8.2 TRACKING A REFERENCE INPUT

It is important, however, to be aware of an additional consideration. The optimal gains at each gain scheduling point should guarantee robust stability and performance; that is, they should guarantee stability and good performance at points near the design equilibrium point. Such robust stability can be verified after the LQ design by using multivariable frequency-domain techniques. These techniques are developed in Section 9.2, where the remarks on robustness to plant parameter variations are particularly relevant to gain scheduling.

8.2 TRACKING A REFERENCE INPUT

In control design we are often interested not in regulating the state near zero, which we discussed in the previous section, but in following a nonzero reference command signal. For example, we may be interested in designing a control system for optimal step-response shaping. This reference-input tracking or servodesign problem is important in the design of command augmentation systems (CAS). In this section and the next we cover tracker design.

It should be mentioned that the optimal linear quadratic (LQ) tracker of modern control is not a causal system (see Chapter 4). It depends on solving an “adjoint” system of differential equations backward in time, and so is impossible to implement. A suboptimal “steady-state” tracker using full state-variable feedback is available, but it offers no convenient structure for the control system in terms of desired dynamics like PI control, washout filters, and so on.

Modified versions of the LQ tracker have been presented in Davison and Ferguson (1981) and Gangsaas et al. (1986). There, controllers of desired structure can be designed since the approaches are output-feedback based. The optimal gains are determined numerically to minimize a PI with, possibly, some constraints.

It is possible to design a tracker by first designing a regulator using, for instance, Table 8.1-1. Then, some feedforward terms are added to guarantee perfect tracking (Kwakernaak and Sivan 1972). The problem with this technique is that the resulting tracker has no convenient structure and often requires derivatives of the reference command input. Moreover, servosystems designed using this approach depend on knowing the DC gain exactly. If the DC gain is not exactly known, the performance deteriorates. That is, the design is not robust to uncertainties in the model.

Here we discuss an approach to the design of tracking control systems that is very useful in several control applications. This approach will allow us to design a servo control system that has any structure desired. This structure will include a unity-gain outer loop that feeds the performance output back and subtracts it from the reference command, thus defining a tracking error $e(t)$ that should be kept small. See Fig. 8.2-1. It can also include compensator dynamics, such as a washout filter or an integral controller. The control gains are chosen to minimize a quadratic performance index (PI). We are able to give explicit design equations for the control gains (see Table 8.2-1), which may be solved using software available in the MATLAB Optimization Toolbox.
A problem with the tracker developed in this section is the need to select the design parameters $Q$ and $R$ in the PI in Table 8.2-1. Later, we show how modified PIs may be used to make the selection of $Q$ and $R$ almost transparent, yielding tracker design techniques that are very convenient for use in aircraft control systems design. We show, in fact, that *the key to achieving required performance using modern design strategies is in selecting an appropriate PI.*

**Tracker with Desired Structure**

In several control designs there is a wealth of experience and knowledge that dictates in many situations what sort of compensator dynamics yield good performance from the point of view of both the controls engineer and the pilot. For example, a washout circuit may be required, or it may be necessary to augment some feedforward channels with integrators to obtain a steady-state error of exactly zero. The control system structures used in classical aircraft design also give good robustness properties. That is, they perform well even if there are disturbances or uncertainties in the system. Thus, the multivariable approach developed here usually affords this robustness. Formal techniques for verifying closed-loop robustness for multivariable control systems are given in Chapter 9.

Our approach to tracker design allows controller dynamics of any desired structure and then determines the control gains that minimize a quadratic PI over that structure. Before discussing the tracker design, let us examine how the compensator dynamics may be incorporated into the system state equations.

A dynamic compensator of prescribed structure may be incorporated into the system description as follows. Consider the situation in Fig. 8.2-1 where the plant...
is described by
\[\dot{x} = Ax + Bu \quad (8.2-1)\]
\[y = Cx \quad (8.2-2)\]
with state \(x(t)\), control input \(u(t)\), and \(y(t)\) the measured output available for feedback purposes. In addition,
\[z = Hx \quad (8.2-3)\]
is a performance output, which must track the given reference input \(r(t)\). The performance output \(z(t)\) is not generally equal to \(y(t)\). It is important to realize that for perfect tracking it is necessary to have as many control inputs in vector \(u(t)\) as there are command signals to track in \(r(t)\) (Kwakernaak and Sivan 1972).

The dynamic compensator has the form
\[\dot{w} = Fw + Ge \quad (8.2-4)\]
with state \(w(t)\), output \(v(t)\), and input equal to the tracking error
\[e(t) = r(t) - z(t). \quad (8.2-5)\]

\(F, G, D,\) and \(J\) are known matrices chosen to include the desired structure in the compensator. The allowed form for the plant control input is
\[u = -Ky - Lv, \quad (8.2-6)\]
where the constant gain matrices \(K\) and \(L\) are to be chosen in the controls design step to result in satisfactory tracking of \(r(t)\). This formulation allows for both feedback and feedforward compensator dynamics.

These dynamics and output equations may be written in augmented form as
\[\frac{d}{dt} \begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} A & 0 \\ -GH & F \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} r \quad (8.2-7)\]
\[\begin{bmatrix} y \\ v \end{bmatrix} = \begin{bmatrix} C & 0 \\ -JH & D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ H \end{bmatrix} r \quad (8.2-8)\]
\[z = [0 \quad H] \begin{bmatrix} x \\ w \end{bmatrix}, \quad (8.2-9)\]
and the control input may be expressed as

\[ u = -[K\ L] \begin{bmatrix} y \\ v \end{bmatrix}. \] (8.2-10)

Note that, in terms of the augmented plant/compensator state description, the admissible controls are represented as a constant output feedback \([K\ L]\). In the augmented description, all matrices are known except the gains \(K\) and \(L\), which need to be selected to yield acceptable closed-loop performance.

A comment on the compensator matrices \(F, G, D, J\) is in order. Often, these matrices are completely specified by the structure of the compensator. Such is the case, for instance, if the compensator contains integrators. However, if it is desired to include a washout or a lead-lag, it may not be clear exactly how to select the time constants. In such cases, engineering judgment will usually give some insight. However, it may sometimes be necessary to go through the design to be proposed, and then if required return to readjust \(F, G, D, J\) and reperform the design.

**LQ Formulation of the Tracker Problem**

By redefining the state, the output, and the matrix variables to streamline the notation, we see that the augmented equations (8.2-7)–(8.2-9) that contain the dynamics of both the plant and the compensator are of the form

\[
\begin{align*}
\dot{x} &= Ax + Bu + Gr \\
y &= Cx + Fr \\
z &= Hx.
\end{align*}
\] (8.2-11)

In this description, let us take the state \(x(t) \in \mathbb{R}^n\), control input \(u(t) \in \mathbb{R}^m\), reference input \(r(t) \in \mathbb{R}^q\), performance output \(z(t) \in \mathbb{R}^q\), and measured output \(y(t) \in \mathbb{R}^p\). The admissible controls (8.2-10) are proportional output feedbacks of the form

\[ u = -Ky = -KCx - KFr, \] (8.2-14)

with constant gain \(K\) to be determined. This situation corresponds to the block diagram in Fig. 8.2-2. Since \(K\) is an \(m \times p\) matrix, we intend to close all the feedback loops simultaneously by computing \(K\).

Using these equations the closed-loop system is found to be

\[
\begin{align*}
\dot{x} &= (A - BKC)x + (G - BKF)r \\
&= A_x x + B_x r.
\end{align*}
\] (8.2-15)

In the remainder of this subsection, we shall use the formulation (8.2-11)–(8.2-14), assuming that the compensator, if required, has already been included
8.2 TRACKING A REFERENCE INPUT

in the system dynamics and demonstrating how to select the constant output feedback gain matrix \( K \) using LQ techniques.

Our formulation differs sharply from the traditional formulations of the optimal tracker problem studied in Chapter 4. Note that (8.2-14) includes both feedback and feedforward terms, so that both the closed-loop poles and compensator zeros may be affected by varying the gain \( K \) (see Example 8.2-1). Thus, we should expect better success in shaping the step response than by placing only the poles. We shall assume henceforth that the reference input \( r(t) \) is a step command with magnitude \( r_0 \). Designing for such a command will yield suitable time-response characteristics. Although our design is based on step-response shaping, it should be clearly realized that the resulting control system, if properly designed, will give good time responses for any arbitrary reference command signal \( r(t) \).

Let us now formulate an optimal control problem for selecting the control gain \( K \) to guarantee tracking of \( r(t) \). Then, we shall derive the design equations in Table 8.2-1, which are used to determine the optimal \( K \). These equations are solved using software like that found in the Optimization Toolbox.

**The Deviation System**

Denote steady-state values by overbars and deviations from the steady-state values by tildes. Then, the state, output, and control deviations are given by

\[
\begin{align*}
\tilde{x}(t) &= x(t) - \overline{x} \quad \text{(8.2-16)} \\
\tilde{y}(t) &= y(t) - \overline{y} = K \tilde{x} \quad \text{(8.2-17)} \\
\tilde{z}(t) &= z(t) - \overline{z} = H \tilde{x} \quad \text{(8.2-18)} \\
\tilde{u}(t) &= u(t) - \overline{u} = -KCx - KFr_0 - (KC\overline{x} - KFr_0) = -KC\tilde{x}(t) \quad \text{(8.2-19)}
\end{align*}
\]
or
\[ \tilde{u} = -K \tilde{y}. \] (8.2-20)

The tracking error \( e(t) = r(t) - z(t) \) is given by
\[ e(t) = \hat{e}(t) + \bar{e} \] (8.2-21)
with the error deviation given by
\[ \hat{e}(t) = e(t) - \bar{e} = (r_0 - Hx) - (r_0 - H\bar{x}) = -H \bar{x} \] (8.2-22)

or
\[ \hat{e} = -\bar{e}. \] (8.2-23)

Since in any acceptable design the closed-loop plant will be asymptotically stable, \( A_c \) is nonsingular. According to (8.2-15), at steady state
\[ 0 = A_c \bar{x} + B_c r_0, \] (8.2-24)
so that the steady-state state response \( \bar{x} \) is
\[ \bar{x} = -A_c^{-1} B_c r_0 \] (8.2-25)
and the steady-state error is
\[ \bar{e} = r_0 - H \bar{x} = (I + HA_c^{-1} B_c) r_0. \] (8.2-26)

To understand this expression, note that the closed-loop transfer function from \( r_0 \) to \( z \) (see (8.2-15) and (8.2-13)) is
\[ H(s) = H(sI - A_c)^{-1} B_c. \] (8.2-27)

The steady-state behavior may be investigated by considering the DC value of \( H(s) \) (i.e., \( s = 0 \)); this is just \(-HA_c^{-1} B_c\), the term appearing in (8.2-24).

Using (8.2-16), (8.2-19), and (8.2-23) in (8.2-15) the closed-loop dynamics of the state deviation are seen to be
\[ \dot{x} = A_c \bar{x} \]
\[ \bar{y} = C \bar{x} \]
\[ \bar{z} = H \bar{x} = -\bar{e} \] (8.2-28)
and the control input to the deviation system (8.2-26) is (8.2-19). Thus, the step-
response shaping problem has been converted to a regulator problem for the deviation system
\[ \dot{\tilde{x}} = A\tilde{x} + B\tilde{u}. \] (8.2-29)

Again, we emphasize the difference between our approach and the traditional one described in Chapter 4. Once the gain \( K \) in (8.2-19) has been found, the control for the plant is given by (8.2-14), which inherently has both feedback and feedforward terms. Thus, no extra feedforward term need be added to make \( e \) zero.

Performance Index

To make the tracking error \( e(t) \) in (8.2-20) small, we propose to attack two equivalent problems: the problem of regulating the error deviation \( \tilde{e}(t) = -\tilde{z}(t) \) to zero, and the problem of making small the steady-state error \( e \). Note that we do not assume a type 1 system, which would force \( e \) to be equal to zero. This can be important in aircraft controls, where it may not be desirable to force the system to be of type 1 by augmenting all control channels with integrators. This augmentation complicates the servo structure. Moreover, it is well known from classical control theory that suitable step responses may often be obtained without resorting to inserting integrators in all the feedforward channels.

To make small both the error deviation \( \tilde{e}(t) = -H\tilde{x}(t) \) and the steady-state error \( e \), we propose selecting \( K \) to minimize the performance index (PI)
\[ J = \int_0^\infty (\tilde{e}^T\tilde{e} + \tilde{u}^TR\tilde{u}) \, dt + \frac{1}{2}e^TVe, \] (8.2-30)
with \( R > 0, V \geq 0 \) design parameters. The integrand is the standard quadratic PI with, however, a weighting \( V \) included on the steady-state error. Note that the PI weights the control deviations and not the controls themselves. If the system is of type 1, containing integrators in all the feedforward paths, then \( V \) may be set to zero since the steady-state error is automatically zero.

Making small the error deviation \( \tilde{e}(t) \) improves the transient response, while making small the steady-state error \( e(t) \) improves the steady-state response. If the system is of type 0, these effects involve a trade-off, so that then there is a design trade-off involved in selecting the size of \( V \). We can generally select \( R = rI \) and \( V = vI \), with \( r \) and \( v \) scalars. This simplifies the design since now only a few parameters must be tuned during the interactive design process. According to (8.2-21), \( \tilde{e}^T\tilde{e} = \tilde{x}^TH^TH\tilde{x} \). Referring to Table 8.1-1, therefore, it follows that the matrix \( Q \) there is equal to \( H^TH \), where \( H \) is known. That is, weighting the error deviation in the PI has already shown us how to select the design parameter \( Q \), affording a considerable simplification.

The problem we now have to solve is how to select the control gains \( K \) to minimize the PI \( J \) for the deviation system (8.2-29). Then, the tracker control for the original system is given by (8.2-14).
We should point out that the proposed approach is suboptimal in the sense that minimizing the PI does not necessarily minimize a quadratic function of the total error \( e(t) = \bar{e} + \hat{e}(t) \). It does, however, guarantee that both \( \hat{e}(t) \) and \( e \) are small in the closed-loop system, which is a design goal.

Solution of the LQ Tracker Problem

It is now necessary to solve for the optimal feedback gain \( K \) that minimizes the PI. The design equations needed are now derived. They appear in Table 8.2-1. By using (8.2-26) and a technique like the one used in Section 8.3 (see problems), the optimal cost is found to satisfy

\[
J = \frac{1}{2} \bar{x}^T(0)P\bar{x}(0) + \frac{1}{2} \bar{e}^T V \bar{e},
\]

(8.2-31)

with \( P \geq 0 \) the solution to

\[
0 = g \equiv A_c^T P + PA_c + Q + C^T K^T R K C,
\]

(8.2-32)

with \( Q = H^T H \) and \( e \) given by (8.2-24).

In our discussion of the linear quadratic regulator we assumed that the initial conditions were uniformly distributed on a surface with known characteristics. While this is satisfactory for the regulator problem, it is an unsatisfactory assumption for the tracker problem. In the latter situation the system starts at rest and must achieve a given final state that is dependent on the reference input, namely (8.2-23). To find the correct value of \( \bar{x}(0) \), we note that, since the plant starts at rest (i.e., \( x(0) = 0 \)), according to (8.2-25)

\[
\bar{x}(0) = -\bar{x},
\]

(8.2-33)

so that the optimal cost (8.2-31) becomes

\[
J = \frac{1}{2} \bar{x}^T P \bar{x} + \frac{1}{2} \bar{e}^T V \bar{e} = \frac{1}{2} \text{tr}(PX) + \frac{1}{2} \bar{e}^T V \bar{e},
\]

(8.2-34)

with \( P \) given by (8.2-32), \( e \) given by (8.2-24), and

\[
X \equiv \bar{x}^T = A_c^{-1} B_c r_0^T B_c^T A_c^{-T},
\]

(8.2-35)

with \( A_c^{-T} = (A_c^{-1})^T \). The optimal solution to the unit-step tracking problem, with (8.2-11) initially at rest, may now be determined by minimizing \( J \) in (8.2-34) over the gains \( K \), subject to the constraint (8.2-32) and equations (8.2-24), (8.2-35).

This algebraic optimization problem can be solved by any well-known numerical method (cf Press et al. 1986, Söderström 1978). A good approach for a fairly small number \( (mp \leq 10) \) of gain elements in \( K \) is the SIMPLEX minimization routine (Nelder and Mead 1964). To evaluate the PI for each fixed value of \( K \) in the iterative solution procedure, one may solve (8.2-32) for \( P \) using...
8.2 TRACKING A REFERENCE INPUT

the `lyap.m` subroutine in MATLAB (Control System Toolbox) and then employ (8.2-34). Software for determining the optimal control gains $K$ can be found in the Optimization Toolbox.

Design Equations for a Gradient-based Solution

As an alternative solution procedure one may use gradient-based techniques (e.g., the Davidson-Fletcher-Powell algorithm [Press et al. 1986]), which are generally faster than non-gradient-based approaches.

To find the gradient of the PI with respect to the gains, define the Hamiltonian

$$H = \text{tr}(PX) + \text{tr}(gS) + \frac{1}{2}e^T V e,$$

(8.2-36)

with $S$ a Lagrange multiplier. Now, using the basic matrix calculus identities

$$\frac{\partial Y^{-1}}{\partial x} = -Y^{-1} \frac{\partial Y}{\partial x} Y^{-1},$$

(8.2-37)

$$\frac{\partial UV}{\partial x} = \frac{\partial U}{\partial x} V + U \frac{\partial V}{\partial x},$$

(8.2-38)

$$\frac{\partial y}{\partial x} = \text{tr} \left[ \frac{\partial y}{\partial z} \cdot \frac{\partial z^T}{\partial x} \right],$$

(8.2-39)

we may proceed as in the previous section, with, however, a little more patience due to the extra terms (see the problems at the end of the chapter), to obtain the necessary conditions for a solution given in Table 8.2-1.

To find $K$ by a gradient minimization algorithm, it is necessary to provide the algorithm with the values of $J$ and $\partial J/\partial K$ for a given $K$. The value of $J$ is given by the expression in Table 8.2-1 for the optimal cost. To find $\partial J/\partial K$ given $K$, solve (8.2-40), (8.2-41) for $P$ and $S$. Then, since these equations hold, $\partial J/\partial K = \partial H/\partial K$, which may be found using (8.2-42).

These equations should be compared to those in Table 8.1-1. Note that the dependence of $X$ on the gain $K$ (see (8.2-45)) and the presence of $e$ in the PI have resulted in extra terms being added in (8.2-42).

Determining the Optimal Feedback Gain

The issues in finding the optimal output-feedback gain $K$ in the tracker problem of Table 8.2-1 are the same as those discussed in connection with the regulator problem of Table 8.1-1: choice of $Q$ to satisfy detectability, choice of solution technique, finding an initial stabilizing gain, and iterative design by tuning $Q$ and $R$. We emphasize that there are only a few design parameters in our approach, namely $r$ and $v$ (since we can generally select $R = rI$, $V = vI$). Thus, it is not difficult or time-consuming to come up with good designs. Much of the simplicity of our approach derives from the fact that $Q$ in the PI is equal to $H^T H$, which is known. Let us now illustrate the servodesign procedure by an example.
OUTPUT FEEDBACK AND STRUCTURED CONTROL

TABLE 8.2-1 LQ Tracker with Output Feedback

<table>
<thead>
<tr>
<th>System model:</th>
<th>[ \dot{x} = Ax + Bu + Gr ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ y = Cx + Fr ]</td>
</tr>
<tr>
<td></td>
<td>[ z = Hx ]</td>
</tr>
<tr>
<td>Control:</td>
<td>[ u = -Ky ]</td>
</tr>
<tr>
<td>Performance index:</td>
<td>[ J = \frac{1}{2} \int_0^\infty \left( \dot{x}^T Q \dot{x} + \dot{u}^T R \dot{u} \right) dt + \frac{1}{2} e^T V e, \text{ with } Q = H^T H ]</td>
</tr>
</tbody>
</table>

Optimal output feedback gain:

\[
0 = \frac{\partial H}{\partial S} = A^T c P + PA_c + Q + C^T K^T R K C \\
0 = \frac{\partial H}{\partial P} = A_c S + S A_c^T + X \\
0 = \frac{1}{2} \frac{\partial H}{\partial K} = R K C S^T - B^T P S C^T + B^T A_c^{-T} (P + H^T V H)^{-1} X^T \\
\]

with \( r \) a unit step of magnitude \( r_0 \) and

\[
\bar{x} = -A_c^{-1} B_c r_0 \\
\bar{y} = C \bar{x} + Fr_0 \\
X = \bar{x}^T = A_c^{-1} B_c r_0 \bar{x}^T B_c^T A_c^{-T},
\]

where

\[ A_c = A - B K C, B_c = G - B K F \]

Optimal cost:

\[ J = \frac{1}{2} \text{tr}(PX) + \frac{1}{2} \bar{y}^T V \bar{y} \]

Example 8.2-1. Normal Acceleration CAS

In this example, we show that, using the LQ design equations in Table 8.2-1, we can close all the loops simultaneously. Thus, the design procedure is more straightforward. We also demonstrate that using LQ design, the algorithm automatically selects the zero of the compensator for optimal performance.

a. Control System Structure

The normal acceleration control system is shown in Fig. 8.2-3, where \( r \) is a reference step input in g’s and \( u(t) \) is the elevator actuator voltage. An integrator has been added in the feedforward path to achieve zero steady-state error. The performance output that should track the reference command \( r \) is \( z = n_z \), so that the tracking error is \( e = r - n_z \). The
8.2 TRACKING A REFERENCE INPUT

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.

The linearized F-16 dynamics about the nominal flight condition are augmented to include the elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

\[
\dot{x} = Ax + Bu + Gr
\]

\[
y = Cx + Fr
\]

\[
z = Hx,
\]

with \(\dot{\varepsilon}(t)\) the integrator output and \(\alpha_F\) the filtered measurements of angle of attack.
The factor of 57.2958 is added to convert angles from radians to degrees. The control input is

\[ u = -Ky = -[k_\alpha \ k_q \ k_e \ k_I]y = -k_\alpha \alpha - k_q q - k_e e - k_I \varepsilon. \]  

(6)

It is desired to select the four control gains to guarantee a good response to a step command \( r \). Note that \( k_\alpha \) and \( k_q \) are feedback gains, while \( k_e \) and \( k_I \) are feedforward gains.

The proportional-plus-integral compensator is given by

\[ k_e + \frac{k_I}{s} = k_e \frac{s + k_I/k_e}{s}, \]  

(7)

which has a zero at \( s = k_I/k_e \). Since the LQ design algorithm will select all four control gains, it will automatically select the optimal location for the compensator zero.

**b. Performance Index and Determination of the Control Gains**

Due to the integrator, the system is of type 1. Therefore, the steady-state error \( \bar{e} \) is automatically equal to zero. A natural PI thus seems to be

\[ J = \frac{1}{2} \int_0^\infty (\bar{e}^2 + \rho \bar{a}^2) \, dt, \]  

(8)

with \( \rho \) a scalar weighting parameter. Since \( \bar{e} = H\bar{x} \), this corresponds to the PI in Table 8.2-1 with

\[ Q = H^T H = \begin{bmatrix} 264.3876 & 15.9153 & -0.7889 & 0 & 0 \\ 15.9153 & 0.9580 & -0.0475 & 0 & 0 \\ -0.7889 & -0.0475 & 0.0024 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \]  

(9)

This is, unfortunately, not a suitable \( Q \) matrix since \((H, A)\) is not observable in open loop. Indeed, according to Fig. 8.2-3 observing the first two states \( \alpha \) and \( q \) can never give information about \( \varepsilon \) in the open-loop configuration (where the control gains are zero). Thus, the integrator state is unobservable in the PI. Since the integrator pole is at \( s = 0 \), \((H, A)\) is undetectable (unstable unobservable pole), so that any design based on (9)
would, in fact, yield a value for the integral gain of \( k_I = 0 \). To correct the observability problem here let us select

\[
Q = H^T H = \begin{bmatrix}
264.3876 & 15.9153 & -0.7889 & 0 & 0 \\
15.9153 & 0.9580 & -0.0475 & 0 & 0 \\
-0.7889 & -0.0475 & 0.0024 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad (10)
\]

where we include a weighting on \( \varepsilon(t) \) to make it observable in the PI.

Now, we selected \( \rho = 1 \) and solved the design equations in Table 8.2-1 for the optimal control gain \( K \). For this \( Q \) and \( \rho \) the feedback matrix was

\[
K = [0.0005 \ -0.1455 \ 1.1945 \ 1.0000] \quad (11)
\]

and the closed-loop poles were

\[
s = -1.24 \pm j0.79 \\
\ -1.28, \ -10.00 \ -20.28. \quad (12)
\]

These yield a system that is not fast enough; the complex pair is also unsuitable in terms of flying qualities requirements.

After repeating the design using several different \( Q \) and \( \rho \), we decided on

\[
Q = H^T H = \begin{bmatrix}
264.3876 & 15.9153 & -0.7889 & 0 & 0 \\
15.9153 & 0.9580 & -0.0475 & 0 & 0 \\
-0.7889 & -0.0475 & 0.0024 & 0 & 0 \\
0 & 0 & 0 & 0 & 100 \\
0 & 0 & 0 & 0 & 100
\end{bmatrix}, \quad (13)
\]

\( \rho = 0.01 \). The decreased control weighting \( \rho \) has the effect of allowing larger control effort and so speeding up the response. The increased weighting on the integrator output \( \varepsilon(t) \) has the effect of forcing \( \eta \) to its final value of \( r \) more quickly, hence also speeding up the response. The increased weighting on the second state component \( q \) has the effect of regulating excursions in \( \tilde{q}(t) \) closer to zero, and hence of providing increased damping.

With this \( Q \) and \( \rho \) the control matrix was

\[
K = [-0.0075 \ -1.0504 \ 25.6504 \ 100.0000] \quad (14)
\]

and the closed-loop poles were at

\[
s = -2.89 \pm j3.76 \\
\ -16.47 \pm j3.76 \\
\ -10. \quad (15)
\]

The closed-loop step response is shown in Fig. 8.2-4; it is fairly fast with an overshoot of 6%. Note the slight delay due to the nonminimum-phase zero. Further tuning of the
elements of $Q$ and $R$ could provide less overshoot, a faster response, and a smaller gain for the angle-of-attack feedback.

According to (7), the compensator zero has been placed by the LQ algorithm at

$$s = -k_i/k_e = -4.06.$$  \hspace{1cm} (16)

c. Discussion

We can now emphasize an important aspect of modern LQ design. As long as $Q \geq 0$, $R > 0$, and $(\sqrt{Q}, A)$ is observable, the closed-loop system designed using Table 8.2-1 is generally stable. Thus, the LQ theory has allowed us to tie the control system design to some design parameters that may be tuned to obtain acceptable behavior—namely, the elements of weighting matrices $Q$ and $R$. If the optimal control gain $K$ does not result in suitable performance in terms of time responses and closed-loop poles, the elements of $Q$ and $R$ can be changed and the design repeated. The importance of this is that for admissible $Q$ and $R$, closed-loop stability is guaranteed. A disadvantage of the design equations in Table 8.2-1 is the need to try different $Q$ and $R$ until suitable performance is obtained, as well as the need for $(H, A)$ to be observable.

Another point needs to be made. Using the control (6) in (2) and using (3), yields the closed-loop plant

$$\dot{x} = (A - BKC)x + (G - BKF)r,$$  \hspace{1cm} (17)

whence the closed-loop transfer function from $r(t)$ to $z(t)$ is

$$H(s) = H(sI - (A - BKC)^{-1}(G - BKF)).$$  \hspace{1cm} (18)

Note that the transfer function numerator depends on the optimal gain $K$. That is, this scheme uses optimal positioning of both the poles and zeros to attain step-response shaping.

FIGURE 8.2-4 Normal acceleration step response.
8.3 TRACKING BY REGULATOR REDESIGN

\textit{d. Selection of Initial Stabilizing Gain}

To initialize the algorithm that determines the optimal $K$ by solving the design equations in Table 8.2-1, it is necessary to find an initial gain that stabilizes the system. In this example, we simply selected gains with signs corresponding to the static loop sensitivity of the individual transfer functions, since this corresponds to negative feedback. The static loop sensitivities from $u$ to $\alpha$ and from $u$ to $q$ are negative, so positive gains were chosen for these loops (note $(A - BK_C)$). The initial gain used was

$$K = \begin{bmatrix} 1 & 1 & -1 & 1 \end{bmatrix}.$$  

\hspace{1cm} (19)

\section*{8.3 TRACKING BY REGULATOR REDESIGN}

In this section we discuss an alternative tracker design technique that amounts to first designing a regulator and then adding some feedforward terms to guarantee tracking behavior. This technique does not have the advantages of the direct design approach of the previous section. There, we were able to

1. Select the form of the compensator, including a unity outer loop to allow feedforward of the error.
2. Simplify the design stage by using only a few design parameters in the PI.

However, the approach to be presented here is simple to understand and may be quite useful in some applications. It will also give us some more insight on the tracking problem.

Let us suppose that the plant-plus-compensator in Fig. 8.2-1 is described, using the technique in Section 8.2, as

$$\dot{x} = Ax + Bu + Er$$  \hspace{1cm} (8.3-1)

$$y = Cx + Fr$$  \hspace{1cm} (8.3-2)

$$z = Hx,$$  \hspace{1cm} (8.3-3)

where $y(t)$ is the measurable output available for feedback and the performance output $z(t)$ is required to track the reference input $r(t)$. The tracking error is

$$e = r - z.$$  \hspace{1cm} (8.3-4)

Thus, this augmented description contains the dynamics of both the plant and the compensator. It is desired to select the control input $u(t)$ so that the tracking error goes to zero.

\textbf{Deviation System}

For perfect tracking, there must exist an ideal plant state $x^*$ and an ideal plant input $u^*$ such that

$$\dot{x}^* = Ax^* + Bu^* + Er$$  \hspace{1cm} (8.3-5)