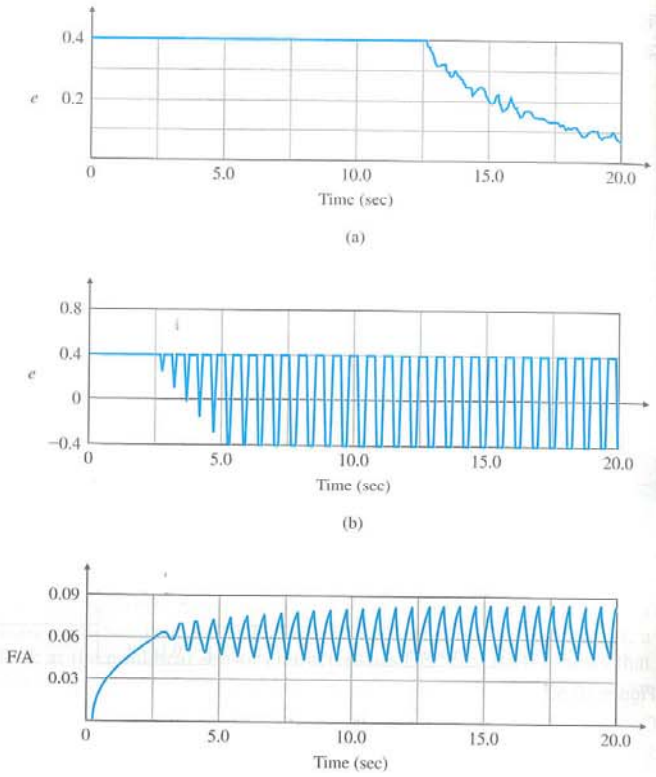
**Figure 10.53**
System response with nonlinear sensor approximation




12.5 sec before the error comes out of saturation and a time constant of almost 5 sec once the linear region is reached. In real automobiles, these systems are operated with much higher gains. To show these effects, a simulation with $K_pK_s = 6.0$ is plotted in Fig. 10.53(b, c). At this gain the linear system is unstable, and up until about 5 sec, the signals grow. The growth halts after 5 sec due to the fact that, as the input to the sensor nonlinearity gets large, the *effective* gain of the sensor decreases due to the saturation, and eventually, a limit cycle is reached. The frequency of this limit cycle corresponds to the point at which the root locus crosses the imaginary axis and has an amplitude such that the total effective gain of $K_pK_{s,eq} = 2.8$. As described in Section 9.3, the effective gain of a saturation for moderately large inputs can be computed and is given by the describing function to be approximately $4N/\pi a$, where $N$ is the saturation level and $a$ is the amplitude of the input signal. Here $N = 0.4$, and if $K_p = 0.1$, then $K_{s,\text{eq}} = 28$. Thus, we predict an input signal amplitude of $a = 4(0.4)/28\pi = 0.018$. This value is closely verified by



the plot of Fig. 10.53(c), the input to the nonlinearity in this case. The frequency of oscillation is also nearly 10.1 rad/sec, as predicted by the root locus in Fig. 10.50.

In the actual implementation of F/A feedback controllers in automobile engines, sensor degradation over thousands of miles of use is of primary concern, because the federal government mandates that the engines meet the exhaust-pollution standards for the first 50,000 mi. In order to reduce the sensitivity of the average setpoint to changes in the sensor output characteristics, manufacturers typically modify the design discussed here. One approach is to feed the sensor output into a relay function [see Fig. 9.6(b)], thus completely eliminating any dependency on the sensor gain at the setpoint. The frequency of the limit cycle is then solely determined by controller constants and engine characteristics. Average steady-state F/A accuracy is also improved. The oscillations in the F/A are acceptable because they are not noticeable to the car's occupants. In fact, the F/A excursions are beneficial to the catalyst operation in reducing pollutants.

Feedback systems have also been used in many other areas of modern automobiles. A car's desired inside temperature is set by the passengers and a feedback system maintains that temperature. Cruise control systems have been augmented with radar so the cruise control will maintain a certain distance behind the car ahead by feedback of that distance to the cruise control. Systems are being developed whereby traffic lane marking sensors are being fed back to the steering to maintain a car in the correct lane. Stability augmentation systems use differential braking based on accelerometer measurements to keep a car upright in violent maneuvers. The list goes on and on, many of which are described in a book by Ulsoy, Peng, and Cakmakci (2012). It is possible that platoons of cars on a freeway will be tightly controlled with small distances between them in order to increase the capacity of freeways in the future. Such a scheme will be a victory for feedback control, but could put a tough test on our legal system if there are serious accidents.

## 10.5 Control of a Quadrotor Drone

Drones, or Unmanned Aerial Vehicles (UAVs) are aircraft that are pilotless. They can be either controlled from the ground or are completely autonomous and have onboard control algorithms that primarily determine their path. Some are winged aircraft, and some are helicopters that rely on one or more rotor blades for lift. Drones are being developed for a very large number of applications. Some of the many examples of current use are aerial photography and video, surveillance, security/police work, search and rescue, farming, defense, and, of course, entertainment. Many other applications are being investigated from package delivery to flying cars. More general research includes optimization of onboard control, artificial intelligence, and swarming approaches where

dozens of drones are in a networked communication system and are being controlled to perform a particular task.

Helicopters with multiple rotors are a popular platform for these vehicles due to the simplicity of the vehicle hardware and maintenance, ability to hover, and the vertical takeoff and landing capability. As a result, much of the sophistication is in the control and navigation software.

STEP 1. *Understand the process.* The quadrotor consists of four rotors, with one pair rotating clockwise (CW), and the other pair rotating counter-clockwise (CCW). An example is shown in Figure 10.54. By independently controlling each rotor's speed, it is possible to command the attitude of the vehicle along with the translation and altitude. By using two CW rotors and two CCW rotors, the reaction torques on the vehicle are canceled, thus eliminating the need for the tail rotor that is required for the typical single-rotor helicopters and simplifying the hardware considerably.

STEP 2. *Select sensors.* 3-D position and velocity sensing is readily available via GPS (or other satellite-based navigation systems) in an earth-fixed coordinate system. In addition, it is reasonably affordable to have inertial sensors (aided by GPS in some cases) that provide pitch, roll, and yaw angles, plus their angular rates. These Inertial Measurement Units (IMUs) contain 3-D accelerometers and 3-D gyros at a minimum, but sometimes contain 3-D magnetometers as well. A package with all these sensors will have the capability to supply accurate attitudes and rates, earth-fixed position and velocity, and body-fixed position and velocity. Having all these sensors in the unit allows the utilization of inexpensive sensors because the redundancy in the sensed

**Figure 10.54**
Quadrotor drone
*Source: Slavoljub Pantelic/Shutterstock*



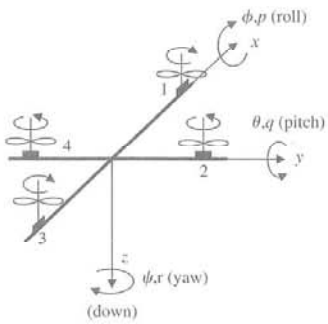
quantities allows for the estimation of indivicual sensor errors. Small quadrotor drones without GPS that only include 3-axis angular rate gyros and 3-axis accelerometers are available for less than $100; however, they require considerable practice to fly them well via remote control. Depending upon UAV requirements and accuracy requirements, sensor packages can range in cost from $100 to $3000 and even greater if very precise positioning is required. For purposes of this example, we will assume the existence of position and velocity in both earth- and body-frames, plus pitch, roll, and yaw angles and rates.

STEP 3. *Select actuators.* The actuators were discussed in STEP 1. The quadrotor drone can be completely controlled by the individual commands to the four motors driving the rotors. As will become apparent in Steps 5 and 8, the 3-D position-time history and yaw angle are controllable with these four inputs. Example 2.5 in Chapter 2 explains the commands necessary to each rotor in order to achieve proper control of roll, pitch, and yaw. Example 5.16 in Chapter 5 explains the use of the actuators to control position as well.

STEP 4. *Make a linear model.* The coordinate system for a quadrotor is shown in Fig. 10.55. It follows the airplane axes in Section 10.3 to some extent; however, in this case, it is arbitrary how the $x$ and $y$ axes are oriented in the plane of the rotors. Note rotors 1 and 3 are rotating in a CW direction, while rotors 2 and 4 are rotating in a CCW direction. Selecting the $x$-axis to be aligned with two of the rotors simplifies the control logic to some extent for purposes of this example; however, most drones normally fly with the rotor arms at 45° to the direction of flight. Quadrotor drones often have cameras installed that are not able to swivel in the $x$-$y$ plane of the drone; rather, they are only able to swivel up and down. Furthermore, the cameras are typically aligned so they look forward in the direction of flight. Therefore, the only way to control the camera viewing aximuth is to yaw the drone.

The quadrotor longitudinal $x$-axis equations, linearized and partially modified for drone use of the full nonlinear aircraft Eqs (10.16)

**Figure 10.55**
Definition of drone coordinates



and (10.17), are contained in Eqs. (10.36a). These linearized equations assume the body-fixed coordinate system stays essentially level through the motion and the pitch and roll angles remain small; that is, the drone stays close to level. Also, for the linear assumptions to be valid, we assume the angular motions are reasonably small. If that were not the case, one would have to use the full equations as shown for the airplane in Eqs. (10.16) and (10.17) where there are multiple nonlinearities involving the angular rates and angles. Another simplification below is that the longitudinal motion ($x, u, \theta$, and $q$) is uncoupled from the lateral motion ($y, v, \phi$, and $p$). If the vehicle is yawing ($r \neq 0$), there will be coupling between the longitudinal and lateral motion as can be seen from Eqs. (10.16) and (10.17). This coupling is ignored in the equations below, and the examples to follow will assume trajectories that do not produce any coupling for purposes of control design. The lontitudinal $x$-axis equations are:

$$\begin{bmatrix} \dot{x} \\ \dot{u} \\ \dot{q} \\ \dot{\theta} \\ \dot{T}_\theta \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & X_u & 0 & -g_o & 0 \\ 0 & M_u & 0 & 0 & M_\theta \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a \end{bmatrix} \begin{bmatrix} x \\ u \\ q \\ \theta \\ T_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ a \end{bmatrix} T_{lon}, \tag{10.36a}$$

$$\begin{bmatrix} \theta_m \\ x_m \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ q \\ \theta \\ T_\theta \end{bmatrix}$$

where

$x = x_m$ = measured position in the drone frame $x$ direction (see Fig. 10.55),

$u$ = velocity in the drone frame $x$ direction (see Fig. 10.55),

$q$ = angular rate about the positive drone frame $y$-axis, or pitch rate,

$\theta = \theta_m$ = measured pitch angle from horizontal,

$X_u$ = partial derivative of the aerodynamic force in $x$, direction with respect to perturbations in $u$,

$M_u$ = partial derivative of the aerodynamic (pitching) moment with respect to perturbations in $u$,

$M_\theta = 1/I_y$,

$T_\theta$ = pitching moment around $+y$ axis from rotors 1 and 3,

$T_{lon}$ = pitching torque command for rotors 1 and 3, produced by $\delta T_1 \cdot d$ and $-\delta T_3 \cdot d$, where $\delta T$ is the thrust change commanded to the rotor and $d$ is the moment arm,

$a$ = lag term representing the delay in the rotors producing the changed thrust and resulting torque,

$g_o$ = gravity (assumes pitch angle is small, hence $\cos(\theta) \simeq 1$).



If there are no aerodynamic terms, i.e., $M_u = 0$, $X_u = 0$, and there is no lag from the rotor commands, this yields $\Theta_m(s)/T_\theta(s) = 1/I_y s^2$ as was the case considered in Example 2.5, Eq. (2.15). If the thrust lag is included as shown in Eq. (10.36a), but again without aero terms, then it produces the transfer function $\Theta_m(s)/T_{lon}(s) = a/I_y/(s^2(s+a))$ as in Examples 5.12 and 5.16.

The lateral $y$-axis equations are:

$$\begin{bmatrix} \dot{y} \\ \dot{v} \\ \dot{p} \\ \dot{\phi} \\ \dot{T}_\phi \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Y_v & 0 & g_o & 0 \\ 0 & L_v & 0 & 0 & L_\phi \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a \end{bmatrix} \begin{bmatrix} y \\ v \\ p \\ \phi \\ T_\phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ a \end{bmatrix} T_{lat}, \tag{10.36b}$$

$$\begin{bmatrix} \phi_m \\ y_m \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ v \\ p \\ \phi \\ T_\phi \end{bmatrix}$$

where

$y = y_m$ = measured position in the drone frame $y$ direction (see Fig. 10.55),

$v$ = velocity in the drone frame $y$ direction (see Fig. 10.55),

$p$ = angular rate about the positive drone frame $x$-axis, or roll rate,

$\phi = \phi_m$ = measured roll-angle from horizontal,

$Y_v$ = partial derivative of the aerodynamic force in $y$ direction with respect to perturbations in $v$,

$L_v$ = partial derivative of the aerodynamic (rolling) moment with respect to perturbations in $v$,

$L_\phi = 1/I_x$,

$T_\phi$ = rolling moment around $+x$-axis from rotors 2 and 4,

$T_{lat}$ = rolling torque command for rotors 2 and 4, produced by $\delta T_2 \cdot d$ and $-\delta T_4 \cdot d$, where $\delta T$ is the thrust change commanded

to the rotor and $d$ is the moment arm,

$a$ = lag term representing the delay in the rotors producing the changed thrust and resulting torque,

$g_o$ = gravity (assumes roll angle is small, hence $\cos(\phi) \simeq 1$).

The rotational - $z$-axis equations are:

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} 1/I_z \\ 0 \end{bmatrix} T_\psi, \qquad (10.36c)$$

$$[\psi_m] = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix}$$

where

$r$ = angular rate about positive drone frame $z$–axis, or yaw rate, (see Fig. 10.55),

$\psi = \psi_m$ = measured azimuth angle of the drone frame $x$-axis with respect to North,

$T_\psi$ = commanded yawing moment from all rotors around + $z$-axis, produced by $\delta T_1, \delta T_2, \delta T_3$ and $\delta T_4 = -T_\psi$. (There is no lag here because the torque is applied directly by the motors.)

For the final dimension, altitude, all the rotors need their speed increased. However, because of the different rotational directions, the torques applied to rotors 1 and 3 will be in the opposite direction compared to rotors 2 and 4. Aerodynamic terms have little effect on these dynamics; therefore, the complete equations are

$$\begin{bmatrix} \dot{h} \\ \dot{w} \\ \dot{F}_h \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & Z_h \\ 0 & 0 & -a \end{bmatrix} \begin{bmatrix} h \\ w \\ F_h \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} F_{alt}, \qquad (10.36d)$$

$$h_m = [1\ 0\ 0] \begin{bmatrix} h \\ w \\ F_h \end{bmatrix}$$

where

$h = h_m$ = vertical position,

$w$ = vertical, $z$-axis, velocity,

$Z_h = 1/m_o$, $m_o$ = mass of the vehicle,

$F_h$ = vertical thrust,

$F_{alt}$ = commanded thrust from all rotors, produced by



$\delta T_1 = \delta T_3 = +F_{alt}$ and $\delta T_2 = \delta T_4 = -F_{alt}$.

To determine the position in an earth-fixed frame for the case when there is no yaw rate, $r$, the transformation matrix based on the rotation of the level body-fixed frame is required, thus

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ +\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \qquad (10.37)$$

where

$x_E$ = position in the earth-fixed frame with the $x$-axis pointing north,

$y_E$ = position in the earth-fixed frame with the $y$-axis pointing east.

For a fixed aximuth of $\psi = 0$, i.e, the $x$-axis is pointing north, and if there are no aero terms, $x_E(s)/\theta(s) = -g_o/s^2$ as we assumed in Example 5.16. As discussed above, for the case when there is a yaw rate, there are cross-coupling terms into the longitudinal and lateral axes as can be seen from Eqs (10.16), plus the kinematic relationships between the body-fixed and earth-fixed frames are significantly more complex (see Greenwood, 1988). For our ensuing example, we have chosen to select a trajectory where the equations above [10.36 and 10.37] are satisfactory. For arbitrary motion of a drone, substantial additional complexity is required to accurately simulate the motion in the body and earth-fixed frames. However, for design of feedback control laws of a drone, use of the four independent axes represented by Eqs. (10.36) are usually adequate.

Note, in the linear models above, the four axes are independent of each other providing the rotor commands are implemented as shown in the equations above, and as diagrammed in Fig. 10.56.

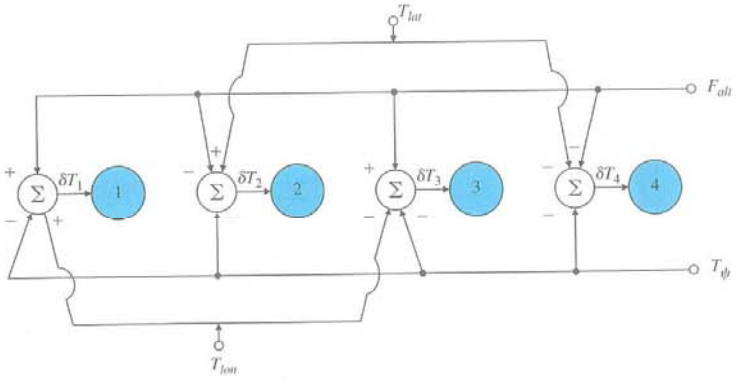



**Figure 10.56**
Required rotor commands for each of the four axes

STEP 5a. *Try a PID controller for a single axis.* For a set of aerodynamic and inertia terms that are representative of a small drone approximately 3 ft in diameter (Berrios, 2017), the parameters used were

$$Mu = 1.1,$$
$$Xu = -0.25,$$
$$M_\theta = 0.02,$$
$$g_o = 32.2,$$
$$a = 20.$$

These values in Eqs. (10.36a) produce the transfer functions,

$$\frac{\theta_m(s)}{T_{lon}(s)} = 0.4\frac{(s+0.25)}{(s-1.6\pm 2.8j)(s+3.4)(s+20)}. \quad (10.38a)$$

$$\frac{x_m(s)}{T_{lon}(s)} = -13\frac{1}{s(s-1.6\pm 2.8j)(s+3.4)(s+20)}. \quad (10.38b)$$

Note there are unstable open-loop poles in this loop. This is typical of helicopters flying near hover and account for the fact that helicopters tend to be difficult to fly near hover. The $(s + 20)$ term is due to the lag in the rotor coming up to the newly commanded speed and thrust; however the remaining set of poles and zero are often referred to as the "hovering cubic".

The use of sisotool allows us to find the PD controller (feedback of $\theta$ and $\dot{\theta}$),

$$D_{c1lon}(s) = 500(s+4) \quad (10.39a)$$

that produces a damping, $\zeta \simeq 0.6$, and $\omega_n \simeq 10$ rad/sec for the oscillatory roots of the this inner loop, as shown by the root locus in Fig. 10.57.

However, we see from the frequencey response in Fig. 10.58 that the system is conditionally stable due to the RHP open-loop roots that resulted from the aerodynamic terms.

**Figure 10.57**
Root locus of the inner loop with compensation given by Eq. 10.39a

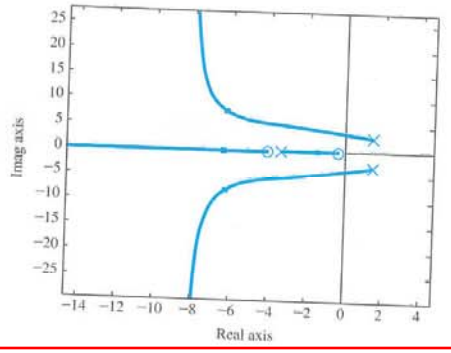
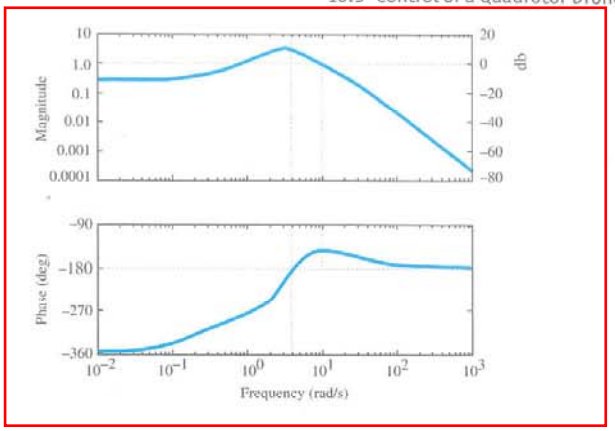





**Figure 10.58**
Open-loop frequency response plots for the inner $\theta$ loop



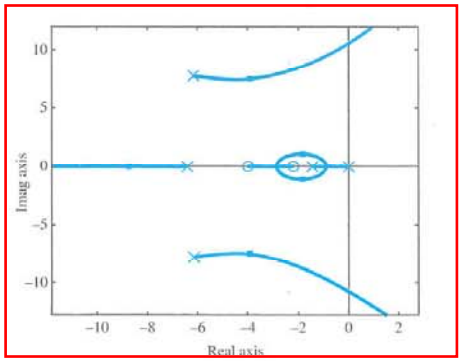

With the compensator as designed, the PM is 40°. Lowering the gain by a factor of 3.2 would cause an instability while there is no upper limit of the gain before instability is reached. Some researchers have found that additional lead in this inner loop substantially improves the performance in the presence of gusty wind conditions.

For the outer loop, we wish to command a change in position; therefore our measurement will be $x$. Using Eq. (10.36a) with the addlition of the inner-loop feedback from Eq. (10.39a), we find that an outer-loop PD controller

$$D_{c2lon}(s) = 0.4(s+2.2), \quad (10.39b)$$

produces the root locus as shown in Fig. 10.59, where the closed-loop roots for the selected gain are indicated. The system response to a step command in $x$ yields a satisfactory response as shown in Fig. 10.60. The rise time is approximately 0.6 seconds and the settling time is approximately 3 seconds, with an overshoot of less than 5%.

**Figure 10.59**
Root locus for the outer $x$-loop including the PD controller $D_{c2lon}(s) = 0.4(s + 2.2)$



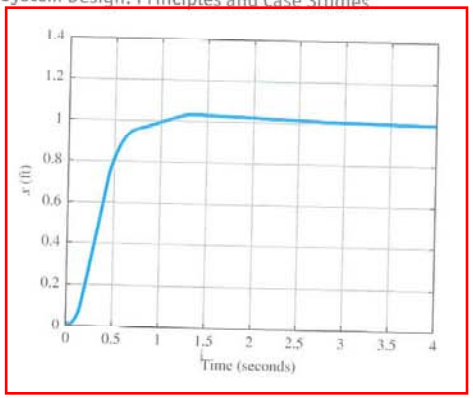



**Figure 10.60**
Step response of $x$ with the PD controller $D_{c2lon}(s) = 0.4(s + 2.2)$ for the outer loop

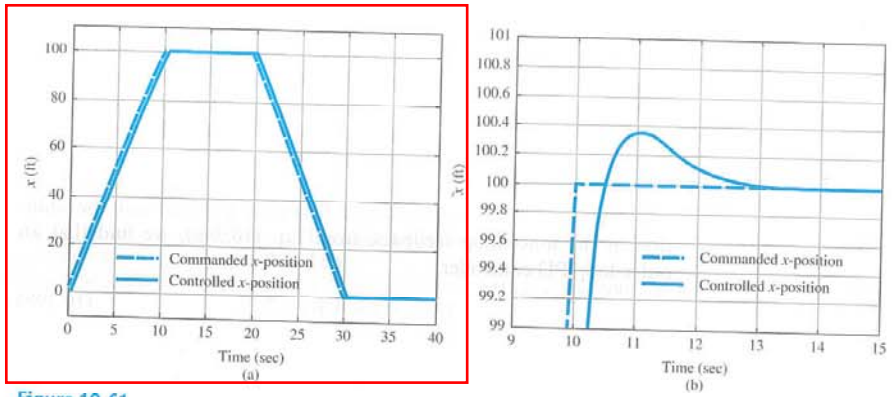



**Figure 10.61**
(a) $x$ response to a commanded trajectory (b) zoom in to a transition

However, a typical trajectory for a drone would involve moving to some location, performing a function, then returing to the starting point. As a partial test of this sort of trajectory in the $x$-direction only, we enter an $x$-command to leave the starting point with a speed of 10 ft/sec for 10 seconds, move along at that speed for 10 seconds, stop for 10 seconds, return at -10 ft/sec and stop after 10 seconds and stay stationary for 10 seconds. Fig. 10.61(a) is a plot that includes the commanded $x$-position, and the system's response to that command. To exhibit the accuracy better, a blow-up of the area near the transition from forward flight to the position hold at 10 seconds is shown in Fig. 10.61(b).

We see that the drone follows the commanded position very well, including the transition where a step change in velocity of 10 ft/sec is commanded and the resulting position error is less than 0.4 ft.



STEP 5b. *Try PID controllers for 2-D motion in the horizontal plane.* Unlike an airplane, a drone is equally comfortable traveling in the $y$-direction as it is traveling in the $x$-direction and equally comfortable going in the positive or negative directions! However, as stated earlier, most drones are equipped with a camera or other sensors that are pointing forward in a particular $x$-$y$ direction fixed in the drone. So there is a preferred direction of flight due to the placement of the observation sensors on the drone and the preferred method of guiding the craft is to rotate the vehicle about the $z$-axis so its sensors point in the direction of flight. Continuing with the evaluation for a trajectory over a path in the $x$-$y$ plane, let's assume the camera is oriented to point along the $x$-axis and can swing down in the $x$-$z$ plane. Initially, we want the drone to start going north so that $\psi = 0$ for the first portion of the trajectory. At a certain point, the camera detects something interesting, stops with the camera pointing straight down, rotates by 180°, moves sideways toward the east (which will be in the $-y$-direction due to the body-frame rotation by 180°), then moves back to the starting place without yaw rotation ($\psi = 180°$) and translation mostly in the $+x$-direction, but enough in the $+y$-direction so it returns to the starting location. For the $y$-direction flight, since the drone is not quite symmetric, the parameters used are:

$$\begin{aligned} Lv &= -0.5, \\ Yv &= -0.2, \\ L_\phi &= 0.016, \\ g_o &= 32.2, \\ a &= 20. \end{aligned}$$

These result in the transfer function of the inner loop

$$\frac{\phi_m(s)}{T_{lat}(s)} = 0.32\frac{(s+0.2)}{(s-1.2\pm 2.2j)(s+2.6)(s+20)}.$$

The use of sisotool allows us to find the inner loop PD controller (feedback of $\phi$ and $p$),

$$D_{c1lat}(s) = 500(s+3), \tag{10.40a}$$

that gives acceptable speed of response and damping. Then closing the outer loop with a PD controller (feedback of $y$ and $v$),

$$D_{c2lat}(s) = 0.25(s+3), \tag{10.40b}$$

completes the controller for commanding movement in the body-frame $y$-direction.

Control of the yaw angle (the azimuth of the body $x$-frame axis) is relatively straight forward since Eqs. (10.36c) show that

$$\frac{\psi_m(s)}{T_\psi(s)} = 0.005\frac{1}{s^2}.$$

The controller,

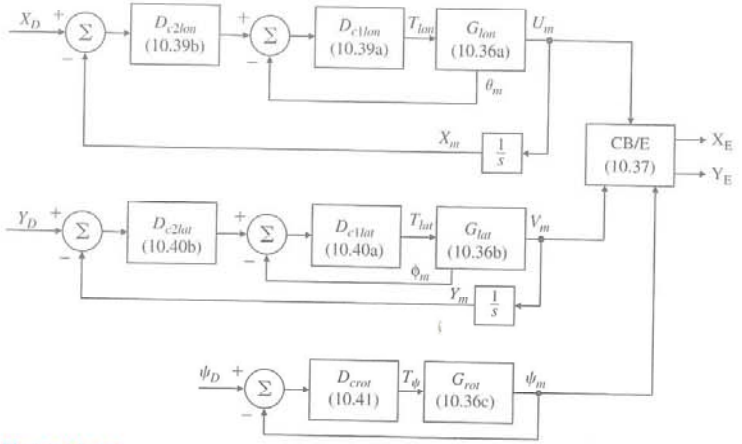



**Figure 10.62**
Block diagram of the controllers for motion in the $x$-$y$ plane

$$D_{crot}(s) = 1500(s+3), \tag{10.41}$$

provides a fast and well damped response, so the complete controller for motion in the $x$-$y$ plane is now designed. Fig. 10.62 shows these control loops in a block diagram.

The specific path chosen is one that goes away from the starting point at 20 ft/sec for 10 seconds along the $x$-axis pointing north, then stops and rotates for 5 seconds around the $z$-axis by 180°, then slides sideways to the east (negative $y$-axis) for 5 seconds at 5 ft/sec, and finally flies for 10 seconds with a velocity along the $+x$-axis of 20 ft/sec (which is now pointing south) and a velocity of 2.5 ft/sec along the $+y$-axis (which is now pointing west) so that the drone will return to the starting point. Fig. 10.63(a) shows the overall trajectory of the drone while 10.63(b) shows a scale change of the trajectory as it starts the trip back to the starting point to illustrate the errors better. The figure shows that the trajectory error never appears to exceed about a foot.

Based on the definition of the $x$-$y$ plane in Fig. 10.55 with the $z$-axis pointing down, and the fact that Matlab plots axes as in Figs. 10.63 as if the $z$-axis was coming out of the page, the trajectories in Figs. 10.63 are based on an observer who is looking up at the $x$-$y$ plane from underneath. That means that, indeed, the $x_E$-axis is pointing north and $y_E$-axis is pointing east as described above.)

Summarizing, we saw from this example that it is not necessary to command changes in the azimuth angle ($\psi$) as it would be for an airplane. However, there are reasons that one would want to command changes in the azimuth, such as orienting a camera which are typically not gimbaled to swivel around the body-frame $z$-axis. In general, if



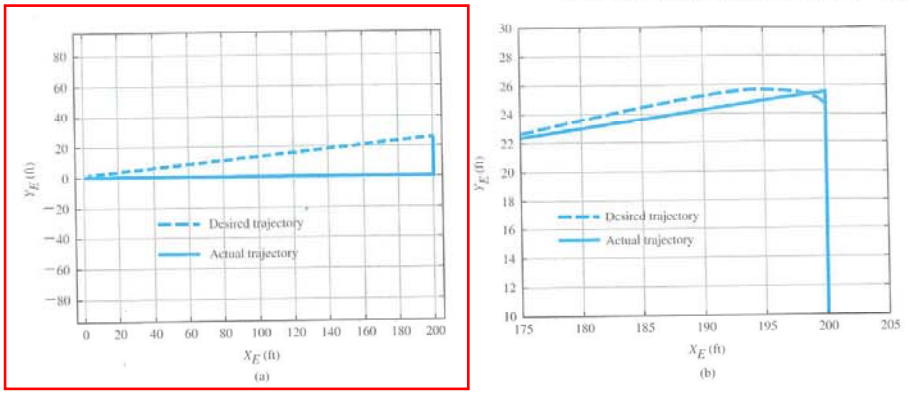



**Figure 10.63**
(a) Trajectory in the earth-fixed frame, (b) blow-up of the trajectory showing errors.

there is continuous motion in the body axis $x$-direction and continual changes in the direction of the body $x$-axis (that is, continual changes in $\psi$), there will be accelerations imparted in the body frame $y$-axis direction thus necessitating adding the coupling terms between the Eqs. (10.36a, b, and c). The trajectory selected above was chosen so those terms did not appear and, therefore, there were no errors induced by the motion. To accurately simulate a more general motion, the nonlinear terms and coupling need to be added; however, it is generally acceptable to use the uncoupled equations for purposes of control system design.

STEP 6. Try an optimal design: The quadrotor longitudinal $x$-axis state-space equations are given by Eq. (10.36a) and with $x$ selected as the system output, the resulting longitudinal transfer function is

$$\frac{x(s)}{T_{lon(s)}} = \frac{-12.88}{s(s-1.6\pm j2.8)(s+3.37)(s+20)}.$$

Selecting the following LQR weighting matrices,

$$\mathbf{Q} = \rho \mathbf{C}^T\mathbf{C}, \mathbf{R} = 1, \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

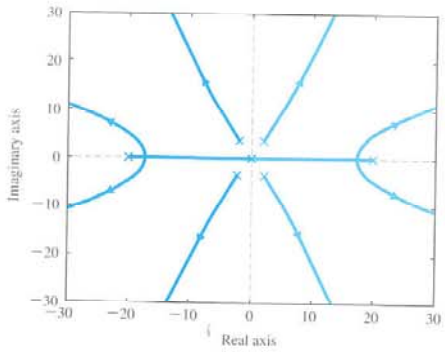
the symmetric root locus (SRL) for the system is shown in Figure 10.64. Choosing $\rho = 1e11$, we compute the state feedback gain matrix as

$$\mathbf{K} = \begin{bmatrix} -49020 & 6204 & 126690 & -316230 & 2.66 \end{bmatrix}.$$

The feedforward gain matrices are computed as

$$\mathbf{N_x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, N_u = 0, \bar{N} = N_u + \mathbf{K}\mathbf{N_x} = -3.1623e5, \mathbf{M} = \mathbf{B}\bar{N}.$$

**Figure 10.64**
Longitudinal SRL for quadrotor




Choosing the following estimator design weighting parameters

$$q = 1e7, \quad \mathbf{B}_1 = \mathbf{B},$$

the estimator gain matrix is determined to be

$$\mathbf{L} = \begin{bmatrix} 148 \\ -44 \\ -23 \\ 17 \\ -170 \end{bmatrix},$$

and results in the dynamic controller transfer function for the longitudinal axis given by

$$D_c(s) = \frac{1.59e7(s + 0.93 \pm j3.1)(s + 3.8)(s + 20)}{(s + 36.9)(s + 24.7 \pm j22.2)(s + 2.2 \pm j24.5)}.$$

The overall closed-loop system equations are

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{BK} - \mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}\bar{N} \\ \mathbf{M} \end{bmatrix} r,$$

$$y = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}.$$

The step response of the longitudinal control is shown in Figures 10.65. The rise time is 0.15 sec, the settling time is 0.8 sec, and there is less than 12% overshoot.

The lateral $y$-axis state-space equations are as in Eq. (10.36b) and result in the transfer function

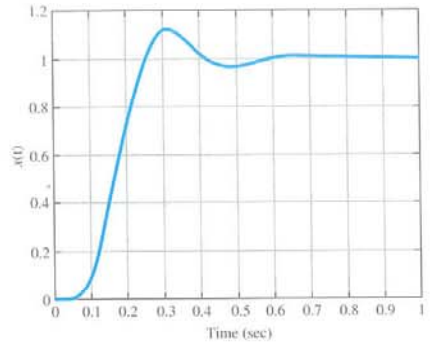
$$\frac{y(s)}{T_{lat(s)}} = \frac{10.3}{s(s - 1.2 \pm j2.2)(s + 2.6)(s + 20)}.$$

Choosing the following LQR weighting matrices,

$$\mathbf{Q} = \rho \mathbf{C}^T \mathbf{C}, \mathbf{R} = 1, \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$



**Figure 10.65**
Longitudinal step response for state-space design




with $\rho = 1e10$, the state feedback gain matrix is computed as

$$\mathbf{K} = \begin{bmatrix} 51641 & 7147 & 139480 & 316230 & 2.5 \end{bmatrix}.$$

The solution to the feedforward gain matrices yields

$$\mathbf{N_x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, N_u = 0, \ \bar{N} = N_u + \mathbf{KN_x} = 3.16e5, \mathbf{M} = \mathbf{B}\bar{N}.$$

Selection of the ensuing estimator design weighting parameters

$$q = 1e10, \quad \mathbf{B}_1 = \mathbf{B},$$

yields the estimator gain matrix

$$\mathbf{L} = \begin{bmatrix} 733 \\ 1539 \\ 261 \\ 38 \\ 52342 \end{bmatrix},$$

and results in the lateral dynamic controller transfer function

$$D_c(s) = \frac{-9.76e7(s + 2 \pm j5.4)(s + 5)(s + 20)}{(s + 46)(s + 30.6 \pm j28.9)(s + 0.78 \pm j31)}.$$

The step response of the lateral-axis and its associated control effort are very similar to Figures 10.65. The rise time is less than 0.15 sec, the settling time is around 0.8 sec, and the overshoot is less than 12%. As expected there is tremendous amount of symmetry between the longitudinal and lateral axes. The rotational $z$-axis state-space equations are as in Eq. (10.36c).

Selecting the ensuing LQR weighting matrices,

$$\mathbf{Q} = \rho \mathbf{C}^T \mathbf{C}, \mathbf{R} = 1, \mathbf{C} = [0 \quad 1],$$

with $\rho = 1e10$, the state feedback gain is computed to be

$$\mathbf{K} = [\ 11246 \quad 316227\ ].$$

The feedforward gain matrices are determined as

$$\mathbf{N_x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, N_u = 0, \bar{N} = N_u + \mathbf{K}\mathbf{N_x} = 3.16e5, \mathbf{M} = \mathbf{B}\bar{N}.$$

Selecting the ensuing estimator design parameters

$$q = 1e10, \mathbf{B}_1 = \mathbf{B},$$

gives the estimator gain matrix

$$\mathbf{L} = \begin{bmatrix} 500 \\ 31 \end{bmatrix},$$

and the dynamic controller transfer function for the rotational dynamics is given by

$$D_c(s) = \frac{-1.56e7(s+10)}{(s+43.9 \pm j43.9)}.$$

The rotational step response is shown in Figure 10.66. The system has a rise time of 0.05 sec, a settling time of 0.2 sec and less than 5% overshoot.

Figure 10.67 shows the trajectory following response of the quadrotor for the same desired trajectory as before. The quadrotor can follow the desired path very accurately and outperforms the classical design as seen from Figure 10.67. To be fair, iterations on the classical design could match this performance.

This example shows the essence of the control issues for the axes involving motion in the $x$-$y$ (horizontal) plane. In addition, there will be a requirement to control the altitude as well, but unlike an airplane

**Figure 10.66**
Rotational axis step response for state-space design

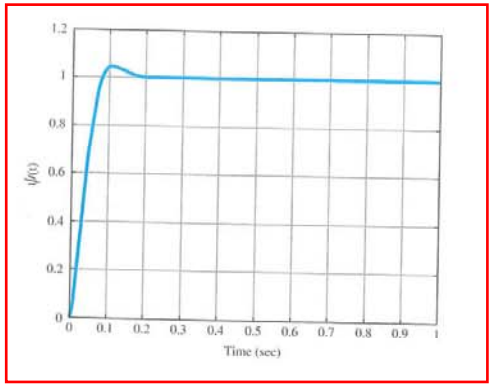
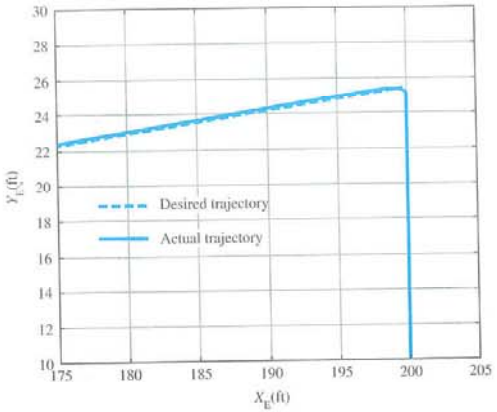





**Figure 10.67**
Trajectory-following response for state-space design




that requires pitch for altitude control, this motion is uncoupled from the other motions for normal fight.

STEP 7. As stated for STEP 5, the control laws determined by any method should be evaluated by a complete and accurate nonlinear simulation to determine the performance considering all known disturbances and the effect of all known model approximations.

## 10.6 Control of RTP Systems in Semiconductor Wafer Manufacturing

Figure 10.68 diagrams the major steps in the manufacture of an ultra-large-scale integrated (ULSI) circuit such as a microprocessor and some of the associated control aspects. Many of the steps described in this process, such as chemical vapor deposition or etching, must be done at closely controlled and timed temperature sequences (Sze, 1988). The standard practice for many years has been to perform these steps in batches on many wafers (silicon disks that contain many chips) at a time to produce large numbers of identical chips. In response to the demand for ever smaller critical dimensions of the devices on the chip, and to give more flexibility in the variety and number of chips to be produced, the makers of the tools for fabrication of integrated circuits are asked to provide more and more precise control of temperature and time profiles during thermal processing. In response to these demands, an important trend is to perform the thermal steps on one wafer at a time in a chamber with cold walls