
```

%Ramaseshan Parthasarathy
%RUID: 157006637
%Lab 4

%Note that in this lab the axes are a bit off, but the plots are
otherwise
%correct.

% The purpose of this lab is to discuss tracking errors and
disturbance by
% creating several plots for comparison. Each involved creating of two
% transfer functions H_err and H_dist.

%Part a constructs tf objects for the system -- Gs, Gc(s), H(s),
H_err(s),
%H_dist(s)
%Part b generates input signals.
% Part c implements PID controller slightly modified.
% The last part we investigate how controlled system respond to
% disturbance.

%part a
u = @(t) double(t >= 0);
t = 0:(20/1000):20;

a = 2; kp = 10; ki = 5; kd = 3;
s = tf('s'); G = 1/(s*(s+a));
Gc = kp + ki/s + kd*s;
H = minreal((Gc * G)/(1 + Gc*G));
H_err = minreal(1/(1 + Gc*G));
H_dist = minreal(G/(1 + Gc*G));

p = roots(H.den{1});
t_const = log(100)/abs(real(p(3)))

y = lsim(H, u(t), t);
figure; plot(t, y, 'b'); grid on;
title('step response, kp = 10, ki = 5, kd = 3');

a = 2; kp = 20; ki = 5; kd = 3;
s = tf('s');
G = 1/(s * (s+a));
Gc = kp + ki/s + kd*s;
H = minreal((Gc*G)/(1 + Gc*G));

y = lsim(H,u(t),t);
figure; plot(t, y, 'b'); grid on;
title('step response, kp = 20, ki = 5, kd = 3');

a = 2; kp = 10; ki = 10; kd = 3;
s = tf('s');
G = 1/(s*(s+a));

```

```

Gc = kp + ki/s + kd*s;
H = minreal((Gc*G)/(1 + Gc*G));

y = lsim(H, u(t), t);
figure; plot(t, y, 'b'); grid on;
title('step response, kp = 10, ki = 10, kd = 3');

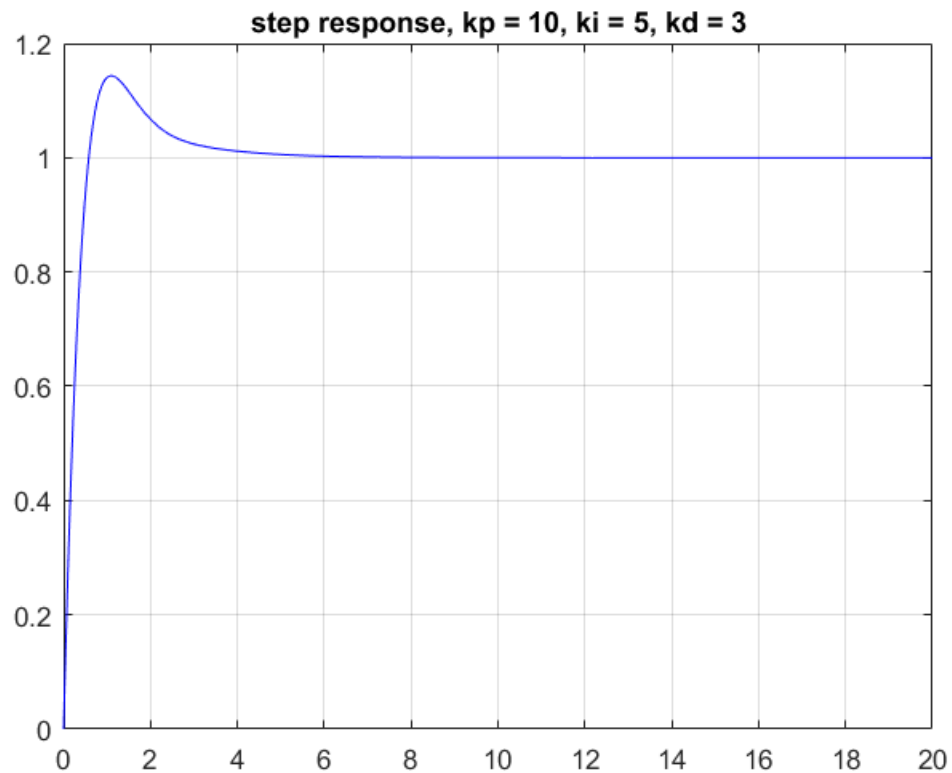
a = 2; kp = 10; ki = 5; kd = 6;
s = tf('s');
G = 1/(s*(s+a));
Gc = kp + ki/s + kd*s;
H = minreal(Gc*G/(1+Gc*G));

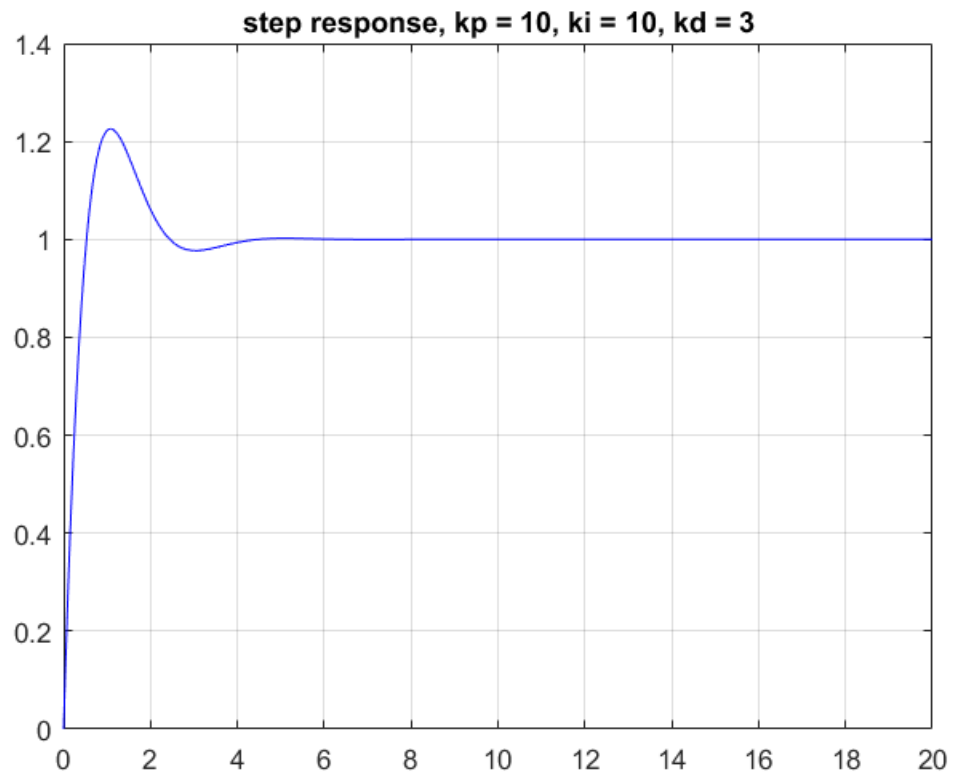
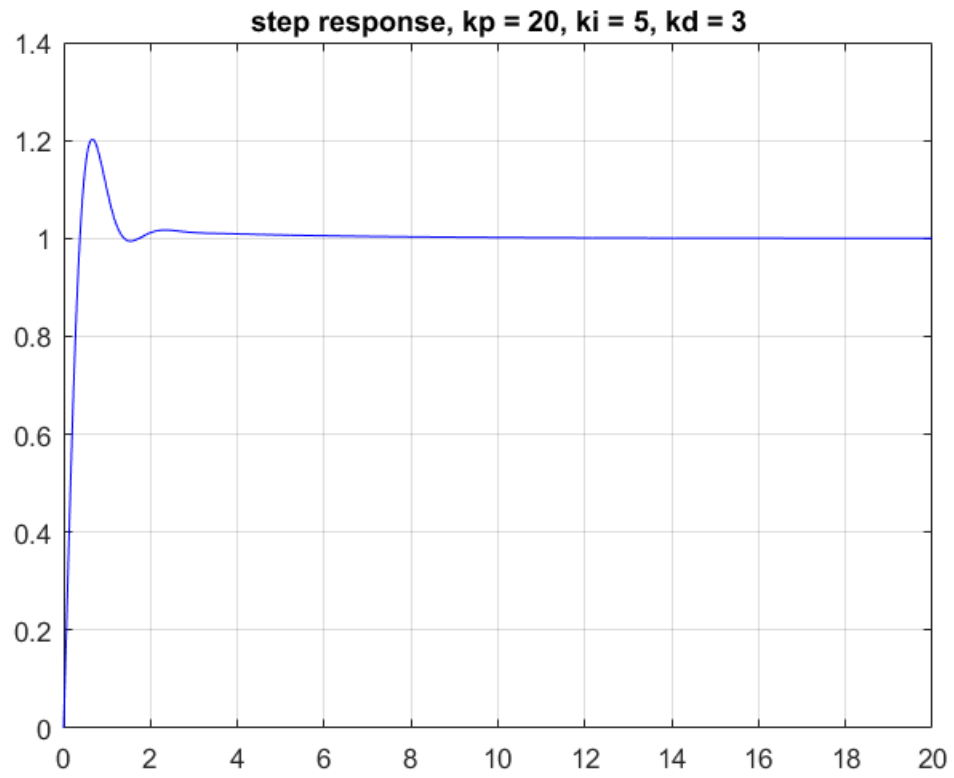
y = lsim(H, u(t), t);
figure; plot(t, y, 'b'); grid on;
title('step response, kp = 10, ki = 5, kd = 6');

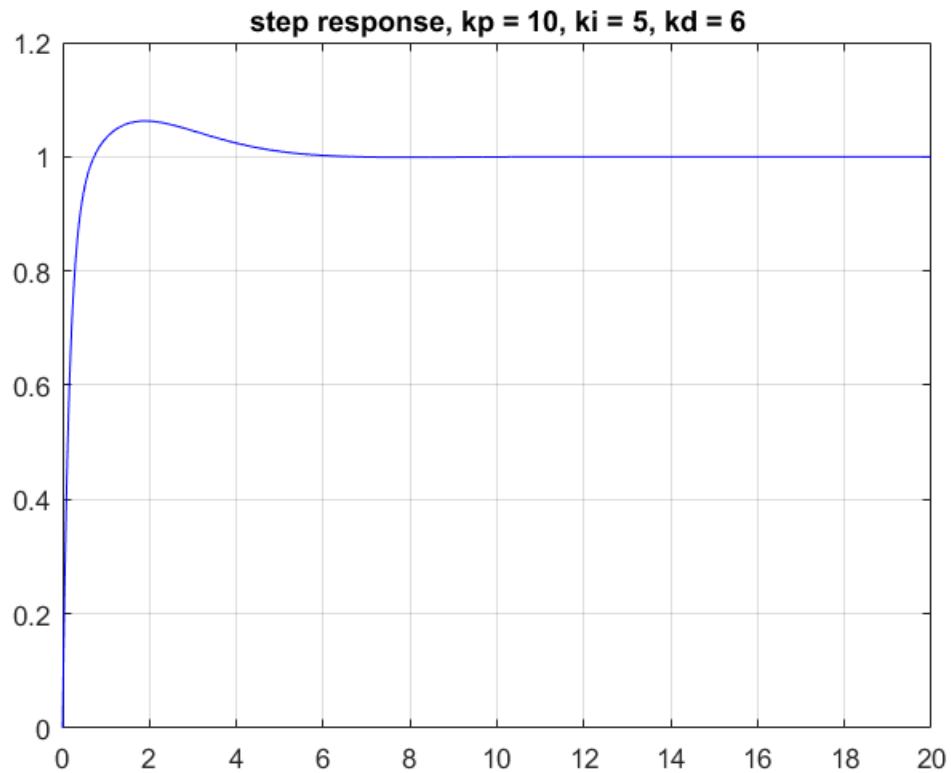
```

t_{const} =

6.3579







```
%part b
a = 2; kp = 10; ki = 5; kd = 3;
s = tf('s');
G = 1/(s*(s+a));
Gc = kp + ki/s + kd*s;
H = minreal(Gc*G/(1+Gc*G));
H_err = minreal(1/(1+Gc*G));

r = u(t) + u(t-10);
y = lsim(H,r,t);
figure; plot(t, r, 'r--', t, y, 'b');
title('tracking step changes');

y_err = lsim(H_err,r,t);
figure; plot(t, y_err, 'b');
title('tracking error');

r = 0.1.*t.*u(t);
y = lsim(H,r,t);
figure; plot(t,r,'--',t,y);
grid on;
title('ramp tracking');

y_err = lsim(H_err,r,t);
figure; plot(t, y_err);
grid on;
```

```

title('tracking error');

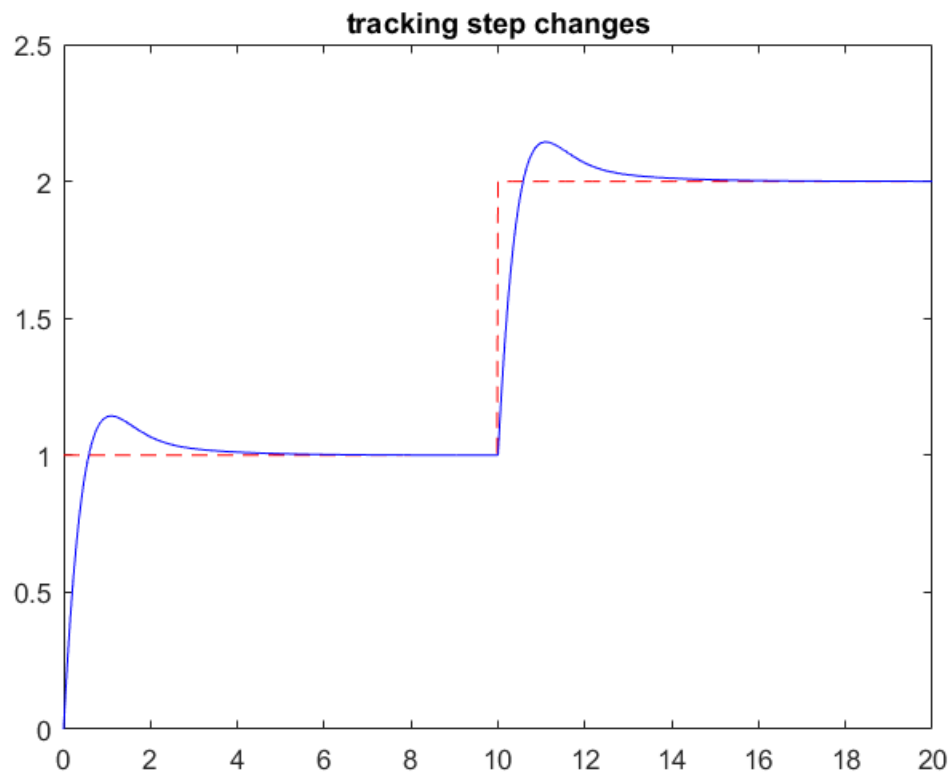
r = atan(0.1*t) .* u(t);
y = lsim(H,r,t);
figure; plot(t, r, '--', t, y);
grid on;
title('ramp tracking with correct angle');

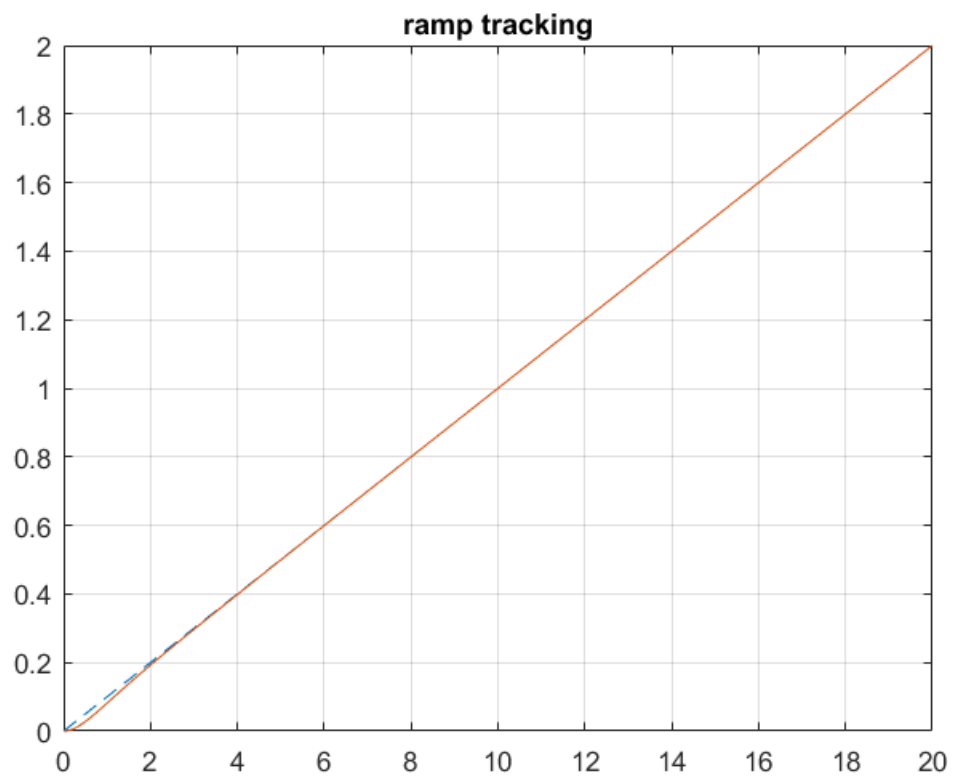
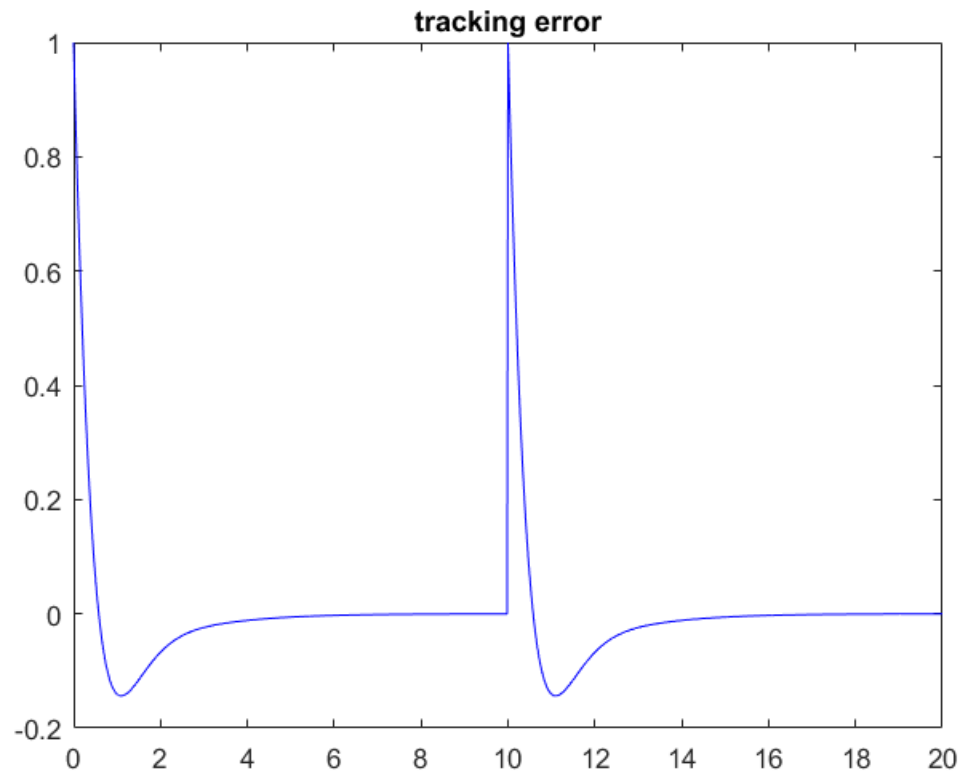
y_err = lsim(H_err,r,t);
figure; plot(t, y_err);
grid on;
title('tracking error');

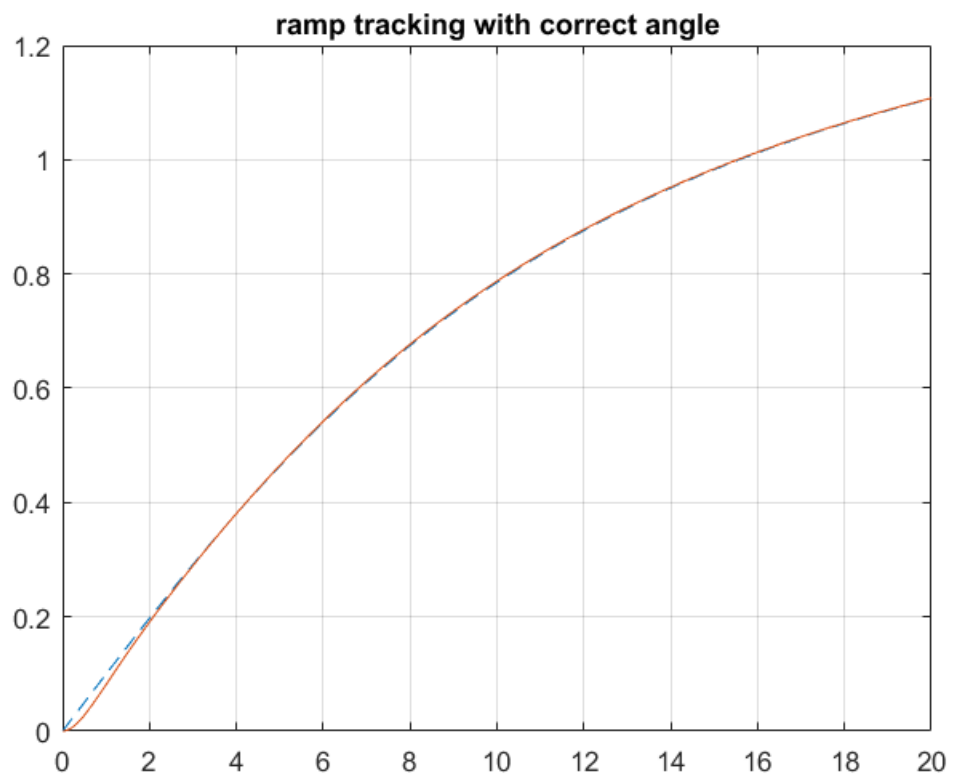
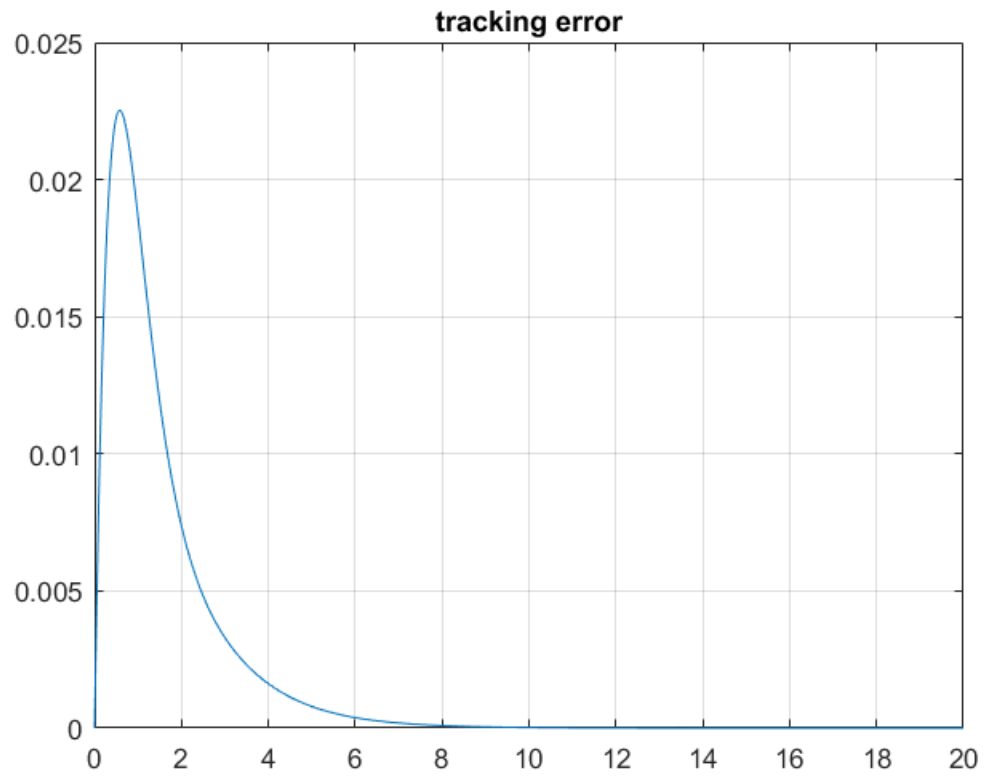
r = (0.04 * t) .* (u(t) - u(t-10)) + (-2 + 0.69*t - 0.07*t.*t
    + 0.0025*t.*t.*t) .* (u(t-10) - u(t-14)) + (0.8+0.2*(t-14)) .*
    (u(t-14)-u(t-21));
y = lsim(H,r,t);
figure; plot(t, r, '--', t, y);
grid on;
title('accelerating case');

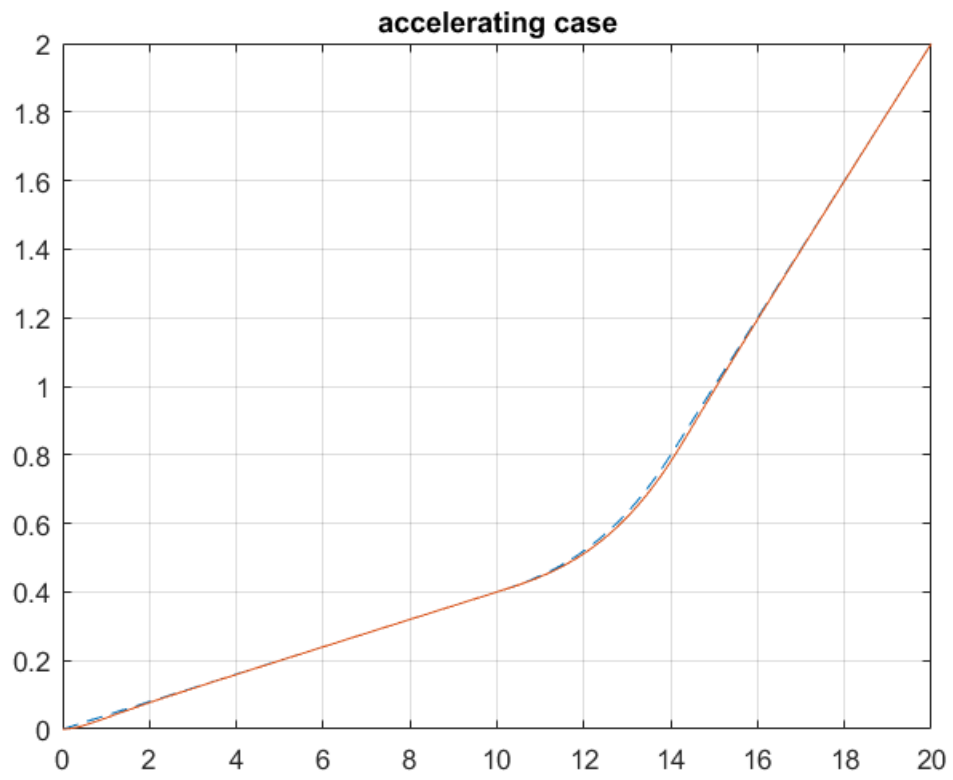
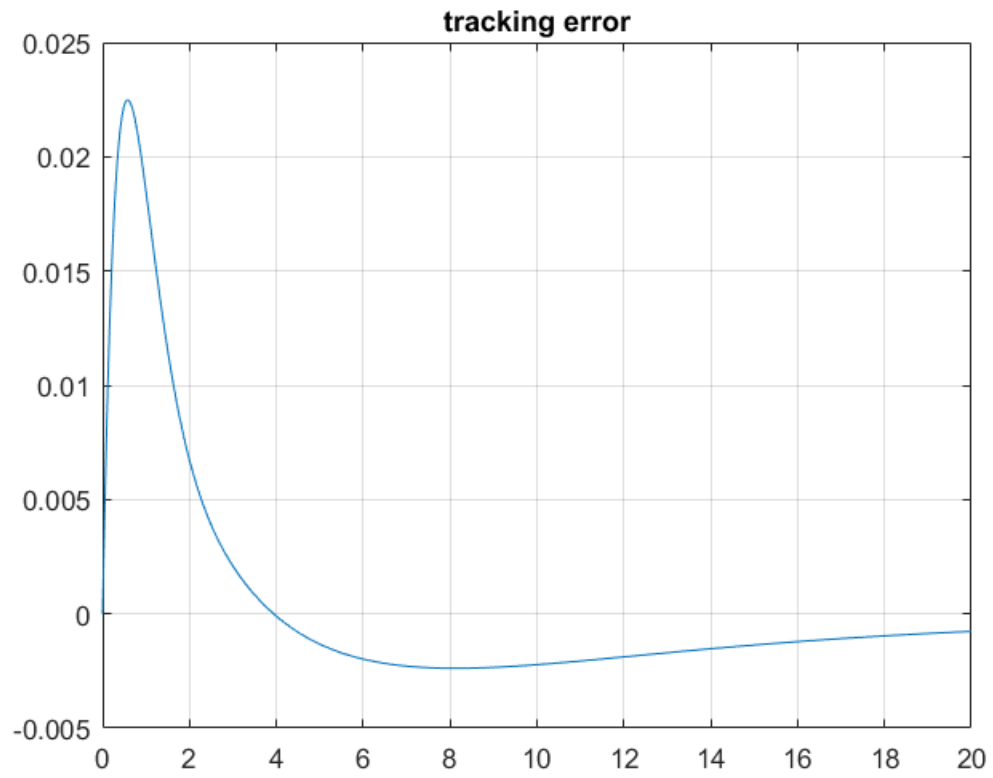
y_err = lsim(H_err,r,t);
figure; plot(t, y_err);
grid on;
title('tracking error');

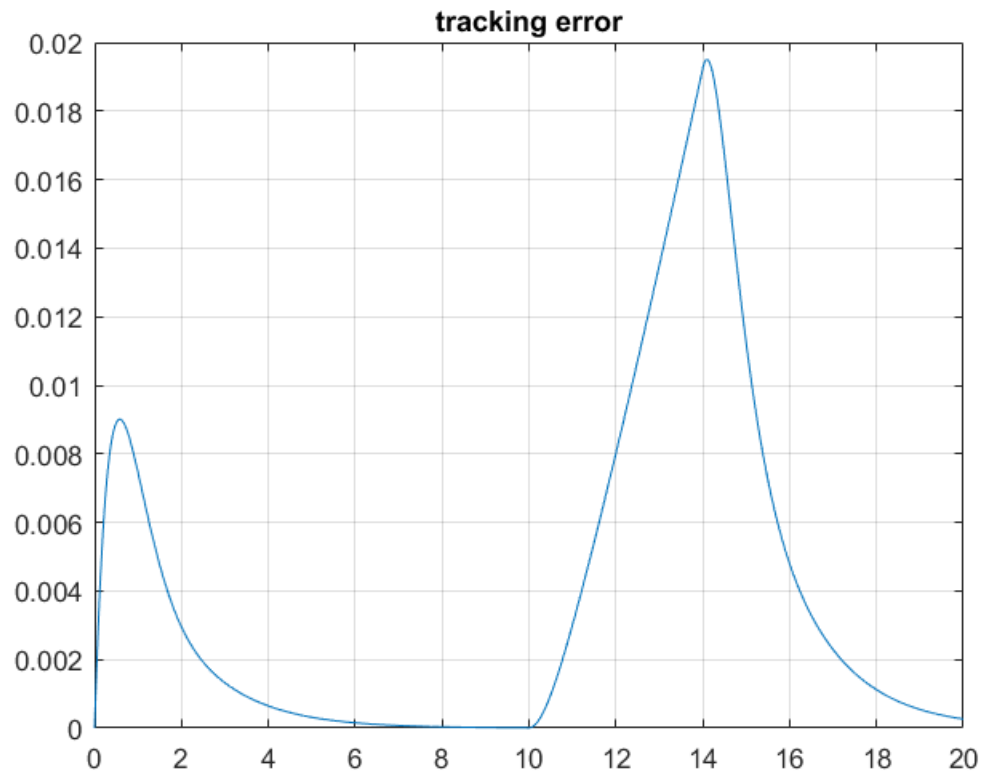
```











```
%part c
tau = 0.05;
Gc = kp + ki/s + kd*s/(tau*s+1);
H_err = minreal(1/(1+Gc*G));

a0 = u(t) + u(t-10);
b0 = lsim(H_err,a0,t);
f0 = lsim(Gc, b0, t);
figure; plot(t, f0);
title('torque f(t) -- step changes');

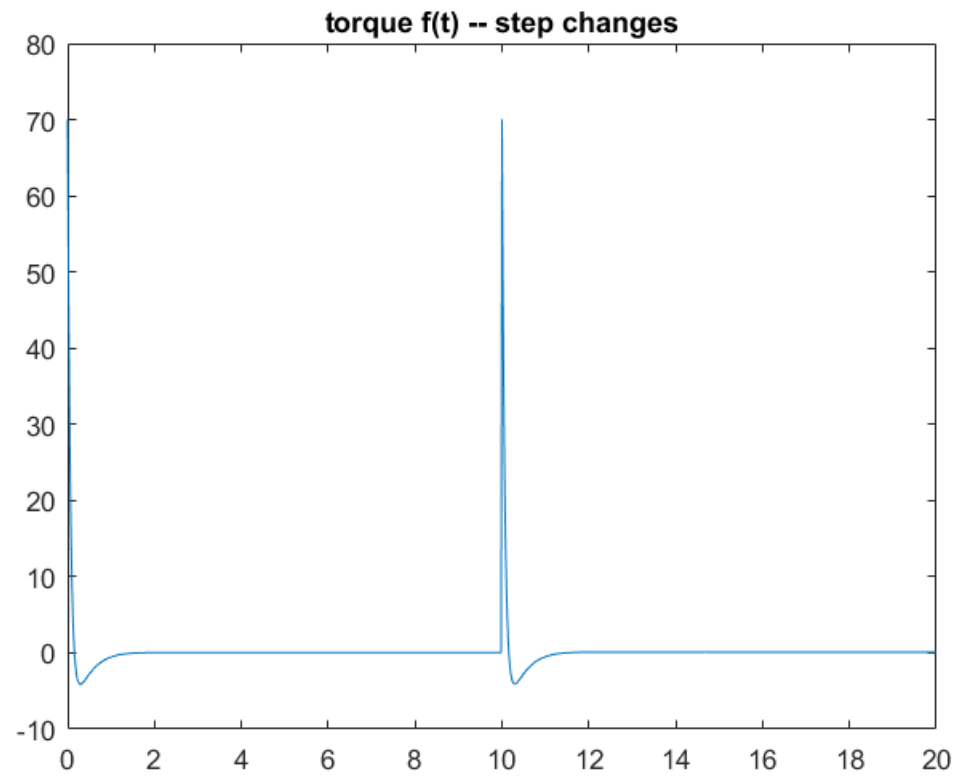
a1 = 0.1.*t.*u(t);
b1 = lsim(H_err,a1,t);
f1 = lsim(Gc, b1, t);
figure; plot(t, f1);
title('torque f(t) -- ramp tracking');

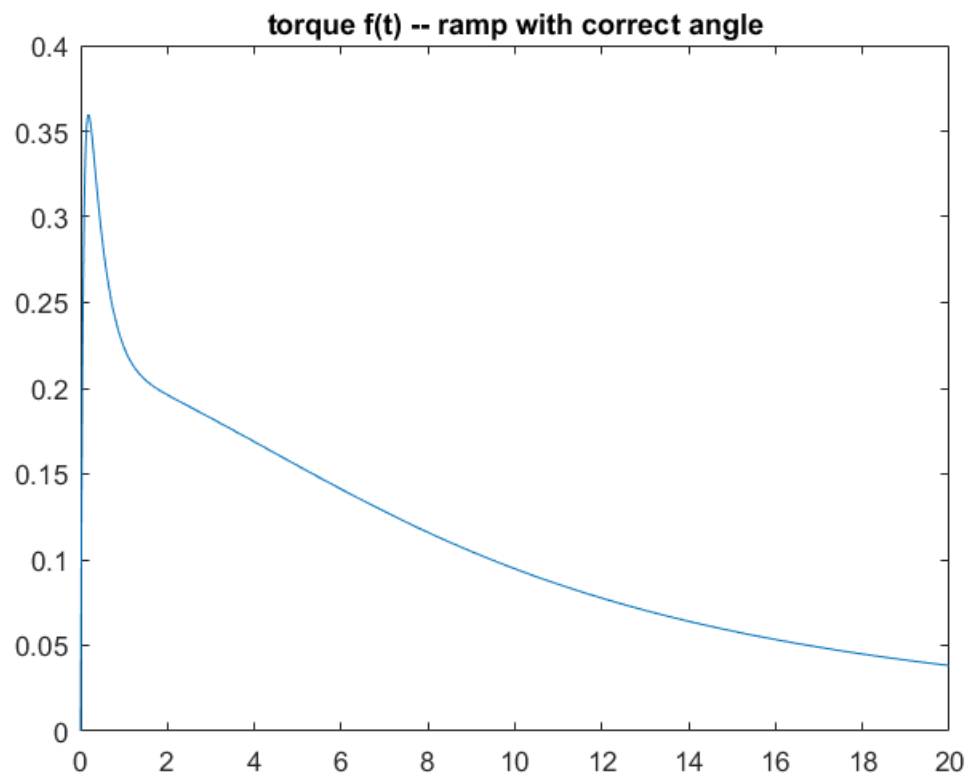
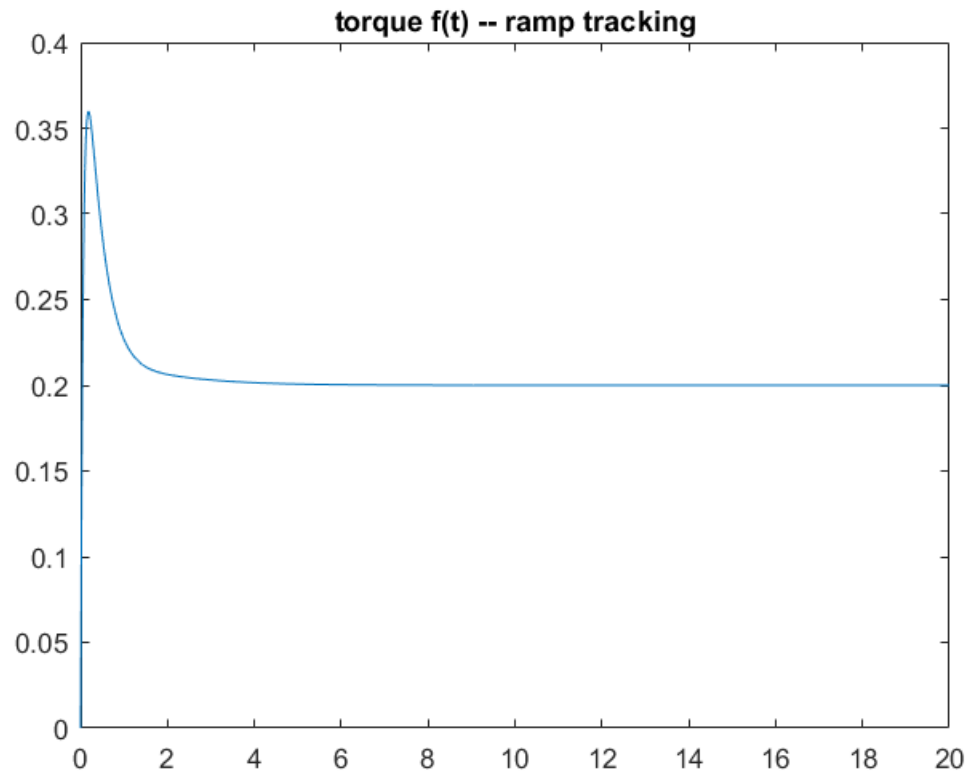
a2 = atan(0.1*t) .* u(t);
b2 = lsim(H_err, a2, t);
f2 = lsim(Gc, b2, t);
figure; plot(t, f2);
title('torque f(t) -- ramp with correct angle');

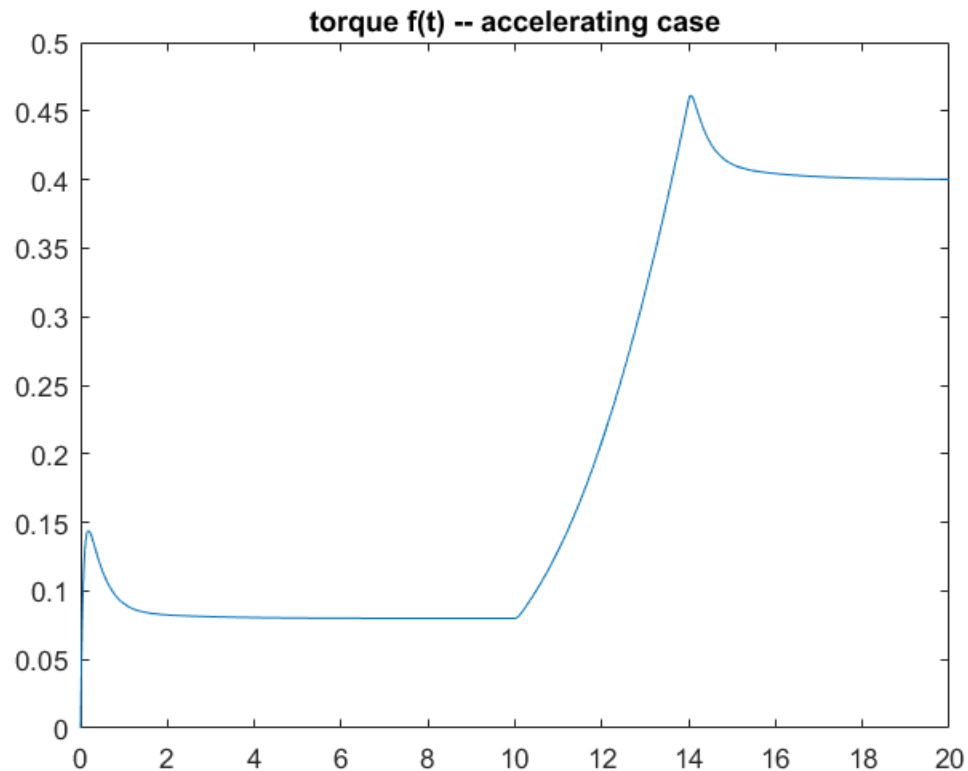
a3 = (0.04 * t) .* (u(t) - u(t-10)) + (-2 + 0.69*t - 0.07*t .* t +
    0.0025*t.* t.* t) .* (u(t-10) - u(t-14)) + (0.8 + 0.2*(t-14)) .*
    (u(t-14)-u(t-21));
```

```
b3 = lsim(H_err, a3, t);  
f3 = lsim(Gc, b3, t);  
figure; plot(t, f3);  
title('torque f(t) -- accelerating case');
```

```
tau = 0; %as per instructions
```







```
%part d
a = 2; kp = 10; ki = 5; kd = 3;
s = tf('s');
G = 1/(s*(s+a));
Gc = kp + ki/s + kd*s;
H = minreal((Gc*G)/(1 + Gc*G));
H_err = minreal(1/(1+Gc*G));
H_dist = minreal(G/(1+Gc*G));

r = u(t) + u(t-10);
f_dist = 2 *(u(t-4) - u(t-6));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind gust -- step changes');

seed = 2016; rng(seed);
f_dist = randn(size(t));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind noise -- step changes');

r = 0.1 .*t .* u(t);
```

```

f_dist = 2*(u(t-4)-u(t-6));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind gust -- ramp');

seed = 2016; rng(seed);
f_dist = randn(size(t));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind noise -- ramp');

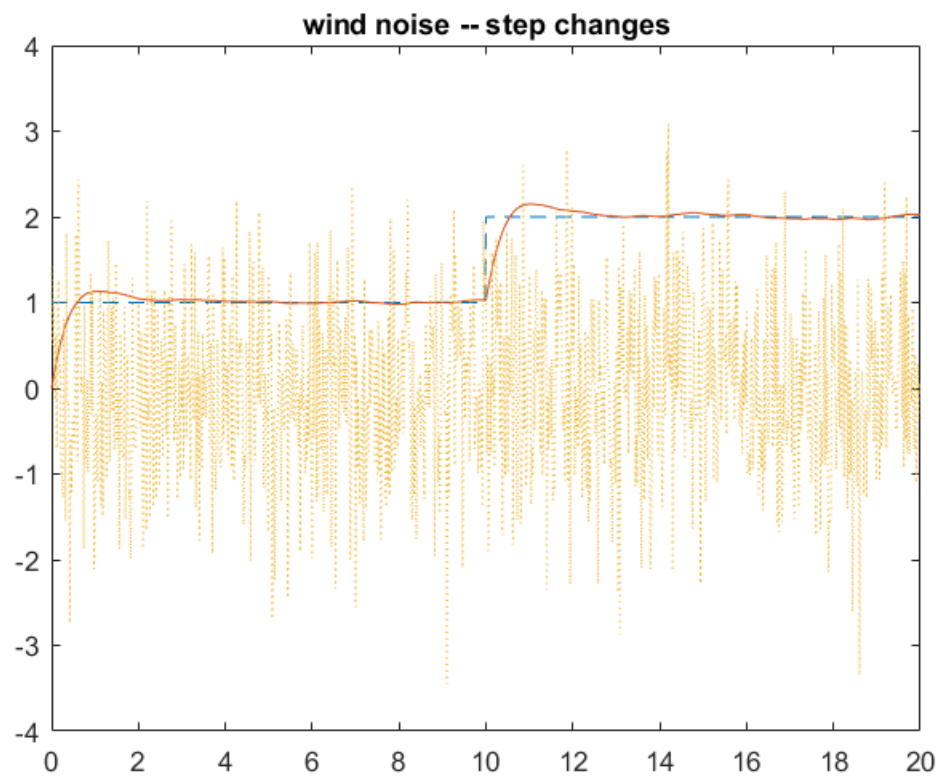
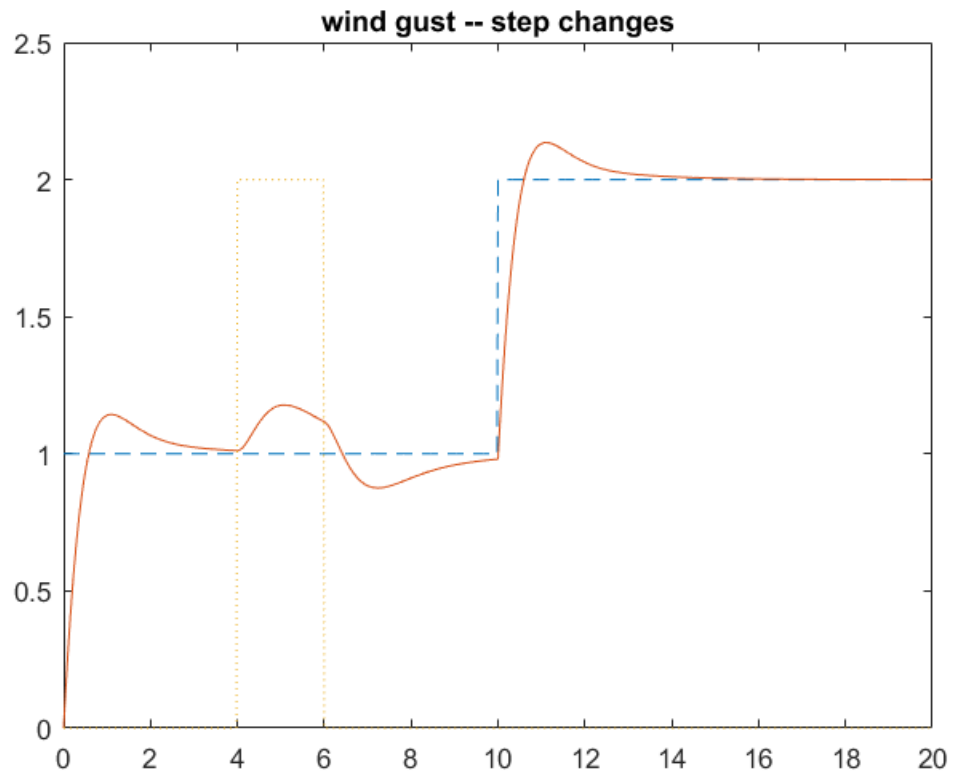
r = atan(0.1*t) .* u(t);
f_dist = 2 * (u(t-4)-u(t-6));
y_dist = lsim(H_dist, f_dist, t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind gust -- ramp with correct angle');

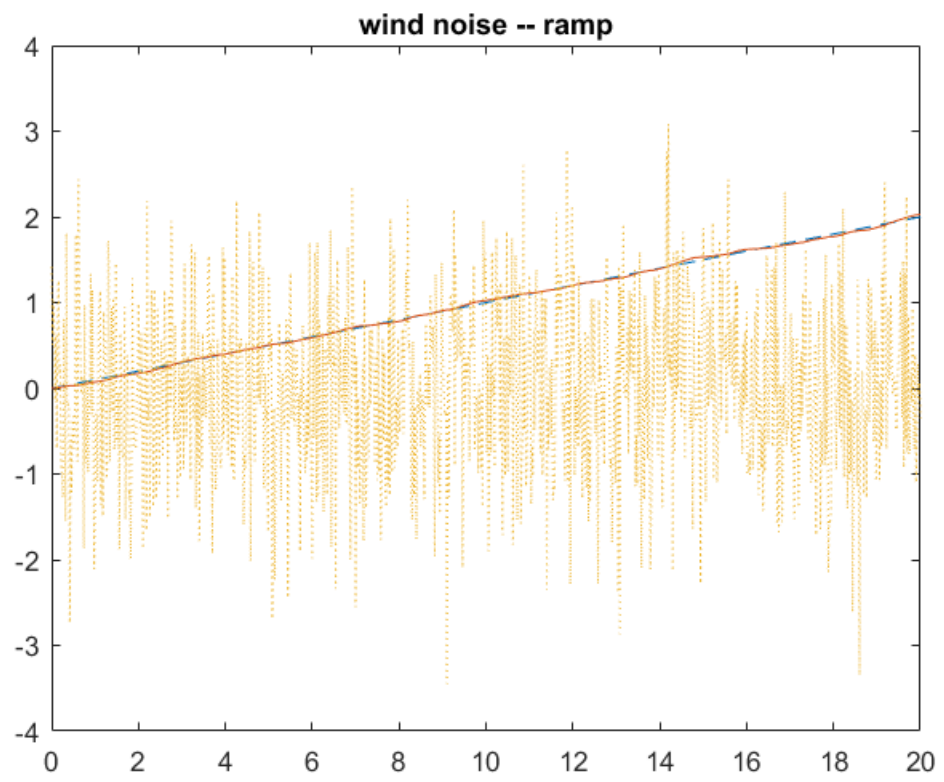
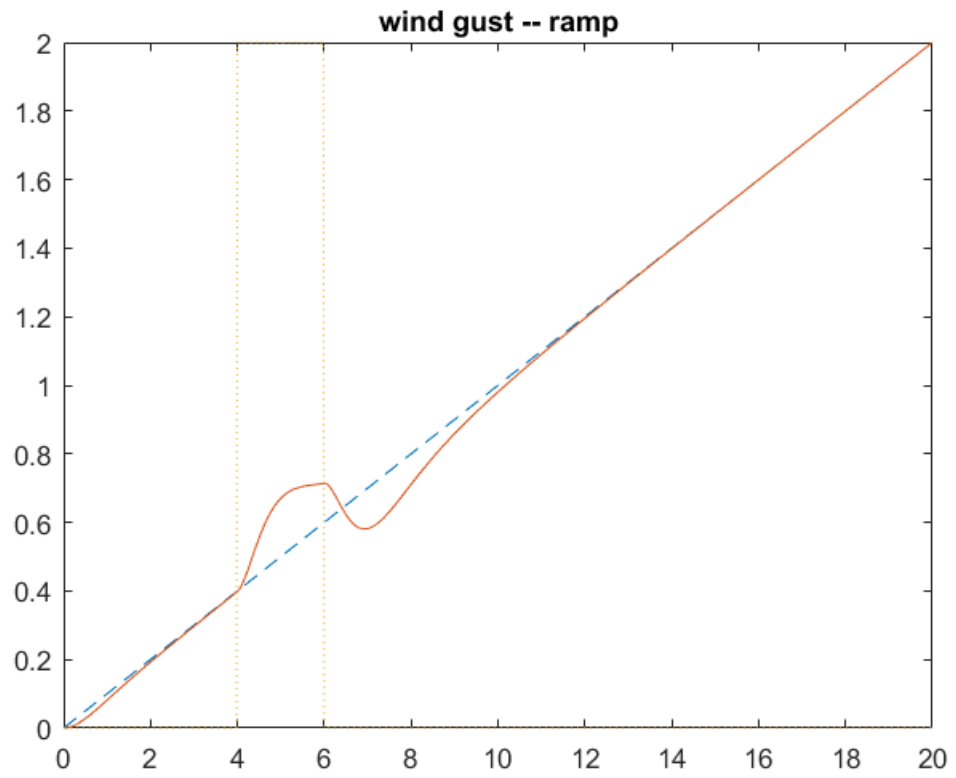
seed = 2016; rng(seed);
f_dist = randn(size(t));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind noise -- ramp with correct angles');

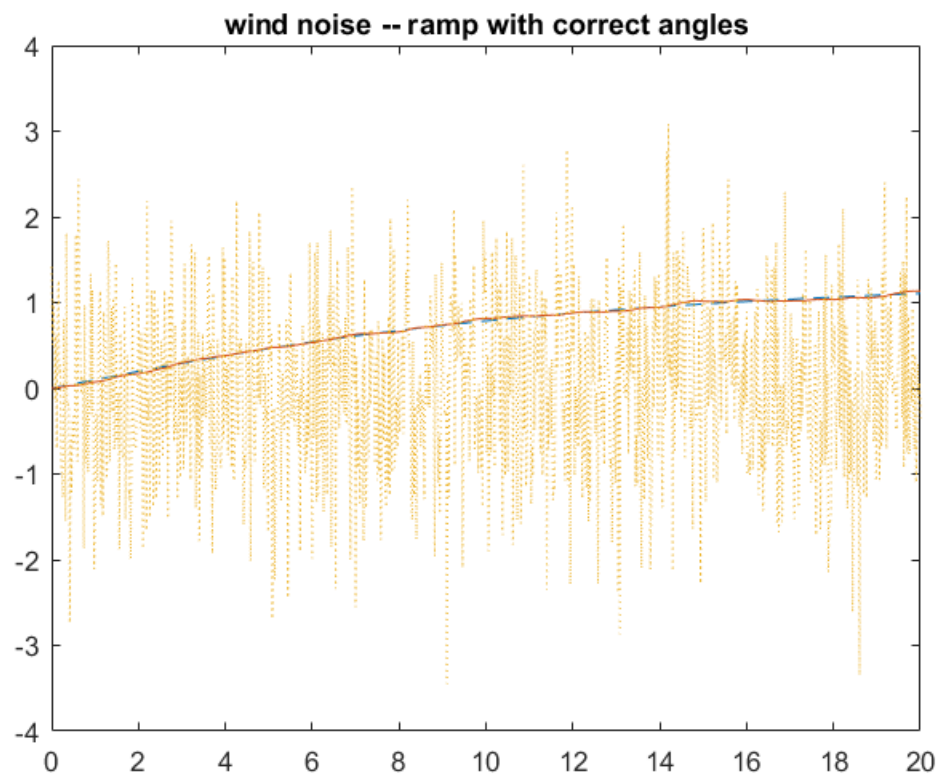
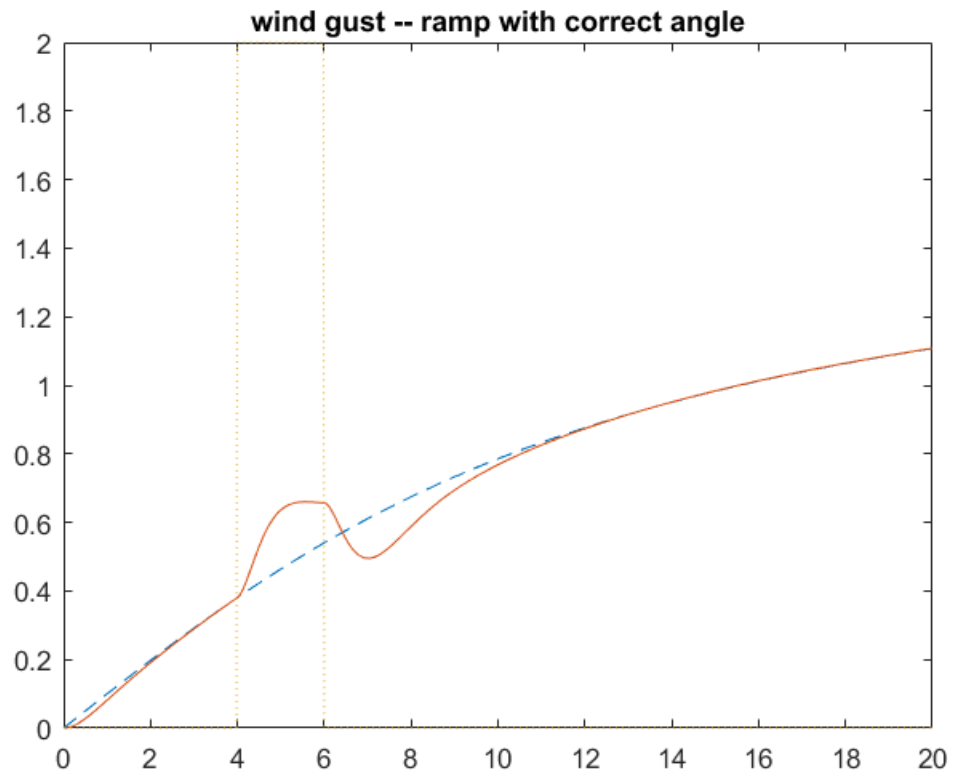
r = (0.04 * t) .* (u(t) - u(t-10)) + (-2 + 0.69*t - 0.07*t.*t +
    0.0025*t.*t.*t) .* (u(t-10) - u(t-14)) + (0.8 + 0.2*(t-14)) .*
    (u(t-14) - u(t-21));
f_dist = 2*(u(t-4)-u(t-6));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H, r, t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind gust -- accelerating');

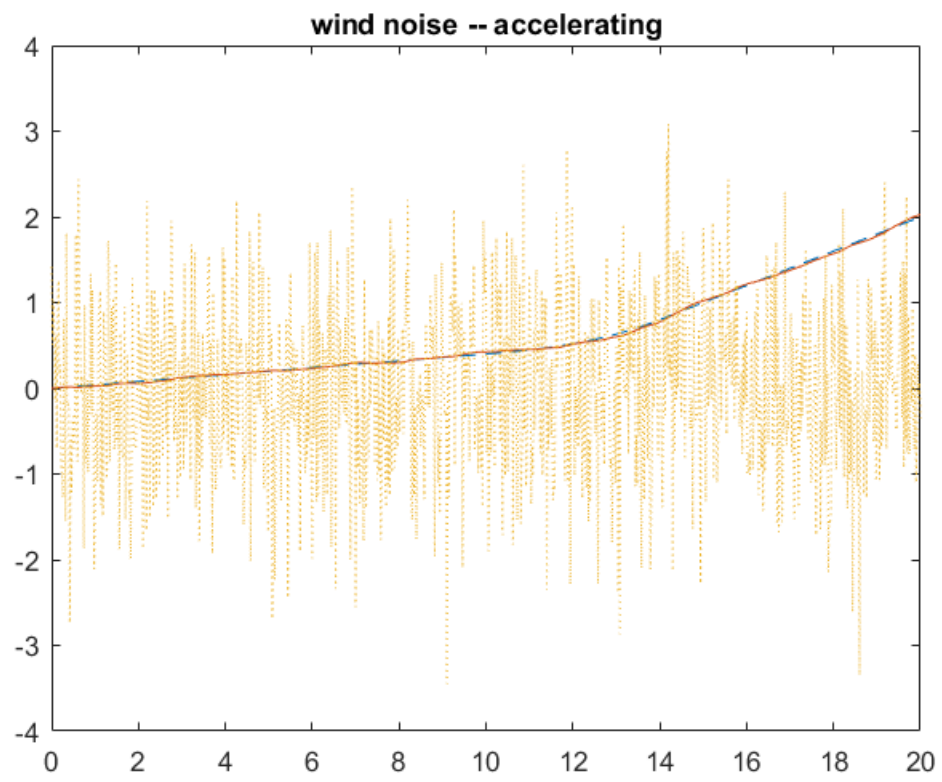
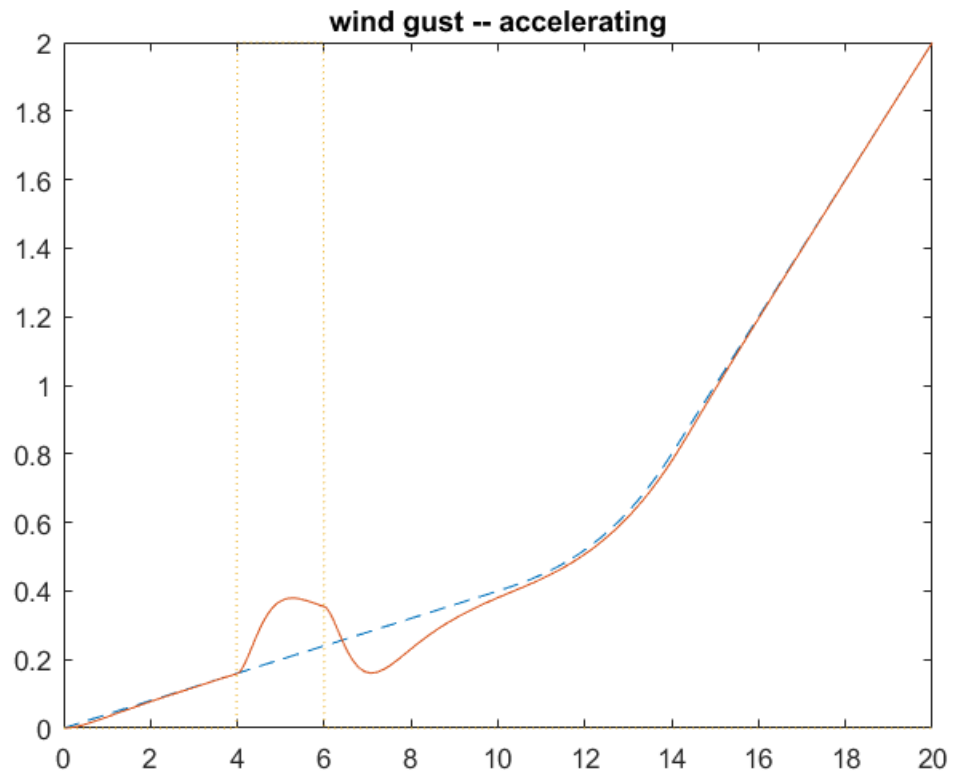
seed = 2016; rng(seed);
f_dist = randn(size(t));
y_dist = lsim(H_dist,f_dist,t);
y = lsim(H,r,t);
y_total = y + y_dist;
figure; plot(t, r, '--', t, y_total, t, f_dist, ':');
title('wind noise -- accelerating');

```









Published with MATLAB® R2016a