332:347 – Linear Systems Lab – Fall 2016

Lab 4 – S. J. Orfanidis

Consider a dish antenna sitting on a rotating base that can be rotated azimuthally by a drive motor to track a flying aircraft. The dynamics of the rotating structure is described by the equations:

$$J\ddot{\theta}(t) = -\beta\dot{\theta}(t) + N(t) + N_{\text{dist}}(t)$$

where $\theta(t)$ is the azimuthal angle, $N(t)$ is the torque applied by the drive motor, $N_{\text{dist}}(t)$ is a torque due to disturbances such as wind gusts or steady wind noise, J is the moment of inertia of the structure, and β is a frictional constant that quantifies an opposing frictional torque that is proportional to the angular velocity $\dot{\theta}$.

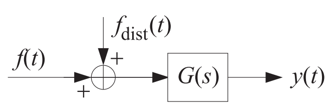
It is desired to design a control system that generates an appropriate torque $N(t)$ such that the angle $\theta(t)$ will follow a desired reference angle $\theta_{\text{ref}}(t)$, that is, $\theta(t) \to \theta_{\text{ref}}(t)$.

For example, if one wishes to point the antenna towards a given angle θ_1, then, $\theta_{\text{ref}}(t) = \theta_1 u(t)$. To point initially towards θ_1 and t_0 seconds later to point towards θ_2, one would choose $\theta_{\text{ref}}(t) = \theta_1 u(t) + (\theta_2 - \theta_1)u(t - t_0)$. Similarly, to track a uniformly moving aircraft, one would choose the ramp function $\theta_{\text{ref}}(t) = \omega_0 t u(t)$, or, more correctly, $\theta_{\text{ref}}(t) = \arctan(\omega_0 t)u(t)$.

By some redefinitions, the above system can be replaced by the following standardized form where the output $y(t)$ represents $\theta(t)$, and $f(t), f_{\text{dist}}(t)$ represent the torque inputs $N(t), N_{\text{dist}}(t)$,

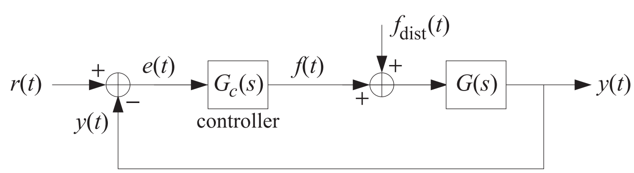
$$\ddot{y}(t) = -a\dot{y}(t) + f(t) + f_{\text{dist}}(t) \quad \Leftrightarrow \quad Y(s) = G(s)\big[F(s) + F_{\text{dist}}(s)\big] \tag{1}$$

where the system transfer function is,



$$\boxed{G(s) = \frac{1}{s(s+a)}} \tag{2}$$

The control system is implemented as the feedback system shown below,



where the overall reference input $r(t)$ represents the desired reference angle $\theta_{\text{ref}}(t)$, and the controller $G_c(s)$ is designed to generate the appropriate torque input $f(t)$ to make the system follow the reference input, i.e., $y(t) \to r(t)$, or for the tracking error signal, $e(t) = r(t) - y(t) \to 0$.

In this lab, you will design a PID controller and experiment with its settings, and also investigate its tracking ability and its robustness in the presence of disturbance inputs. The PID controller has the transfer function:

$$\boxed{G_c(s) = k_p + \frac{k_i}{s} + k_d s} \tag{3}$$

The overall transfer relationships from the two inputs $R(s)$, $F_{\text{dist}}(s)$ to the two outputs $Y(s)$ and $E(s)$ are as follows:

$$\boxed{\begin{aligned} Y(s) &= H(s)R(s) + H_{\text{dist}}(s)F_{\text{dist}}(s) \\ E(s) &= H_{\text{err}}(s)R(s) - H_{\text{dist}}(s)F_{\text{dist}}(s) \end{aligned}} \tag{4}$$

where,

$$
\begin{aligned}
H(s) &= \frac{G_c(s)G(s)}{1+G_c(s)G(s)} = \frac{k_d s^2 + k_p s + k_i}{s^3 + (a+k_d)s^2 + k_p s + k_i} = \text{closed-loop} \\
H_{\text{err}}(s) &= \frac{1}{1+G_c(s)G(s)} = \frac{s^2(s+a)}{s^3 + (a+k_d)s^2 + k_p s + k_i} = \text{error} \\
H_{\text{dist}}(s) &= \frac{G(s)}{1+G_c(s)G(s)} = \frac{s}{s^3 + (a+k_d)s^2 + k_p s + k_i} = \text{disturbance}
\end{aligned}
\tag{5}
$$

Regarding the choice of the PID parameters k_p, k_i, k_d, we note the following: (a) k_i must be nonzero in order to guarantee zero steady-state error, i.e., $e(t) \to 0$, for both the step and ramp inputs, (b) increasing k_i will increase the overshoot and the settling time and decrease the rise time, (c) increasing k_d will decrease the overshoot and the settling time, and (d) increasing k_p will decrease the rise time but increase the overshoot.

The steady-state tracking error due to a particular reference input $r(t)$ can be calculated with the help of the final-value theorem of Laplace transforms, that is,

$$
\lim_{t\to\infty} e(t) = \lim_{s\to 0}\big[sE(s)\big] = \lim_{s\to 0}\big[sH_{\text{err}}(s)R(s)\big] = \lim_{s\to 0}\left[\frac{s^3(s+a)R(s)}{s^3 + (a+k_d)s^2 + k_p s + k_i}\right]
\tag{6}
$$

For a step input $r(t) = u(t)$, or a ramp input $r(t) = t\,u(t)$, we have $R(s) = 1/s$, or $R(s) = 1/s^2$, respectively, and for these Eq. (6) implies that the tracking error will be zero provided $k_i \neq 0$. With the above background information, please carry out the following experiments:

a. This part requires that you produce 6 graphs. Starting with the parameter values,

$$
a = 2, \quad k_p = 10, \quad k_i = 5, \quad k_d = 3
\tag{7}
$$

construct the *transfer function objects* for the system $G(s)$, controller $G_c(s)$, closed-loop feedback system $H(s)$, tracking error $H_{\text{err}}(s)$, and disturbance transfer function $H_{\text{dist}}(s)$, using, for example, the MATLAB code:

```
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));
Herr = minreal(1/(1+Gc*G));
Hdist = minreal(G/(1+Gc*G));
```

where the **minreal** function removes any possible common factors from the numerator and denominator transfer functions, resulting in a minimal realization—this happens for example in the case $k_i = 0$ in which some s factors can be canceled.

First, determine the *poles* of the closed-loop transfer function $H(s)$ and from the pole lying closest to the imaginary axis on the s-plane, calculate the 40-dB *time constant* of $H(s)$. Note that the poles can be determined by using the function **roots** or **pzmap**, e.g.,

```
p = roots(H.den{1});     % H.den{1} is the vector of denominator coefficients of H(s)
p = pzmap(H);
```

Next, define t as a vector of 1001 equally-spaced time samples spanning the interval $0 \le t \le 20$. Using the **lsim** function, *calculate and plot* the unit-step response of $H(s)$ over this time range. Is the observed transient time consistent with the 40-dB time constant? As in previous labs, you may find it useful to define the unit-step function as,

```
u = @(t) double(t>=0);
```

Then, increase the PID parameters by doubling their values one at a time, and *plot* the corresponding step responses, and comment on the effect of such changes.

b. This part requires that you produce 10 graphs. For the parameter values defined in Eq. (7), and for the same time range as in part (a), generate the following four reference input signals describing the typical reference angle situations mentioned at the beginning of this handout:

$$
\begin{aligned}
r(t) &= u(t) + u(t-10) && \text{(switches from } r=1 \text{ to } r=2 \text{ at } t=10\text{)} \\
r(t) &= 0.1\,t\,u(t) && \text{(uniformly moving aircraft)} \\
r(t) &= \arctan(0.1\,t)\,u(t) && \text{(uniformly moving with correct angle)} \\
r(t) &= \begin{cases} 0.04\,t\,, & 0 \le t \le 10 \\ -2 + 0.69\,t - 0.07\,t^2 + 0.0025\,t^3\,, & 10 \le t \le 14 \\ 0.8 + 0.2\,(t-14)\,, & 14 \le t \le 20 \end{cases} && \text{(accelerating)}
\end{aligned}
\tag{8}
$$

The fourth case, emulates a situation where the aircraft is moving at constant speed until $t = 10$ and then between $t = 10$ and $t = 14$, it accelerates to a new speed. The expression between $10 \le t \le 14$ is the cubic Hermite interpolation polynomial (see z-transform file **sztable.pdf** on Sakai) that interpolates smoothly between the two speeds.

For each of the four $r(t)$ inputs, compute the corresponding output $y(t)$ of the closed-loop system H, using the function **lsim**:

```
y = lsim(H,r,t);
```

On the same graph, plot both $y(t)$ and $r(t)$ with different linestyles, observing whether the controlled system is capable of following the desired input reference setting. On a separate graph, plot the tracking error signal $e(t)$ versus t.

For the particular case of the ramp input $r(t) = 0.1\,t\,u(t)$, set temporarily $k_i = 0$, and recompute and plot the system output $y(t)$ and error $e(t)$, noting that the steady-state error $e(t)$ is no longer zero, although the slope of the output does follow the slope of the reference input. After this part, set k_i back to its non-zero value.

c. This part requires that you produce 4 graphs. Because of the difficulty in implementing the derivative term $k_d s$ of the PID controller, the following modified variant is often used:

$$G_c(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{\tau s + 1} \tag{9}$$

where τ is a very small quantity. In this case, the closed-loop transfer function becomes,

$$H(s) = \frac{(k_d + \tau k_p)s^2 + (k_p + \tau k_i)s + k_i}{\tau s^4 + (\tau a + 1)s^3 + (a + k_d + \tau k_p)s^2 + (k_p + \tau k_i)s + k_i} \tag{10}$$

Set $\tau = 0.05$ and use the PID parameters of Eq. (7).

Determine the transfer function $H_f(s)$ from the overall input $r(t)$ to the controller's torque output $f(t)$ and for all four choices of $r(t)$ of Eq. (8), compute the torque $f(t)$ and plot it versus t.[†] This will give you a sense of the actual input being applied to the controlled system $G(s)$ that causes it to follow the reference input $r(t)$. Set $\tau = 0$ after this part is complete.

d. This part requires that you produce 8 graphs. Here, you will investigate how the controlled system responds to a disturbance. Consider two types of disturbances, one imitating a wind gust lasting for a brief period of time, say, $4 \le t \le 6$, and the other imitating steady wind noise. They can be generated by the following MATLAB code, for the same length-1001 vector of t's that you defined in part (a),

[†] MATLAB will complain if you tried to do this part with $\tau = 0$.

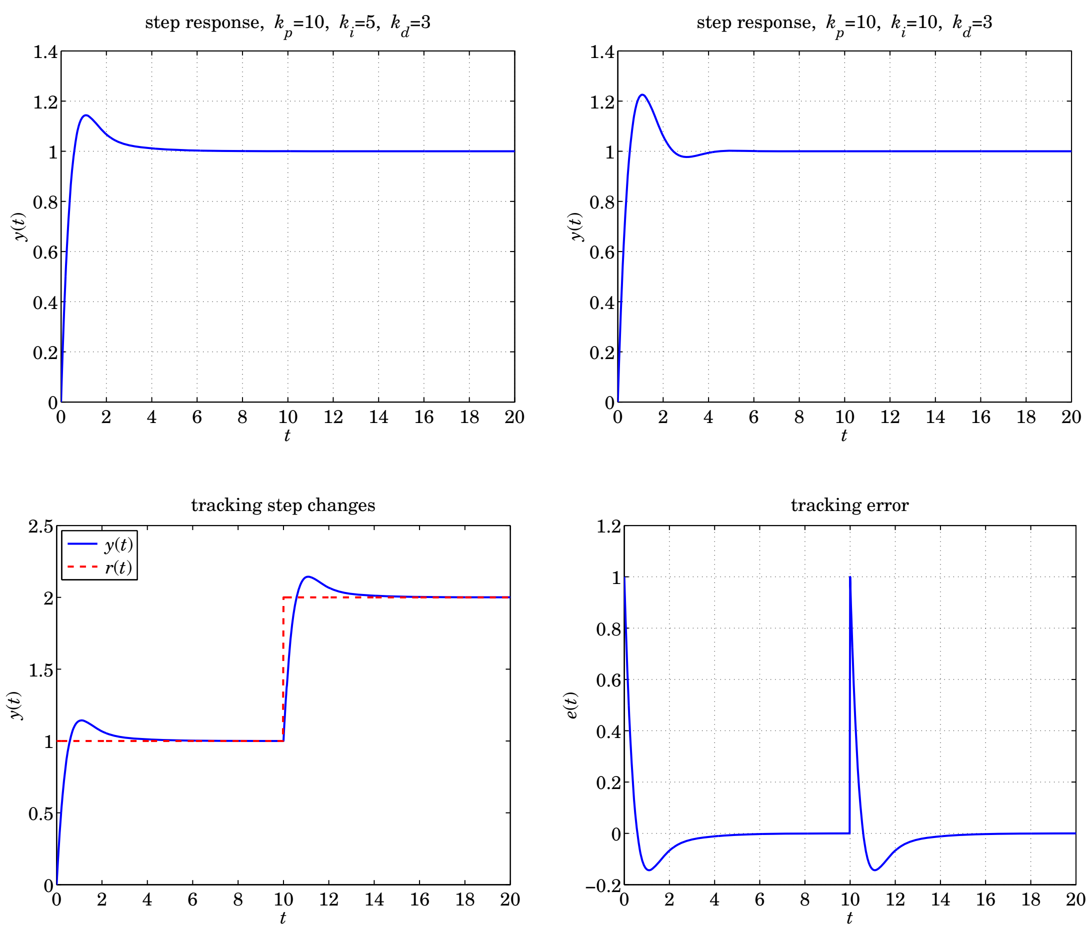
```
fdist = 2*(u(t-4)-u(t-6));          % wind gust

seed=2016; rng(seed);               % initialize random number generator
fdist = randn(size(t));             % zero-mean, unit-variance noise
```

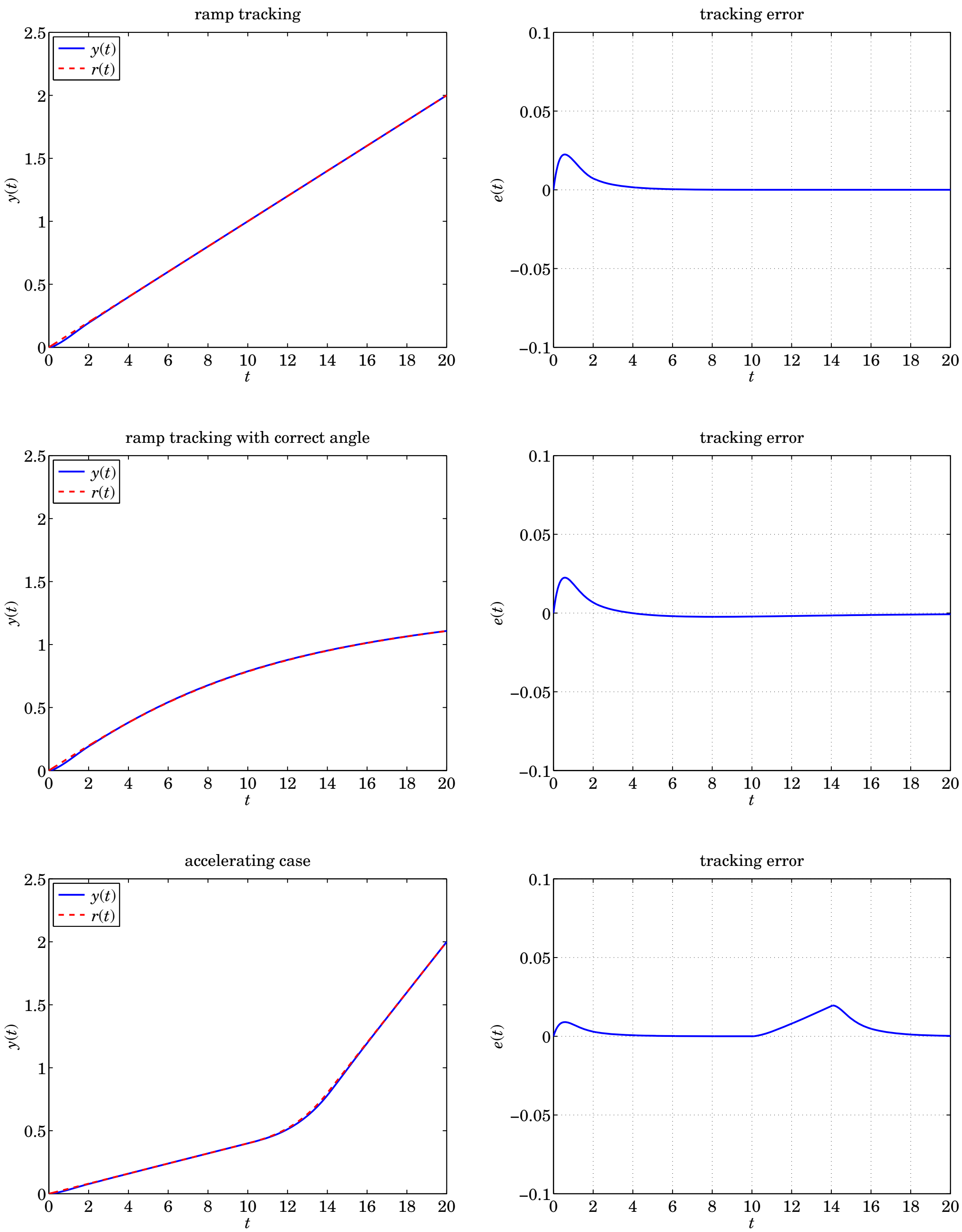
For each type of disturbance, compute the corresponding system output using the disturbance transfer function H_{dist}, and add it to the previously obtained output from each of the four reference signals $r(t)$ to get the total system output:

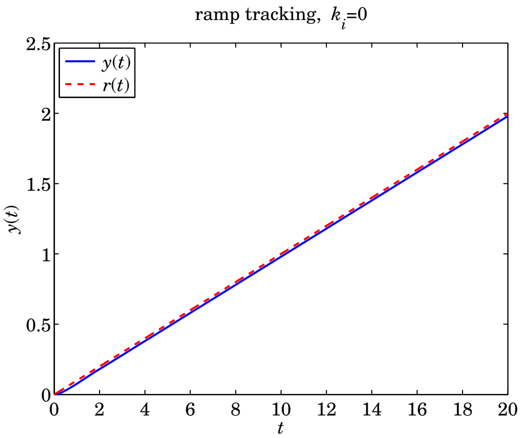
```
ydist = lsim(Hdist,fdist,t);
y = lsim(H,r,t);
ytot = y + ydist;
```

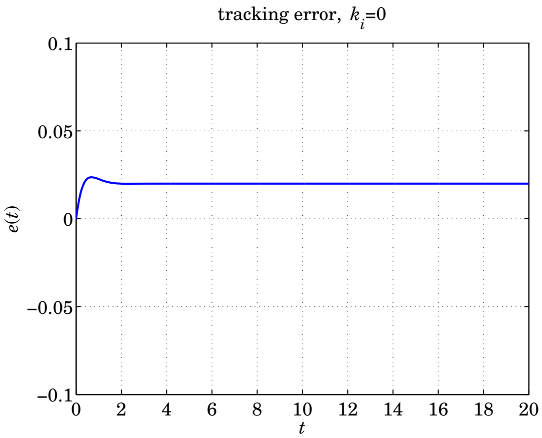
For each of the resulting eight cases (2 disturbances $\times$ 4 reference signals), plot the signals $y_{\text{tot}}(t)$ and $r(t)$ on the same graph, observing how the system recovers (or not) from the disturbance.

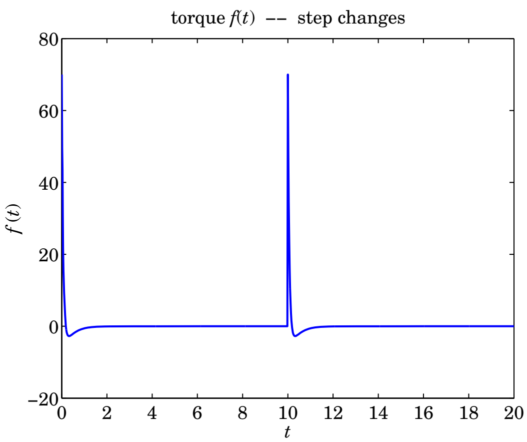
Typical Outputs

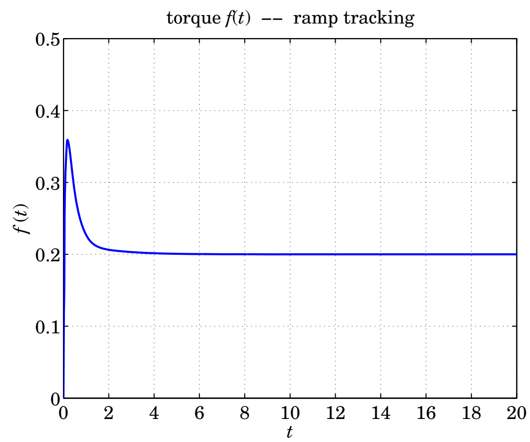


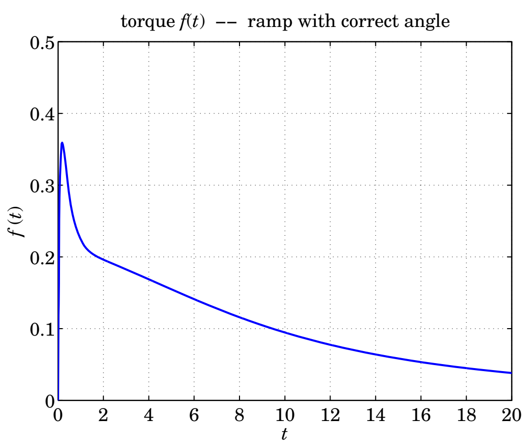

ramp tracking
tracking error
ramp tracking with correct angle
tracking error
accelerating case
tracking error
y(t)
r(t)
e(t)
t
0
2
4
6
8
10
12
14
16
18
20
0
0.5
1
1.5
2
2.5
−0.1
−0.05
0
0.05
0.1

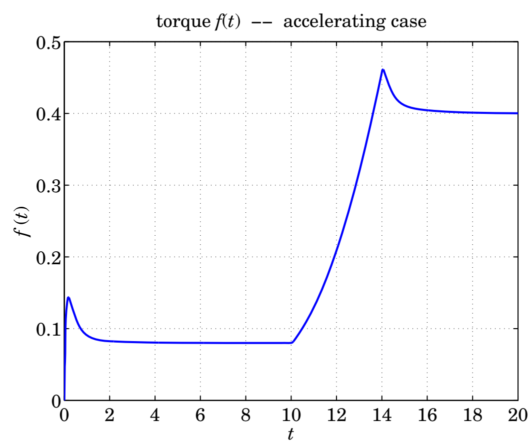

ramp tracking, k_i=0
$y(t)$
$r(t)$
$y(t)$
t

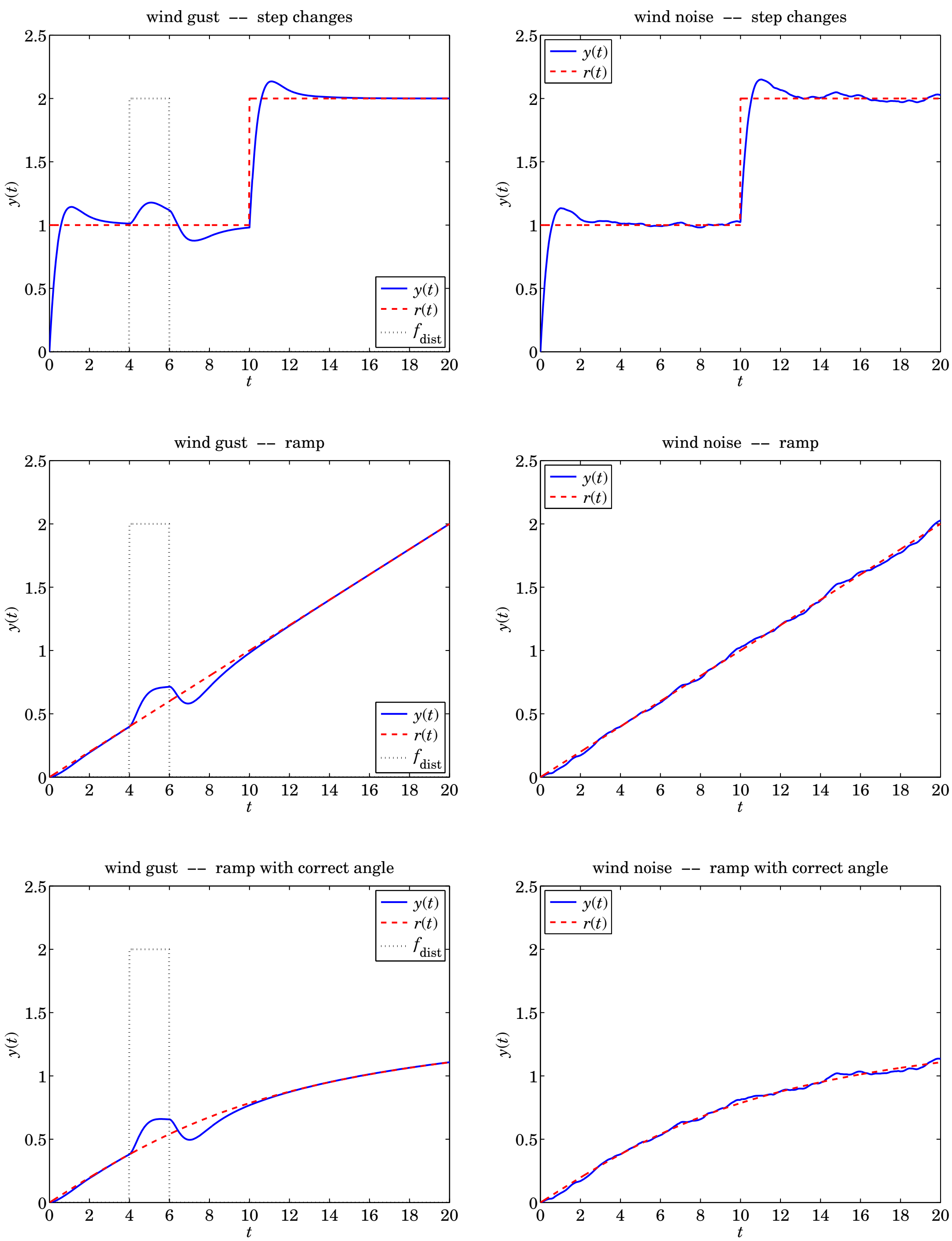

tracking error, k_i=0
$e(t)$
t

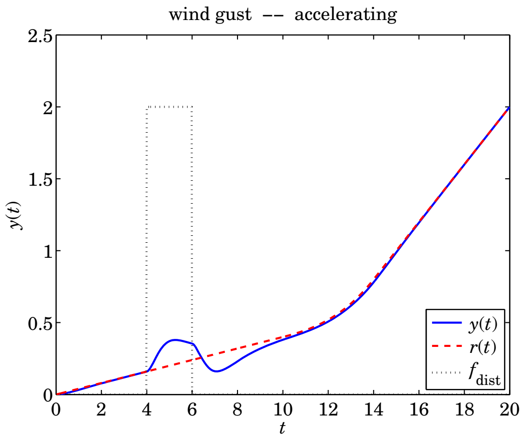

torque $f(t)$ -- step changes
$f(t)$
t

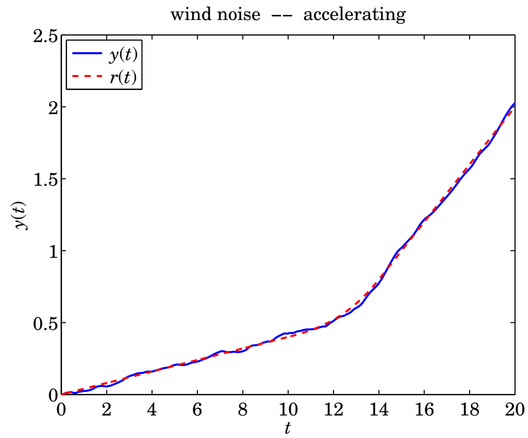

torque $f(t)$ -- ramp tracking
$f(t)$
t


torque $f(t)$ -- ramp with correct angle
$f(t)$
t


torque $f(t)$ -- accelerating case
$f(t)$
t


wind gust -- step changes
wind noise -- step changes
wind gust -- ramp
wind noise -- ramp
wind gust -- ramp with correct angle
wind noise -- ramp with correct angle
$y(t)$
$r(t)$
f_{dist}
t


wind gust -- accelerating
y(t)
t
y(t)
r(t)
f_dist
0
2
4
6
8
10
12
14
16
18
20
0.5
1
1.5
2
2.5


wind noise -- accelerating
y(t)
t
y(t)
r(t)
0
2
4
6
8
10
12
14
16
18
20
0.5
1
1.5
2
2.5