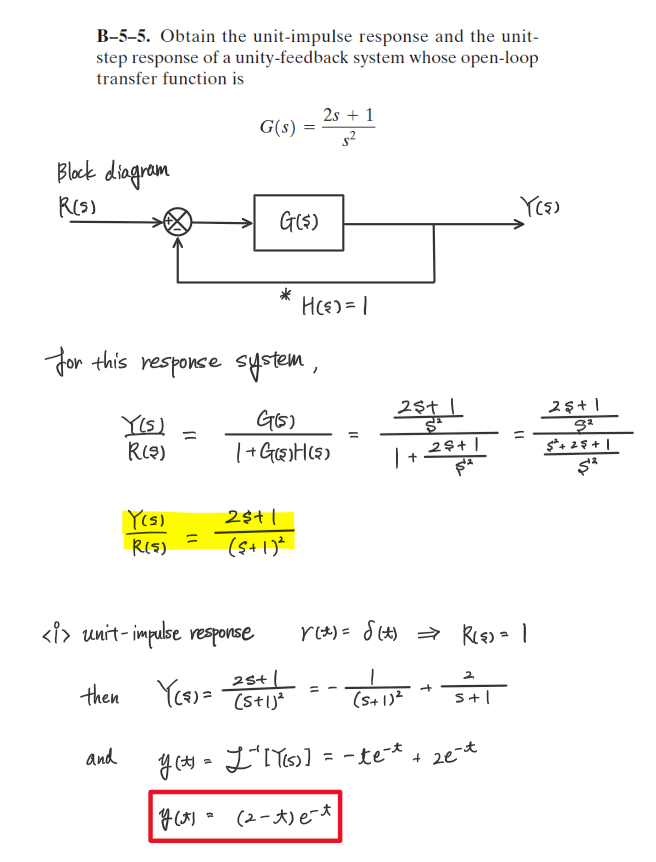
HW5: Stability and Error Analysis

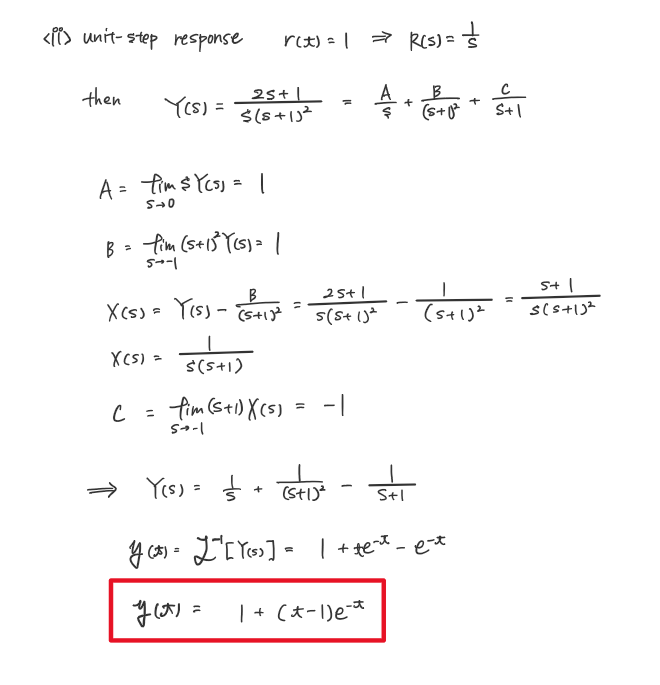
Dr. Sun

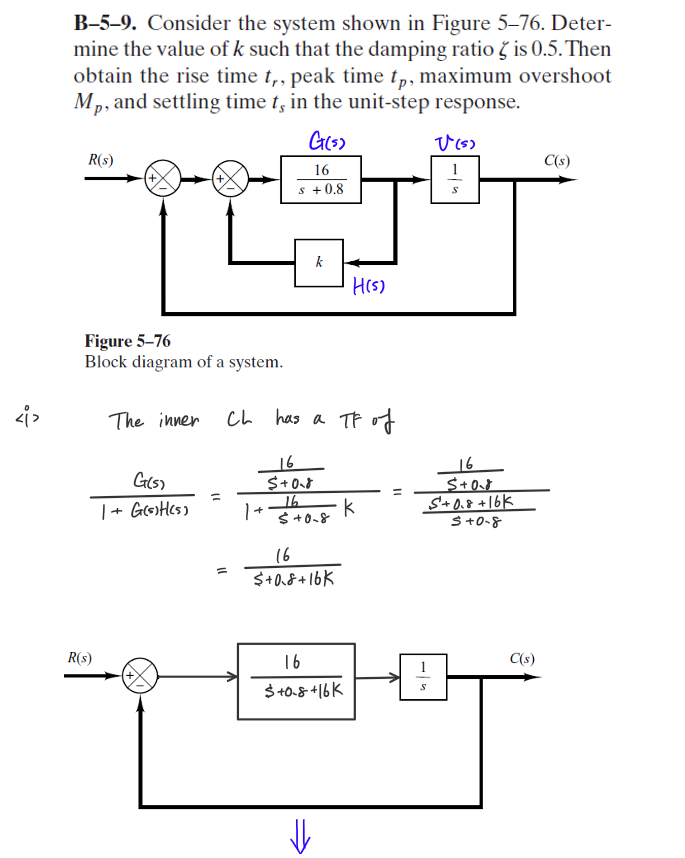
Tomoki Koike

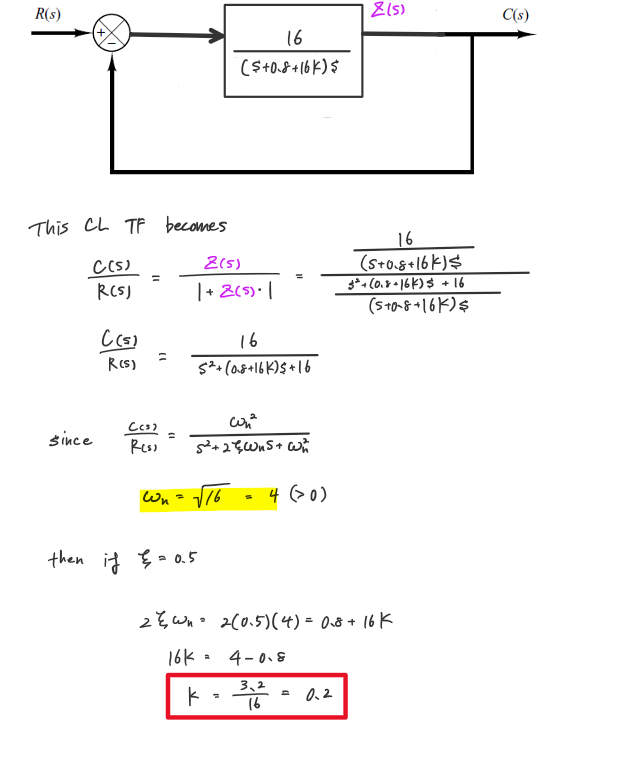
Friday, February 21, 2020

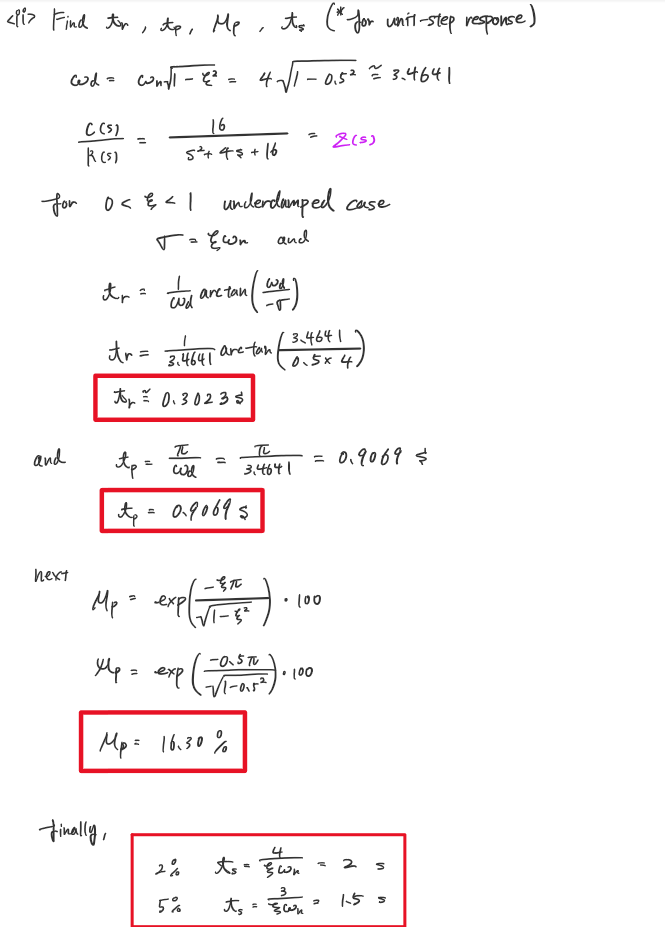


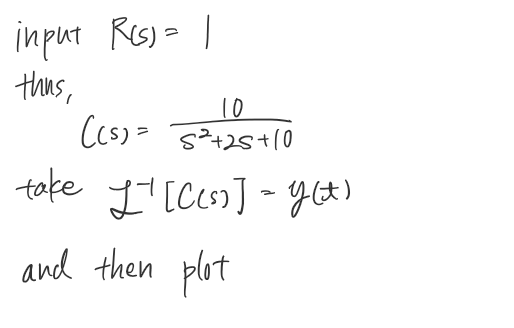
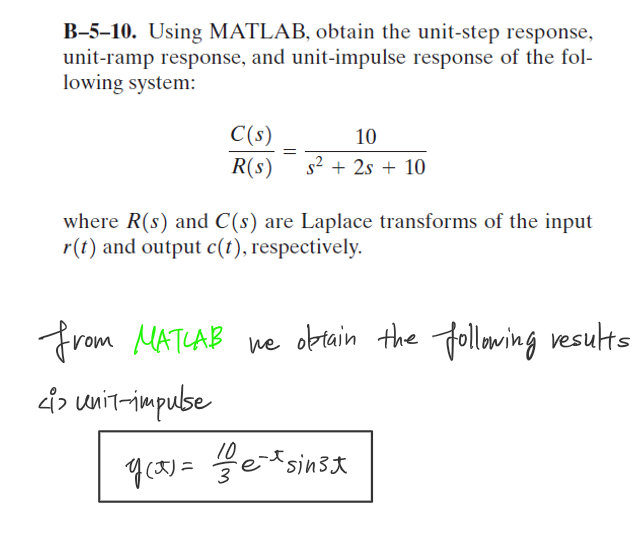






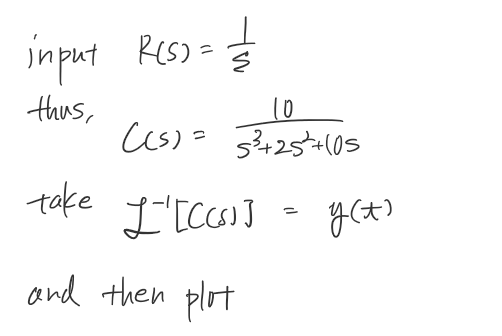
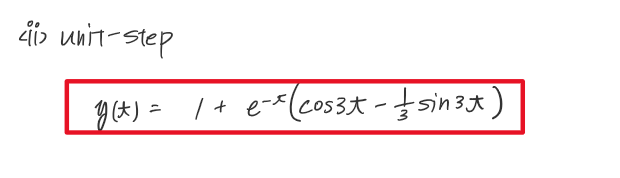






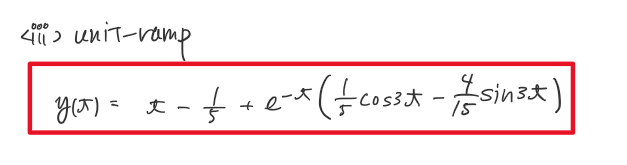
A close up of a map

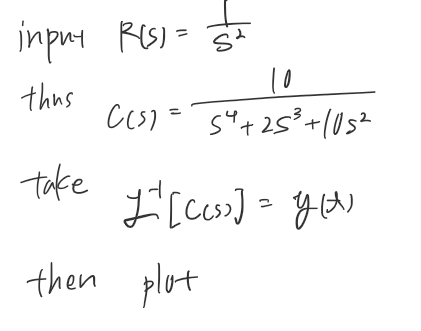
Description automatically generated



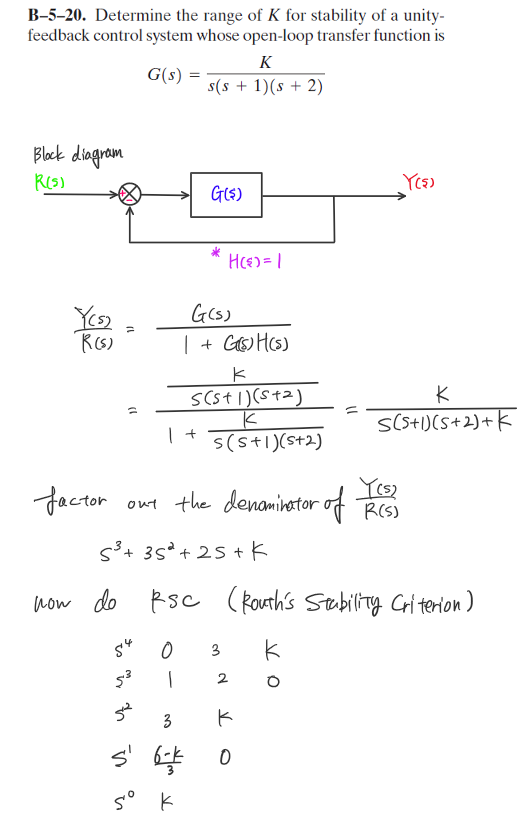
A close up of a map

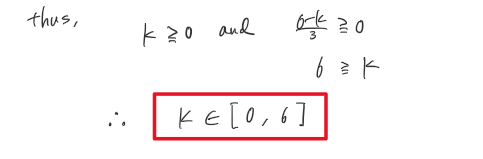
Description automatically generated

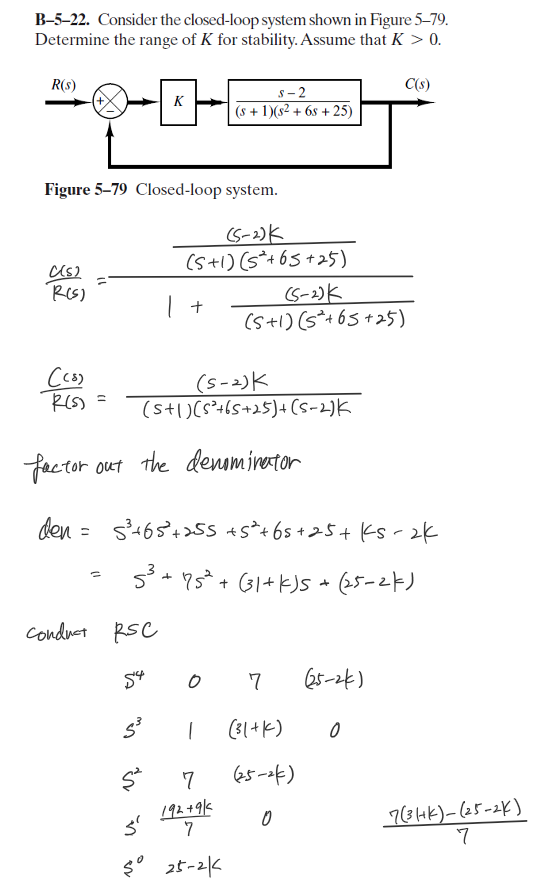


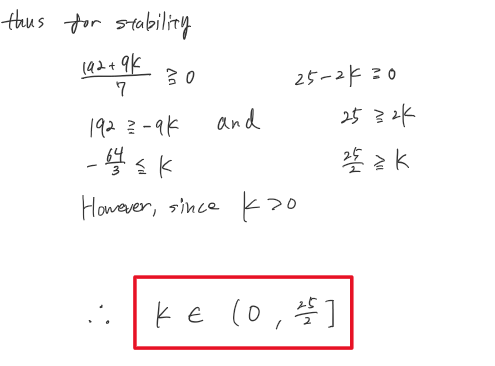
A close up of a map

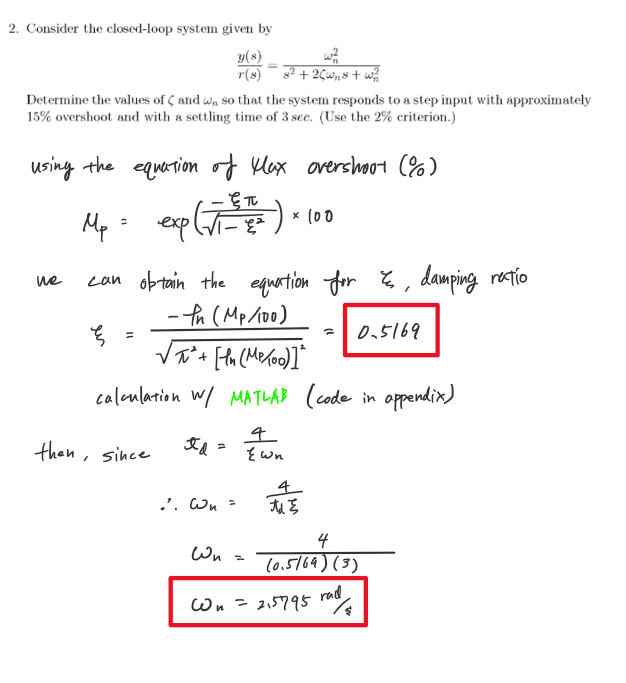
Description automatically generated

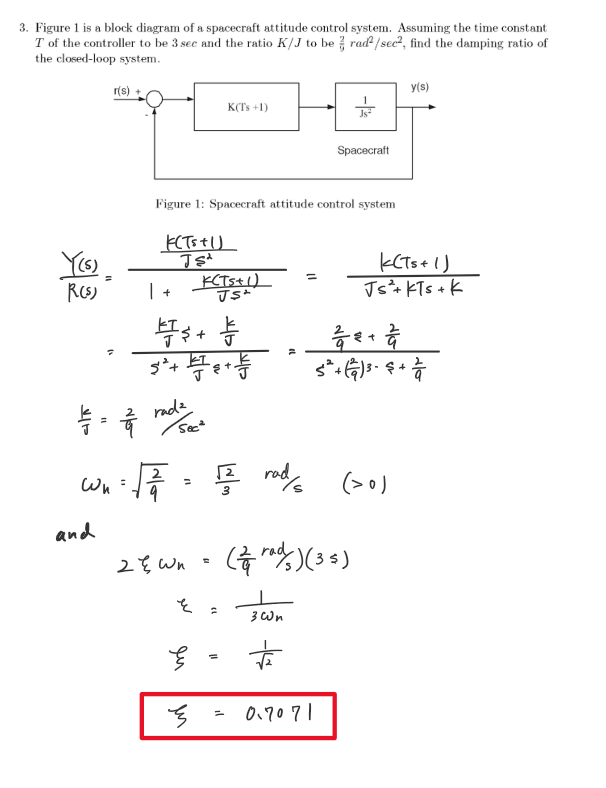












APPENDIX

**AAE364 HW5 MATLAB CODE**

**problem 1 >> B-5-10**

s = tf('s');

G = 10/(s^2+2\*s+10);

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE364\matlab\matlab\_output';

% Impulse reponse

impF = 1;

[num\_imp, den\_imp] = tfdata(G\*impF);

output\_impulse = return\_inverseLaplace\_expression(num\_imp, den\_imp)

fig1 = figure("Renderer","painters");

impulse(G);

saveas(fig1, fullfile(fdir,'HW5\_b\_5\_10\_impulse.png'));

% Step Response

stepF = 1/s;

[num\_step, den\_step] = tfdata(G\*stepF);

output\_step = return\_inverseLaplace\_expression(num\_step, den\_step)

fig2 = figure("Renderer","painters");

step(G);

saveas(fig2, fullfile(fdir,'HW5\_b\_5\_10\_step.png'));

% Ramp Response

rampF = 1/s^2;

[num\_ramp, den\_ramp] = tfdata(G\*rampF);

output\_ramp = return\_inverseLaplace\_expression(num\_ramp, den\_ramp)

fig3 = figure("Renderer","painters");

step(G / s);

ylim([0, 10])

xlim([0, 10])

hold on

plot(linspace(0,10,20), linspace(0,10,20), '-r')

hold off

title('Ramp Response')

saveas(fig3, fullfile(fdir,'HW5\_b\_5\_10\_ramp.png'));

**problem 2**

MOS = 15; % percent

zeta = calc\_zetaFromMOS\_or\_MOSFromzeta(MOS, "zeta")

function inverted\_expr = return\_inverseLaplace\_expression(num, den)

%{

inputs: 1) num: numerator of the transfer function times input

function G(s)\*R(s)

2) den: denominator of the transfer function times input

function G(s)\*R(s)

outputs: 1) inverted\_expr: returns the expression for the inverse

laplace equation of the output laplace equation

%}

syms s t % Invoke Symbolic Math Toolbox

snum = poly2sym(num, s); % Symbolic Numerator Polynomial

sden = poly2sym(den, s); % Symbolic Denominator Polynomial

G\_time\_domain = ilaplace(snum/sden); % Inverse Laplace Transform

G\_time\_domain = simplify(G\_time\_domain, 'Steps',10); % Simplify To Get Nice Result

inverted\_expr = collect(G\_time\_domain, exp(-t)); % Optional Further Factorization

end

function output = calc\_zetaFromMOS\_or\_MOSFromzeta(MOS\_or\_zeta, type)

%{

inputs: 1) MOS\_or\_zeta: maximum overshoot or zeta (damping ratio) input

the one of the two will be chosen depending on the second

input "type"

2) type: string "MOS" or "zeta" indicates what output the

user requires

outputs: 1) output: returns either the MOS or zeta

%}

if type == "MOS"

zeta = MOS\_or\_zeta;

output = exp(-zeta\*pi/sqrt(1-zeta^2))\*100;

elseif type == "zeta"

MOS = MOS\_or\_zeta;

output = -log(MOS/100)/sqrt(pi^2 + (log(MOS/100))^2);

end

end