AAE 364: Controls Systems Analysis

HW7: Root Locus Analysis

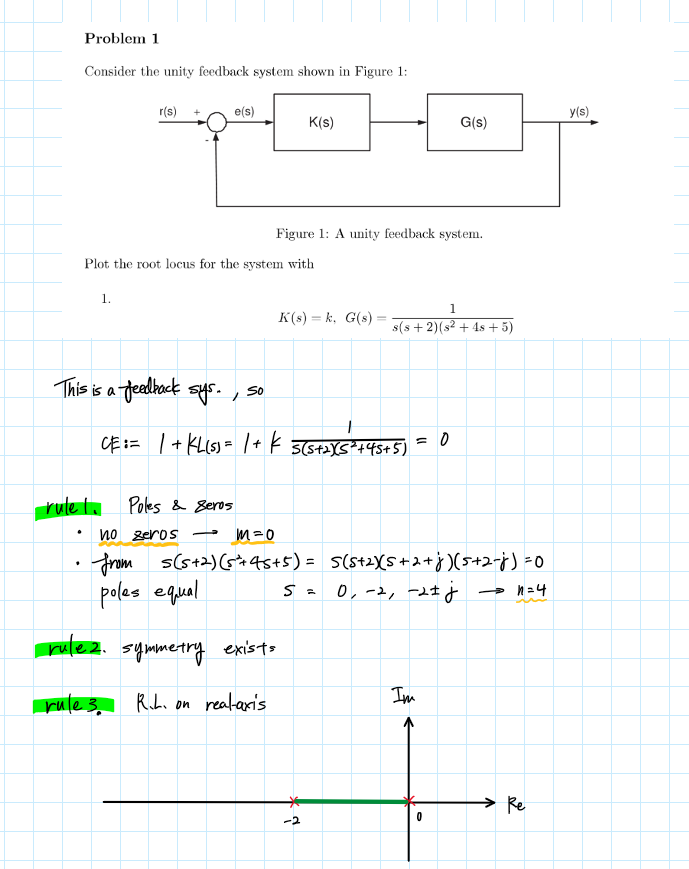
Dr. Sun

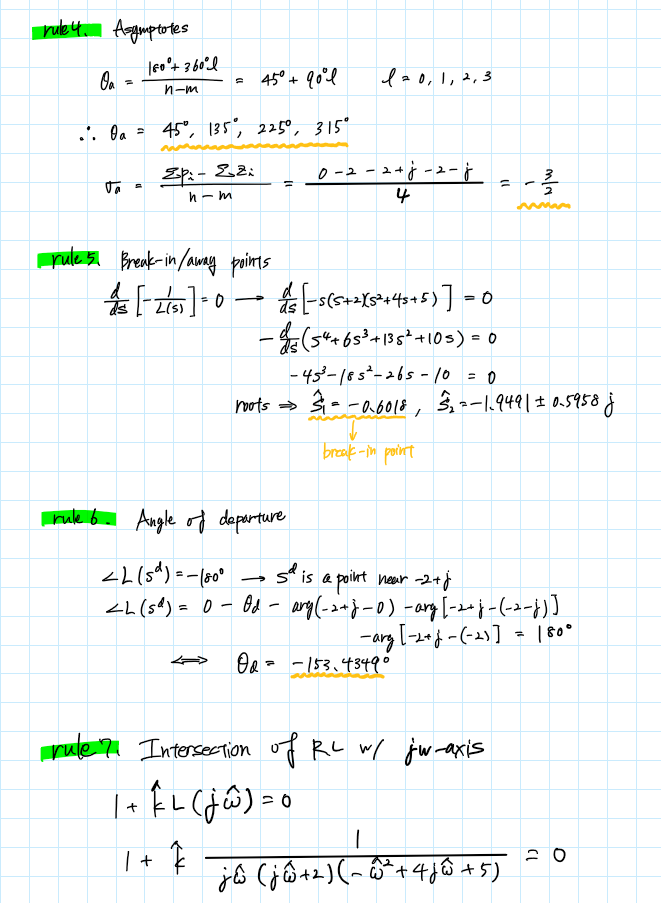
School of Aeronautical & Astronautical Engineering

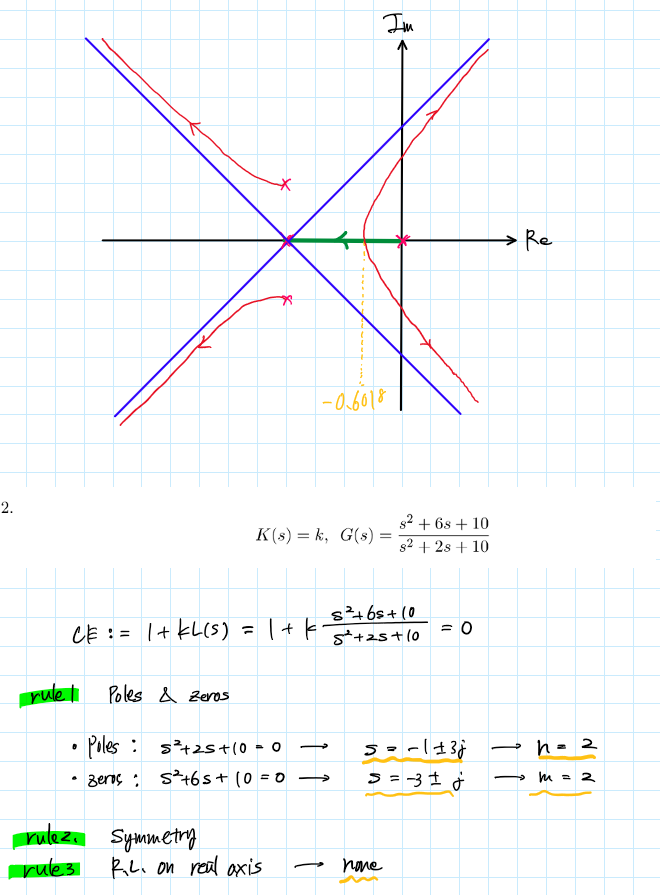
Purdue University

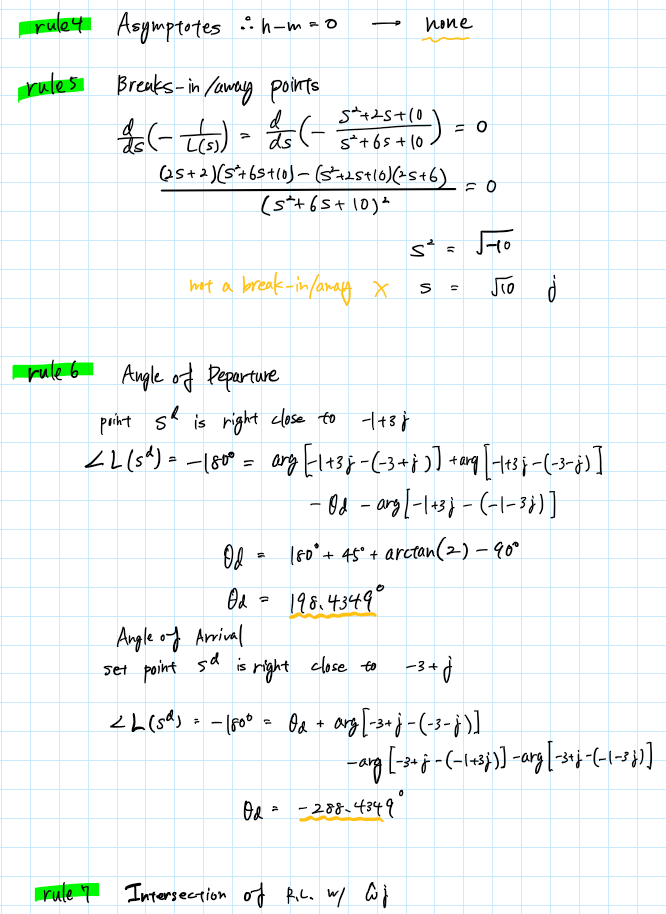
Tomoki Koike

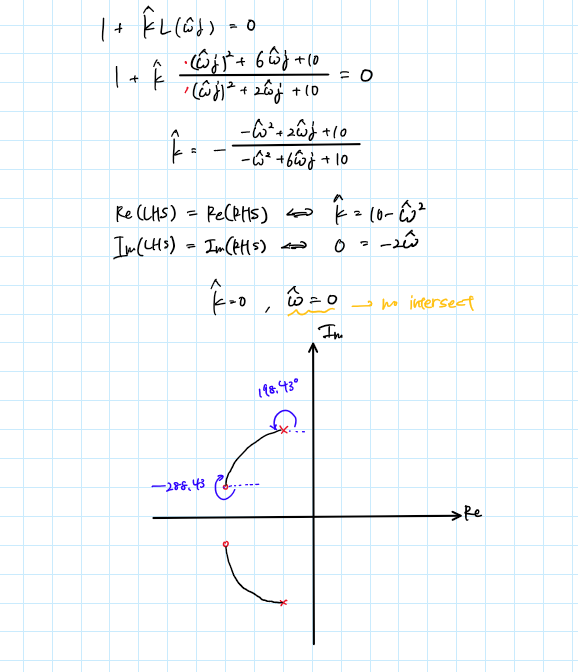
Friday March 13th 2020

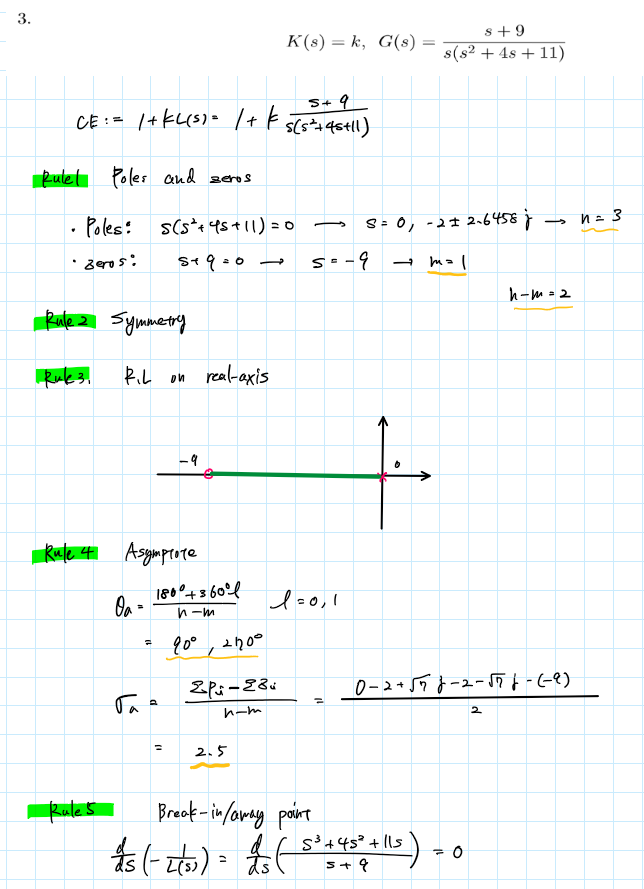


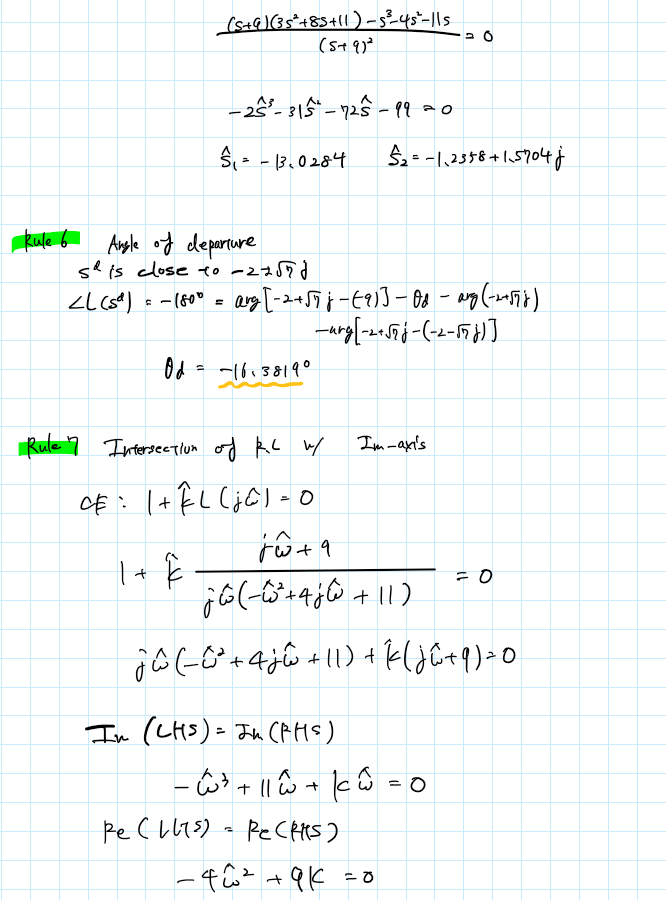


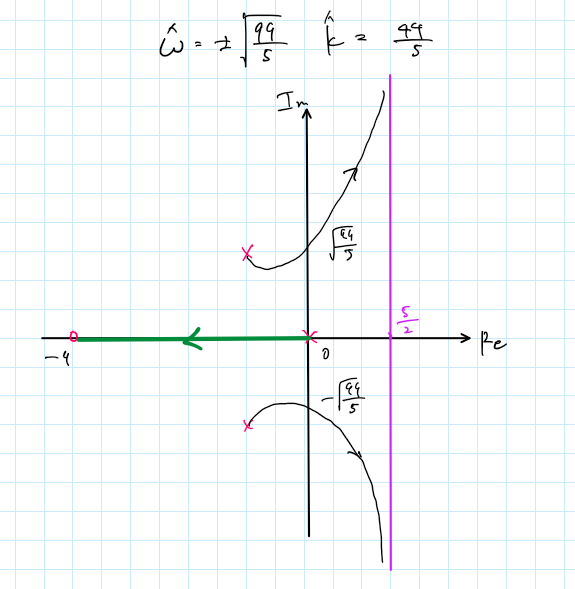


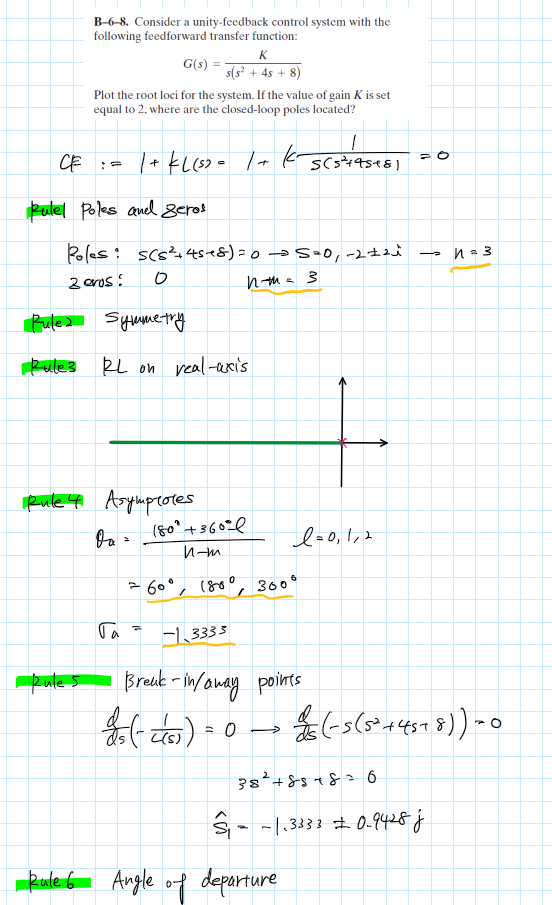


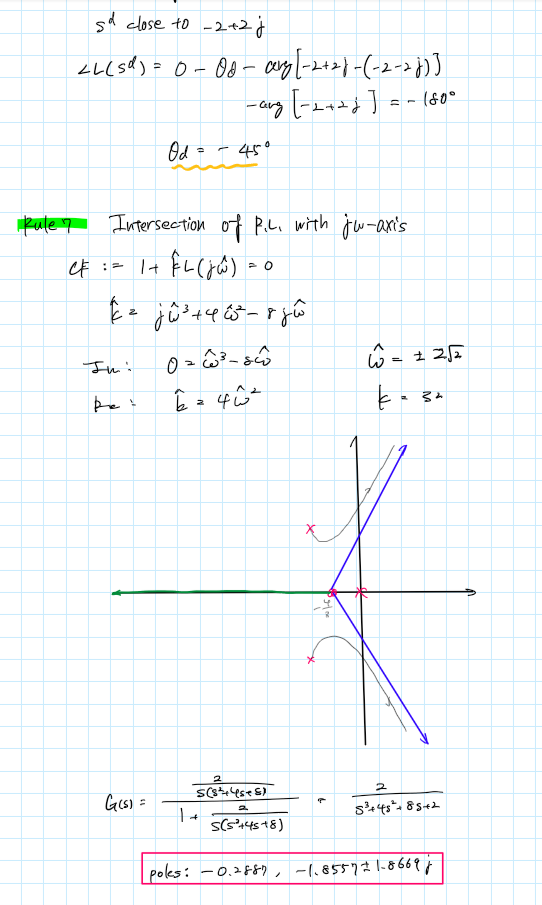


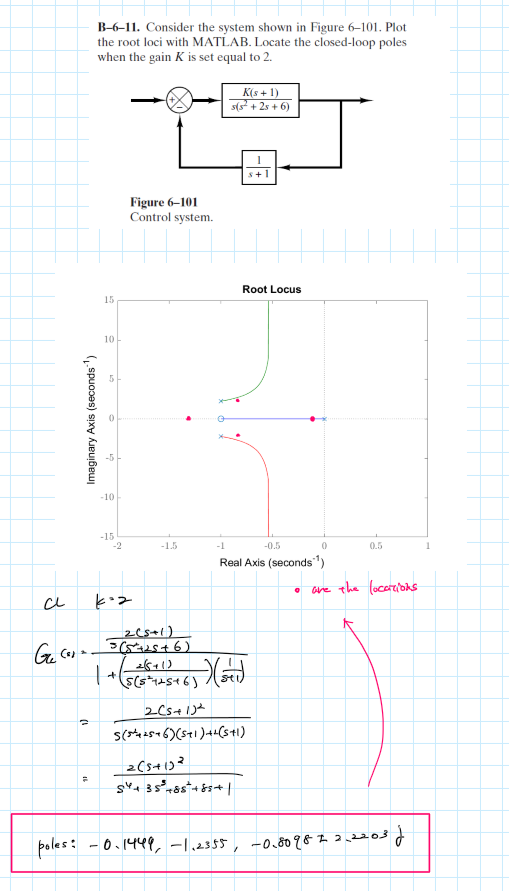


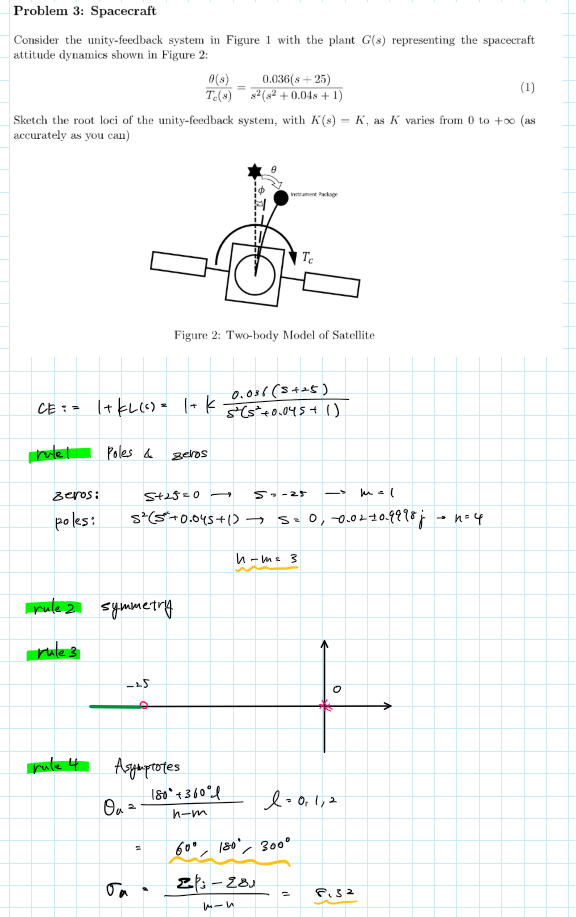


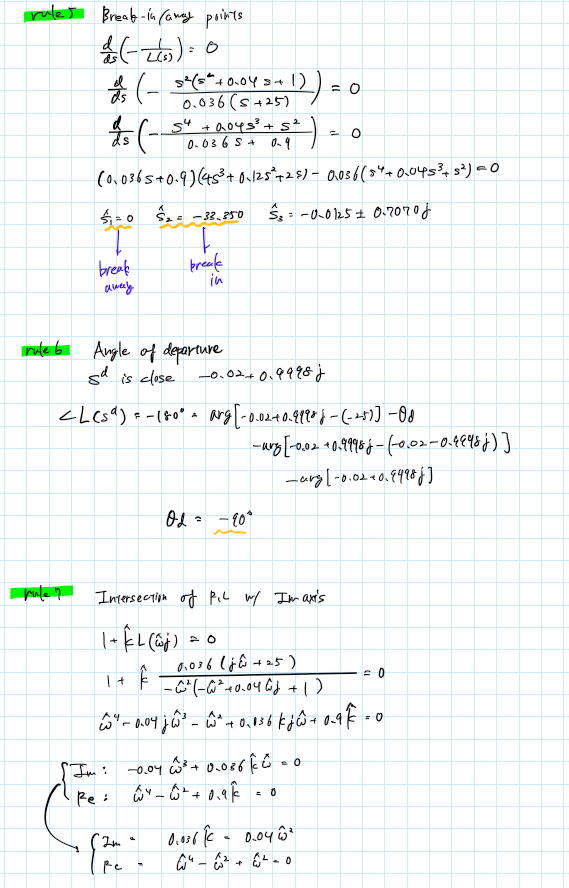


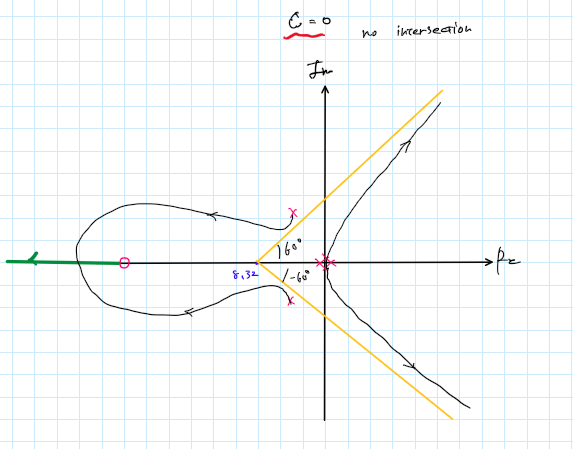












Appendix

**AAE364 HW7 MATLAB CODE**

clear all; close all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE364\matlab\matlab\_output\hw7';

set(groot, 'defaulttextinterpreter',"latex");

set(groot, 'defaultAxesTickLabelInterpreter',"latex");

set(groot, 'defaultLegendInterpreter',"latex");

**P1**

% 1

num = [1];

den = conv([1 0],[1 2]);

den = conv(den,[1 4 5])

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(2), "deg", "poles")

[k,w] = intersection\_IM\_axis(num,den)

L = tf(num, den)

fig1 = figure(1);

rlocus(L)

saveas(fig1, fullfile(fdir, 'p1\_1.png'));

% 2

num = [1 6 10];

den = [1 2 10];

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(1), "deg","poles")

theta\_d\_arr = departure\_angle\_calc(zrs,poles,zrs(1),"deg","zeros")

[k,w] = intersection\_IM\_axis(num,den)

L = tf(num,den);

fig2 = figure(2);

rlocus(L)

saveas(fig2, fullfile(fdir, 'p1\_2.png'));

% 3

num = [1 9];

den = conv([1 0],[1 4 11])

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(2), "deg","poles")

[k,w] = intersection\_IM\_axis(num,den)

L = tf([1 9], [1 4 11 0]);

fig3 = figure(3);

rlocus(L)

saveas(fig3, fullfile(fdir, 'p1\_3.png'));

**P2**

% B-6-8

num = [1];

den = conv([1 0],[1 4 8])

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(2), "deg","poles")

[k,w] = intersection\_IM\_axis(num,den)

L = tf([0 1], [1 4 8 0]);

fig4 = figure(4);

rlocus(L)

saveas(fig4, fullfile(fdir, 'Figure4.png'));

% B-6-11

num = [1 1];

den = conv([1 0],[1 2 6])

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(2), "deg","poles")

[k,w] = intersection\_IM\_axis(num,den)

L = tf([1 1], [1 2 6 0]);

fig5 = figure(5);

rlocus(L)

saveas(fig5, fullfile(fdir, 'Figure5.png'));

**P3**

num = 0.036\*[1 25];

den = conv([1 0 0],[1 0.04 1])

poles = roots(den)

zrs = roots(num)

[angs,sigma] = RL\_asymptote(zrs,poles)

bi\_pt = break\_in\_away\_pt(num,den)

theta\_d = departure\_angle\_calc(zrs, poles, poles(3), "deg","poles")

[k,w] = intersection\_IM\_axis(num,den)

L = tf(num, den);

fig6 = figure(6);

rlocus(L)

saveas(fig6, fullfile(fdir, 'Figure6.png'));

function theta = departure\_angle\_calc(zrs, poles, obj, angle\_type, obj\_type)

%{

zrs: the zrs of the transfer function

poles: the poles of the transfer function

obj: the aimed pole to find the departure angle

%}

if obj\_type == "poles"

idx = find(poles==obj);

poles(idx) = [];

else

idx = find(zrs==obj);

zrs(idx) = [];

end

theta = 0;

if not(isempty(zrs))

for i = 1:length(zrs)

theta = theta + angle(obj - zrs(i));

end

end

for i = 1:length(poles)

theta = theta - angle(obj - poles(i));

end

if obj\_type == "poles"

theta = theta + deg2rad(180);

else

theta = -deg2rad(180) - theta;

end

if angle\_type == "deg"

theta = rad2deg(theta);

end

end

function rts = break\_in\_away\_pt(num,den)

[q, d] = polyder(-den,num)

rts = roots(q)

rts = rts(rts==real(rts));

end

function [angs, sigma] = RL\_asymptote(zrs, poles)

n = length(poles)

m = length(zrs)

angs = zeros([1,n-m]);

for i = 0:(n-m)-1

angs(i+1) = (180 + 360\*i)/(n - m);

end

sigma = (sum(poles) - sum(zrs))/(n - m);

end

function [K, W] = intersection\_IM\_axis(num, den)

syms k w

n = length(den);

f1 = 0;

f2 = 0;

if rem(n,2) == 1

for i = 1:2:n

if rem(i-1,4) == 0

f1 = f1 + den(i)\*w^(n-i);

else

f1 = f1 + den(i)\*w^(n-i)\*(-1);

end

end

for i = 2:2:n-1

if rem(i-1,4) == 1

f2 = f2 + den(i)\*w^(n-i);

else

f2 = f2 + den(i)\*w^(n-i)\*(-1);

end

end

elseif rem(n,2) == 0

for i = 1:2:n-1

if rem(i-1,4) == 0

f1 = f1 + den(i)\*w^(n-i);

else

f1 = f1 + den(i)\*w^(n-i)\*(-1);

end

end

for i = 2:2:n

if rem(i-1,4) == 1

f2 = f2 + den(i)\*w^(n-i);

else

f2 = f2 + den(i)\*w^(n-i)\*(-1);

end

end

end

n = length(num);

p1 = 0;

p2 = 0;

if rem(n,2) == 1

for i = 1:2:n

if rem(i-1,4) == 0

p1 = p1 + num(i)\*w^(n-i);

else

p1 = p1 + num(i)\*w^(n-i)\*(-1);

end

end

for i = 2:2:n-1

if rem(i-1,4) == 1

p2 = p2 + num(i)\*w^(n-i);

else

p2 = p2 + num(i)\*w^(n-i)\*(-1);

end

end

elseif rem(n,2) == 0

for i = 1:2:n-1

if rem(i-1,4) == 0

p1 = p1 + num(i)\*w^(n-i);

else

p1 = p1 + num(i)\*w^(n-i)\*(-1);

end

end

for i = 2:2:n

if rem(i-1,4) == 1

p2 = p2 + num(i)\*w^(n-i);

else

p2 = p2 + num(i)\*w^(n-i)\*(-1);

end

end

end

eqn1 = k\*p1 == f1

eqn2 = k\*p2 == f2

a = vpasolve([eqn1 eqn2], [k w]);

K = double(a.k);

W = double(a.w);

end

A close up of a map

Description automatically generatedMATLAB PLOTS

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

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated