[redacted]

1) a) $\boxed{G_{yr}(s) = \dfrac{C(s)P(s)}{1+C(s)P(s)}}$

$e = r - y$

$$G_{er}(s) = \frac{E(s)}{R(s)} = \frac{R(s)-Y(s)}{R(s)} = 1 - \frac{Y(s)}{R(s)} = 1 - G_{yr}(s) = \frac{1}{1+C(s)P(s)}$$

$$\boxed{G_{er}(s) = \frac{1}{1+C(s)P(s)}}$$

b) $R(s) = \frac{r}{s}$

$$Y(s) = G_{yr}(s)\,R(s) = \frac{C(s)P(s)}{1+C(s)P(s)} \cdot \frac{r}{s}$$

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} s \frac{C(s)P(s)}{1+C(s)P(s)} \frac{r}{s} = \lim_{s\to 0} \frac{C(s)P(s)}{1+C(s)P(s)} r$$

$$\lim_{s\to 0} C(s) = \infty$$

$$\lim_{s\to 0} \frac{C(s)P(s)}{1+C(s)P(s)} = 1$$

$$\therefore \lim_{t\to\infty} y(t) = 1 \cdot r = r$$

$$E(s) = G_{er}(s)\,R(s) = \frac{1}{1+C(s)P(s)} \frac{r}{s}$$

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+C(s)P(s)} \frac{r}{s} = \lim_{s\to 0} \frac{1}{1+C(s)P(s)} r = 0 \cdot r = 0$$

$$\therefore \lim_{t\to\infty} e(t) = 0$$

c) $C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}$

$$G_{yr}(s) = \frac{C(s)P(s)}{1+C(s)P(s)} = \frac{\left(\frac{k_d s^2 + k_p s + k_i}{s}\right)\left(\frac{b_0}{s^2+a_1 s+a_0}\right)}{1+\left(\frac{k_d s^2 + k_p s + k_i}{s}\right)\left(\frac{b_0}{s^2+a_1 s+a_0}\right)} = \frac{\left(\frac{b_0(k_d s^2 + k_p s + k_i)}{s(s^2+a_1 s+a_0)}\right)}{\left(\frac{s(s^2+a_1 s+a_0)+b_0(k_d s^2+k_p s+k_i)}{s(s^2+a_1 s+a_0)}\right)}$$

$$= \frac{b_0(k_d s^2 + k_p s + k_i)}{s(s^2+a_1 s+a_0)+b_0(k_d s^2+k_p s+k_i)} = \frac{b(s)}{a(s)}$$

$$a(s) = s^3 + (a_1 + b_0 k_d)s^2 + (a_0 + b_0 k_p)s + b_0 k_i$$

$$= (s-p_1)(s-p_2)(s-p_3) = s^3 + c_2 s^2 + c_1 s + c_0$$

$c_2 = a_1 + b_0 k_d \Rightarrow k_d = \frac{c_2 - a_1}{b_0}$

$c_1 = a_0 + b_0 k_p \Rightarrow k_p = \frac{c_1 - a_0}{b_0}$

$c_0 = b_0 k_i \Rightarrow k_i = \frac{c_0}{b_0}$

Since the poles can be arbitrarily selected, the constants $c_i$ will also be arbitrary based on the poles. Since the PID gains are based on $c_i$, the poles of $G_{yr}$ can be arbitrarily assigned by adjusting the PID gains.

d) $q(s) = (s+2)(s-(-1+i))(s-(-1-i))$

$$= (s+2)(s^2-(-1+i)s-(-1-i)s+(-1+i)(-1-i))$$

$$= (s+2)(s^2-(-s+is)-(-s-is)+1-i^2)$$

$$= (s+2)(s^2+s\cancel{-is}+s\cancel{+is}+2)$$

$$= (s+2)(s^2+2s+2)$$

$$= s^3+2s^2+2s^2+4s+2s+4$$

$$= s^3+4s^2+6s+4$$

$c_2 = 4 \quad c_1 = 6 \quad c_0 = 4$

$$k_i = \frac{c_0}{b_0} = \frac{4}{1} = 4 \Rightarrow \boxed{k_i = 4}$$

$$k_p = \frac{c_1 - a_0}{b_0} = \frac{6+2}{1} = 8 \Rightarrow \boxed{k_p = 8}$$

$$k_d = \frac{c_2 - a_1}{b_0} = \frac{4-2}{1} = 2 \Rightarrow \boxed{k_d = 2}$$

2) (I) $P_1(s) = \frac{1}{s(s+10)(s+20)}$

Open-loop poles: $0, -10, -20$

$\hookrightarrow$ NOT a, c

No open-loop zeros $\Rightarrow$ NOT d

(I) $\rightarrow$ b

(II) $P_2(s) = \frac{s^2+9}{s(s+10)(s+20)}$

NOT b

Open-loop poles: $0, -10, -20$

$\hookrightarrow$ NOT a, c

This system has open-loop zeros at $\pm 3i$

$\boxed{(II) \rightarrow d}$

(III) $P_3(s) = \frac{s^2+9}{s(s+10)}$

NOT b, d

Open-loop Poles: $0, -10$

Open-loop Zeros: $\pm 3i$

a has poles on imaginary axis but (III) has only real poles $\Rightarrow$ NOT a

$\boxed{(III) \rightarrow c}$

(IV) $P_4(s) = \frac{s^2+9}{s(s+10)(s^2+25)}$

Open-loop Poles: $0, -10, \pm 5i$

Open-loop zeros: $\pm 3i$

(IV) $\rightarrow$ a

3) $y_m = rF(s)$

$e = y_m - y$

$u = eC(s)$

$\mu = u + v$

$\eta = \mu P(s)$

$y = \eta + w$

$y = \eta + w = P\mu + w = P(u+v) + w = Pu + Pv + w$

$= PCe + Pv + w$

$= PC(y_m - y) + Pv + w$

$= PC(Fr - y) + Pv + w$

$= PCFr + Pv + w - PCy$

$y(1+PC) = PCFr + Pv + w$

$$\boxed{y = \frac{PCF}{1+PC} r + \frac{P}{1+PC} v + \frac{1}{1+PC} w}$$ ✓

$u = Ce = C(y_m - y) = C(rF - \eta - w) = C(Fr - P\mu - w) = C(Fr - P(u+v) - w)$

$= CFr - CPv - Cw - CPu$

$u(1+PC) = CFr - PCv - Cw$

$$\boxed{u = \frac{CF}{1+PC} r - \frac{PC}{1+PC} v - \frac{C}{1+PC} w}$$

$e = y_m - y = Fr - \eta - w = Fr - P\mu - w = Fr - P(u+v) - w = Fr - P(Ce+v) - w$

$e = Fr - Pv - w - PCe$

$e(1+PC) = Fr - Pv - w$

$$\boxed{e = \frac{F}{1+PC} r - \frac{P}{1+PC} v - \frac{1}{1+PC} w}$$

$\mu = u + v = Ce + v = C(y_m - y) + v = C(Fr - \eta - w) + v = C(Fr - P\mu - w) + v$

$\mu = CFr - Cw + v - PC\mu$

$\mu(1+PC) = CFr - Cw + v$

$$\boxed{\mu = \frac{CF}{1+PC} r + \frac{1}{1+PC} v - \frac{C}{1+PC} w}$$

$\eta = P\mu = Pu + Pv = PCe + Pv = PC(y_m - y) + Pv = PC(Fr - \eta - w) + PV$

$\eta = PCFr + Pv - PCw - PC\eta$

$\eta(1+PC) = PCFr + Pv - PCw$

$$\boxed{\eta = \frac{PCF}{1+PC} r + \frac{P}{1+PC} v - \frac{PC}{1+PC} w}$$

4) a) $m\frac{dv}{dt} = -cv + b\tau + F_{hill} \xrightarrow{s} msY(s) = -cY(s) + b\tau(s) + F_{hill}$

$$Y(s)[ms+c] = b\tau(s) + F_{hill}$$

$$Y(s) = \frac{b\tau(s)}{ms+c} + d \Rightarrow \text{total disturbance from hill}$$

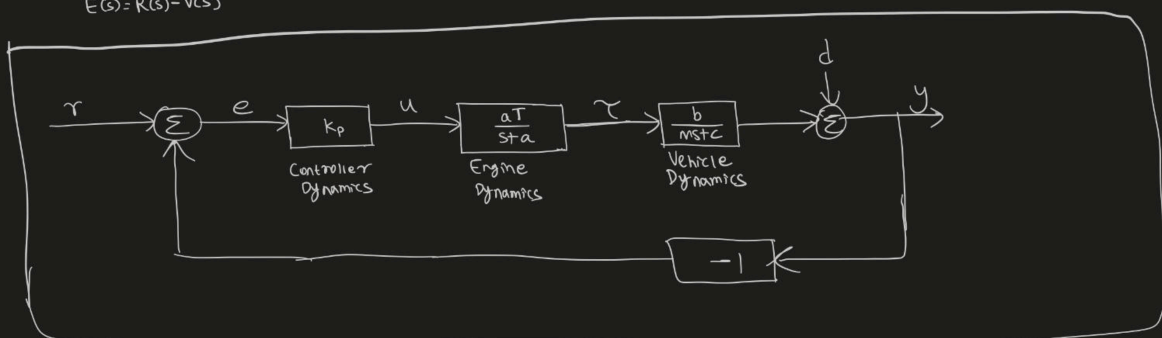
$$\frac{d\tau}{dt} = a(-\tau + Tu) \xrightarrow{s} s\tau(s) = -a\tau(s) + aTU(s)$$

$$[s+a]\tau(s) = aTu(s)$$

$$\tau(s) = \frac{aT}{s+a}U(s)$$

$U(s) = k_pE(s)$

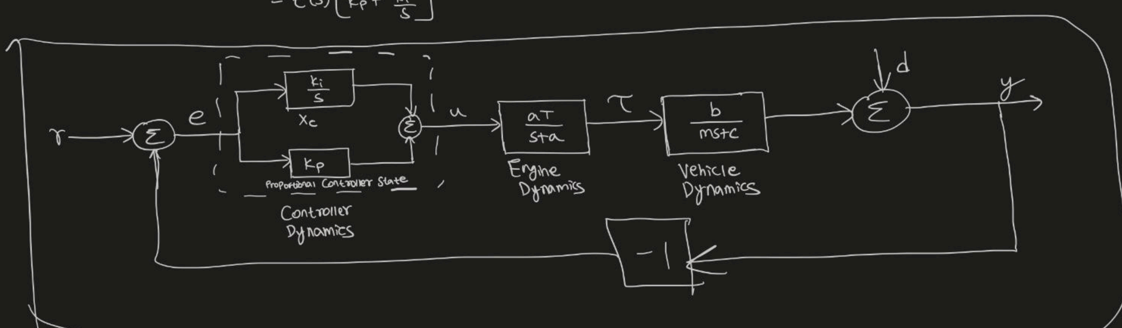
$E(s) = R(s) - V(s)$




c) $\frac{dx_c}{dt} = r - v \quad u = k_pe + k_ix_c$

$sX_c(s) = R(s) - V(s)$

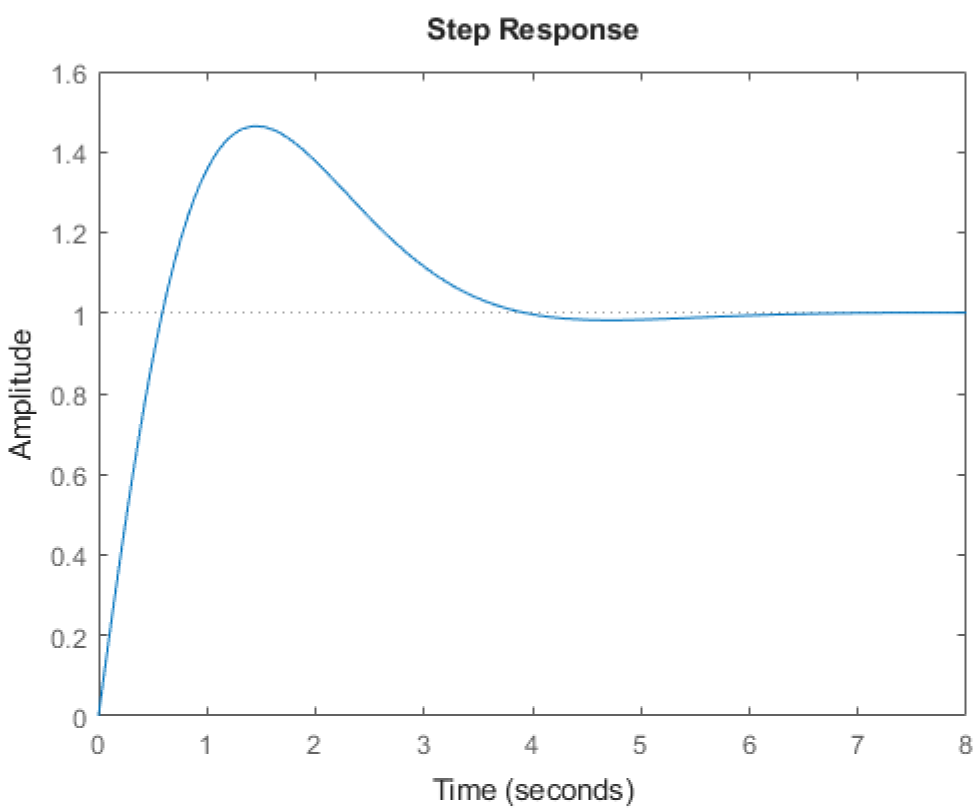
$$U(s) = k_pE(s) + k_iX_c(s)$$
$$= k_pE(s) + k_i\left(\frac{1}{s}[R(s) - V(s)]\right)$$
$$= E(s)\left[k_p + \frac{k_i}{s}\right]$$

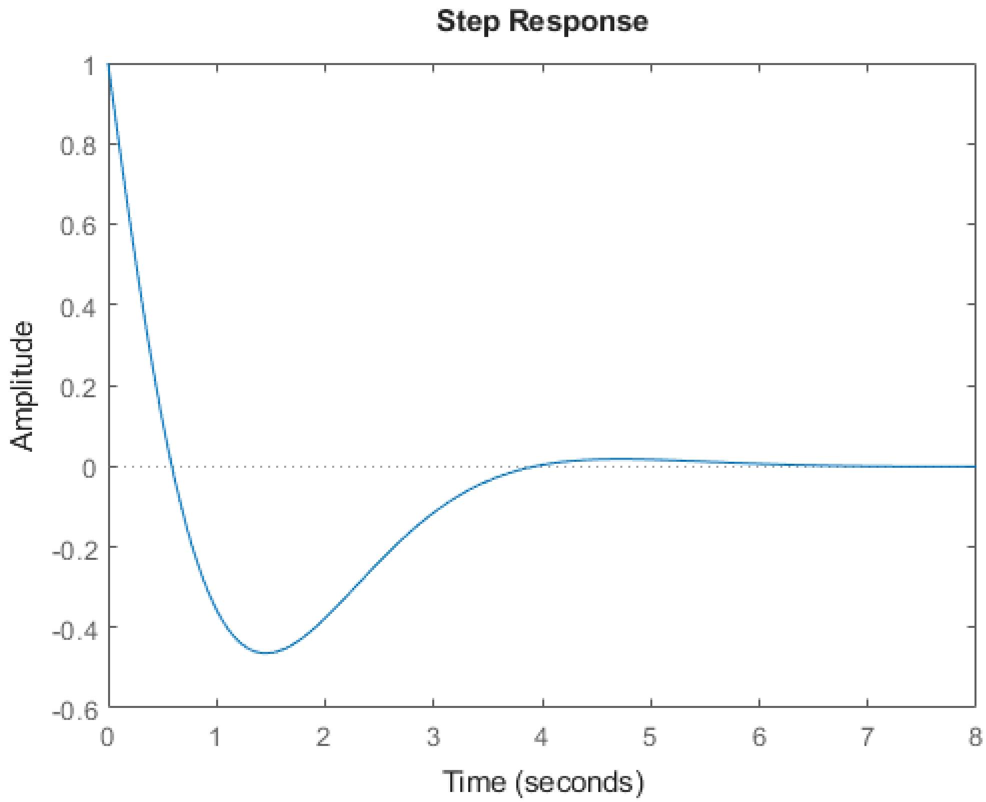



Questions 1d, 4b, 4d, 4e, and 4f are answered in the sections below.

# 1d)

```
b0 = 1;
a1 = 2;
a0 = -2;
ki = 4;
kp = 8;
kd = 2;
numP = b0;
denP = [1 a1 a0];
P = tf(numP, denP);
numC = [kd kp ki];
denC = [1 0];
C = tf(numC, denC);
Gyr = feedback(P*C, 1);
figure;
step(Gyr);
Ger = 1-Gyr;
figure;
step(Ger);

%The response of y(t) reaches the reference in its steady state and the
%error response reaches 0 at steady state.
```






*Published with MATLAB® R2024b*

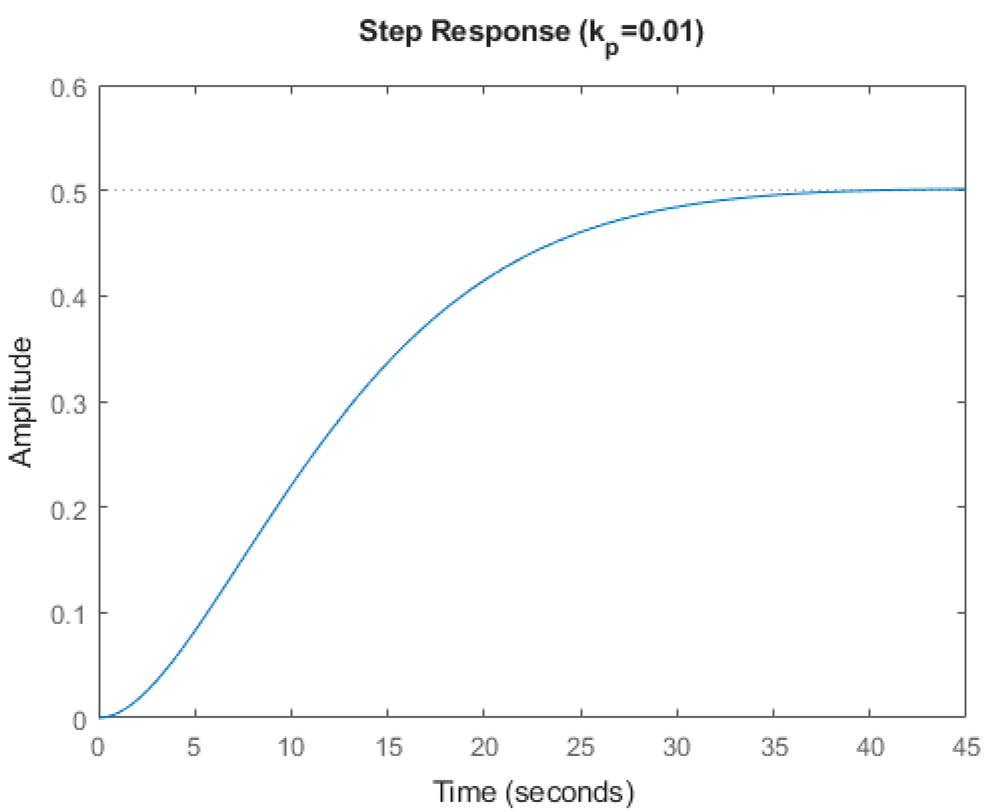
```
% -------------------------------------------------------------------------
%
%  ECE171A: HW7 Problem 4  -- sample code
%  Writen by Yang Zheng
%
% -------------------------------------------------------------------------

% Note: you can also simulate this problem using ODE
%       as what you have done in previous assignments.
%       For LTI systems, we can do the simulations using
%       transfer functions which are easier for coding.

close all
```

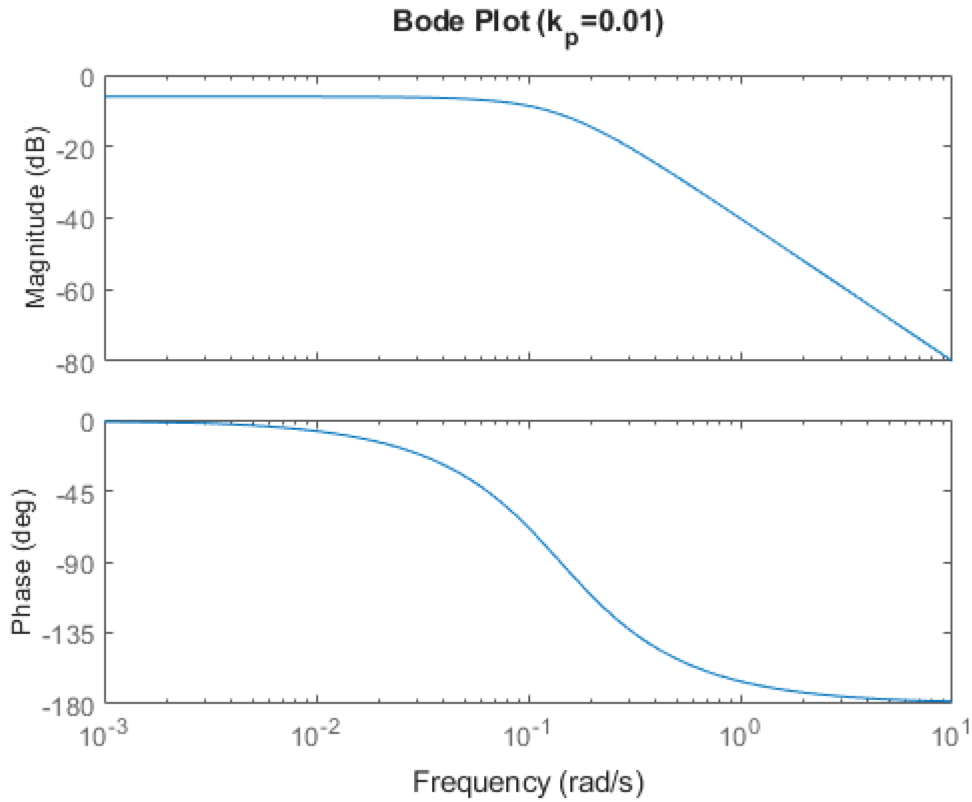
# 4b)

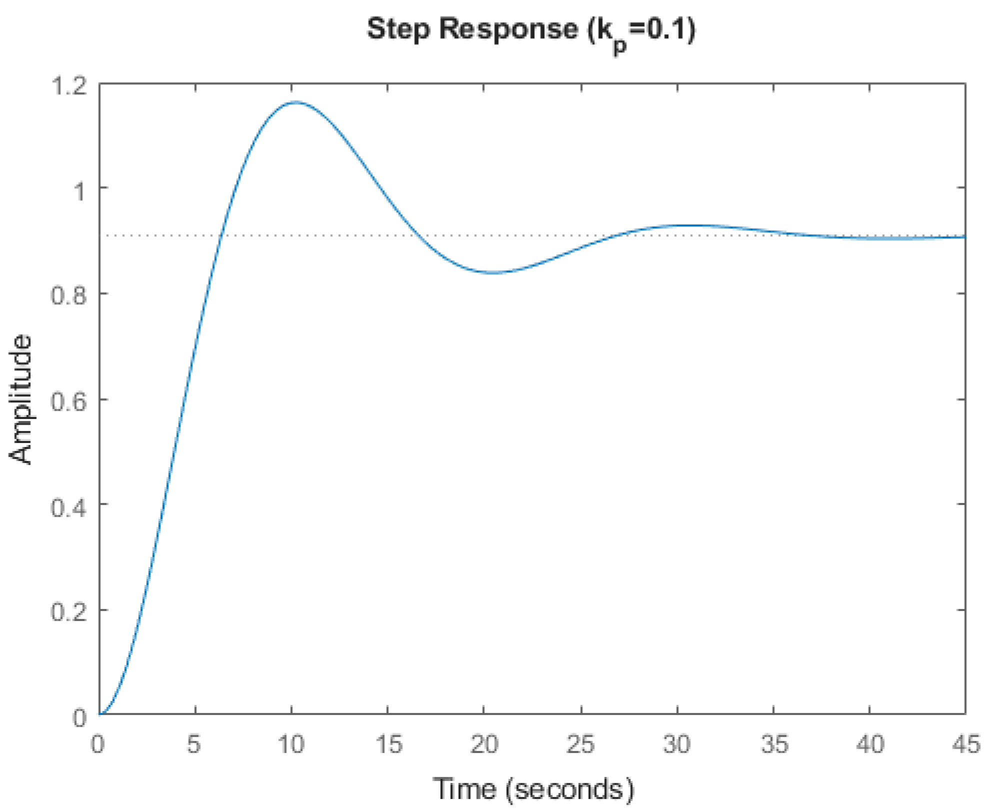
System parameters

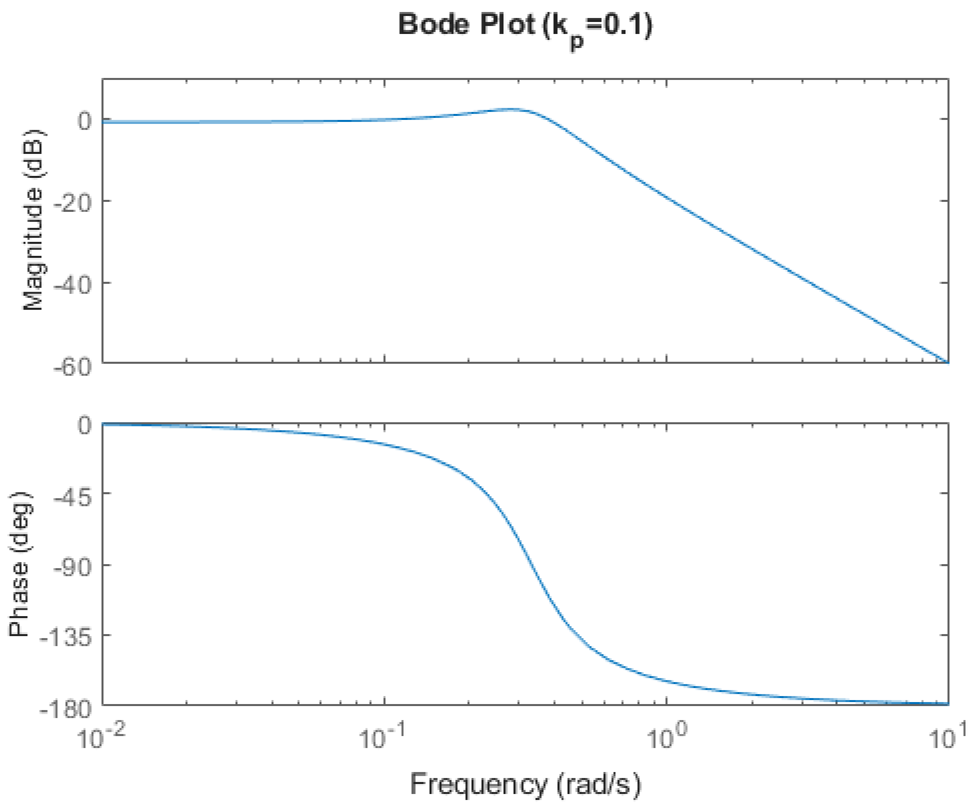
```
m = 1000;    % mass
c = 50;      % damping coefficient
b = 25;      % conversion factor
a = 0.2;     % lag coefficient
T = 200;     % another conversion factor

% Dynamics
s = tf('s');

P1 = b/(m*s+c);  % transfer function for tau to v
P2 = a*T/(s+a);  % transfer function for u to tau

%Case 1: kp=0.01:
% P controller
kp  = 0.01;
ki  = 0; %0.005;
C   = kp + ki/s;

L   = C*P2*P1;          % Loop transfer function
Gyr1 = feedback(L,1);   % closed transfer function from r to y

% step response
figure;
step(Gyr1);
title('Step Response (k_p=0.01)');

% freqency response
figure;
bode(Gyr1);
title('Bode Plot (k_p=0.01)');


%Case 2: kp=0.1:
% P controller
```

```
kp   = 0.1;
ki   = 0; %0.005;
C    = kp + ki/s;

L    = C*P2*P1;          % Loop transfer function
Gyr2 = feedback(L,1);    % closed transfer function from r to y

% step response
figure;
step(Gyr2);
title('Step Response (k_p=0.1)');


% freqency response
figure;
bode(Gyr2);
title('Bode Plot (k_p=0.1)');

% The case with kp = 0.01 results in a step response that reaches steady
% state without as many oscillations as the case with kp = 0.1. For the
% frequency reponses, both cases have a phase decrease from 0 to -180
% degrees and overall, the shapes of both the magnitude and frequency
% responses are the same. However, when kp has a higher value, the
% magnitude response has more of a peak like shape at the crossover
% frequency before dipping and the phase response has a more obvious
% dip shape at this frequency.
```




Bode Plot (kp=0.01)
Magnitude (dB)
0
-20
-40
-60
-80
Phase (deg)
0
-45
-90
-135
-180
10-3
10-2
10-1
100
101
Frequency (rad/s)


Step Response (kp=0.1)
Amplitude
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
35
40
45
Time (seconds)



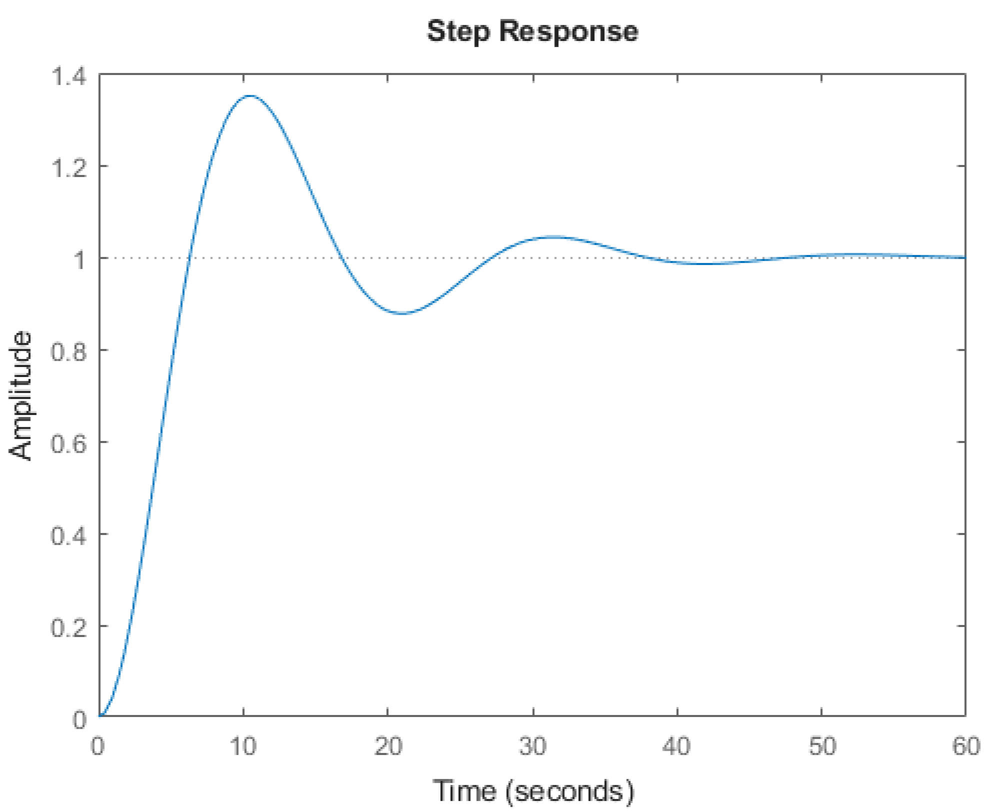

# 4d)

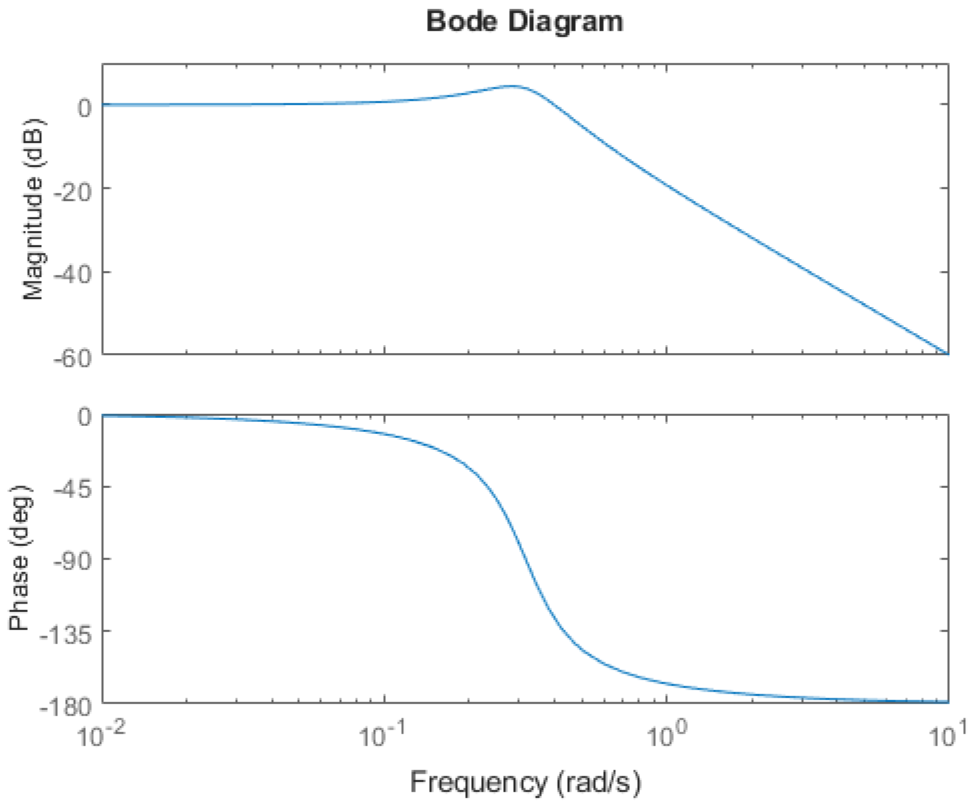
Dynamics

```
s = tf('s');

P1 = b/(m*s+c);  % transfer function for tau to v
P2 = a*T/(s+a);  % transfer function for u to tau

% PI controller
kp  = 0.1;
ki  = 0.005;
C   = kp + ki/s;

L   = C*P2*P1;          % Loop transfer function
Gyr3 = feedback(L,1);   % closed transfer function from r to y
disp(Gyr3);
% step response
figure; step(Gyr3)

% freqency response
figure; bode(Gyr3)

%When integral control is added, the step response eventually reaches the
%steady state value instead of just oscillating around that region of the
%graph as it did in proportional control. At lower frequencies, the value
```

```
%of the magnitude response is larger (only a little larger in this case due
%to the small value of ki). Also, for both the magnitude and phase
%responses, the crossover frequency increases resulting in a wider
%magnitude and phase plot.
```

```
  tf with properties:

       Numerator: {[0 0 100 5]}
     Denominator: {[1000 250 110 5]}
        Variable: 's'
         IODelay: 0
      InputDelay: 0
     OutputDelay: 0
       InputName: {''}
       InputUnit: {''}
      InputGroup: [1×1 struct]
      OutputName: {''}
      OutputUnit: {''}
     OutputGroup: [1×1 struct]
           Notes: [0×1 string]
        UserData: []
            Name: ''
              Ts: 0
        TimeUnit: 'seconds'
    SamplingGrid: [1×1 struct]
```





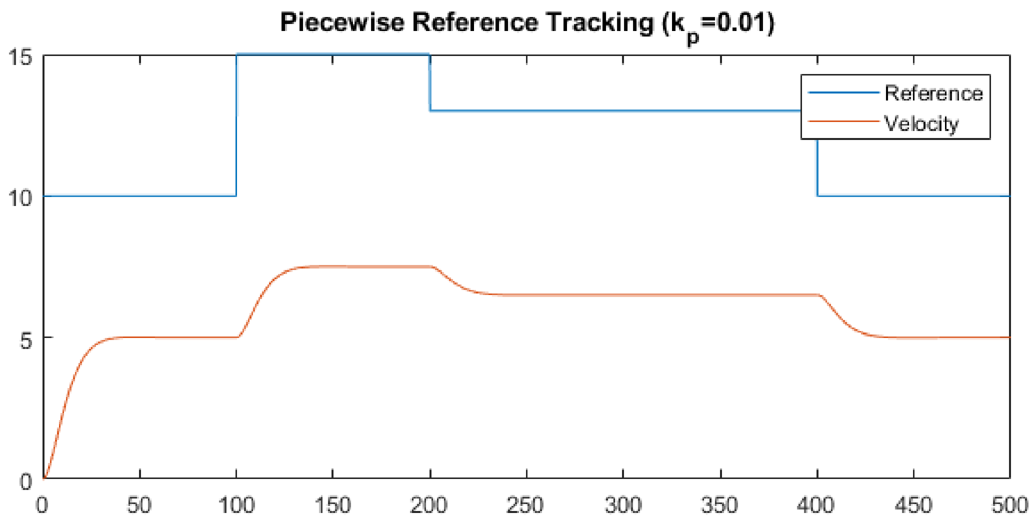

# 4e)

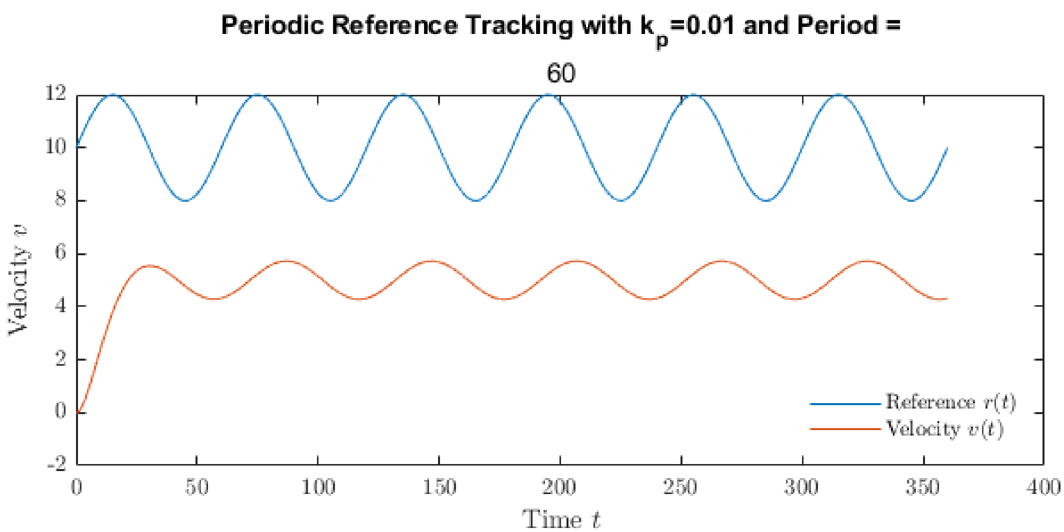
```
%Case 1: kp = 0.01:
% ----------------------------------------------
%          Time domain simulation
% ----------------------------------------------

% Case 1: peice-wise constant reference
t = 0:0.01:500;
r = zeros(length(t),1);
for i = 1:length(t)
    if t(i) <= 100
        r(i) = 10;
    elseif t(i) <= 200
        r(i) = 15;
    elseif t(i) <= 400
        r(i) = 13;
    else
        r(i) = 10;
    end
end
y = lsim(Gyr1,r,t,0);
figure; plot(t,r,t,y);
title('Piecewise Reference Tracking (k_p=0.01)');
legend('Reference', 'Velocity');
```

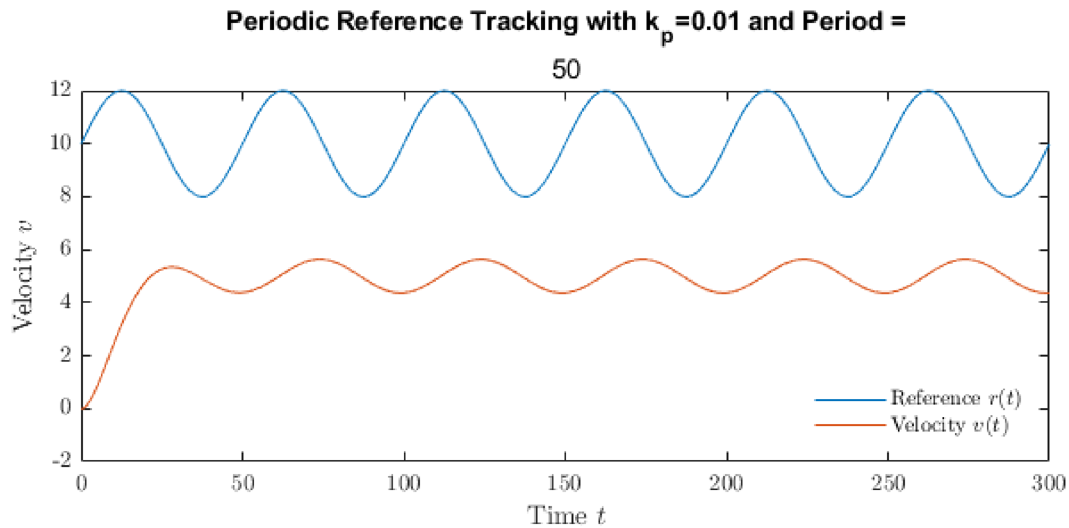
```
set(gcf,'Position',[100 100 700 300])

% Case 2: periodic references
for period = [60, 50, 40, 20]
    t = 0:0.01:6*period;        % time range -- 6 periods
    r = 10 + 2*sin(2*pi/period.*t);
    y = lsim(Gyr1,r,t,0);
    figure; plot(t,r,t,y);
    title('Periodic Reference Tracking with k_p=0.01 and Period =',
num2str(period));
    set(gcf,'Position',[100 100 700 300])
    h = legend('Reference $r(t)$','Velocity $v(t)$',...
        'location','southeast','Interpreter','latex');
    set(h,'box','off');
    xlabel('Time $t$','Interpreter','latex');
    ylabel('Velocity $v$','Interpreter','latex');
    set(gca,'TickLabelInterpreter','latex','fontsize',10);
end


%Case 2: kp = 0.1:
% ----------------------------------------------
%         Time domain simulation
% ----------------------------------------------

% Case 1: peice-wise constant reference
t = 0:0.01:500;
r = zeros(length(t),1);
for i = 1:length(t)
    if t(i) <= 100
        r(i) = 10;
    elseif t(i) <= 200
        r(i) = 15;
    elseif t(i) <= 400
        r(i) = 13;
    else
        r(i) = 10;
    end
end
y = lsim(Gyr2,r,t,0);
figure; plot(t,r,t,y);
title('Piecewise Reference Tracking (k_p=0.1)');
set(gcf,'Position',[100 100 700 300])

% Case 2: periodic references
for period = [60, 50, 40, 20]
    t = 0:0.01:6*period;        % time range -- 6 periods
    r = 10 + 2*sin(2*pi/period.*t);
    y = lsim(Gyr2,r,t,0);
    figure; plot(t,r,t,y);

    set(gcf,'Position',[100 100 700 300])
    h = legend('Reference $r(t)$','Velocity $v(t)$',...
        'location','southeast','Interpreter','latex');
```

```
    set(h,'box','off');
    xlabel('Time $t$','Interpreter','latex');
    ylabel('Velocity $v$','Interpreter','latex');
    title('Periodic Reference Tracking with k_p=0.1 and Period =',
num2str(period));
    set(gca,'TickLabelInterpreter','latex','fontsize',10);
end


%Case 3: PI Control
% ----------------------------------------------
%         Time domain simulation
% ----------------------------------------------

% Case 1: peice-wise constant reference
t = 0:0.01:500;
r = zeros(length(t),1);
for i = 1:length(t)
    if t(i) <= 100
        r(i) = 10;
    elseif t(i) <= 200
        r(i) = 15;
    elseif t(i) <= 400
        r(i) = 13;
    else
        r(i) = 10;
    end
end
y = lsim(Gyr3,r,t,0);
figure; plot(t,r,t,y);
title('Piecewise Reference Tracking (PI Control)');
set(gcf,'Position',[100 100 700 300])

% Case 2: periodic references
for period = [60, 50, 40, 20]
    t = 0:0.01:6*period;        % time range -- 6 periods
    r = 10 + 2*sin(2*pi/period.*t);
    y = lsim(Gyr3,r,t,0);
    figure; plot(t,r,t,y);

    set(gcf,'Position',[100 100 700 300])
    h = legend('Reference $r(t)$','Velocity $v(t)$',...
        'location','southeast','Interpreter','latex');
    set(h,'box','off');
    xlabel('Time $t$','Interpreter','latex');
    ylabel('Velocity $v$','Interpreter','latex');
    title('Periodic Reference Tracking with PI Control and Period =',
num2str(period));
    set(gca,'TickLabelInterpreter','latex','fontsize',10);
end


%Proportional control with a small kp does not accurately track the
%reference. The output velocity is much further away than the reference
```

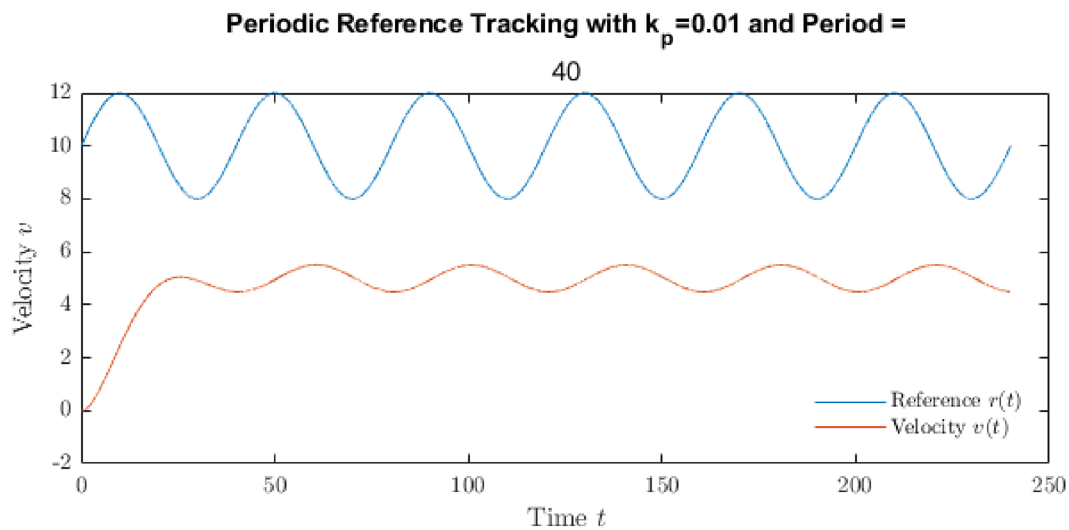
```
%velocity. When the proportional constant is increased, the controller does
%a better job at tracking the reference with the velocity values being
%closer to the reference. However, this case isn't nearly as accurate the
%PI controller case. Adding integral control allows the controller to track
%the reference much more accurately and at some points, almost exactly.
```



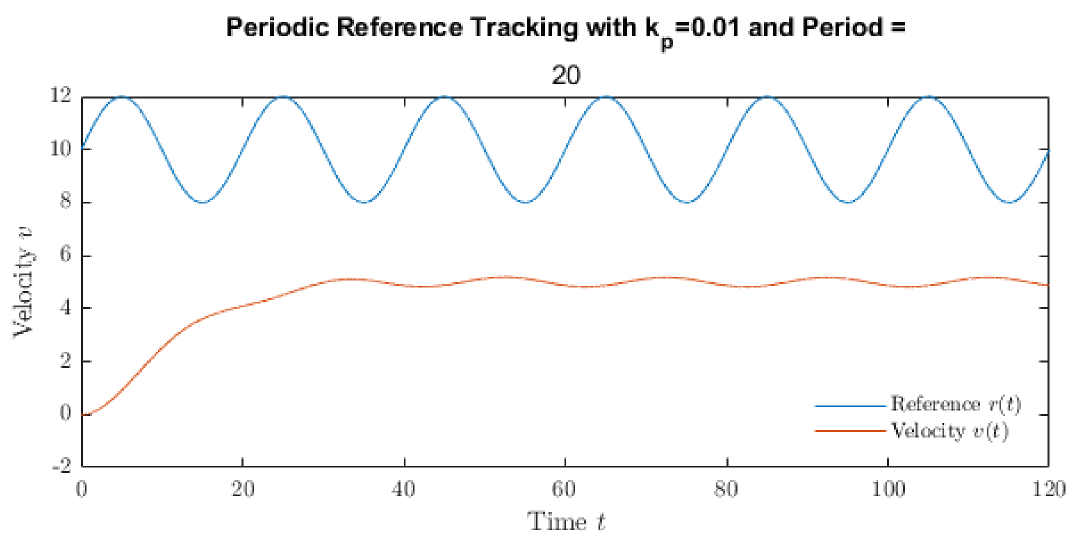





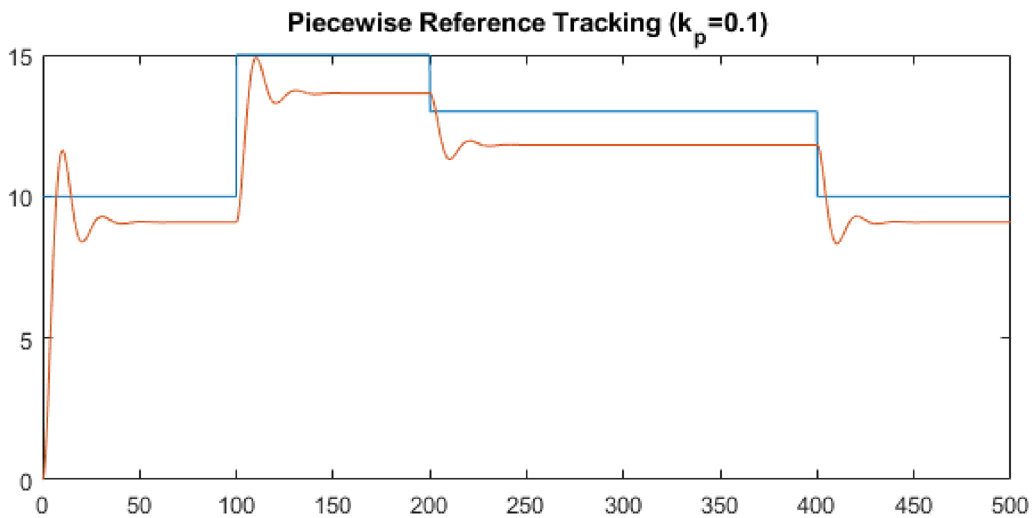
Periodic Reference Tracking with k_p=0.01 and Period = 50
Velocity v
Time t
Reference r(t)
Velocity v(t)

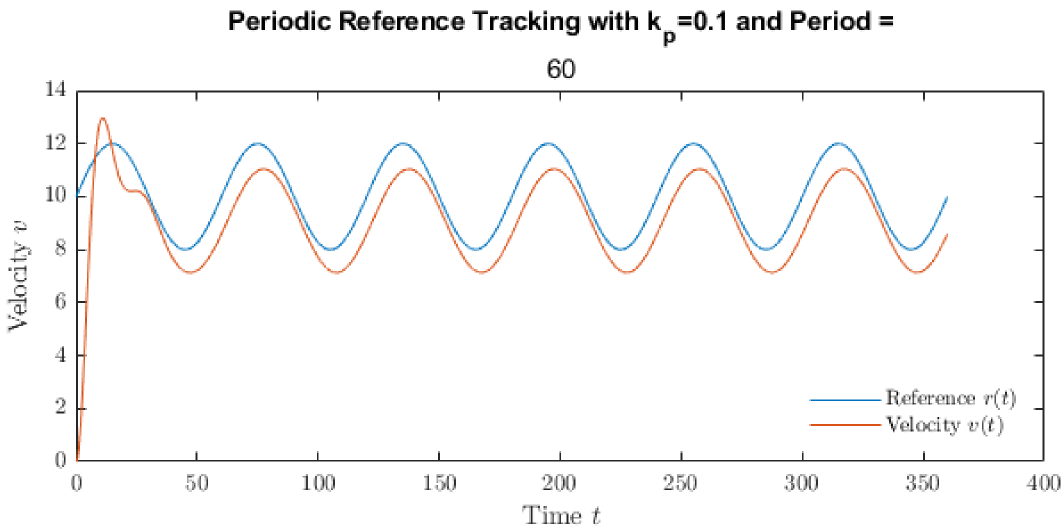
Periodic Reference Tracking with k_p=0.01 and Period = 40
Velocity v
Time t
Reference r(t)
Velocity v(t)

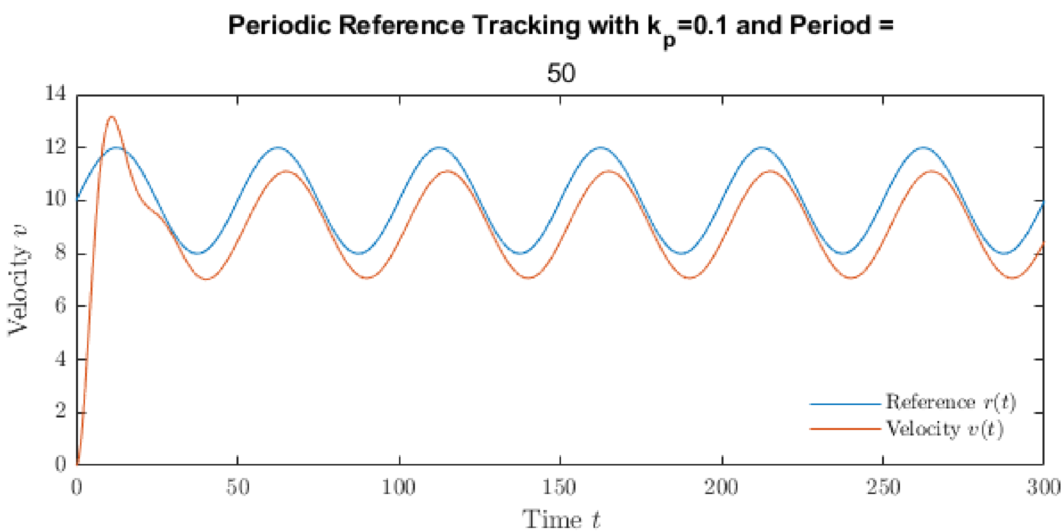
Periodic Reference Tracking with k_p=0.01 and Period = 20
Velocity v
Time t
Reference r(t)
Velocity v(t)



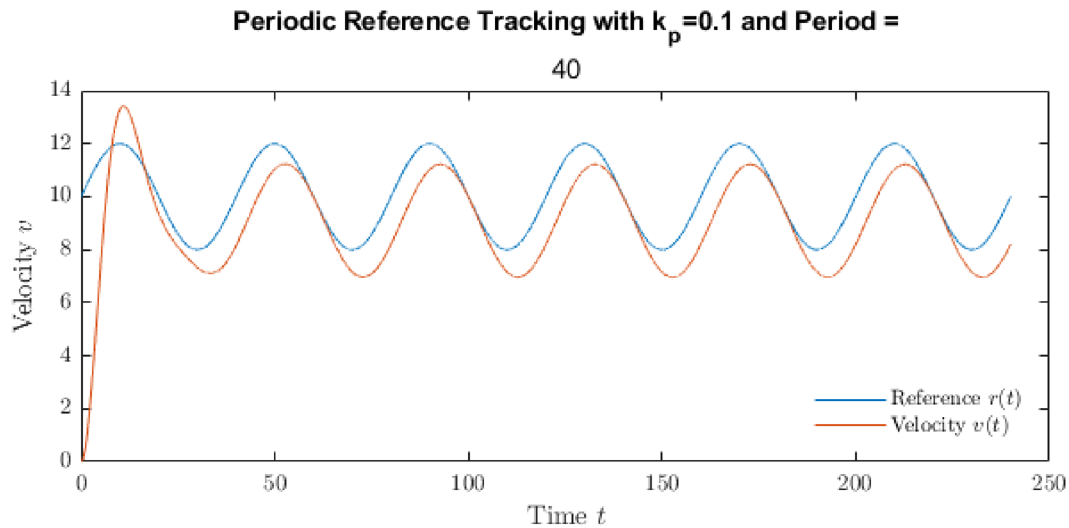
Piecewise Reference Tracking ($k_p$=0.1)

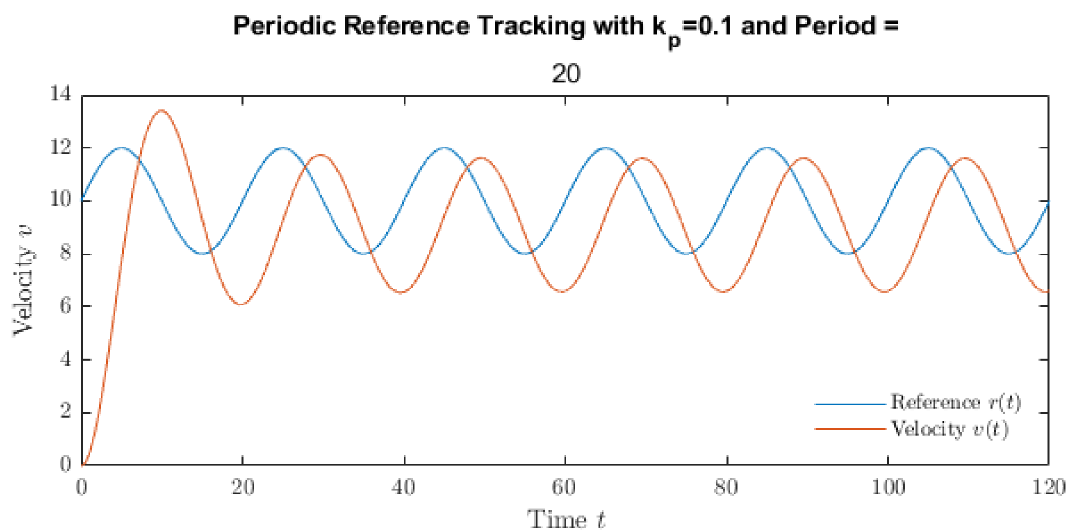
Periodic Reference Tracking with $k_p$=0.1 and Period = 60
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$

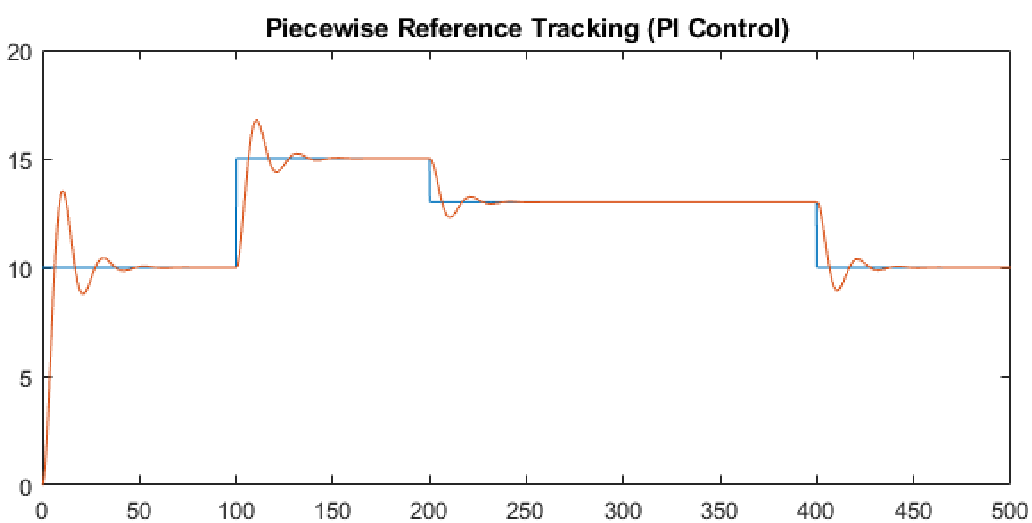
Periodic Reference Tracking with $k_p$=0.1 and Period = 50
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$



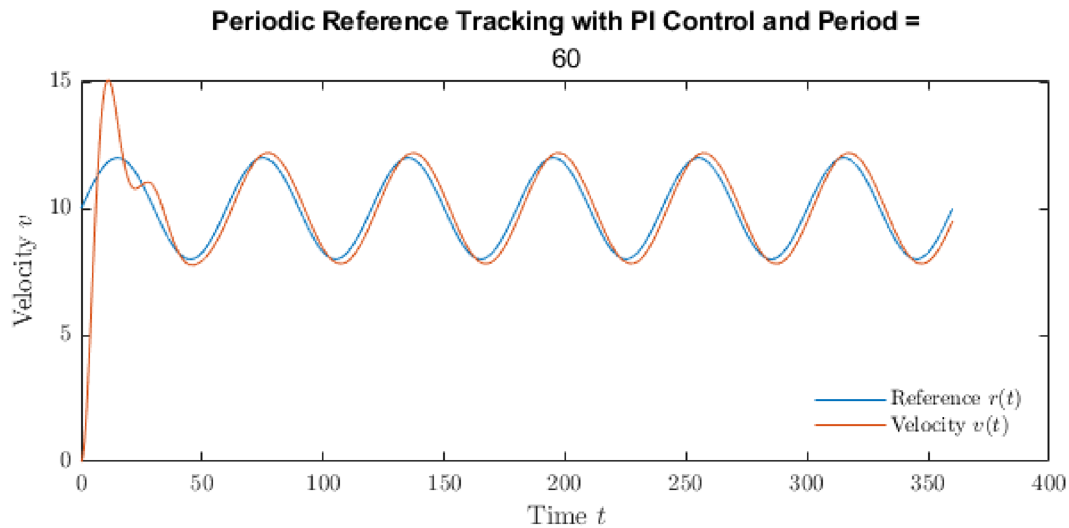
Periodic Reference Tracking with $k_p$=0.1 and Period = 40
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$

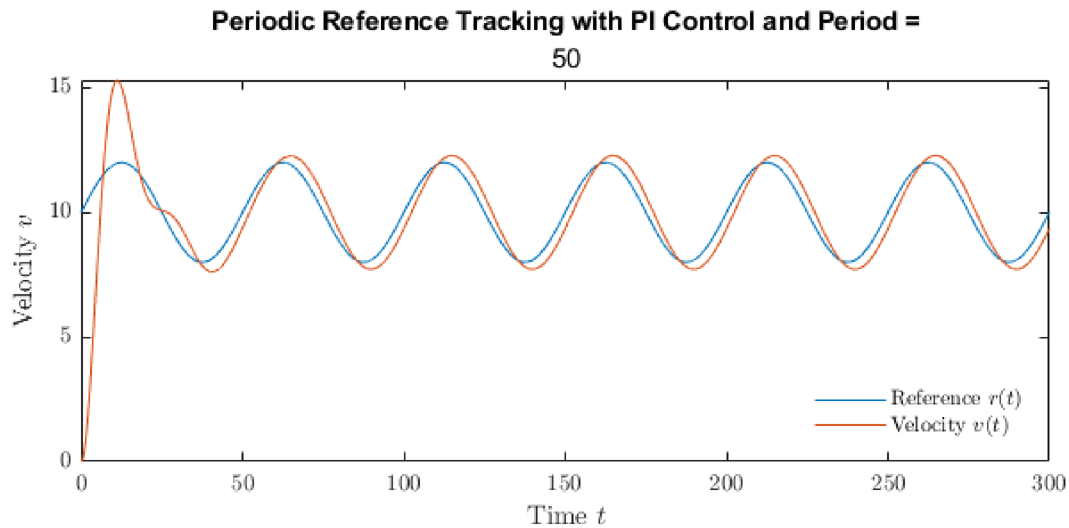
Periodic Reference Tracking with $k_p$=0.1 and Period = 20
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$

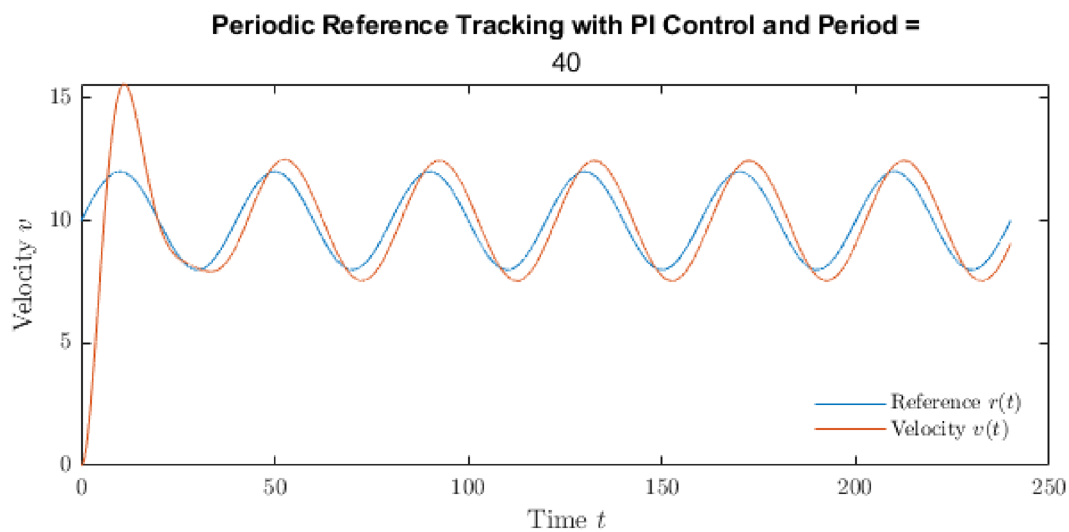
Piecewise Reference Tracking (PI Control)



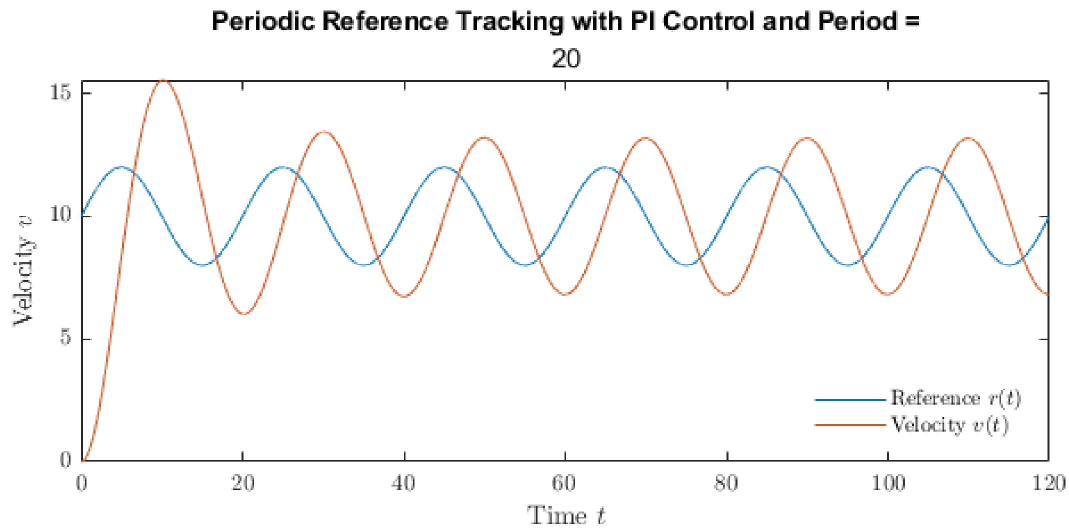
Periodic Reference Tracking with PI Control and Period =
60
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$

Periodic Reference Tracking with PI Control and Period =
50
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$

Periodic Reference Tracking with PI Control and Period =
40
Velocity $v$
Time $t$
Reference $r(t)$
Velocity $v(t)$




# 4f)

```
%To ensure safety in this system, it is important to have accurate
%reference tracking. For this, the kp and ki values must be properly
%adjusted. After experimenting with this simulation, it can be seen that
%proportional control is accurate for higher values of kp. However, one
%disadvantage of having higher values is the damping that will occur
%immediately after activating the system. In cruise control, this high
%damping will result in reference tracking errors that could lead to an
%accident. In order to get more accurate reference tracking with smaller
%proportional constants, it helps to have integral control. However, with
%integral control, higher constants will cause much more damping than lower
%constants, and it is much more sensitive to parameter changes than a
%proportional controller, so it's important to keep ki low. The best
%controller is one with a kp value high enough to avoid damping while also
%being able to perform reference tracking accurately and a ki value that is
%low enough to do the same. Finding this will require an optimization
%algorithm that finds the constants that result in the lowest amount of
%damping(possibly using peak difference).
```

*Published with MATLAB® R2024b*