MECH467 Prelab #3

Ratthamnoon Prakitpong
#63205165

A1.

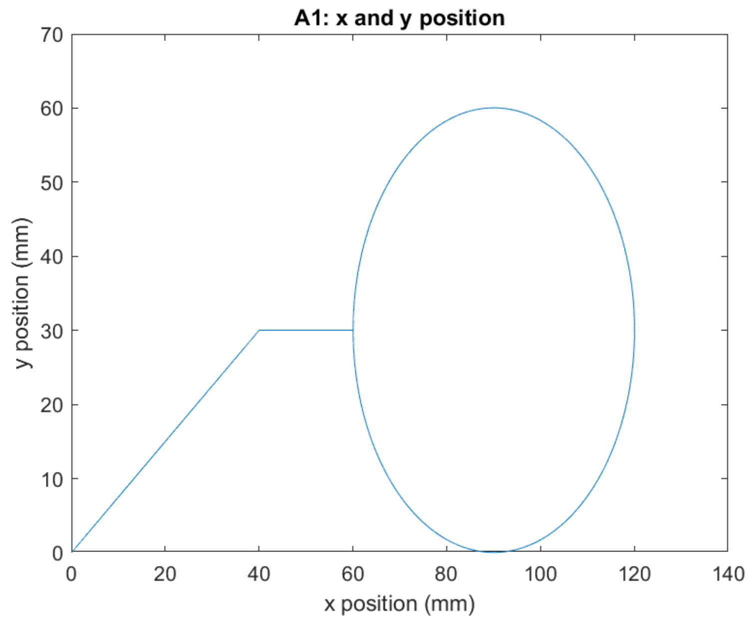


Fig A1: X and Y position of sample trajectory

A2.

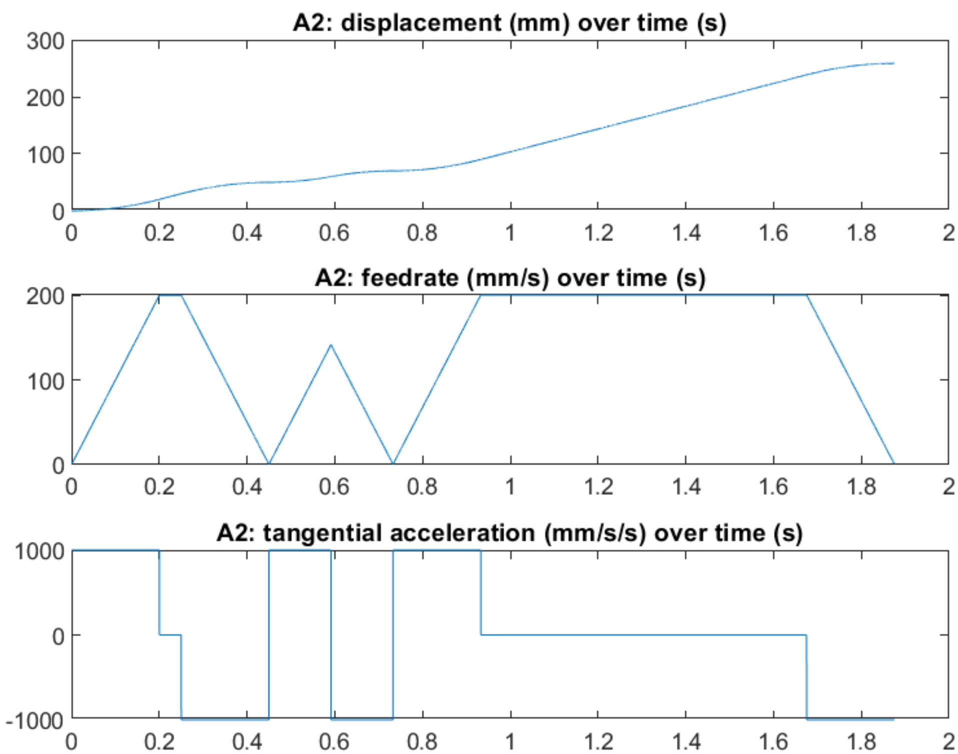


Fig A2: Displacement, feedrate, and tangential acceleration of sample trajectory

A3.

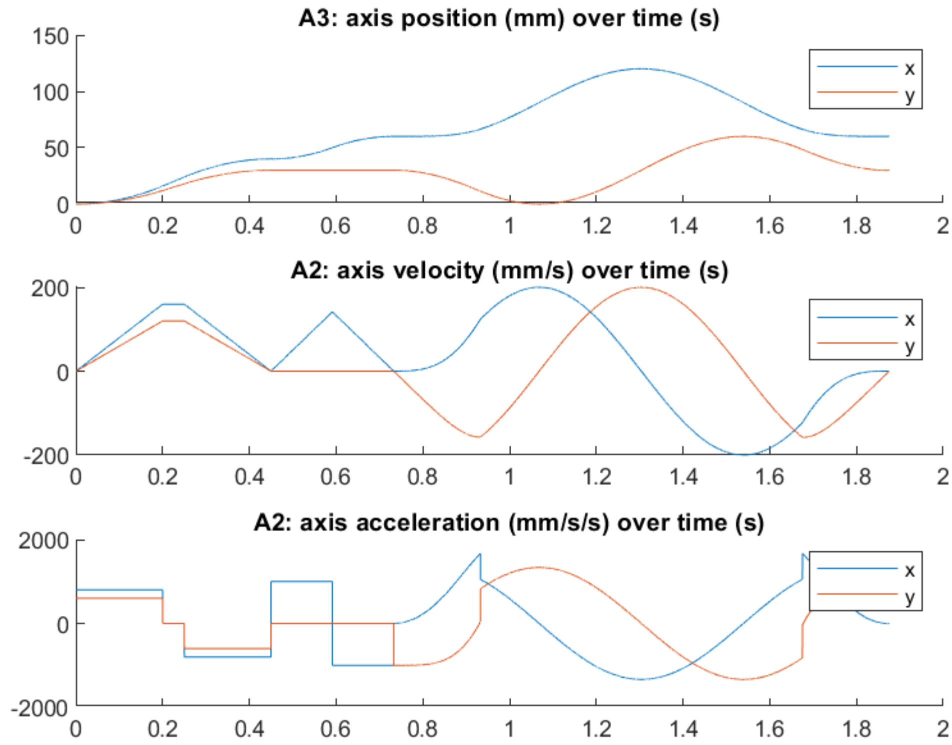


Fig A3: Axis position, velocity, and acceleration of sample trajectory

B1.

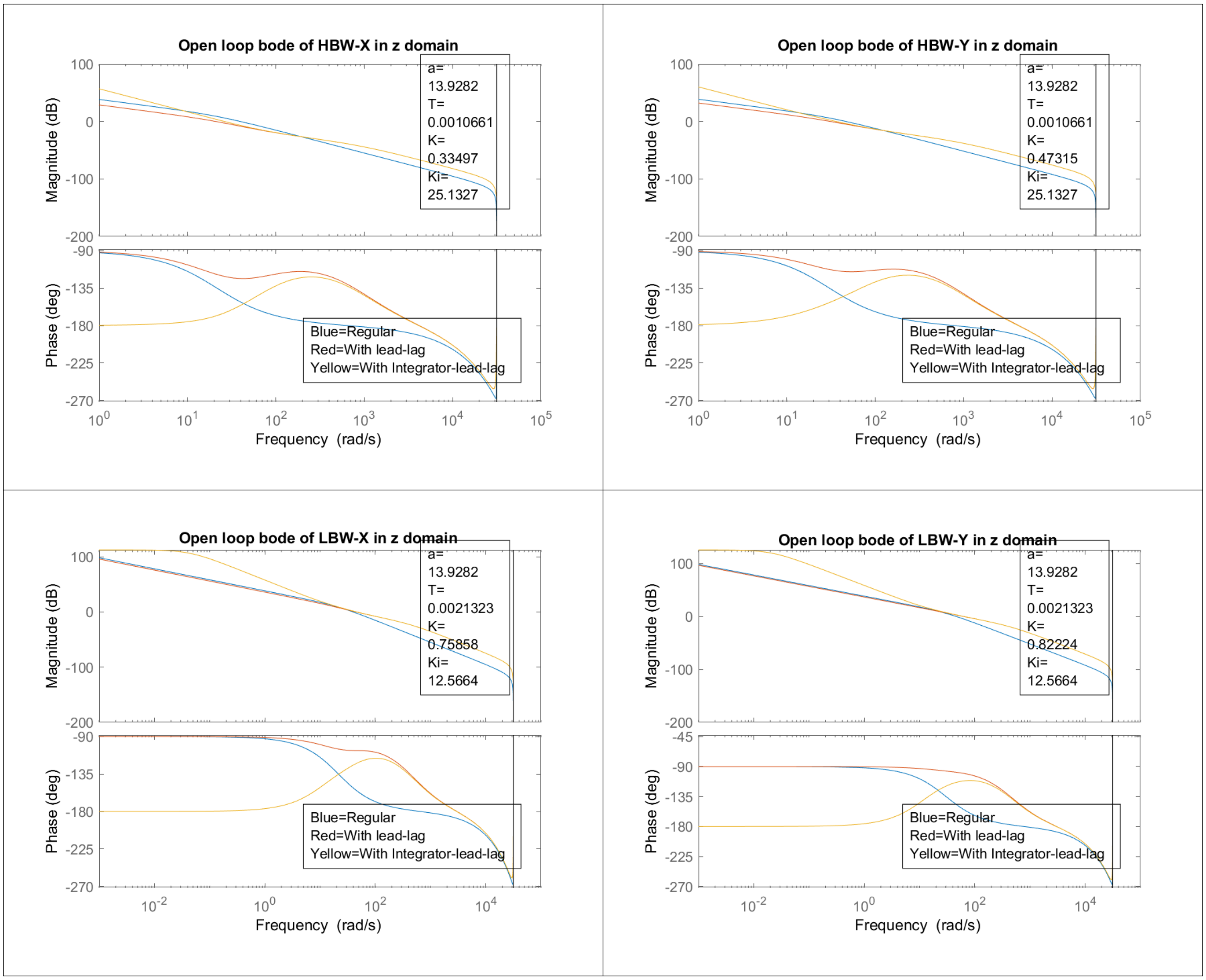


Fig B1: Lead-Lag-Integrator of X and Y axis motors, for LBW and HBW

The parameters of LLI controllers are written in the plot. They can be used in the equations like shown here:

$$\text{Lead-Lag} = K\frac{1+\alpha Ts}{1+Ts}$$

$$\text{Integrator} = \frac{K_i + s}{s}$$

B2.

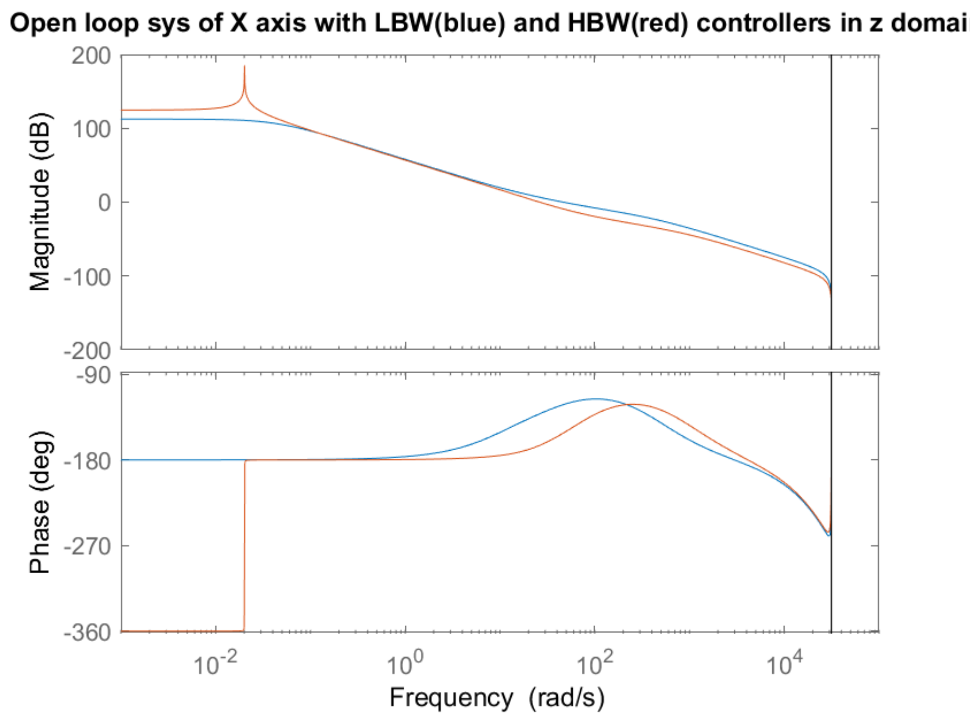


Fig B2.1: Open loop system of X axis with LBW and HBW controllers, in z domain

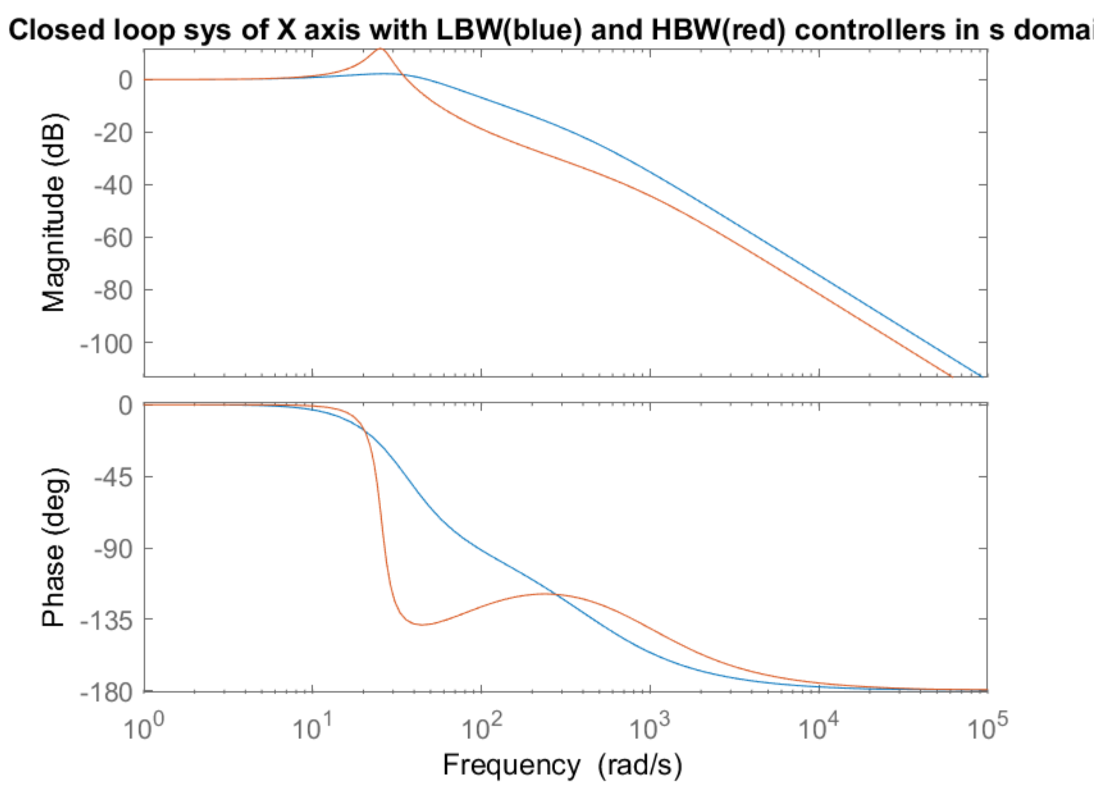


Fig B2.2: Close loop system of X axis with LBW and HBE controllers, in s domain

B3.

System	Domain	Bandwidth(rad/s)	Zero	Pole	RiseTime(s)	Overshoot(%)
LBW-X	z	67.6374	[-1;1;1]	[1;1+0.0026i;1-0.0026i;0.96]	0.0259	23.0245
LBW-X	s	67.4119	[-34;-13]	[-4.3e+02;-24+26i;-24-26i;-14]	0.025995	22.9486
LBW-Y	z	90.1982	[-1;1;1]	[1;1+0.0012i;1-0.0012i;0.96]	0.0202	13.9894
LBW-Y	s	89.8381	[-34;-13]	[-4e+02;-43+13i;-43-13i;-16]	0.020406	13.949
HBW-X	z	40.4458	[-1;1;0.99]	[1+0.0025i;1-0.0025i;1;0.91]	0.041	67.1054
HBW-X	s	40.4492	[-67;-25]	[-9.3e+02;-23;-3.4+25i;-3.4-25i]	0.04174	66.9174
HBW-Y	z	55.572	[-1;1;0.99]	[1+0.0032i;1-0.0032i;1;0.91]	0.0305	47.9008
HBW-Y	s	55.5764	[-67;-25]	[-9.2e+02;-9.9+32i;-9.9-32i;-28]	0.030648	47.6812

Table B3: Bandwidth, zeros, poles, rise time, and overshoot for each combination of bandwidth, axis, and domain in a closed loop system

C1.

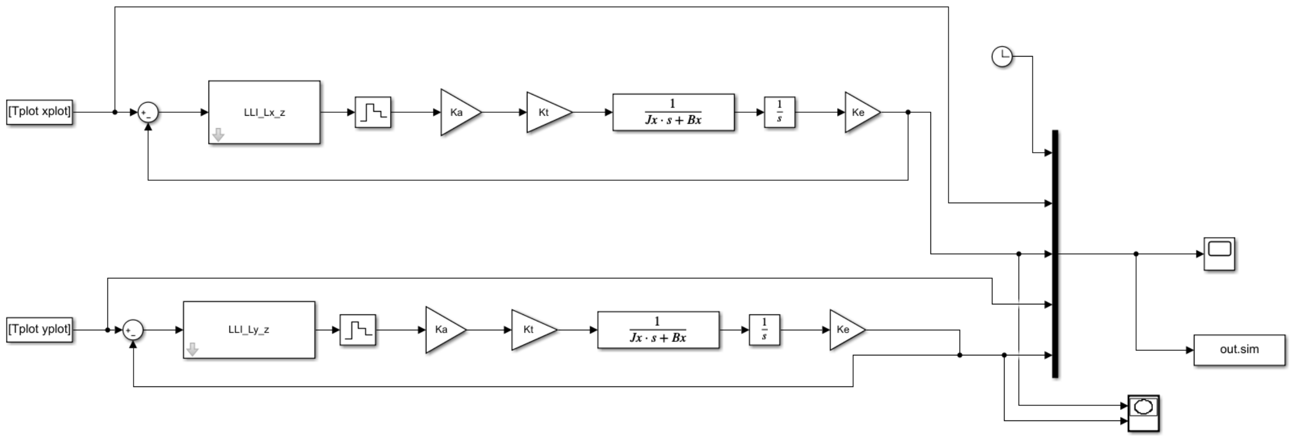


Fig C1.1: Simulink model

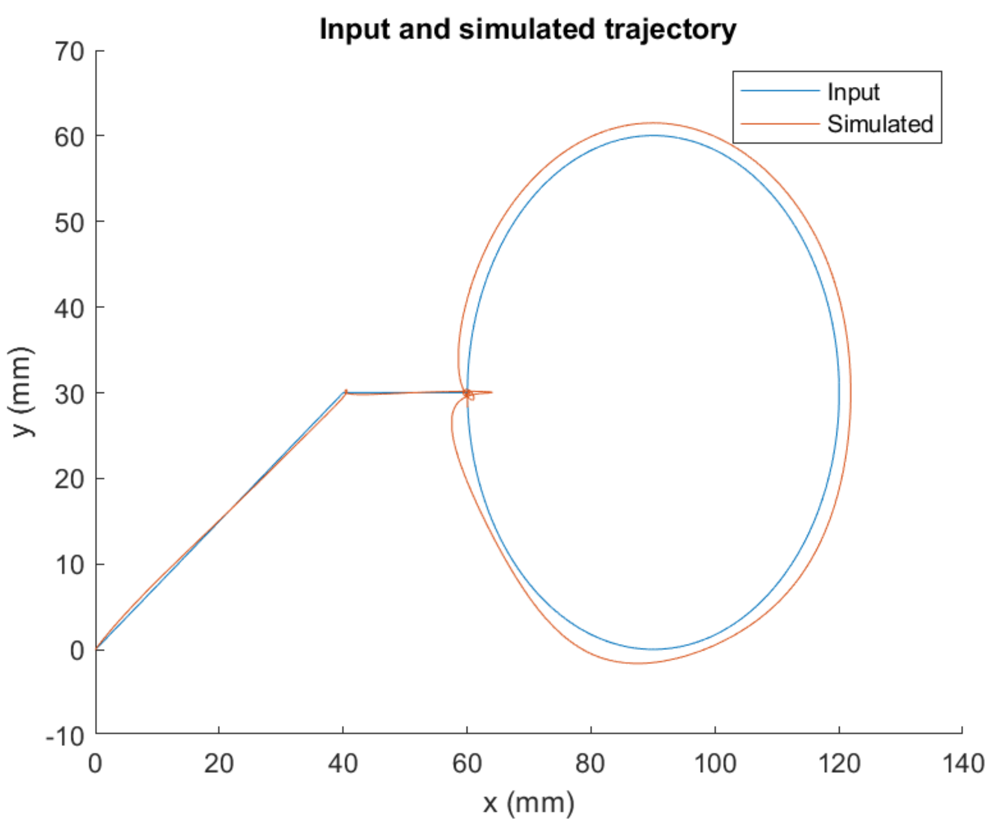


Fig C1.2: Input and simulated toolpath

Controllers has some issues at the point where the circle starts due to overshoot of P2-to-P3 path, and consequently causes misalignment in the rest of the tool path. To reduce error, it is recommended that gain is increased. Likely that the rule-of-thumb used to get Ki (Ki = wc / 10) resulted in Ki that is not large enough to eliminate steady state error.

C2.

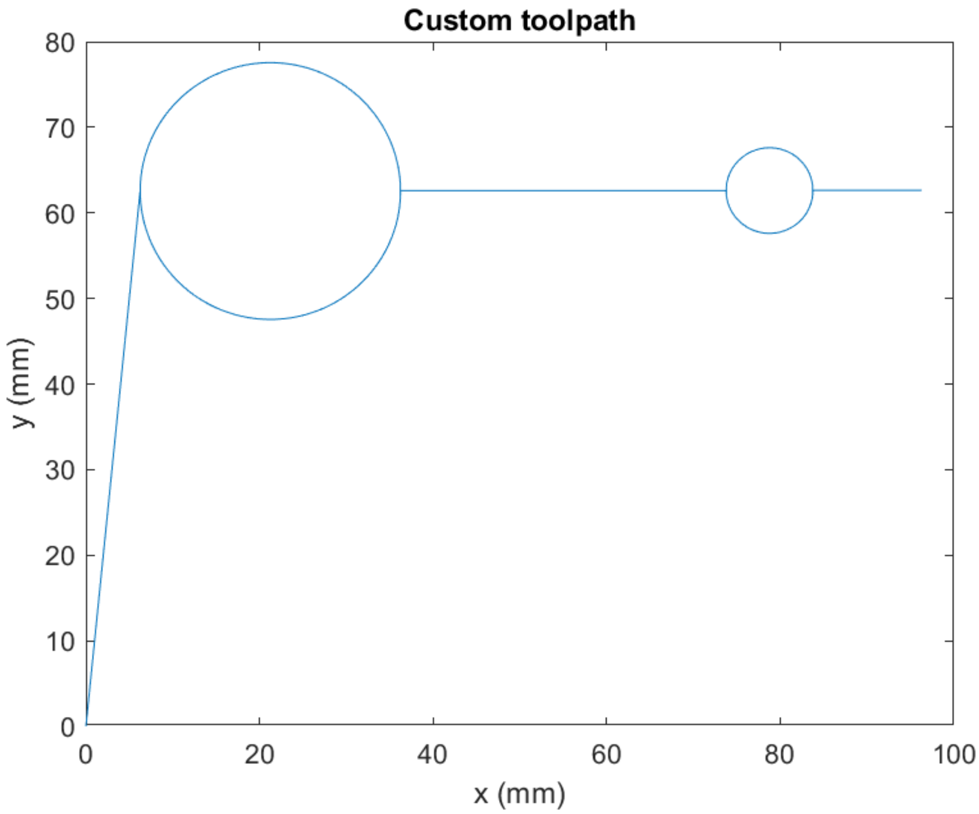


Fig C2: Custom toolpath

The toolpath takes multiple cuts around the circles to ensure nicer finish.

Appendix:

1. Part A Matlab code
2. Part B Matlab code
3. Part C Matlab code
4. myTraj.m (attached separately to this pdf)

Contents

init vars

```
A = 1000; % mm/s^2
fc = 200; % mm/s
T = 0.1; % ms
T = T * 0.001; % s

P1 = [0 0];
P2 = [40 30];
P3 = [60 30];
P4 = [90 30];
```

P1 -> P2

```
[T1, T2, T3] = calc(50); % sqrt(30*30+40*40)=50
T_total = T1+T2+T3;

x_ratio = 4/5;
y_ratio = 3/5;

t = 0;
s = 0;
sdot = 0;
sdotdot = A;
xr = 0;
yr = 0;
vxr = 0;
vyr = 0;
axr = A*x_ratio;
ayr = A*y_ratio;

count = 1;
data1 = zeros(T_total/T, 10);
while t <= T_total-T
    data1(count, 1) = t;
    data1(count, 2) = s;
    data1(count, 3) = sdot;
```

```
    data1(count, 4) = sdotdot;
    data1(count, 5) = xr;
    data1(count, 6) = yr;
    data1(count, 7) = vxr;
    data1(count, 8) = vyr;
    data1(count, 9) = axr;
    data1(count, 10) = ayr;

    if t < T1
        sdotdot = A;
    elseif t >= T1 && t < T1+T2
        sdotdot = 0;
    else
        sdotdot = -A;
    end

    sdot = sdot + T*sdotdot;
    s = s + sdot*T;

    xr = s*x_ratio;
    yr = s*y_ratio;
    vxr = sdot*x_ratio;
    vyr = sdot*y_ratio;
    axr = sdotdot*x_ratio;
    ayr = sdotdot*y_ratio;

    t = t + T;
    count = count + 1;
end
```

P2 -> P3

```
[T1, T2, T3] = calc(20); % 60-40=20
T_total = T1+T2+T3;

x_ratio = 1;
y_ratio = 0;

t = 0;
s = 0;
sdot = 0;
sdotdot = A;
xr = 0;
yr = 0;
vxr = 0;
vyr = 0;
axr = A*x_ratio;
ayr = A*y_ratio;

count = 1;
data2 = zeros(T_total/T, 10);
```

```
while t <= T_total-T
    data2(count, 1) = t+data1(end, 1);
    data2(count, 2) = s+data1(end, 2);
    data2(count, 3) = sdot+data1(end, 3);
    data2(count, 4) = sdotdot;
    data2(count, 5) = xr+data1(end, 5);
    data2(count, 6) = yr+data1(end, 6);
    data2(count, 7) = vxr+data1(end, 7);
    data2(count, 8) = vyr+data1(end, 8);
    data2(count, 9) = axr;
    data2(count, 10) = ayr;

    if t < T1
        sdotdot = A;
    elseif t >= T1 && t < T1+T2
        sdotdot = 0;
    else
        sdotdot = -A;
    end

    sdot = sdot + T*sdotdot;
    s = s + sdot*T;

    xr = s*x_ratio;
    yr = s*y_ratio;
    vxr = sdot*x_ratio;
    vyr = sdot*y_ratio;
    axr = sdotdot*x_ratio;
    ayr = sdotdot*y_ratio;

    t = t + T;
    count = count + 1;
end
```

P3 circle

```
L = 2*3.1415*30;
[T1, T2, T3] = calc(L); % 90-60=30
T_total = T1+T2+T3;

t = 0;
s = 0;
sdot = 0;
sdotdot = A;
xr = 0;
yr = 0;
vxr = 0;
vyr = 0;
axr = A*x_ratio;
ayr = A*y_ratio;
```

```
count = 1;
data3 = zeros(T_total/T, 10);
while t <= T_total-T
    if t < T1
        sdotdot = A;
    elseif t >= T1 && t < T1+T2
        sdotdot = 0;
    else
        sdotdot = -A;
    end

    sdot = sdot + T*sdotdot;
    s = s + sdot*T;

    [x_ratio, y_ratio] = getRatios(s/L);

    vxr_prev = vxr;
    vyr_prev = vyr;
    vxr = sdot*x_ratio;
    vyr = sdot*y_ratio;

    R = 30;
    xr = xr + vxr*T;
    yr = yr + vyr*T;

    axr = (vxr - vxr_prev)/T;
    ayr = (vyr - vyr_prev)/T;

    data3(count, 1) = t+data2(end, 1);
    data3(count, 2) = s+data2(end, 2);
    data3(count, 3) = sdot+data2(end, 3);
    data3(count, 4) = sdotdot;
    data3(count, 5) = xr+data2(end, 5);
    data3(count, 6) = yr+data2(end, 6);
    data3(count, 7) = vxr+data2(end, 7);
    data3(count, 8) = vyr+data2(end, 8);
    data3(count, 9) = axr;
    data3(count, 10) = ayr;

    t = t + T;
    count = count + 1;
end
```

data for plot

```
data = [data1; data2; data3];
```

plot

```
plot(data(:, 5), data(:, 6));
title('A1: x and y position');
xlabel('x position (mm)');
ylabel('y position (mm)');
saveas(gcf, 'qA1.png');
clf;
txy.t = data(:, 1);
txy.x = data(:, 5);
txy.y = data(:, 6);
save sampleTraj txy

subplot(3,1,1);
plot(data(:,1), data(:,2));
title('A2: displacement (mm) over time (s)');
subplot(3,1,2);
plot(data(:,1), data(:,3));
title('A2: feedrate (mm/s) over time (s)');
subplot(3,1,3);
plot(data(:,1), data(:,4));
title('A2: tangential acceleration (mm/s/s) over time (s)');
saveas(gcf, 'qA2.png');
clf;

subplot(3,1,1);
hold on;
plot(data(:,1), data(:,5));
plot(data(:,1), data(:,6));
title('A3: axis position (mm) over time (s)');
legend('x', 'y');
subplot(3,1,2);
hold on;
plot(data(:,1), data(:,7));
plot(data(:,1), data(:,8));
title('A2: axis velocity (mm/s) over time (s)');
legend('x', 'y');
subplot(3,1,3);
hold on;
plot(data(:,1), data(:,9));
plot(data(:,1), data(:,10));
title('A2: axis acceleration (mm/s/s) over time (s)');
legend('x', 'y');
saveas(gcf, 'qA3.png');
clf;
```

helper

```
function [x_r, y_r] = getRatios(r)
    x_r = 0;
    y_r = 0;
    if r < .25
        theta = (pi/2)*(r/.25);
        x_r = sin(theta);
        y_r = -cos(theta);
    elseif r >= .25 && r < 0.5
        theta = (pi/2)*((r-0.25)/.25);
        x_r = cos(theta);
        y_r = sin(theta);
    elseif r >= 0.5 && r < 0.75
        theta = (pi/2)*((r-0.5)/.25);
        x_r = -sin(theta);
        y_r = cos(theta);
    else
        theta = (pi/2)*((r-0.75)/.25);
        x_r = -cos(theta);
        y_r = -sin(theta);
    end
end

function [T1, T2, T3] = calc(L)
```

```
    A = 1000; % mm/s^2
    fc = 200; % mm/s
    T = 0.1; % ms
    T = T * 0.001; % s

    T1 = fc/A;
    T3 = T1;
    s_init = A*T1*T1/2; % this is 20
    T2 = (L-2*s_init)/fc;
    T2 = ceil(T2/T)*T;
    if T2 < 0
        T2 = 0;
        s_init = L/2;
        T1 = sqrt(2*s_init/A);
        T1 = ceil(T1/T)*T;
        T3 = T1;
    end
end
```

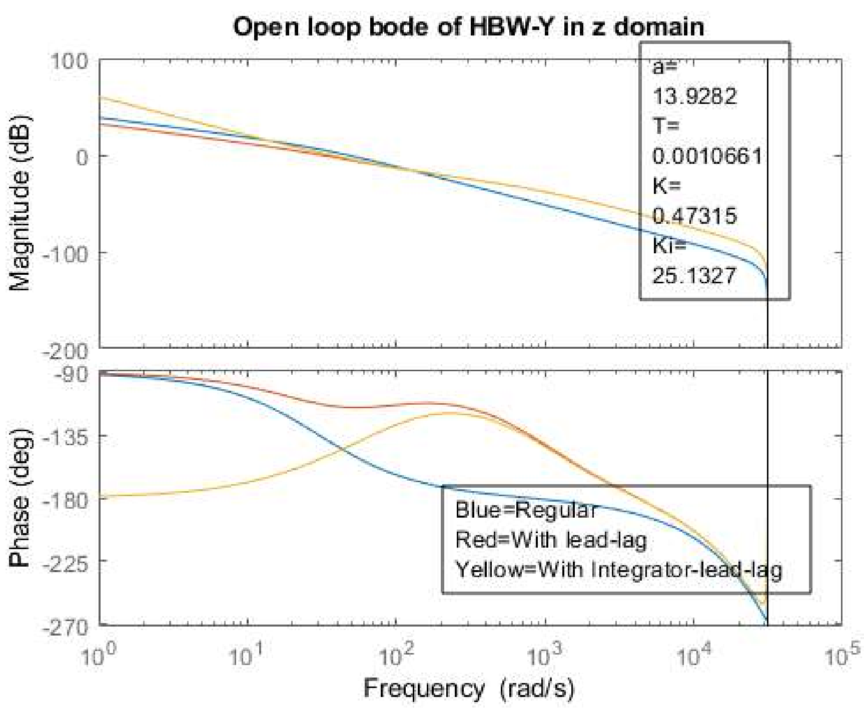
Published with MATLAB® R2020b

Contents


```
toPlot = [20, 60, 2.4, .000436, .0094; 20, 60, 1.7, .0003, .0091; 40, 60, 9.5, .000436, .0094; 40, 60, 6.5, .0003, .0091];
names = {'LBW-X', 'LBW-Y', 'HBW-X', 'HBW-Y'};
```

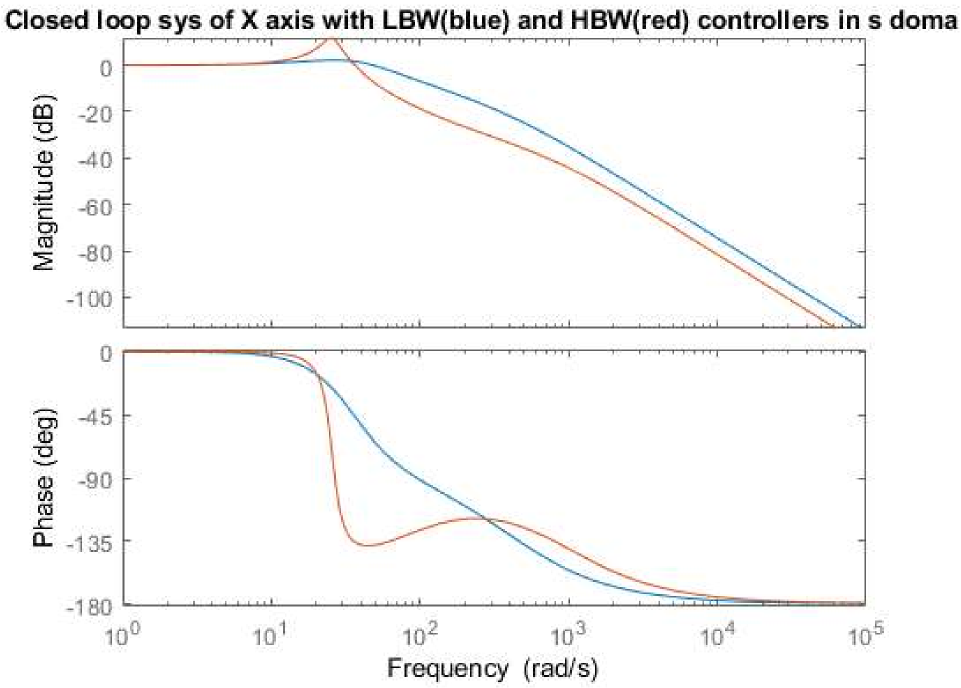
1

```
for i = 1:4
    B1(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5),names{i});
end
```



2

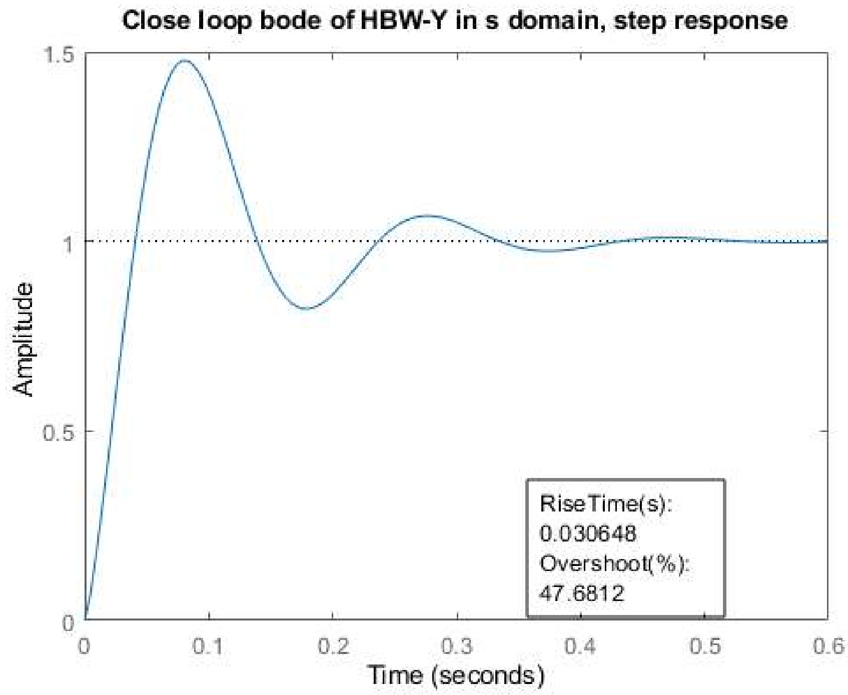
```
clf;
i = 1;
B2_open_z(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5));
i = 3;
B2_open_z(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5));
title('Open loop sys of X axis with LBW(blue) and HBW(red) controllers in z domain');
saveas(gcf, 'qB2-1.png');
clf;
i = 1;
B2_close_s(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5));
i = 3;
B2_close_s(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5));
title('Closed loop sys of X axis with LBW(blue) and HBW(red) controllers in s domain');
saveas(gcf, 'qB2-2.png');
```



3

```
table = [];
for i = 1:4
    info1 = B3_z(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5),names{i});
    info2 = B3_s(toPlot(i, 1),toPlot(i, 2),toPlot(i, 3),toPlot(i, 4),toPlot(i, 5),names{i});
    table = [table; info1; info2];
end
disp(table);
```

```
  Columns 1 through 4

    {'LBW-X'}    {["67.6374"]}    {'[-1;1;1]'    }    {'[1;1+0.0026i;1-…'}
    {'LBW-X'}    {["67.4119"]}    {'[-34;-13]'   }    {'[-4.3e+02;-24+2…'}
    {'LBW-Y'}    {["90.1982"]}    {'[-1;1;1]'    }    {'[1;1+0.0012i;1-…'}
    {'LBW-Y'}    {["89.8381"]}    {'[-34;-13]'   }    {'[-4e+02;-43+13i…'}
    {'HBW-X'}    {["40.4458"]}    {'[-1;1;0.99]'}    {'[1+0.0025i;1-0.…'}
    {'HBW-X'}    {["40.4492"]}    {'[-67;-25]'   }    {'[-9.3e+02;-23;-…'}
    {'HBW-Y'}    {["55.572" ]}    {'[-1;1;0.99]'}    {'[1+0.0032i;1-0.…'}
    {'HBW-Y'}    {["55.5764"]}    {'[-67;-25]'   }    {'[-9.2e+02;-9.9+…'}

  Columns 5 through 6

    {["0.0259"   ]}    {["23.0245"]}
    {["0.025995"]}    {["22.9486"]}
    {["0.0202"   ]}    {["13.9894"]}
    {["0.020406"]}    {["13.949" ]}
    {["0.041"    ]}    {["67.1054"]}
    {["0.04174" ]}    {["66.9174"]}
    {["0.0305"   ]}    {["47.9008"]}
    {["0.030648"]}    {["47.6812"]}
```



functions

```
function B1(w, phi, mag, Je, Be, name)
    Ke = 1.59;
    Ka = 1;
    Kt = .49;
    T = .0001;

    w = 2*pi*w; % hz -> rad/s
    phi = phi * pi/180; % deg -> rad
    K = 10^(-mag/20); % found that with K=1, mag at w is -22.4dB, so we need to shift by +22.4dB
    a = (1+sin(phi))/(1-sin(phi));
    t = 1/(sqrt(a)*w);
    C = tf([K*a*t K], [t 1]);

    %{
    disp(a);
    disp(t);
    disp(K);
%}
    Ki = w/10;
    G = tf([1 Ki], [1 0]);
    %disp(Ki);

    H = tf((Ke*Ka*Kt),[Je Be 0]);   % continuous time
    Hd = c2d(H,T);                  % discrete time
    clf;
    hold on;
    bode(Hd);

    % a
    CHd = c2d(C*H, T);
    bode(CHd);

    % b
    GCHd = c2d(G*C*H, T);
    bode(GCHd);

    title(['Open loop bode of ' name ' in z domain']);
    text1 = ['Blue=Regular' newline 'Red=With lead-lag' newline 'Yellow=With Integrator-lead-lag'];
    text2 = ['a=' strtrim(string(a)) 'T=' strtrim(string(t)) 'K=' string(K) 'Ki=' string(Ki)];
```

```
    annotation('textbox', [0.5, 0.2, 0.1, 0.1], 'String', text1);
    annotation('textbox', [0.7, 0.8, 0.1, 0.1], 'String', text2);
    saveas(gcf, ['qB1-' name '.png']);
end

function B2_open_z(w, phi, mag, Je, Be)
    Ke = 1.59;
    Ka = 1;
    Kt = .49;
    T = .0001;

    w = 2*pi*w; % hz -> rad/s
    phi = phi * pi/180; % deg -> rad
    K = 10^(-mag/20); % found that with K=1, mag at w is -22.4dB, so we need to shift by +22.4dB
    a = (1+sin(phi))/(1-sin(phi));
    t = 1/(sqrt(a)*w);
    C = tf([K*a*t K], [t 1]);

    Ki = w/10;
    G = tf([1 Ki], [1 0]);

    H = tf((Ke*Ka*Kt),[Je Be 0]);   % continuous time
    hold on;

    % b
    GCHd = c2d(G*C*H, T);
    bode(GCHd);
end

function B2_close_s(w, phi, mag, Je, Be)
    Ke = 1.59;
    Ka = 1;
    Kt = .49;
    T = .0001;

    w = 2*pi*w; % hz -> rad/s
    phi = phi * pi/180; % deg -> rad
    K = 10^(-mag/20); % found that with K=1, mag at w is -22.4dB, so we need to shift by +22.4dB
    a = (1+sin(phi))/(1-sin(phi));
    t = 1/(sqrt(a)*w);
    C = tf([K*a*t K], [t 1]);

    Ki = w/10;
    G = tf([1 Ki], [1 0]);

    H = tf((Ke*Ka*Kt),[Je Be 0]);   % continuous time
    hold on;

    % b
    GCH = G*C*H;
    bode(feedback(GCH,1));
end

function info = B3_z(w, phi, mag, Je, Be, name)
    Ke = 1.59;
    Ka = 1;
    Kt = .49;
    T = .0001;

    w = 2*pi*w; % hz -> rad/s
    phi = phi * pi/180; % deg -> rad
    K = 10^(-mag/20); % found that with K=1, mag at w is -22.4dB, so we need to shift by +22.4dB
    a = (1+sin(phi))/(1-sin(phi));
    t = 1/(sqrt(a)*w);
    C = tf([K*a*t K], [t 1]);

    Ki = w/10;
    G = tf([1 Ki], [1 0]);
```

```matlab
    clf;
    hold on;

    H = tf((Ke*Ka*Kt),[Je Be 0]);   % continuous time
    Hd = c2d(H,T,'zoh');                  % discrete time
    bode(feedback(Hd, 1));

    % a
    CHd = c2d(C, T, 'tustin')*Hd;
    bode(feedback(CHd, 1));

    % b
    GCHd = c2d(G*C, T, 'tustin')*Hd;
    bode(feedback(GCHd, 1));

    sys = feedback(GCHd, 1);
    bw = string(bandwidth(sys));
    ze = mat2str(zero(sys),2);
    po = mat2str(pole(sys), 2);

    title(['Close loop bode of ' name ' in z domain']);
    text1 = ['Yellow=With Integrator-lead-lag' 'bandwidth(rad/s)=' bw 'zero:' ze 'pole:' po];
    text2 = ['a=' strtrim(string(a)) 'T=' strtrim(string(t)) 'K=' string(K) 'Ki=' string(Ki)];
    annotation('textbox', [0.6, 0.4, 0.1, 0.1], 'String', text1);
    annotation('textbox', [0.7, 0.8, 0.1, 0.1], 'String', text2);
    saveas(gcf, ['qB3-z-' name '.png']);

    clf;
    sys = feedback(GCHd, 1);
    step(sys);
    S = stepinfo(sys);
    annotation('textbox', [0.6, 0.2, 0.1, 0.1], 'String', strucToStr(S));
    title(['Close loop bode of ' name ' in z domain, step response']);
    saveas(gcf, ['qB3-z-step-' name '.png']);

    info = {name, bw, ze, po, string(S.RiseTime), string(S.Overshoot)};
end


function info = B3_s(w, phi, mag, Je, Be, name)
    Ke = 1.59;
    Ka = 1;
    Kt = .49;
    T = .0001;

    w = 2*pi*w; % hz -> rad/s
    phi = phi * pi/180; % deg -> rad
    K = 10^(-mag/20); % found that with K=1, mag at w is -22.4dB, so we need to shift by +22.4dB
    a = (1+sin(phi))/(1-sin(phi));
    t = 1/(sqrt(a)*w);
    C = tf([K*a*t K], [t 1]);

    Ki = w/10;
    G = tf([1 Ki], [1 0]);

    H = tf((Ke*Ka*Kt),[Je Be 0]);   % continuous time
    clf;
    hold on;
    bode(feedback(H, 1));
    bode(feedback(C*H, 1));
    bode(feedback(G*C*H, 1));

    sys = feedback(G*C*H, 1);
    bw = string(bandwidth(sys));
    ze = mat2str(zero(sys),2);
    po = mat2str(pole(sys), 2);

    title(['Close loop bode of ' name ' in s domain']);
    text1 = ['Yellow=With Integrator-lead-lag' 'bandwidth(rad/s)=' string(bw) 'zero:' mat2str(ze,2) 'pole:' mat2str(po,2)];
```

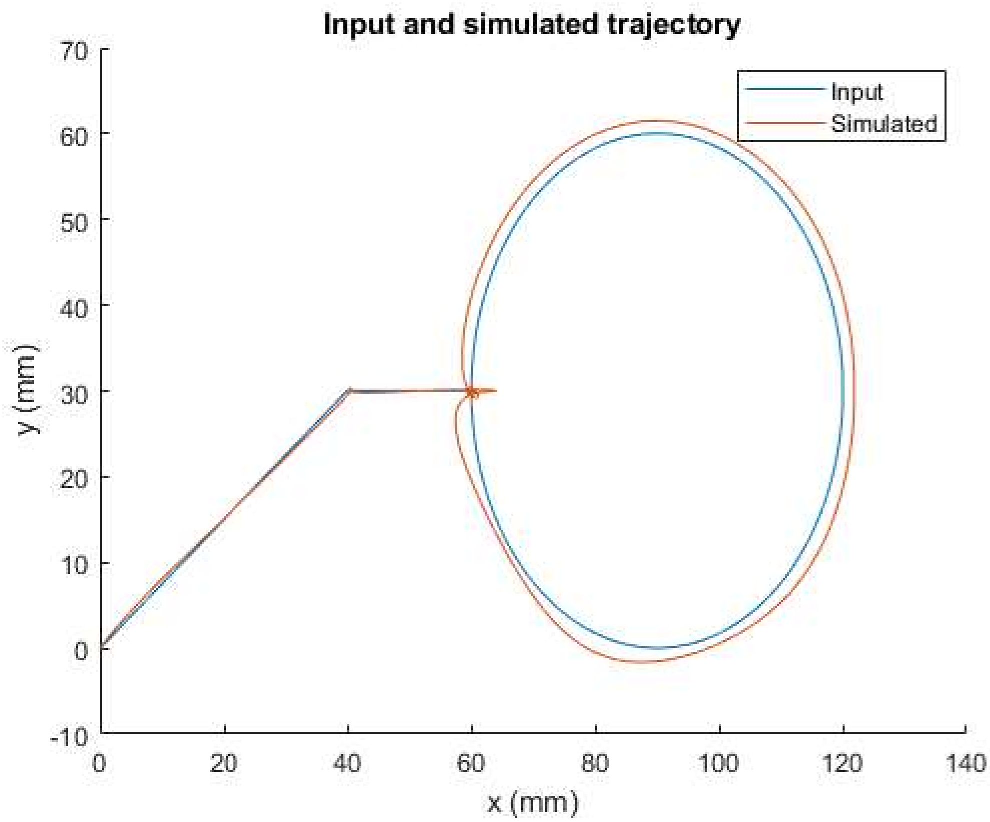
```
    text2 = ['a=' strtrim(string(a)) 'T=' strtrim(string(t)) 'K=' string(K) 'Ki=' string(Ki)];
    annotation('textbox', [0.6, 0.4, 0.1, 0.1], 'String', text1);
    annotation('textbox', [0.7, 0.8, 0.1, 0.1], 'String', text2);
    saveas(gcf, ['qB3-s-' name '.png']);

    clf;
    sys = feedback(G*C*H, 1);
    step(sys);
    S = stepinfo(sys);
    annotation('textbox', [0.6, 0.2, 0.1, 0.1], 'String', strucToStr(S));
    title(['Close loop bode of ' name ' in s domain, step response']);
    saveas(gcf, ['qB3-s-step-' name '.png']);

    info = {name, bw, ze, po, string(S.RiseTime), string(S.Overshoot)};
end

function str = strucToStr(struc)
    str = ['RiseTime(s): ' string(struc.RiseTime) 'Overshoot(%): ' string(struc.Overshoot)];
end
```

Published with MATLAB® R2020b

Contents

1

```
% init variables to be loaded into simulink model
T = 0.0001;
Ka = 1;
Kt = 0.49;
Ke = 1.59;
Jx = 0.000436;
Bx = 0.0094;
Jy = 0.0003;
By = 0.0091;

% use LBW LLI controller
a = 13.9282;
T_ = 0.0021323;
Kx = 0.75858;
Ky = 0.82224;
Ki = 12.5664;

LL = tf([a*T_ 1],[T_ 1]);
I = tf([1 Ki],[1 0]);
LLI_Lx_z = Kx*c2d(LL*I, T, 'tustin');
LLI_Ly_z = Ky*c2d(LL*I, T, 'tustin');

data = load('sampleTraj.mat');
Tplot = data.txy.t;
xplot = data.txy.x;
yplot = data.txy.y;

% run this in console after simulink is finished
% output = out.sim
% save simTraj output

% run after simulink sim finished
toPlot = load('simTraj.mat');
clf;
hold on;
title('Input and simulated trajectory');
plot(toPlot.output.Data(:,2), toPlot.output.Data(:,4));
plot(toPlot.output.Data(:,3), toPlot.output.Data(:,5));
xlabel('x (mm)');
ylabel('y (mm)');
legend('Input', 'Simulated');
saveas(gcf, 'qC1.png');
```



2

```
data = linearFromZero(5, 50);
ti = data(end, 1);
xi = data(end, 5);
yi = data(end, 6);
data = [data; circleCut(xi, yi, ti, 15)];
ti = data(end, 1);
xi = data(end, 5);
yi = data(end, 6);
data = [data; circleCut(xi, yi, ti, 5)];
traj.t = data(:,1);
traj.x = data(:,5);
traj.y = data(:,6);
save myTraj traj;
clf;
plot(data(:,5), data(:,6));
title('Custom toolpath');
xlabel('x (mm)');
ylabel('y (mm)');
saveas(gcf, 'qC2.png');

function data1 = linearFromZero(xf, yf)
    A = 250;
    T = 0.1;
    T = T * 0.001;

    xf = xf/2;
    yf = yf/2;

    L = ceil(sqrt(xf^2 + yf^2));
    [T1, T2, T3] = calc(L);
    T_total = T1+T2+T3;
```

```
    x_ratio = xf/L;
    y_ratio = yf/L;

    t = 0;
    s = 0;
    sdot = 0;
    sdotdot = A;
    xr = 0;
    yr = 0;
    vxr = 0;
    vyr = 0;
    axr = A*x_ratio;
    ayr = A*y_ratio;

    count = 1;
    data1 = zeros(T_total/T, 10);
    while t <= T_total
        data1(count, 1) = t;
        data1(count, 2) = s;
        data1(count, 3) = sdot;
        data1(count, 4) = sdotdot;
        data1(count, 5) = xr;
        data1(count, 6) = yr;
        data1(count, 7) = vxr;
        data1(count, 8) = vyr;
        data1(count, 9) = axr;
        data1(count, 10) = ayr;

        if t < T1
            sdotdot = A;
        elseif t >= T1 && t < T1+T2
            sdotdot = 0;
        else
            sdotdot = -A;
        end

        sdot = sdot + T*sdotdot;
        s = s + sdot*T;

        xr = s*x_ratio;
        yr = s*y_ratio;
        vxr = sdot*x_ratio;
        vyr = sdot*y_ratio;
        axr = sdotdot*x_ratio;
        ayr = sdotdot*y_ratio;

        t = t + T;
        count = count + 1;
    end
end

function data = circleCut(xi, yi, ti, r)
    A = 250;
    T = 0.1;
    T = T * 0.001;

    L = 2*3.1415*r;
    [T1, T2, T3] = calc(L);
    T_total = T1+T2+T3;
```

```
x_ratio = 1;
y_ratio = 0;

t = 0;
s = 0;
sdot = 0;
sdotdot = A;
xr = 0;
yr = 0;
vxr = 0;
vyr = 0;
axr = A*x_ratio;
ayr = A*y_ratio;

count = 1;
data1 = zeros(T_total/T, 10);
while t <= T_total
    if t < T1
        sdotdot = A;
    elseif t >= T1 && t < T1+T2
        sdotdot = 0;
    else
        sdotdot = -A;
    end

    sdot = sdot + T*sdotdot;
    s = s + sdot*T;

    [x_ratio, y_ratio] = getRatios(s/L);

    vxr_prev = vxr;
    vyr_prev = vyr;
    vxr = sdot*x_ratio;
    vyr = sdot*y_ratio;

    xr = xr + vxr*T;
    yr = yr + vyr*T;

    axr = (vxr - vxr_prev)/T;
    ayr = (vyr - vyr_prev)/T;

    data1(count, 1) = t+ti;
    data1(count, 2) = s;
    data1(count, 3) = sdot;
    data1(count, 4) = sdotdot;
    data1(count, 5) = xr+xi;
    data1(count, 6) = yr+yi;
    data1(count, 7) = vxr;
    data1(count, 8) = vyr;
    data1(count, 9) = axr;
    data1(count, 10) = ayr;

    t = t + T;
    count = count + 1;
end

[T1, T2, T3] = calc(r); % 60-40=20
T_total = T1+T2+T3;

x_ratio = 1;
y_ratio = 0;
```

```
    t = 0;
    s = 0;
    sdot = 0;
    sdotdot = A;
    xr = 0;
    yr = 0;
    vxr = 0;
    vyr = 0;
    axr = A*x_ratio;
    ayr = A*y_ratio;

    count = 1;
    data2 = zeros(T_total/T, 10);
    while t <= T_total
        data2(count, 1) = t+data1(end, 1);
        data2(count, 2) = s+data1(end, 2);
        data2(count, 3) = sdot+data1(end, 3);
        data2(count, 4) = sdotdot;
        data2(count, 5) = xr+data1(end, 5);
        data2(count, 6) = yr+data1(end, 6);
        data2(count, 7) = vxr+data1(end, 7);
        data2(count, 8) = vyr+data1(end, 8);
        data2(count, 9) = axr;
        data2(count, 10) = ayr;

        if t < T1
            sdotdot = A;
        elseif t >= T1 && t < T1+T2
            sdotdot = 0;
        else
            sdotdot = -A;
        end

        sdot = sdot + T*sdotdot;
        s = s + sdot*T;

        xr = s*x_ratio;
        yr = s*y_ratio;
        vxr = sdot*x_ratio;
        vyr = sdot*y_ratio;
        axr = sdotdot*x_ratio;
        ayr = sdotdot*y_ratio;

        t = t + T;
        count = count + 1;
    end

    data = [data1; data2];
end

function [x_r, y_r] = getRatios(r)
    x_r = 0;
    y_r = 0;
    if r < .25
        theta = (pi/2)*(r/.25);
        x_r = sin(theta);
        y_r = -cos(theta);
    elseif r >= .25 && r < 0.5
        theta = (pi/2)*((r-0.25)/.25);
        x_r = cos(theta);
```

```
        y_r = sin(theta);
    elseif r >= 0.5 && r < 0.75
        theta = (pi/2)*((r-0.5)/.25);
        x_r = -sin(theta);
        y_r = cos(theta);
    else
        theta = (pi/2)*((r-0.75)/.25);
        x_r = -cos(theta);
        y_r = -sin(theta);
    end
end

function [T1, T2, T3] = calc(L)
    A = 100; % mm/s^2
    fc = 250; % mm/s
    T = 0.1; % ms
    T = T * 0.001; % s

    T1 = fc/A;
    T3 = T1;
    s_init = A*T1*T1/2; % this is 20
    T2 = (L-2*s_init)/fc;
    T2 = ceil(T2/T)*T;
    if T2 < 0
        T2 = 0;
        s_init = L/2;
        T1 = sqrt(2*s_init/A);
        T1 = ceil(T1/T)*T;
        T3 = T1;
    end
end
```