Chapter 3
Command Tracking and the Robust Servomechanism

3.1 Introduction

Most industrial control problems require the control system to accurately track commands. This requirement distinguishes these problems from regulation in which the state is driven to zero. From classical control theory, we know that in order to track a constant command with zero error, we need to add integral error control action into the controller. For single-input single-output (SISO) systems, the loop transfer function $L(s)$ can be written as

$$L(s) = \frac{K(b_0 s^m + \cdots + b_{m-1}s + 1)}{s^p(a_0 s^n + \cdots + a_{n-1}s + 1)} \tag{3.1}$$

where the gain K and the polynomial coefficients a_i and b_i are real constants. The *type* of the control system depends upon the order p of the pole of $L(s)$ at $s = 0$. The number of finite zeros, their location, or the location of the poles are not important to specify the system type. The system type p, where $p = 0, 1, 2, \cdots$ indicates how many integrators are present in the control system. We know that in order to track a constant command $r(t) = \text{constant}$, and to produce zero steady-state error, an integrator is needed, $p \geq 1$, creating (at a minimum) a type 1 system. In order to track a type 1 input, the control system will need two integrators, creating a type 2 system. Thus, to track commands accurately, the class of commanded signals must be known, and the controller must be augmented with enough integrators to produce zero steady-state errors.

When these integrators are added to the control system for command tracking, they also provide disturbance rejection within the same class, that is, a type 1 control system can track constant commands and reject constant disturbances. Similarly, a type 2 system can track ramp inputs and reject ramp disturbances.

Basically, the augmentation of the system with these integrators for command tracking requires embedding into the system a model of the class of signals that the system will track. This is often referred to as the *internal model principle* [1].

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_3,

For instance, when tracking a constant command and adding a single integrator, we have embedded the command generation internal model $\dot{r} = 0$ into the system.

In the previous chapter, we have illustrated the use of linear quadratic optimal control theory to design a controller and examined the excellent stability properties provided by that method. The linear quadratic regulator (LQR) forces the system state to go to zero, forming a type 0 control system. If one wants to track a constant command using such an LQR controller, the system would have a steady-state offset error to the command. We know from Eq. (3.1) that in order to track a constant command with zero error, we need to add an integrator, creating a type 1 control system.

A natural extension of the LQR method presented in the previous chapter would be to add an integral control action into the controller to produce zero steady errors, while tracking constant commands. The number of integrators that would need to be added depends upon the commanded signal (whether it is a constant, a ramp, or other type of signal).

This chapter presents a systematic process for building an augmented state space model called the *servomechanism design model* [2]. This state space description embeds a model of the class of signals to be tracked, such that when optimal control theory is applied, the state regulation provides accurate tracking of the selected class of external commands. This system is then decomposed into two parts: a servo tracking controller for command following and a state feedback component for stabilization. In aerospace, this approach is often used to design flight control systems for both manned and unmanned aerial vehicles. The resulting control architecture provides accurate command tracking and a robust control system design with predictable and robust performance. The meaning of controller robustness was introduced in Chap. 2. It requires the minimum singular value of the return difference matrix having magnitude greater than 1. This topic of robustness in the frequency domain is covered in great detail later in Chap. 5.

3.2 The Servomechanism Design Model

Consider the following finite dimensional linear-time-invariant state space model:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew \\ y &= Cx + Du \end{aligned} \tag{3.2}$$

with an unknown bounded disturbance w and with the signals $x \in R^{n_x}$, $u \in R^{n_u}$, and $y \in R^{n_y}$ representing the system state, control, and output, respectively. We assume that the system is both controllable and observable. We would like a preselected subset of the output vector y to track the command input vector $r \in R^{n_r}$, and we assume that the dimension of r is less than or equal to the number of the system outputs (i.e., $n_y \geq n_r$). It is also assumed that the p^{th} order differential equation for $r(t)$ is given, with the following model:

Table 3.1 Internal models for command generation

Command signal $r(t)$	Differential equation	Model parameters
Constant	$\dot{r} = 0$	$p = 1, a_1 = 0$
Ramp	$\ddot{r} = 0$	$p = 2, a_1 = a_2 = 0$
Sinusoid	$\ddot{r} = -\omega_0^2 r$	$p = 2, a_1 = 0, a_2 = -\omega_0^2$

$$\overset{(p)}{r} = \sum_{i=1}^{p} a_i \overset{(p-i)}{r} \tag{3.3}$$

where the scalar coefficients a_i are known and the superscript (i) denotes the i^{th} derivative. Using the model in (3.3), examples for typical signals are shown in Table 3.1.

The polynomial formed by the Laplace transformation of (3.3) is

$$a(s) = s^p + \sum_{i=1}^{p} a_i s^{p-i}, \tag{3.4}$$

and it gives a known class of inputs without the knowledge of their magnitudes. Our control goal is to track this command with zero steady-state error. For disturbance inputs, we assume the same model as $r(t)$:

$$\overset{(p)}{w} = \sum_{i=1}^{p} a_i \overset{(p-i)}{w} \tag{3.5}$$

where $w_0 = w(0)$ is unknown.

Let us define the tracking error signal as

$$e = y_c - r \tag{3.6}$$

where $y_c \in R^{n_r}$ is the subset of the output y to be controlled and $e \in R^{n_r}$. The error signal is defined here as $e = y_c - r$ so that we can apply negative feedback of the errors and their derivatives in forming the feedback control. We will also arrange the output vector so that the first n_r variables in y define y_c. Thus,

$$y = \begin{bmatrix} y_c^T & y_{nc}^T \end{bmatrix}^T \tag{3.7}$$

where y_{nc} are output variables that are not controlled. The model for $y_c \in R^{n_r}$ is

$$y_c = C_c x + D_c u \tag{3.8}$$

This is the *regulated* system output. Tracking in y_c is the same as regulation in e; therefore, the objective is to make the error approach zero $e \to 0\,(y_c \to r)$, as $t \to \infty$, in the presence of unmeasurable disturbance w, in a robust manner with respect to the plant description. Taking (3.6) and differentiating p times, the resulting differential equation for the error may be written as

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = \overset{(p)}{y_c} - \sum_{i=1}^{p} a_i \overset{(p-i)}{y_c} - \left(\overset{(p)}{r} - \sum_{i=1}^{p} a_i \overset{(p-i)}{r} \right) \tag{3.9}$$

From (3.3), the bracketed term in the right side of (3.9) will be zero. Using (3.8), we have

$$\overset{(p-i)}{y_c} = C_c \overset{(p-i)}{x} + D_c \overset{(p-i)}{u} \tag{3.10}$$

Substituting this into (3.9) yields

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = C_c \left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right] + D_c \left[\overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \right] \tag{3.11}$$

This system represents a set of coupled ordinary differential equations. Let ξ and μ be defined as

$$\xi = \overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \tag{3.12}$$

$$\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \tag{3.13}$$

The error equation is

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = C_c \xi + D_c \mu \tag{3.14}$$

Differentiating (3.12), we get

$$\dot{\xi} = \overset{(p+1)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i+1)}{x} \tag{3.15}$$

Using (3.2) for $\dot{x}$ results in

$$\dot{\xi} = A \underbrace{\left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right]}_{\xi} + B \underbrace{\left[\overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \right]}_{\mu} + E \underbrace{\left[\overset{(p)}{w} - \sum_{i=1}^{p} a_i \overset{(p-i)}{w} \right]}_{=0} \tag{3.16}$$

where (3.5) shows the last term to be zero. We can rewrite (3.16) as

$$\dot{\xi} = A\xi + B\mu \tag{3.17}$$

which is the original system model minus the disturbances.

The servomechanism design model is formed by creating a new state space model, containing the error dynamics and the system model from (3.17). The new state vector is z, and its components are the errors $e, \cdots, \overset{(p-1)}{e}$, with the vector ξ. The error is a linear combination of ξ and μ from (3.14). Augmenting z with these derivatives and ξ defines z to be

$$z = \begin{bmatrix} e \\ \dot{e} \\ \vdots \\ \overset{(p-1)}{e} \\ \xi \end{bmatrix} \tag{3.18}$$

This new state vector z has dimension $(n_x + p \times n_r)$. Differentiating (3.18) yields the *servomechanism design model*:

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.19}$$

where $\tilde{A}$ and $\tilde{B}$ are given by

$$\tilde{A} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & 0 & I & 0 \\ a_p I & a_{p-1} I & \cdots & a_2 I & a_1 I & C_c \\ 0 & \cdots & \cdots & \cdots & 0 & A \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ D_c \\ B \end{bmatrix} \tag{3.20}$$

The robust servomechanism LQR solution is obtained by applying linear quadratic regulator theory to (3.19). By regulating z, we regulate to zero both e, its $(p-1)$ derivatives, and ξ. In steady state, this allows the state vector x to be nonzero, in which case, $C_c x + D_c u = r$. This control formulation adds the desired integral control action acting on the command error.

3.2.1 Controllability of the Servomechanism Design Model

If we apply the Hautus controllability tests to the servomechanism design model in (3.19), for the system to be controllable, we must have

$$\operatorname{rank}\left[sI - \tilde{A} | \tilde{B}\right] = n_x + n_r \times p \tag{3.21}$$

where s evaluates at each of the eigenvalues of $\tilde{A}$. This matrix has $n_x + n_r \times p$ rows that must all be independent to have full rank. To derive this requirement, simply use elementary row and column operations to transform $\left[sI - \tilde{A} | \tilde{B}\right]$ into the following:

$$\left[\begin{array}{cc} sI - \tilde{A} & \tilde{B} \end{array}\right] = \left[\begin{array}{cccccc} sI & -I & 0 & 0 & 0 & 0 \\ 0 & sI & -I & 0 & 0 & 0 \\ 0 & 0 & sI & -I & 0 & 0 \\ 0 & 0 & 0 & \frac{a(s)}{s^p} I & -C & D \\ 0 & 0 & 0 & 0 & sI - A & B \end{array}\right]$$

$$a(s) = s^p - \sum_{i=1}^{p} a_i s^{p-i} \tag{3.22}$$

Clearly, the first $n_r \times (p-1)$ rows are independent. From the last row, $[sI - A | B]$ must be full rank which says that the original system model must be controllable. Considering the last two rows, if $s = s_i$ such that $a(s_i) = 0$ (a zero of $a(s)$), then we must have

$$\operatorname{rank}\left[\left[\begin{array}{cc} -C_c & D_c \\ s_i I - A & B \end{array}\right]\right] = n_x + n_r \tag{3.23}$$

For this to occur, the multivariable zeros or transmission zeros of the original system must not equal any zeros of $a(s)$ and $n_u \geq n_r$. To summarize are the following:

1. The original system (A, B) must be controllable.
2. The number of controls must be greater than the number of signals to track, $n_u \geq n_r$.
3. The original system (A, B, C_c, D_c) must not have any transmission zeros common with the polynomial $a(s)$.

For control design, we can often relax the controllability requirement to that of stabilizability. For stabilizability, the original system (A, B) must be stabilizable, and conditions (2) and (3) above must also be satisfied.

Example 3.1 Constant Command Tracking Consider a constant command r. According to (3.4), this gives $\dot{r} = 0$ $(p = 1)$, with $a_1 = 0$. The command error is $e = y_c - r$. The servomechanism design model using (3.19) is given by

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \dot{x} \end{bmatrix}, \mu = \dot{u}$$
$$\tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} D_c \\ B \end{bmatrix} \tag{3.24}$$

Example 3.2 Sinusoidal Command Tracking Consider a sinusoidal command $r(t) = \sin(\omega t)$. This gives $\ddot{r} = -\omega^2 r$, $(p = 2)$, with $a_1 = 0$, $a_2 = -\omega_0^2$, (see Eq. (3.4)). The command error is $e = y_c - r$. The state space system using (3.19) is given by

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \xi \end{bmatrix}, \xi = \ddot{x} - \omega^2 x, \mu = \ddot{u} - \omega^2 u,$$
$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & C_c \\ 0 & 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ D_c \\ B \end{bmatrix} \tag{3.25}$$

Example 3.3 Constant Command Tracking in a Scalar System Knowledge from classical control tells us that a type 1 controller is needed to track a constant command. Using a scalar system, this example will build a state space model and illustrate how to design an integral control for tracking constant commands. Consider the following scalar system:

$$\dot{x} = -2x + u + w$$
$$y = x \tag{3.26}$$

where x is the state, u is the control, and w is a nonmeasurable constant disturbance. Hence, $A = [-2]$, $B = [1]$, $E = [1]$, $C = [1]$, and $D = [0]$. The goal is for the output y (same as the state x) to track a constant command r, with zero steady-state error. The constant command is modeled using (3.3) as

$$\dot{r} = 0, p = 1, a_1 = 0 \tag{3.27}$$

The robust servo design model is

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.28}$$

with

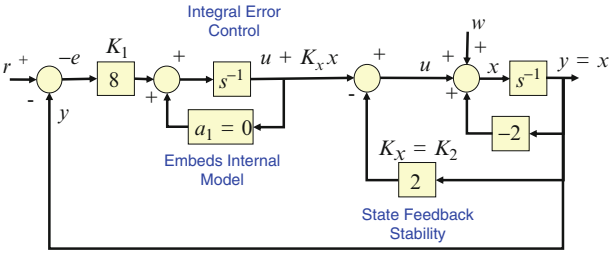

Fig. 3.1 Example 3.3 block diagram of the control and system dynamics

$$\tilde{A} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}; \tilde{B} = \begin{bmatrix} D \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.29}$$

in which we see that $(\tilde{A}, \tilde{B})$ form a controllable pair. The feedback control law is $\mu = -Kz$. It is desired that the closed-loop dynamics have a characteristic polynomial of $\phi_{\text{cl}}(s) = (s+2)^2 + 4 = s^2 + 4s + 8$ (pole placement problem). The feedback control is

$$\mu = -[K_1 \quad K_2] \begin{bmatrix} e \\ \dot{x} \end{bmatrix} \tag{3.30}$$

The closed-loop system is $\dot{z} = (A - BK)z$ with characteristic polynomial $\phi_{\text{cl}}(s) = \det[sI - \tilde{A} + \tilde{B}K]$. Substitute for $(\tilde{A}, \tilde{B})$ keeping the gains as parameters, expand the determinant, and equate to the desired closed-loop characteristic polynomial

$$\det[sI - \tilde{A} + \tilde{B}K] = s^2 + (2 + K_2)s + K_1 = s^2 + 4s + 8 \tag{3.31}$$

Equating coefficients of s yields two equations in the two unknown gains that can be solved for the gains $[K_1 \quad K_2] = [8 \quad 2]$. The control $u = \int \mu$ and is

$$\begin{aligned} u = -K \int z \, \mathrm{d}t &= -[8 \quad 2] \begin{bmatrix} \int e \, dt \\ x \end{bmatrix} \\ &= -8 \int e \, dt - 2x + \text{constant of integration} \end{aligned} \tag{3.32}$$

In the implementation, the constant of integration is ignored. Figure 3.1 illustrates the system, (controller, plant, and disturbance).

3.3 The Robust Servomechanism LQR

In Chap. 2, it was shown that the state feedback infinite-time linear quadratic regulator has excellent stability and robustness properties. In this section, this approach is applied to the servomechanism design model from the previous section

to form the robust servomechanism LQR (RSLQR). The RSLQR gain matrix K_c that is produced from the solution of the algebraic Riccati equation forms the state feedback control given as

$$\mu = -K_c z \tag{3.33}$$

which, when integrated p times, is implemented using integral control for command tracking and state feedback for stabilization.

The RSLQR design problem uses the servomechanism design model written as

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.34}$$

where z and μ are defined in (3.18) and (3.13), respectively. LQR control theory is applied to (3.34), using the performance index (PI),

$$J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) \mathrm{d}\tau \tag{3.35}$$

where $Q = Q^T \geq 0, R = R^T > 0$, $\left(\tilde{A}, \tilde{B}\right)$ is stabilizable, and $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ is detectable. For the infinite-time problem, the optimal steady-state control law for μ using state feedback is formed by solving the algebraic Riccati equation (ARE) using Q and R from (3.35), given as

$$P\tilde{A} + \tilde{A}^T P - P\tilde{B}R^{-1}\tilde{B}^T P + Q = 0 \tag{3.36}$$

The resulting steady-state $n_u \times (n_r + n_x)$-dimensional feedback controller gain matrix is

$$K_c = R^{-1}\tilde{B}^T P \tag{3.37}$$

with the state feedback control given as $\mu = -K_c z$. The gain matrix K_c is partitioned in the same manner as the vector z in (3.18), written as

$$K_c = \begin{bmatrix} K_p & K_{p-1} & \cdots & K_1 & K_x \end{bmatrix} \tag{3.38}$$

Substituting the definition of z into (3.33) yields

$$\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} = -\sum_{i=1}^{p} K_i \overset{(p-i)}{e} - K_x \left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right] \tag{3.39}$$

Integrating (3.39) p-times gives the control solution u for the original system model in (3.2) as

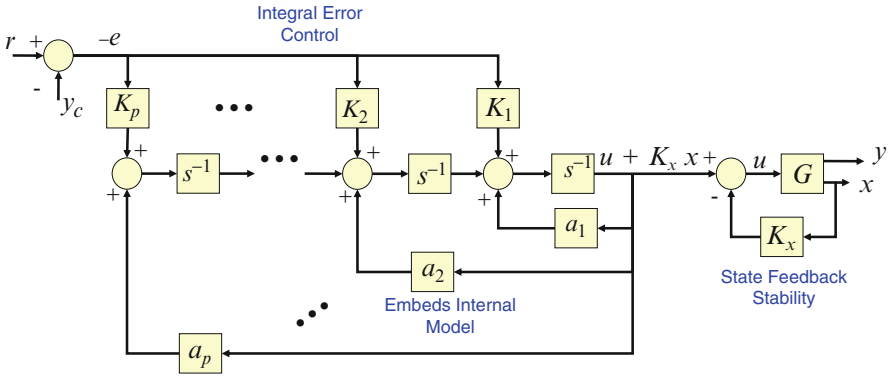

Fig. 3.2 Robust servomechanism block diagram

$$u = -K_x x + \sum_{i=1}^{p} s^{-i}\left(a_i\left(\overset{(p-i)}{u} + K_x \overset{(p-i)}{x}\right) - K_i \overset{(p-i)}{e}\right) \tag{3.40}$$

Figure 3.2 is a block diagram illustrating the system of (3.2) (represented as G) connected to the robust servomechanism state feedback control law.

The state feedback term $(-K_x x)$ enforces closed-loop stability of the plant. The p integrators and their gains provide integral error control, and the coefficients a_i embed the internal model of the signal being tracked. So, the closed-loop system is

$$\dot{z} = \left(\tilde{A} - \tilde{B}K_c\right)z + Fr \tag{3.41}$$

where $F = \left[-I_{n_u \times n_u} \quad 0_{n_u \times n_x}\right]^T$. The RSLQR closed-loop design using state feedback is guaranteed to be globally exponentially stable, and it will force the system-regulated output track the command signal $r(t)$, with zero steady-state error.

In Chap. 1, we introduced plant (1.35) and controller (1.36) state space models. These models were then coupled to form a closed-loop simulation model and loop gain frequency domain analysis models. We want to implement the RSLQR control from (3.40) using the controller given by

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{3.42}$$

The control in (3.40) is a state feedback control $(y = x)$. Substituting into (3.42), we have

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} x + B_{c2} r \\ u &= C_c x_c + D_{c1} x + D_{c2} r \end{aligned} \tag{3.43}$$

with

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \left[\begin{bmatrix} 0 & I_{n_r} & \cdots & 0 \\ \vdots & \ddots & \cdots & 0 \\ 0 & 0 & \cdots & I_{n_r} \\ a_p I_{n_r} - D_c K_p & \cdots & \cdots & a_1 I_{n_r} - D_c K_1 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_c - D_c K_x \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I_{n_r} \end{bmatrix} \right] \\ \begin{bmatrix} -K_p & \cdots & -K_2 & -K_1 \end{bmatrix} \quad [-K_x] \quad [0] \tag{3.44}$$

Example 3.4 The Robust Servo Controller for Example 3.3 In Example 3.3, the control u was given as

$$u = -8 \int e \, dt - 2x \tag{3.45}$$

The state space model for the controller using (3.44) is

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1] & [-1] \\ [-8] & [-2] & [0] \end{bmatrix} \tag{3.46}$$

In industrial applications, the commanded signal $r(t)$ is often assumed to be a constant. For example, in flight control, such a command could represent the stick force coming from a pilot or the guidance command coming from the outer-loop steering algorithms. Even though these command signals are not actually constant, designing and implementing a type 1 control system has proven very effective in most applications, and the RSLQR will provide zero steady-state error command tracking.

To achieve good transient response characteristics, tuning of the LQR PI matrices Q and R is required. Understanding how these matrices affect the control gains and how the control gains influence the closed-loop system response is key to achieving a good design.

It is important in the design of a realistic control system to be mindful of the "size" of the feedback gains in K_c. In aerospace applications, gains that are too large amplify sensor noise, drive the actuators with high rates, and cause issues and challenges with flexible body dynamics, called structural mode interaction. The feedback gains K_c depend upon the numerical values in Q and R. As $\|Q\|_2$ becomes large, the gains get large; as $\|R\|_2$ is made small, the gains get large; thus, $\|K_c\|_2 \sim \|Q\|_2 / \|R\|_2$.

3.3.1 Summary

Dynamics: $\dot{x} = Ax + Bu + Ew$

$y = Cx + Du$

Command model: $\overset{(p)}{r} = \sum_{i=1}^{p} a_i \overset{(p-i)}{r}$; Disturbance model: $\overset{(p)}{w} = \sum_{i=1}^{p} a_i \overset{(p-i)}{w}$

State model: $\xi = \overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x}$; Control model: $\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u}$

Augmented state vector: $z = \begin{bmatrix} e & \dot{e} & \cdots & \overset{(p-1)}{e} & \xi \end{bmatrix}$

Performance index: $J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) d\tau \qquad Q = Q^T \geq 0, R = R^T > 0$

Control design model: $\dot{z} = \tilde{A}z + \tilde{B}\mu$; $\left(\tilde{A}, \tilde{B}\right)$ controllable. $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ detectable.

$$\tilde{A} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & 0 & I & 0 \\ \alpha_p I & \alpha_{p-1} I & \cdots & \alpha_2 I & \alpha_1 I & C \\ 0 & \cdots & \cdots & \cdots & 0 & A \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ D \\ B \end{bmatrix}$$

ARE: $P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P = 0 \qquad \mu = -R^{-1}\tilde{B}^T P z = -K_c \mathrm{z}$

Control: $u = -K_x x + \sum_{i=1}^{p} s^{-i} \left(a_i \left(\overset{(p-i)}{u} + K_x \overset{(p-i)}{x} \right) - K_i \overset{(p-i)}{e} \right)$

Controller: $\dot{x}_c = A_c x_c + B_{c1} x + B_{c2} r$

$u = C_c x_c + D_{c1} x + D_{c2} r$

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & I_{n_r} & \cdots & 0 \\ \vdots & \ddots & \cdots & 0 \\ 0 & 0 & \cdots & I_{n_r} \\ a_p I_{n_r} - D_c K_p & \cdots & \cdots & a_1 I_{n_r} - D_c K_1 \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_c - D_c K_x \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I_{n_r} \end{bmatrix} \\ \begin{bmatrix} -K_p & \cdots & -K_2 & -K_1 \end{bmatrix} & [-K_x] & [0] \end{bmatrix}$$

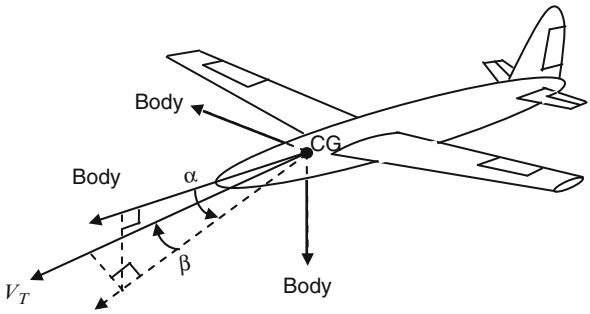

Fig. 3.3 Unmanned aircraft

The following example will illustrate how to choose parameters within Q and how to select a design that performs well, has a reasonable bandwidth, and does not result in high actuator rates. The processes for selecting the LQR penalty weights form LQR design charts that show important time domain and frequency domain metrics plotted versus loop gain crossover frequency. Viewing this information, while using the design charts, allows the control system engineer to select the desired bandwidth of the design and to perform the necessary trade studies required to meet the desired closed-loop system performance design goals. This process also prevents large feedback gains from being selected, which can introduce challenges later in the simulation and analysis of the control system.

Example 3.5 LQR Q-Matrix Parameter Selection Using Design Charts Consider the pitch-plane dynamics of an unmanned aircraft (Fig. 3.3), given as

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta + M_q q\end{aligned} \tag{3.47}$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant and will design an RSLQR controller with integral control. We will design a constant gain matrix K_c for a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration $A_z(\text{ft}/s^2)$ is given by

$$A_z = -V\dot{\gamma} = VZ_\alpha \alpha + VZ_\delta \delta \tag{3.48}$$

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate (3.48) to form the differential equation for $\dot{A}_z$ and then substitute for $\dot{\alpha}$ from (3.47). This produces

$$\begin{aligned}\dot{A}_z &= Z_\alpha A_z + VZ_\alpha q + VZ_\delta \dot{\delta}_e \\ \dot{q} &= \frac{M_\alpha}{VZ_\alpha}A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_e\end{aligned} \tag{3.49}$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_e = -2\zeta_a \omega_a \dot{\delta}_e + {\omega_a}^2(\delta_c - \delta_e) \tag{3.50}$$

Combining (3.49) and (3.50) forms the plant model written in matrix form as

$$\begin{bmatrix} \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha & VZ_\alpha & 0 & VZ_\delta \\ M_\alpha/VZ_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \delta_c \tag{3.51}$$

Since $r = \text{constant}$, $\dot{r} = 0$, and $p = 1$, then we need to add an integrator to form our type 1 controller. The state vector (Eq. 3.18) for the robust servomechanism design model is

$$z = \begin{bmatrix} e & \dot{x}^T \end{bmatrix}^T \tag{3.52}$$

with the design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ given as

$$\begin{bmatrix} \dot{e} \\ \ddot{A}_z \\ \ddot{q} \\ \dddot{\delta}_e \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha & \mathrm{VZ}_\alpha & 0 & \mathrm{VZ}_\delta \\ 0 & M_\alpha/\mathrm{VZ}_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \ddot{\delta}_e \\ \dddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \dot{\delta}_c \tag{3.53}$$

where $z = \begin{bmatrix} e & \dot{A}_z & \dot{q} & \ddot{\delta}_e & \dddot{\delta}_e \end{bmatrix}^T \in R^5$. At a flight condition of Mach 0.3, 5,000 ft altitude, and a trim angle-of-attack α of 5 degrees, the plant model data (stability and control derivatives) are

$$\begin{aligned} Z_\alpha &= -1.05273(1/\mathrm{s}) \\ Z_\delta &= -0.0343(1/\mathrm{s}) \\ M_\alpha &= -2.3294\left(1/\mathrm{s}^2\right) \\ M_q &= -1.03341\left(1/\mathrm{s}^2\right) \\ M_\delta &= -1.1684\left(1/\mathrm{s}^2\right) \\ V &= 329.127(\mathrm{ft/s}) \\ \omega_a &= 2\pi * 13.(\mathrm{rad/s}) \\ \zeta_a &= 0.6 \end{aligned} \tag{3.54}$$

Substituting the data into (3.53) yields

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -6672. & -98.02 \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6672. \end{bmatrix} \tag{3.55}$$

If we check the controllability of the pair $\left(\tilde{A}, \tilde{B}\right)$, we find the system to be controllable.

The objective in the design of the gain matrix is to track the acceleration command with zero error without using large gains. The design begins by equating $R = 1$ and selecting a Q matrix that penalizes the error state e in (3.53). Thus, the performance index in (3.35) is

$$J = \int_0^\infty \left(z^T Q z + \mu^2\right) \mathrm{d}\tau \tag{3.56}$$

We start by inserting the parameter q_{11} in the (1,1) element

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ & 0 & & 0 & \\ & & 0 & & \\ & 0 & & 0 & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{q} \\ A_z \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}, \tag{3.57}$$

and set the other matrix elements to zero. This will penalize the error in tracking the command. Substituting (3.57) into (3.56) gives the performance index as

$$J = \int_0^\infty \left(q_{11} e^2 + \mu^2\right) d\tau \tag{3.58}$$

If we check the observability of the pair $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$, we find the system to be observable through this choice of Q.

The LQR design charts are formed by sweeping q_{11} values from small to large, solving for the feedback gains for each value of q_{11}, and examining the closed-loop system properties. The computation steps are the following:

1. Set the value of q_{11} in Q from (3.57).
2. Solve the ARE in (3.36) for P.
3. Compute the feedback gain matrix K_c in (3.37).
4. Form the closed-loop system in (3.41).

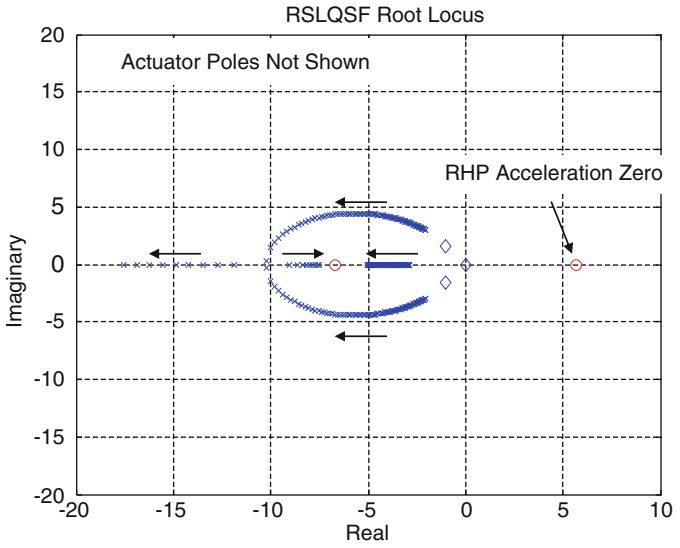

Fig. 3.4 RSLQR short-period dynamics root locus. Actuator poles at $-49.0 \pm 65.3j$ not shown

5. Simulate the closed-loop system to a step command and extract time domain performance metrics. These are rise time, settling time, percent command overshoot, percent command undershoot, max control, and control rate.
6. Evaluate the loop transfer function at the plant input and extract frequency domain metrics. These are loop gain crossover frequency, minimum singular values of $I + L$ and $I + L^{-1}$ (L at the plant input) versus frequency, and $\|S\|_\infty$ and $\|T\|_\infty$ for the commanded variable (S and T are formed using L at the plant output).
7. Loop back to step 1 and increase q_{11} until the numerical range is complete.

For this command tracking system, it is desired to track the acceleration command with zero error, and minimize the rise time and settling time, all in response to the command, without driving the control surface actuators with large gains. Large gains will cause large actuator deflections and rates, which are not desirable. This creates a trade study, in which the bandwidth must be limited in order not to exceed actuator limitations. Also, large gains amplify sensor noise, reduce stability margins, and make the system sensitive to unmodeled high-frequency dynamics (like flexible body modes).

For this flight condition, the range of the LQR penalty q_{11} is selected to be $q_{11} = \left[10^{-2}, \quad 10^{0.5}\right]$, using 100 design points. For a linear system, the response will depend upon the location of the closed-loop poles in the s-plane. Looping through the above calculations, the eigenvalues of the closed-loop system matrix $\left(\tilde{A} - \tilde{B}K_c\right)$ are plotted to form a root locus. The data are shown in Fig. 3.4.

Also plotted are the open-loop poles (diamonds), the commanded variable A_z/δ_e, and the system transfer function zeros, which include a nonminimum phase zero (right half plane (RHP) zero). The open-loop dynamics are stable at this flight condition, with the poles located in the left half plane (LHP). The two finite zeros of the acceleration transfer function are -6.73 and 5.69. As discussed in Chap. 2 on asymptotic properties of regulators and the root square locus, Fig. 3.4 shows the RHP zero at 5.69 mirrored into the LHP. Two of the closed-loop poles, one from the integrator and the other from the short period, are approaching this region on the negative real axis. The remaining short-period pole moves out to infinity along the negative real axis. The control actuator poles, not shown in the figure, move toward infinity along asymptotes at 45 degrees.

The time domain performance metrics of interest here are 63% rise time, 95% settling time, percent overshoot, percent undershoot (because the system is nonminimum phase), max actuator deflection, and max actuator rate in response to a constant step command. The frequency domain performance metrics are loop gain crossover frequency ω_c in Hz, the minimum of the minimum singular value of the return difference dynamics, denoted $\underline{\sigma}(I+L)$, and the minimum of the minimum singular value of the stability robustness matrix $I+L^{-1}$, denoted $\underline{\sigma}\left(I+L^{-1}\right)$. The metric $\underline{\sigma}(I+L) = 1/\|S\|_\infty$ and $\underline{\sigma}\left(I+L^{-1}\right) = 1/\|T\|_\infty$ (see Chap. 5, Sect. 5.2 for definitions). These metrics, plotted versus ω_c, are used to determine how the increasing bandwidth of the system affects the system characteristics, indicating a desired value for q_{11}.

As with most control system design procedures, there is not a single answer to determining a set of gains that are acceptable. It is for the designer to make a reasonable selection. Once a suitable design is chosen, the associated gain matrix K_c is then stored in a table to create a gain-scheduled control for real-time implementation. Figure 3.5 shows the rise time and settling time plotted against loop gain crossover frequency ω_c.

As ω_c increases, the system responds more quickly to the step command. As seen from the figure, there is a diminishing return in terms of speed of response as the bandwidth increases. This is also evident from the root locus in Fig. 3.4. As the dominant poles approach the zero locations at -6.73 and -5.69, the change in the pole location diminishes with the increasing gains. The poles headed toward infinity along the asymptotes continue to move, but their contribution to the response $\left(e^{\lambda t}\right)$ dies quickly as the eigenvalues get large and negative. This indicates that large gains are not needed to make the system respond quickly.

Figure 3.6 shows the percent overshoot, percent undershoot, max elevon (tail actuated control surface) deflection per g commanded, and max elevon rate per g commanded versus the loop gain crossover frequency ω_c.

At lower values of ω_c, the response slightly overshoots the command, causing an overshoot. Command overshoot in flight control systems needs to be minimized in order to maintain limits and placards on the aircraft. As the integrator gain increases (as q_{11} increases), above 2.1 Hz ω_c, the response has no command overshoot. This metric by itself indicates a desire for larger gains. The percent undershoot,

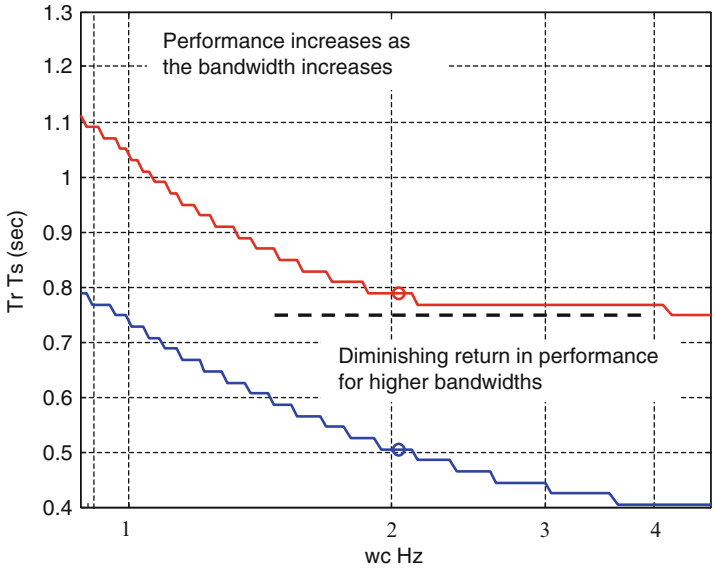

Fig. 3.5 Rise time (*blue*) and settling time (*red*) versus loop gain crossover frequency

characteristic of nonminimum phase responses, continues to increase with increasing ω_c. This response characteristic is undesirable and also needs to be minimized. Unfortunately, it increases with increasing ω_c. This metric indicates a desire for lower gains. Both the max deflection and max rate increase with increasing ω_c. It is critical in flight control systems not to have excessive deflections and rates in response to changes in the command. Electric actuators typically used in unmanned aircraft systems draw current proportional to the peak rate (at these normal operating conditions). High rates then cause significant power draw. Also, if the surface becomes rate saturated, this nonlinear effect can significantly degrade stability. As shown in the figure, the deflection and rate increase almost exponentially with increasing ω_c. These metrics also indicate a need for lower gains. As seen in this figure, some of the metrics tend toward increasing the gains, and some tend toward decreasing the gains.

Figure 3.7 shows two frequency response metrics: the minimum of the minimum singular value of the return difference dynamics $\underline{\sigma}(I+L)$ and the minimum of the minimum singular value of the stability robustness matrix $\underline{\sigma}\left(I+L^{-1}\right)$.

As is characteristic of LQR state feedback designs (discussed in Chap. 2), the $\underline{\sigma}(I+L)$ is equal to unity for all q_{11} design values. This metric is not particularly useful for developing state feedback designs but is critical when output feedback is used. The $\underline{\sigma}\left(I+L^{-1}\right)$, which is the inverse of the infinity norm of the complementary sensitivity function, is a measure of the damping in the dominant poles of the closed-loop system. We would like to maximize $\underline{\sigma}\left(I+L^{-1}\right)$. The figure shows that this metric tends to favor larger gains.

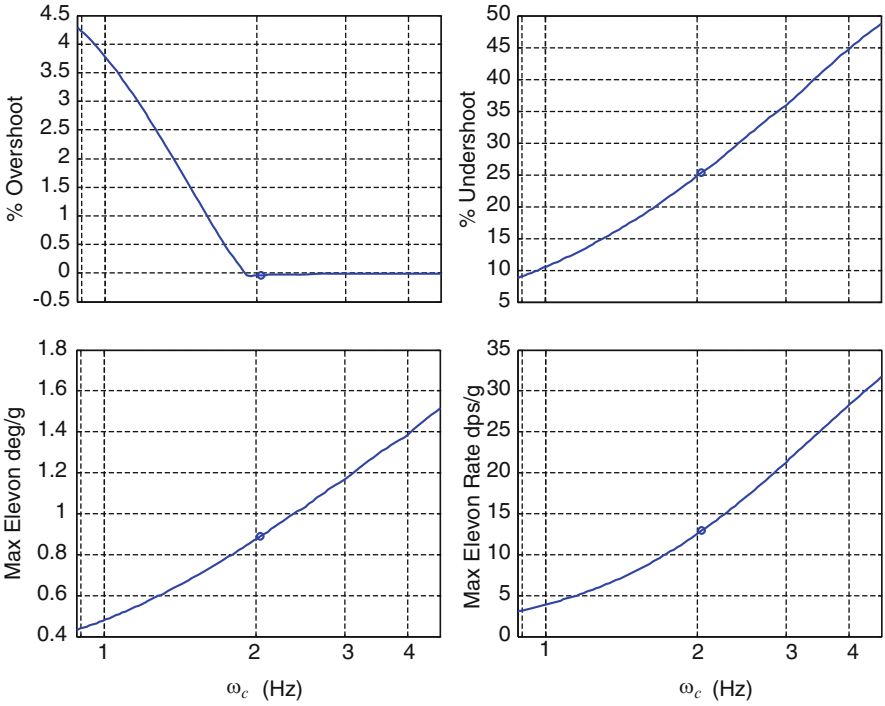

Fig. 3.6 Percent overshoot, undershoot, and max elevon deflection and rate versus loop gain crossover frequency ω_c

In balancing the positive and negative trends indicated by these metrics, a design condition $q_{11} = 0.2448$ was selected. This is the value of q_{11} where the percent overshoot first approaches zero. For this design condition, the states A_z, q, δ_e, and $\dot{\delta}_e$ are plotted versus time in Fig. 3.8.

Note that there is no overshoot to the unit command. For this approach flight condition, the response is quick, without the use of large gains.

The gain matrixK_c is

$$K_c = [0.4948\ 0.1790 - 14.0605\ 2.2089\ 0.0018] \tag{3.59}$$

The controller implementing this design is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{3.60}$$

where $y = \begin{bmatrix} A_z & q & \delta_e & \dot{\delta}_e \end{bmatrix}^T$, $r = A_{zc}$, and

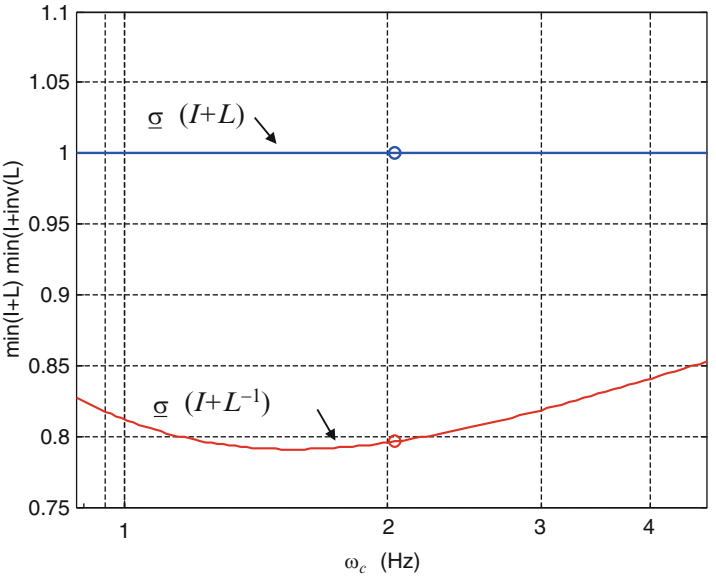

Fig. 3.7 Singular value frequency domain metrics versus loop gain crossover frequency

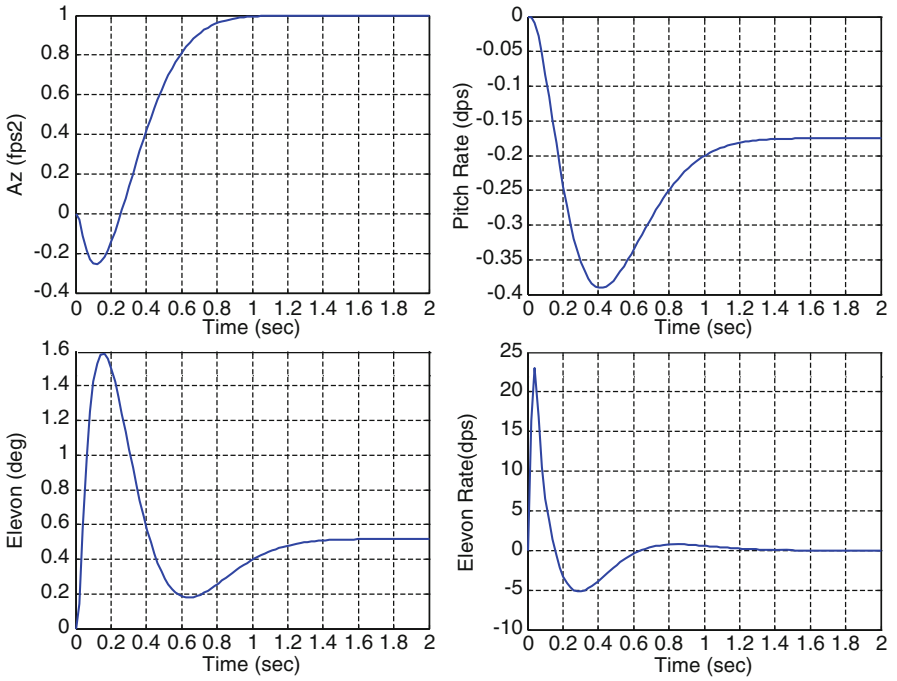

Fig. 3.8 States of the system responding to a unit acceleration step command

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1 \quad 0 \quad 0 \quad 0] & [-1] \\ [-0.4948] & [-0.179 \quad 14.0605 \quad -2.2089 \quad -0.0018] & [0] \end{bmatrix} \tag{3.61}$$

3.4 Conclusions

Depending upon the signal to be tracked, a certain number of integrators are needed to provide zero steady-state tracking error. In this chapter, we discussed how to formulate this problem within a state space framework and how to use optimal control to design the command tracking control system. In Chap. 2, we discussed the excellent frequency domain properties of LQR controllers. For our robust servomechanism controllers, we have these same excellent properties.

One of the key takeaways from the chapter should be the development of design charts for selecting numerical weights in optimal control problems. It is very easy to use too large of numerical weightings in the LQR performance index, and these large weights would lead to high gains. It is critical to be able to determine the bandwidth that is needed in the design to meet performance requirements and not to drive the control actuation system too hard.

3.5 Exercises

Exercise 3.1. A linearized suspended ball model is described by

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

(a) Use state feedback to stabilize the system producing closed-loop eigenvalues at $-1, -1/2$.
(b) The ball position x_1 can be measured using a photocell, but the velocity x_2 is more difficult to obtain. Suppose, therefore, that $y = x_1$. Design a full-order observer having poles at -4 and -5 and use the observer feedback to produce closed-loop eigenvalues at $-1/2, -1, -4, -5$.
(c) Repeat (b) using a first-order observer with pole at -6. Give a block diagram showing the controller as a single transfer function.
(d) Repeat this same design problem using the robust servo approach, obtaining integral control.

Exercise 3.2. Consider the design of a longitudinal (pitch-plane) autopilot. Using the robust servo formulation, design a pitch autopilot commanding a constant angle-of-attack α. Use the following dynamic model as the nominal plant model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{bmatrix} \delta_\mathrm{e}$$

and use data for $\frac{Z_\alpha}{V} = -1.21;\ \frac{Z_\delta}{V} = -0.1987; M_\alpha = 44.2506; (M_\delta = -97.2313)$.

(a) Design the autopilot to track a constant angle-of-attack command. Use the LQR approach outlined in Sect. 3.2.
(b) Design an autopilot to track a sinusoidal angle-of-attack command.

Exercise 3.3. Consider the longitudinal dynamics of a transport aircraft as given in Chap. 1, Exercise 1.2. Design a robust servo LQR control to track a constant speed command and a constant angle-of-attack command.

Exercise 3.4. Consider the lateral-directional dynamics of a transport aircraft as given in Chap. 1, Exercise 1.4. Design a robust servo LQR control to track a constant stability axis roll-rate p_s command (see Eq. (1.22)). Assume $\alpha_0 = 6\,\mathrm{deg}$.

References

1. Francis, B.A., Wonham, W.M.: The internal model principle of control theory. Automatica **12**(5), 457–465 (1976)
2. Davidson, E.J., Copeland, B.: Gain margin and time lag tolerance constraints applied to the stabilization problem and robust servomechanism problem. IEEE Trans. Autom. Control **30**(3), 229–239 (1985)