

Chapter 4

Engine Control by Robust State Feedback

Abstract This chapter reviews linear multivariable theory and introduces polytopic system descriptions of plant variability. The chapter also presents various methods for MIMO state-feedback synthesis, such as: LQR, \mathcal{H}_2 , \mathcal{H}_∞ and mixed-objective optimization with regional pole placement constraints. A simplified \mathcal{H}_∞ compensator synthesis method is presented for SISO systems. Matlab code and simulations using the CMAPSS nonlinear engine model are included.

The purpose of this chapter is to provide an overview of robust, multivariable techniques that are directly applicable to the GTE control problem. As pointed out in Chap. 3, fixed controllers cannot be expected to preserve engine limits or to operate satisfactorily across the whole flight envelope, less so if the engine health parameters are subject to changes and unmodeled dynamics exist. However, various approaches to gain scheduling are based on a set of fixed controllers. It is therefore essential to discuss the salient features of robust multivariable control and provide practical design guidelines. The chapter assumes familiarity with the state-space pole-placement concept, at a minimum.

Actual engines incorporate real-time sensing of fan and core speeds, which are the states of the dynamic model. This fact opens the doors to many techniques based on state measurement feedback, such as the linear quadratic regulator (LQR) and sliding mode control. Although most state feedback techniques admit observer-based extensions, the performance and robustness properties attained with measurement feedback are partially lost when a state estimator is introduced. A classical example of this effect is given by the performance and robustness losses associated with observer-based implementations of linear quadratic regulation. Loop transfer recovery techniques [29] aim to restore the lost performance and robustness properties.

The chapter begins by introducing essential concepts such as singular values and signal and system norms and their computation using Matlab. Robustness and performance are then addressed using classical LQR theory. A characterization of the uncertain design plant, along with a corresponding formulation of the control objectives is then developed using the tools of \mathcal{H}_2 and \mathcal{H}_∞ robust state feedback synthesis. A classical LQR controller and robust \mathcal{H} -norm based controllers are

designed for the 40k-class engine, using W_F , VSV, and VBV as actuators. The designs are then simulated in CMAPSS-40k.

4.1 Overview of Multivariable Systems Theory

Given a multi-input, multi-output (MIMO) system described by a set of state-space matrices (A, B, C, D) , the single-input, single-output (SISO) transfer function $G_{ij}(s)$ from the j -th input to the i -th output is found using the formula:

$$G_{ij}(s) = C_i(sI - A)^{-1}B_j + D_{ij}, \quad (4.1)$$

where C_i is the i -th row of C , B_j is the j -th column of B , and D_{ij} is the (i, j) entry of matrix D . The arrangement of the G_{ij} in a matrix is the *transfer matrix* $G(s)$. The poles of the individual entries G_{ij} are a subset of the eigenvalues of A , where pole-zero cancelations may have occurred in individual entries. The *transmission zeroes* or *multivariable zeroes* of $G(s)$ have a special definition, and they generally differ from the zeroes of the individual $G_{ij}(s)$. The zeroes of a transfer matrix are given by the set of values of s for which $G(s)$ loses rank. Mathematically, $s = z$ is a zero of $G(s)$ with multiplicity k if there exist k linearly independent vectors v so that $G(z)v = 0$. In Matlab, the transmission zeroes are readily found with the `tzzero` command.

In this book, we limit ourselves to *rational* transfer matrices; that is, matrices whose entries are ratios of polynomials. A rational transfer matrix is *proper* if its individual entries are so, that is, if they do not have more zeroes than poles. A set of state-space matrices (A, B, C, D) always results in a rational, proper transfer matrix.

4.1.1 Example

Take the 40k engine model matrices of CMAPSS-40k at Ground Idle listed in Appendix C. Define the control input vector as $u = [\Delta W_F \ \Delta VSV]^T$ and choose two outputs as $y = [\Delta EPR \ \Delta T_{48}]^T$. The following Matlab code is used to find the transmission zeroes:

```
%Assumes A,B,C,D are in the workspace
sysSS=ss(A,B,C,D); %Create system
sysTM=zpk(sysSS) %Find and display transfer matrix
tzzero(sysTM) %Display transmission zeroes
```

The reader can verify that the transfer matrix returned in `sysTM` is:

$$G(s) = \frac{\begin{bmatrix} 289.0525(s + 3.925)(s + 1.918) & 0.1332(s - 95.1)(s + 3.471) \\ -10.9483(s + 1.274)(s + 4.64) & 0.1837(s^2 + 0.7075s + 4.929) \end{bmatrix}}{(s + 3.992)(s + 2.439)}.$$

The transmission zeroes are $\{2.9121, -0.5659, -3.9919, -2.4389\}$, while the zeroes of the individual entries are $\{-3.925, -1.918, 95.1, -3.471, -1.274, -4.64, -0.3538 \pm 2.1918i\}$, a different set. A right-half plane transmission zero is found at 2.9121. As in the SISO case, such nonminimum phase zeroes introduce fundamental limitations to attainable performance and may lead to design difficulties.

To illustrate the definition of transmission zero, evaluate the transfer matrix at one of the zeroes, for instance at $z = 2.9121$. The reader can verify that $G(z)$ is very close to being singular. Numerical precision prevents the exact verification of the rank loss property. In contrast, if the zeroes of the individual transfer function entries are used for z , the rank of $G(z)$ is always 2.

4.1.2 Singular Values

Singular values are fundamental indicators of the effects of input directionality on system outputs. In one of their interpretations, they extend the notion of frequency response to multi-input systems. Recalling basic linear algebra, a vector v of constant unit length and variable direction will, in general, give rise to vectors w of different lengths upon the linear operation $w = Tv$, where T is a constant matrix. The lengths obtained for w as v is varied are bounded above and below by two quantities known as the *maximum and minimum singular values* of T :

$$\bar{\sigma}(T) = \max_{\|v\|=1} \|Tv\|, \quad (4.2)$$

$$\underline{\sigma}(T) = \min_{\|v\|=1} \|Tv\|. \quad (4.3)$$

Every m -by- n matrix T with complex entries can be decomposed as $T = U\Sigma V^*$, where U and V are unitary matrices and Σ is a diagonal matrix of the form

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},$$

where

$$\Sigma_1 = \begin{bmatrix} \bar{\sigma} & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \underline{\sigma} \end{bmatrix}.$$

The diagonal entries of Σ_1 , $\bar{\sigma} = \sigma_1 > \sigma_2 > \dots > \sigma_p = \underline{\sigma}$ are the singular values, with $p = \min(m, n)$. The notation V^* is used for the complex conjugate transpose of V , and a unitary matrix is such that $V^*V = VV^* = I$. In Matlab, the singular value decomposition is found with the command `[U, S, V] = svd(T)`.

4.1.3 The Infinity Norm of a Linear System

In control systems, notions of “size” are customarily defined for transfer matrices, leading to bounds on the magnitude of the output as the input is varied under constant norm. One of such definitions is given by the *infinity norm* $\|G\|_\infty$, as follows: Suppose a p -by- m transfer matrix G is excited with an input vector $u(t)$ with sinusoidal components $u_j(t)$, $j = 1, 2, \dots, m$, all sharing the same frequency w but with possibly different amplitudes and phase shifts: $u_j(t) = U_j \sin(\omega t + \phi_j)$. Supposing output components reach a steady-state of sinusoidal oscillation of the form $y_i(t) = Y_i \sin(\omega t + \theta_i)$, $i = 1, 2, \dots, p$, an amplification ratio can be defined as $\|Y\|/\|U\|$, where $U = [U_1 \ U_2 \ \dots \ U_m]$ and $Y = [Y_1 \ Y_2 \ \dots \ Y_p]$. The infinity norm of G , denoted $\|G\|_\infty$, can be interpreted as the least upper bound (supremum) of the amplification ratio as frequency, input component amplitudes and phases are varied. That is,

$$\|G\|_\infty = \sup_{\phi_j, U_j, w} \frac{\|Y\|}{\|U\|}.$$

The frequency-domain definition of $\|G\|_\infty$ allows its calculation through the maximum singular value:

$$\|G\|_\infty = \sup_w \bar{\sigma}\{G(jw)\}. \quad (4.4)$$

For practical purposes, w is swept in a range, calculating the maximum and minimum singular values of $G(jw)$ at each point. The overall maximum of $\bar{\sigma}(G(jw))$ over all frequencies is the value of $\|G\|_\infty$. In Matlab (Robust Control Toolbox), an efficient numerical routine is implemented in the `hinfnorm` command. Alternatively, a routine using `svd` and `freqresp` can be easily programmed. As an example, the following sequence of Matlab commands plots the maximum and minimum singular values of the transfer function of Example 4.1.1 (assuming system matrices are available in the workspace):

```
>> sys=pck(A,B,C,D);
>> w=logspace(-1,4,500); %create a vector of 500 logarithmically
spaced frequencies between 10^(-1) and 10^4
>> Gf=frsp(sys,w); %calculate frequency response
>> [u,s,v]=svsd(Gf); %calculate SV decomposition at each
frequency
>> vplot('liv,lm',s); %plot singular values. The norm can be
visually extracted from the plot
>> hinfnorm(sys) %accurate calculation via bisection search
```

The reader can verify that $\|G\|_\infty = 289$ for the transfer matrix of Example 4.1.1. Needless to say, this number is affected by the choice of units of measurement for the inputs and outputs of the system. The infinity norm may also be calculated directly with `norm(sysTM,inf)`.

4.1.4 The 2-Norm of a Linear System

The infinity norm is a measure of the *peak* amplification produced by the linear system as input frequency and direction are varied. The 2-norm measures system amplification in terms of *root mean square averages* rather than peaks:

$$\|G\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} \{G^*(j\omega)G(j\omega)\} d\omega}. \quad (4.5)$$

The definition shows that a finite 2-norm can only be obtained if $G(\infty) = 0$, that is, if the transfer matrix is strictly proper. For rational, strictly proper transfer matrices with state-space realization $(A, B, C, 0)$, where A is stable, the 2-norm calculation reduces to

$$\|G\|_2^2 = \text{trace}(B^*QB) = \text{trace}(CPC^*), \quad (4.6)$$

where P and Q are the *observability and controllability Gramians*, obtained from the following Lyapunov equations:

$$\begin{aligned} AP + PA^* + BB^* &= 0, \\ A^*Q + QA + C^*C &= 0. \end{aligned}$$

The Matlab command `h2norm` incorporates these calculations. The transfer matrix of Example 4.1.1 does not have a finite 2-norm, since it is not strictly proper. The reader may verify this fact by inspection, or numerically, using `norm(system, 2)`.

4.2 Robust State Feedback Synthesis

The GTE control problem, although very challenging in many respects, has a simplifying feature: *the states of the linearized plant are measurable in real-time*. That is, fan and core speeds may be used directly in the calculation of feedback laws. This permits the application of several state-based techniques for control without the need for state estimation. In this section, we are interested in assessing the stability of a feedback arrangement of plant and controller, not only when plant parameters are known, but when they are subject to bounded, uncertain variations. Figure 4.1 shows the basic state-feedback arrangement considered in this section. The linearized engine plant is described in state-space form as

$$\dot{x} = Ax + Bu + \Gamma w, \quad (4.7)$$

$$y = Cx + Du + \Lambda w + n, \quad (4.8)$$

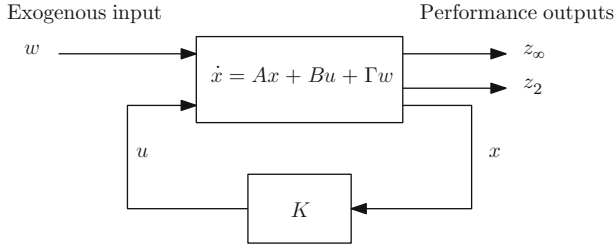


Fig. 4.1 State feedback configuration

where $x = [\Delta N_f \quad \Delta N_c]^T$ is the state vector, $u = [\Delta W_F \quad \Delta VSV \quad \Delta VB V]^T$ is the vector of control inputs and w is a vector that can be used to include the effect of disturbances. In linearized models obtained with CMAPSS, w is used to represent a vector of *health parameter inputs*, which can be regarded as disturbances. As described in Sect. 2.1.3, w can be used to simulate the effects of aging and deterioration of engine components. Output vector y may be defined to reflect a combination of sensed variables, outputs that need to be regulated and quantities that need to be monitored or maintained between certain limits. A sensor noise vector n may also be considered in certain control problems. A *steady operating point* is a set of states \bar{x} such that $A\bar{x} + B\bar{u} + \Gamma\bar{w} = 0$ for some fixed inputs \bar{u} and \bar{w} . That is, given a constant health parameter vector and a constant vector of actuator inputs, the engine reaches a steady operating condition with corresponding constant values of fan and core speeds.

4.2.1 Polytopic Description of System Uncertainty

In addition to exogenous inputs, system (4.7) is subject to variations in all its describing matrices. In the GTE, these changes arise mainly from two sources:

- **Intrinsic nonlinearity:** the dynamic relationship between control inputs and states is nonlinear, even when operating the engine at fixed environmental conditions (altitude, aircraft speed, ambient temperature). Therefore, system matrices will vary with steady-state linearization point.
- **Parametric variations:** given the same set of steady states and inputs, different system matrices will be obtained through linearization if environmental conditions are changed.

It is important to note that engine health status could have been modeled as parameters of functions f_1 and f_2 in (2.1) and (2.2). Changes in these parameters would be reflected in changes to the linearized system matrices. Rather, the designers of CMAPSS have chosen to capture faults and deterioration effects by means of exogenous inputs collected in vector w .

4.2.1.1 Scheduled vs. Robust Control

The variations in A, B, C, D, Γ and Λ are *deterministic* by nature. That is, they can be traced to the steady operating point in a repeatable way. In a scheduled control approach, this knowledge about the variation of system matrices with operating point is used to construct a controller whose gains are tailored to operating region. In contrast, when system matrix variations are not excessively wide, the designer may choose to ignore this knowledge and regard the variations as uncertain, but contained within certain bounds. This approach, termed “robust feedback synthesis” derives fixed controllers whose sensitivity to plant parameter changes is minimized. The complexity associated with gain-scheduled control implementations is thus avoided. As the next sections will demonstrate, robust stability with fixed linear compensators is feasible in GTE control systems. Consistent transient response qualities across the flight envelope are not obtainable with fixed compensators, however. For this reason, gain-scheduled control is used extensively in GTE control systems. Gain scheduling is covered in Chap. 5.

In this section, we focus on the robust feedback stabilization problem, that is, we attempt to find a fixed gain K capable of maintaining closed-loop stability regardless of variations in system matrices. An appropriate description of the uncertain state-space plant is given by the following *polytopic system*:

$$\dot{x} = A(\alpha)x + B(\alpha)u + \Gamma(\alpha)w, \quad (4.9)$$

where $A(\alpha)$, $B(\alpha)$ and $\Gamma(\alpha)$ are restricted to vary in a *polytope* of matrices, defined as a convex combination of k vertices A_i and B_i , $i = 1, 2, \dots, k$:

$$A(\alpha) = \sum_{i=1}^k \alpha_i A_i,$$

$$B(\alpha) = \sum_{i=1}^k \alpha_i B_i,$$

$$\Gamma(\alpha) = \sum_{i=1}^k \alpha_i \Gamma_i,$$

$$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k],$$

$$\sum_{i=1}^k \alpha_i = 1.$$

Thus, each instance of A , B and Γ can be understood as made up of “percentage” contributions of the vertices. These contributions are reflected in weight vector

α , whose entries are termed *polytopic coordinates*. Note that the influence of exogenous inputs through Γ is not relevant to stability analysis. Similar definitions apply to output matrices $C(\alpha)$, $D(\alpha)$, and $\Lambda(\alpha)$.

To generate a polytopic description of a GTE model, the designer must choose a number of representative steady operating points and perform system linearization. Simulation packages such as the CMAPSS family include a pre-defined set of such *flight conditions*, which provide good coverage of the entire flight envelope.

4.2.2 Nominal and Robust Stability

Recall that a rational transfer matrix $G(s)$ is *stable* if its entries are transfer functions whose poles have negative real parts. In a state-space realization of $G(s)$ given by the quadruple (A, B, C, D) , stability is equivalent to the condition that all the eigenvalues of A have negative real parts. If no pole-zero cancellations occur, the eigenvalues of A match the set of poles of the entries of G . When the feedback control law $u = -Kx$ is applied to system (4.7), the resulting closed-loop system has the form

$$\dot{x} = (A - BK)x + \Gamma w \quad (4.10)$$

Thus, *nominal stability* is ensured if K can be chosen so that all eigenvalues of $A - BK$ have negative real parts. The pair (A, B) is called *stabilizable* when a stabilizing gain K exists. Several tests for nominal stabilizability are possible. A simple test for stabilizability is accomplished by evaluating the *modal controllability* of (A, B) as follows: the pair is first transformed to its controllability form (\bar{A}, \bar{B}) :

$$\bar{A} = \left[\begin{array}{c|c} A_{nc} & 0 \\ \hline A_{21} & A_c \end{array} \right], \quad \bar{B} = \left[\begin{array}{c} 0 \\ B_c \end{array} \right].$$

This transformation arises from the change of state coordinates $z = Tx$, where $T = T^T$ is an orthogonal transformation matrix. This change, referred to as a *similarity transformation*, results in system matrices $\bar{A} = TAT^T$ and $\bar{B} = TB$, which share the same stabilizability properties as the original pair (A, B) . Matrix A_{nc} describes the dynamics of the *uncontrollable subspace*, a set of state variables that are unaffected by the control input. The pair (A, B) is stabilizable if and only if A_{nc} is stable. The similarity transformation T required to put the system in controllability form can always be calculated. In Matlab, T , \bar{A} and \bar{B} are found directly using the `ctrbf` command.

Note that stabilizable systems need not be controllable, but controllability implies stabilizability. Controllable systems possess an empty uncontrollable subspace, and A_{nc} does not exist. When a system is controllable, a stabilizing K may be found by pole-placement methods or more sophisticated robust synthesis methods, considered later in this chapter. When the system is only stabilizable, however,

a number of poles equal to the dimension of the controllable subspace may be arbitrarily placed. The pole-placement problem is solved in z -coordinates using (A_c, B_c) , resulting in a gain K_z yielding desired pole locations for $A_c - B_c K_z$. The feedback gain to be used in the original system is then calculated as $K = [* | K_z]T$, where $*$ denotes arbitrary values having no effect on the achieved pole locations. These values do have an impact on the magnitude of the control effort, however. For this reason, their choice is no longer immaterial in optimal feedback synthesis approaches such as LQR or \mathcal{H}_∞ control.

4.2.3 Quadratic Stability of Polytopic Systems

Consider a polytopic system of the form

$$\dot{x} = A(\alpha)x \quad (4.11)$$

with vertices $A_i, i = 1, 2, \dots, k$ and $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]$. The system is *quadratically stable* if all trajectories $x(t)$ starting from an arbitrary initial condition $x(0)$ converge to the origin as $t \rightarrow \infty$ and, in addition, there exists a quadratic function $V(x) = x^T P x$, which decreases when evaluated at $x(t)$. Quadratic stability is equivalent to the existence of a k -by- k symmetric matrix P satisfying the following set of *Lyapunov inequalities* [31]:

$$\begin{aligned} A_i^T P + P A_i &< 0, \quad i \in \{1, 2, \dots, k\} \\ P &> 0. \end{aligned} \quad (4.12)$$

The inequality signs correspond to *linear matrix inequalities*, or LMI. The notation $X > Y$ used in conjunction with matrices X and Y means that $X - Y$ is a *positive-definite* matrix. A positive-definite matrix P defines a quadratic function $f(x) = x^T P x$ such that $f(x) > 0$ for all vectors $x \neq 0$. One test for positive-definiteness is given by the requirement that all eigenvalues of P be positive. Similarly, a negative-definite matrix defines a quadratic function that does not take positive values, and negative-definiteness is equivalent to the requirement that all eigenvalues of P be negative. For more details, the reader is referred to [32].

The quadratic stability of polytopic system (4.11) is readily evaluated in Matlab. Assuming that the vertices are available in the workspace as variables A_1, A_2, \dots, A_k , the following sequence of commands attempts to find P :

```
s1=ltisys(A_1)
s2=ltisys(A_2)
....
sk=ltisys(A_k)
polysys=psys([s1 s2 ... sk])
[tmin,P]=quadstab(polysys)
```

Quadratic stability can be conservative, as it is based on a *single* quadratic Lyapunov function defined by P . It is also possible to seek a piecewise-quadratic Lyapunov function to reduce conservativeness. The interested reader is referred, for instance, to Johansson [33] for more information.

We now focus on the polytopic description of (4.9). We wish to determine whether a state feedback gain K exists that results in quadratic stability of the closed-loop system resulting from applying the control $u = -Kx$. This is the *quadratic stabilizability* problem, which has also been formulated in terms of LMIs [31]. Quadratic stabilizability is equivalent to the existence of a symmetric, positive-definite matrix Q and a matrix Y such that

$$QA_i^T + A_iQ - B_iY - Y^T B_i^T < 0, \quad i \in \{1, 2, \dots, k\}. \quad (4.13)$$

The search for Q and Y given vertices A_i and B_i is formulated as an *LMI feasibility problem*. Feasibility can be evaluated in Matlab using the `fesasp` command. Given feasible matrices Q and Y , a stabilizing feedback gain can be computed as $K = YQ^{-1}$. We defer details regarding the computation of feedback gains to Sects. 4.4 and below, where synthesis methods are considered that include robust stability and performance requirements.

4.3 Performance Measures

Consider again the feedback arrangement of Fig. 4.1. Performance outputs z_2 and z_∞ are chosen by designers to measure the effects of disturbances (vector w) on variables of interest. Denoting the transfer matrix from w to z_2 as $G_{w,z_2}(s)$, a measure of closed-loop system performance is given by the norm of $G_{w,z_2}(s)$: small values indicate that the system state and the computed control input are not very sensitive to disturbances. Minimizing the 2-norm of $G_{w,z_2}(s)$ corresponds to an optimal *linear quadratic control* problem to be discussed in the next section. Denoting the transfer matrix from w to z_∞ as $G_{w,z_\infty}(s)$, another measure of closed-loop system performance is given by the norm of $G_{w,z_\infty}(s)$. It should be noted that w can be extended to contain *reference commands* in addition to disturbances. Thus, if the control objective includes the requirement that the output y track a reference command $r(t)$, vector w can be defined as $w = [d \ r]^T$, and good tracking accuracy will be obtained by minimizing the norm of $G_{w,z}(s)$, where z is defined as the tracking error: $z = r - y$. Minimizing the infinity norm of G_{w,z_∞} corresponds to an \mathcal{H}_∞ *gain synthesis problem*, to be discussed in Sect. 4.6. The designer's role is to define z in a manner that best reflects the performance being sought.

4.4 LQR State Feedback Synthesis

Consider first the nominal system of (4.7), where A and B are fixed and no disturbance input is considered. Define a performance output as $z = Cx + Du$ such that $D^T D$ is invertible. The *linear quadratic regulator problem*, or LQR, is to find the control $u(t)$ that minimizes a performance measure given by

$$J = \int_0^{\infty} z(t)^T z(t) dt. \quad (4.14)$$

Note that the value of J depends on the initial conditions $x(0)$. Under some conditions [28–30] which include stabilizability of (A, B) , the solution takes the form of a state-feedback law $u(t) = -Kx(t)$ with a uniquely defined value of K as follows:

$$(D^T D)^{-1}(B^T P + D^T C) = K, \quad (4.15)$$

$$A^T P + PA - (PB + C^T D)(D^T D)^{-1}(B^T P + D^T C) + C^T C = 0. \quad (4.16)$$

The optimal solution for K is the same regardless of $x(0)$. Equation (4.16) is known as the *algebraic Riccati equation*, and it yields a unique, positive-definite solution for P under the assumption that (A, B) is stabilizable. In Matlab, the Riccati equation can be solved with the `are` command. Alternatively, the `lqr` command computes P and the optimal gain K . Calling $Q = C^T C$, $R = D^T D$ and $N = C^T D$, we see that $z^T z = x^T Q x + u^T R u + 2x^T N u$. Given an initial condition $x(0)$, a unique state trajectory $x(t)$ is obtained under the control $u = -Kx$. The term $x^T Q x$ penalizes excessive deviations the components of $x(t)$ from zero, positive or negative. Since the term is under an integral sign and the integral takes only positive values, trajectories which converge to zero slowly will be penalized more than those rapidly approaching zero. Similar penalties are applied to the control vector u , but using a different weighting matrix. The designer can use Q and R to manage the tradeoff between a fast response without excessive peaking and the magnitude of the control effort required to produce it. The cross-term $2x^T N u$ is not easily linked to response performance features and may be safely omitted in most practical situations.

4.4.1 LQR with Regional Eigenvalue Constraints

A commonly-used tuning approach is to use diagonal Q and R matrices, individually penalizing state and control components by adjusting the diagonal entries of these matrices. This method is not always satisfactory, and many trial-and-error simulations could be needed. The designer may instead require that all closed-loop

eigenvalues have real parts less than a specified negative number γ . The time constants of the closed-loop eigenvalues can thus be made as small as desired, accelerating the speed of response. To do this, simply use $A + \gamma I$ in place of A when calculating the gain from (4.15) and (4.16) [34]. This method, however, does not restrict the imaginary part of the closed-loop poles, potentially leading to unacceptably low damping ratios. To address this limitation, a circular region with center at $-\gamma + 0i$ and radius ρ may be specified. The solution is found by using $\frac{1}{\rho}(A + \gamma I, B)$ as state-space description of a *discrete-time* LQR problem. For more details, the reader is referred to [34].

4.4.2 The Cheap LQR Problem and Performance Limits

Maintaining a low level of control effort is not crucial in certain problems. This may occur, for instance, when the scaling of matrix B is such that small values of u contribute significant changes to \dot{x} , and the allowable values of u – defined by actuator limits – are large. The performance measure used in conjunction with the cheap LQR problem is given by

$$J = \int_0^{\infty} x^T Q x + \rho u^T R u dt, \quad (4.17)$$

where $\rho \rightarrow 0$. Recalling that $Q = C^T C$, the achievable performance (i.e., the minimum attainable value of J) is influenced by the zero locations of the transfer matrix defined as $H(s) = C(sI - A)^{-1} B$. As shown in [35, 36], J will approach zero as $\rho \rightarrow 0$ if and only if $H(s)$ is minimum-phase. In this case, the minimum-phase zeroes are canceled by closed-loop poles in the limit, as $\rho \rightarrow 0$. When $H(s)$ contains non-minimum-phase zeroes, J has a nonzero minimum value and the closed-loop poles approach their mirror images relative to the imaginary axis. Any excess poles approach infinity along asymptotes that remain in the left-half of the complex plane.

The designer must therefore exercise care in choosing C to ensure that $H(s)$ is minimum-phase and performance improves uniformly as $\rho \rightarrow 0$.

4.4.3 LQR Robustness Properties

The solution for the optimal feedback gain K enjoys strong stability robustness properties. Specifically, an increasing gain margin of infinity, a decreasing gain margin of 50% and a phase margin of $\pm 60^\circ$ are guaranteed for each control channel in conjunction with *any* choice of Q and R . That is, if we consider the SISO transfer functions from input components u_i to performance output components z_i , an unlimited gain increment or a 50% gain reduction will be tolerated without compromise to closed-loop stability. Similarly, a phase shift can be introduced (due

to unmodeled dynamics or signal delays) in a range of $\pm 60^\circ$ without destabilizing the loop.

In the GTE control problem, such robustness properties are beneficial if actuator dynamics are to be ignored. Many actuators can be modeled as a first-order transfer function, introducing phase lags of up to 90° . Thus, if the closed-loop bandwidth is designed at appropriately small values, the phase lag due to actuator dynamics can be kept larger than -60° and stability can be maintained. Similarly, the nominal gain of the actuator transfer function can be reflected in plant matrix B . Gain reductions of up to 50% due to uncertainties or changes to actuator dynamics will be tolerated.

4.4.4 Polytopic Systems

The previous sections assume that (A, B) is a fixed, known plant description. To design LQR controllers for the GTE under this assumption, a single representative steady operating point would have to be chosen, and a single gain K would be calculated for use in the entire operating envelope. If this were to be done, robust stability of the resulting polytopic system with vertices $(A_i - B_i K)$ could be readily evaluated using conditions (4.12). Even if stability is verified, performance could be unsatisfactory. A more reasonable approach is to use information about all vertices to seek a control law, which minimizes an integral quadratic cost of the form of (4.14). The optimal solution is given by a gain-scheduled state-feedback control $u = -K_i x$, where K_i are derived from a set of Riccati equations.

4.5 \mathcal{H}_2 State Feedback Synthesis

Consider the nominal system of (4.7) and suppose only performance output z_2 is considered in Fig. 4.1, defined as $z_2 = Cx + D_{zu}u + D_{zw}w$. The closed-loop system arising from the application of state-feedback control law $u = -Kx$ is given by

$$\begin{aligned}\dot{x} &= (A - BK)x + \Gamma w, \\ z_2 &= (C - D_{zu}K)x + D_{zw}w.\end{aligned}$$

The transfer matrix from the exogenous input vector w to the performance output z_2 is then

$$G_{w,z_2}(s) = (C - D_{zu}K)(sI - (A - BK))^{-1}\Gamma + D_{zw}.$$

The \mathcal{H}_2 feedback gain synthesis problem is to be formulated as

Find K so that $(A - BK)$ is stable and $\|G_{w,z_2}\|_2 < \nu$ for some $\nu > 0$.

That is, we seek a stabilizing feedback gain that maintains the influence of the exogenous inputs on the performance outputs below a prescribed level ν . For example, choosing z_2 to be the incremental HPC stall margin ΔSmHPC would result in a closed-loop controller, which minimizes the sensitivity of stall margin to health parameter changes, in addition to rendering the closed-loop system stable.

4.5.1 Optimal \mathcal{H}_2 Synthesis

In an \mathcal{H}_2 synthesis problem, a prescribed norm value ν is targeted by solving the following LMI feasibility problem in variables X , Q , and K :

$$\begin{bmatrix} (A - BK)X + X(A - BK)^T & \Gamma \\ & \Gamma^T - I \end{bmatrix} < 0, \quad (4.18)$$

$$\begin{bmatrix} Q & (C - D_{zu}K)X \\ X^T(C - D_{zu}K)^T & X \end{bmatrix} > 0, \quad (4.19)$$

$$\text{trace}(Q) < \nu^2. \quad (4.20)$$

A numerical solution to the above LMI problem is incorporated in Matlab's `msfsyn` command to be discussed in the example at the end of this chapter. An *optimal* \mathcal{H}_2 state feedback synthesis procedure is to attempt the LMI feasibility problem repeatedly, with decreasing values of ν , until the problem is no longer feasible.

4.5.1.1 LQR and \mathcal{H}_2 Equivalence

When all states are available for measurement, the \mathcal{H}_2 state feedback synthesis problem is equivalent to an LQR problem. Further, when only a vector of noisy measurements is available, the \mathcal{H}_2 problem is equivalent to a *linear quadratic Gaussian*, or LQG problem, whose solution is given by the combination of a state estimator (Kalman filter) and a state feedback law computed on the basis of the state estimates. Here, we focus in the *state measurement feedback* case, where all states are available for computation of the feedback law. Recalling that $\|G_{w,z_2}\|_2$ is finite only when the transfer matrix is strictly proper, we only consider the case $D_{zw}=0$. It can be shown [28] that the optimal \mathcal{H}_2 state feedback synthesis problem with plant matrices (A, B, Γ) and performance output matrices (C, D_{zu}) is equivalent to an LQR problem with $Q=C'C$, $R=D_{zu}'D_{zu}$ and $N=C'D_{zu}$. The gain K obtained as the solution of the LQR problem will be the same as the one resulting from \mathcal{H}_2 optimization regardless of Γ , but the optimum norm will be affected by it.

To illustrate the equivalence in detail, consider the following example system

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} -1 \\ 3 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}.$$

We leave it to the reader to verify that the solution to the LQR problem with $Q = C'C$, $R = D'_{zu}D_{zu}$, and $N = C'D_{zu}$ is given by

$$K = \begin{bmatrix} 1.13 & 0.59 \\ -1.01 & 0.54 \end{bmatrix}$$

and that the solution to the optimum \mathcal{H}_2 gain synthesis problem (see following section) gives the same K as above regardless of Γ . If the given Γ is used, the optimum cost is 0.4484.

4.5.2 Polytopic Systems

The \mathcal{H}_2 state feedback synthesis problem for polytopic systems is also tractable and can be solved through LMI feasibility. The definition of 2-norm of (4.5) is no longer appropriate for polytopic systems. Instead, the equivalent LQG definition of 2-norm is used, taking the maximum over all instances of system matrices ($A(\alpha)$, $B(\alpha)$, $C(\alpha)$):

$$\|G\|_2^2 = \max_{\alpha} \lim_{T \rightarrow \infty} E \left\{ \frac{1}{T} \int_0^T y^T y \, dt \right\}, \quad (4.21)$$

where y is the output produced by polytopic system G when the input is a white noise with unit covariance. Thus, a *worst-case* approach is adopted when calculating the 2-norm of a polytopic system. A prescribed 2-norm can be targeted by solving an LMI feasibility problem whose complexity increases with the number of vertices. For further details, the reader is referred to [31]. The `msf_syn` command in Matlab admits polytopic system descriptions.

4.6 \mathcal{H}_∞ State Feedback Synthesis

The \mathcal{H}_2 synthesis approach delineated above seeks to maintain the influence of exogenous inputs on a designer-defined performance output below a prescribed value. The 2-norm is used as a measure of the “strength” of this influence. The

\mathcal{H}_∞ state feedback synthesis problem described here has the same objective, with the distinction that the infinity norm is used. Also, the definition of the performance output admits a nonzero D_{zw} term. Consider the nominal system of (4.7) and a performance output $z_\infty = Cx + D_{zu}u + D_{zw}w$. The transfer matrix from the exogenous input vector w to the performance output z_∞ is then

$$G_{w,z_\infty}(s) = (C - D_{zu}K)(sI - (A - BK))^{-1}\Gamma + D_{zw}.$$

The \mathcal{H}_∞ feedback gain synthesis problem is formulated as

Find K so that $(A - BK)$ is stable and $\|G_{w,z_\infty}\|_\infty < \gamma$ for some $\gamma > 0$.

The corresponding LMI feasibility problem is formulated as

$$\begin{bmatrix} (A - BK)X + X(A - BK)^T & \Gamma & X(C - D_{zu})^T \\ & \Gamma^T & -I & D_{zw}^T \\ & (C - D_{zu}K)X & D_{zw} & -\gamma^2 I \end{bmatrix} < 0, \quad (4.22)$$

$$X > 0. \quad (4.23)$$

As in the \mathcal{H}_2 case, an optimal \mathcal{H}_∞ state feedback synthesis procedure is to attempt the LMI feasibility problem repeatedly, with decreasing values of γ , until the problem is no longer feasible. This process, commonly known as γ -iteration, is incorporated in Matlab's `hinfsyn` command. Alternatively, `msfsyn` can be used. The latter command also works with polytopic systems.

4.6.1 Polytopic Systems

The infinity norm of a polytopic system G is regarded as the maximum time-invariant norm taken over all instances of system matrices $(A(\alpha), B(\alpha), C(\alpha), D(\alpha))$ with transfer function $G_\alpha(s)$:

$$\|G\|_\infty = \max_\alpha \sup_w \bar{\sigma} \{G_\alpha(jw)\}. \quad (4.24)$$

As in the \mathcal{H}_2 case, a prescribed infinity norm can be targeted by solving an LMI feasibility problem whose complexity increases with the number of vertices. The `msfsyn` command in Matlab admits polytopic system descriptions. For further details, the reader is referred to [31] and Matlab's Robust Control Toolbox documentation.

4.7 $\mathcal{H}_2/\mathcal{H}_\infty$ Feedback Synthesis with Regional Pole Placement

The reader may question the need for the \mathcal{H}_2 synthesis approach, given that \mathcal{H}_∞ synthesis pursues the same general objective, and does not have the restriction $D_{zw} = 0$. An intuitive justification for considering both norms can be found in their input/output interpretations. Minimizing the infinity norm amounts to reducing the peak of the closed-loop magnitude response across all frequencies. By design, the magnitude response for frequencies other than the one corresponding to the peak magnitude will also be suppressed. When high-frequency actuator dynamics are left unmodeled, good tracking and disturbance rejection calls for more selective control actions according to frequency: the controller should have an enhanced reaction to low-frequency error components, while being more insensitive to high frequency components. Subjecting all frequencies to the same peak magnitude constraint is conservative, and may lead to poor results. For this reason, frequency shaping is generally a necessary step in \mathcal{H}_∞ methods. In contrast, the definition of 2-norm given in (4.21) indicates that minimizing $\|G_{w,z_2}\|_2$ implies reducing the *time average* (in a root-mean-square sense) of the performance output z_2 . The definition given in (4.5) shows that a frequency-average is also implied. The relationship to LQR control and its integral quadratic performance measure gives the \mathcal{H}_2 approach a time-domain interpretation, allowing the designer to manage the performance vs. control effort constraint in a direct way. The reader wishing to examine the theoretical bases of \mathcal{H}_∞ and \mathcal{H}_2 control in detail is referred to standard material, for instance [28–30]. These references also provide detailed coverage of frequency-shaped \mathcal{H}_∞ methods.

It is possible to consider \mathcal{H}_2 and \mathcal{H}_∞ synthesis objectives simultaneously by defining two sets of performance outputs z_2 and z_∞ . The designer can balance the importance of each minimization by defining a weighted objective of the form

$$M = a\|G_{w,z_\infty}\|^2 + b\|G_{w,z_2}\|^2, \quad (4.25)$$

where a and b are non-negative weights. In the polytopic case, the norms are understood to be worst-case values, i.e., the maximum over all instances of system matrices. Again, minimization of the weighted objective is carried out under LMI feasibility constraints [37]. Unlike direct pole-placement design, norm-based minimization as presented until now does not provide a mechanism to enforce a location constraint for the closed-loop poles. Such *regional eigenvalue constraint* can be incorporated as an added LMI constraint, and the minimization problem remains tractable under some relaxing assumptions, leading to suboptimal solutions. The `msfsyn` command can be used to find a suboptimal solution to the mixed norm minimization problem, with the option of including a regional eigenvalue placement constraint. Placement regions in the complex plane can be of various shapes. The designer can obtain generally faster response times by choosing a constraint of the form $\text{Re}(s) < p$, where s denotes a closed-loop pole and p is a negative number. When the constraint is fulfilled, every closed-loop pole has a time constant lesser

than $1/|p|$. A circular region also be used to prevent poles having excessively low damping. The example at the end of this chapter illustrates the use of `msfsyn` in detail, as pertaining to the GTE control problem.

4.7.1 State-Feedback Setpoint Regulation and Input Integration

The state-feedback control law $u = -Kx$ is designed to regulate the plant state from an arbitrary initial condition $x(0)$ to the origin. This law is adequate for GTE control problems where fan speed must be driven from one steady point to another. Indeed, recall that the linearized plant state is given by fan and core speed deviations from a steady operating state $[\bar{N}_f \ \bar{N}_c]^T$: $x = [\Delta N_f \ \Delta N_c]^T$, where $\Delta N_f = N_f - \bar{N}_f$ and $\Delta N_c = N_c - \bar{N}_c$. In nonlinear simulation or actual realtime operation, x is formed by shifting the sensed N_f and N_c by the intended reference states \bar{N}_f and \bar{N}_c . Thus, $x = 0$ and $u = -Kx = 0$ at the beginning of the maneuver. Since u is the incremental actuator input, i.e., $u^T = [W_F - \bar{W}_F \ VSV - \bar{VSV} \ VBV - \bar{VBV}]$, constant inputs \bar{W}_F , \bar{VSV} and \bar{VBV} are being applied to the engine at the initial time. A setpoint change is produced when the reference states and inputs are suddenly changed to values defining a new steady operating point. The state will no longer be zero and regulatory control action will begin according to $u = -Kx$. If the closed-loop system is stable, x and u will be driven to zero again. The actual inputs applied to the engine must be shifted by their references, that is, $[W_F \ VSV \ VBV]^T = -Kx + [\bar{W}_F \ \bar{VSV} \ \bar{VBV}]^T$. *The control system must thus include a pre-calculated list of the required references for each desired pair of fan and core speed setpoints so that the sensed variables and the calculated control can be appropriately shifted.* Figure 4.2 shows the structure of the control law as applied to the nonlinear engine. From a nominal linear model standpoint, a steady operating point is defined by the equation

$$0 = A\bar{x} + B\bar{u}.$$

When A has no poles at zero (as is the case for the open-loop engine), the equilibrium state corresponding to a constant input \bar{u} can be calculated as

$$\bar{x} = -A^{-1}B\bar{u}. \quad (4.26)$$

The equation for the reverse operation is

$$B\bar{u} = -A\bar{x}. \quad (4.27)$$

If there are less inputs than states, i.e., if $m < n$, system (4.27) is underdetermined and infinitely many solutions for u exist. If $m = n$ and B is invertible, a unique solution can be found. If $m > n$ the system is overdetermined and solutions for \bar{u} exist if $A\bar{x}$ is in the column span of B . This condition cannot be assumed to hold

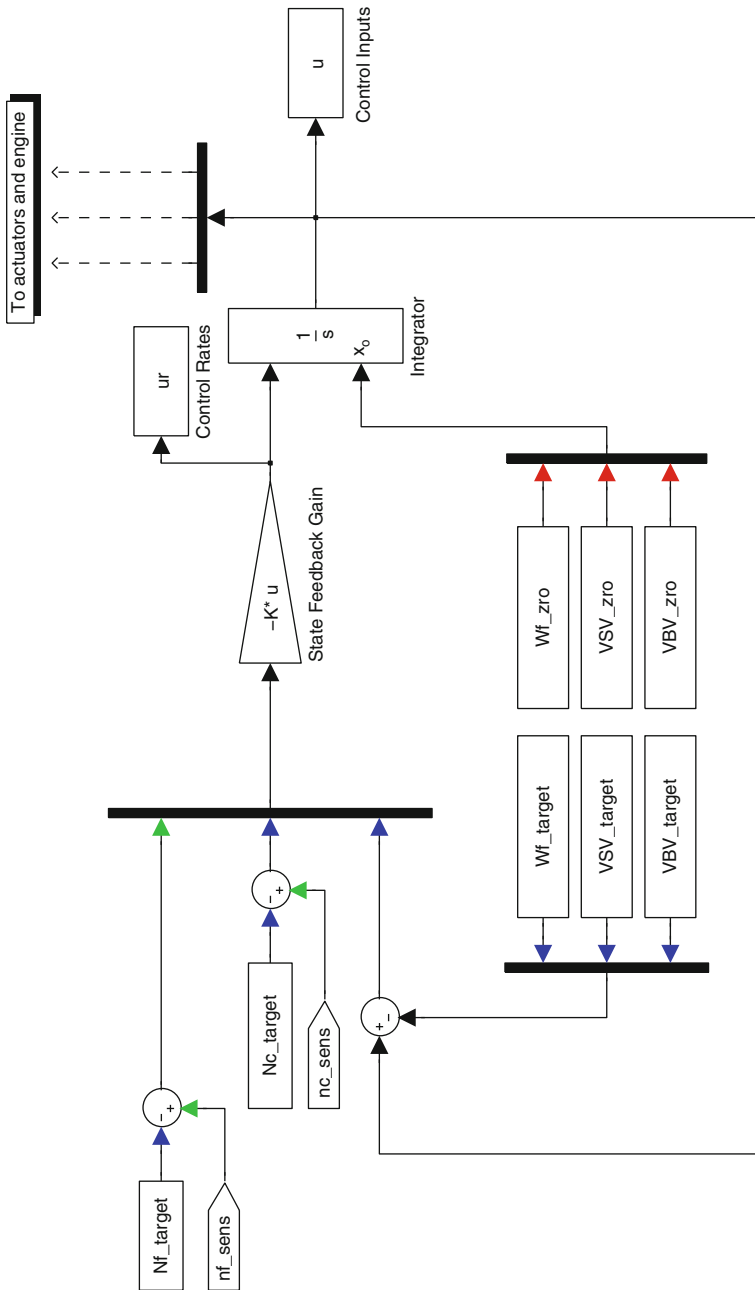


Fig. 4.2 State feedback with input integration in nonlinear engine simulation

in general. When considering three control inputs and two states for the GTE, no solutions to (4.27) can be assumed to exist for an arbitrary choice of \bar{x} . In addition to the structural difficulty arising from the use of three actuators with a 2-state plant, the reader must keep in mind that linearized models are only approximations to the actual nonlinear engine. The calculations for the triples $(\bar{W}_F, \bar{V}S\bar{V}, \bar{V}B\bar{V})$ that yield a desired pair (\bar{N}_f, \bar{N}_c) must be performed using an accurate steady-state solver for the nonlinear engine.

Disturbances and intrinsic engine nonlinearity limit the tracking accuracy of the state feedback law $u = -Kx$. The effect of disturbances on the regulated variables can be minimized by using the synthesis approaches of this chapter, but not eliminated entirely. The ability of the control system to track varying reference commands and reject disturbances can be enhanced by including integral control action. As mentioned in Chap. 3, constant disturbances produce zero steady-state error if the control loop is of type 1, that is, if it includes a free integrator. Input integration is commonly-used in actual GTE control implementations and is found in the built-in controllers designed for CMAPSS. To achieve integral control, the following *augmented plant* description is used:

$$\dot{x} = Ax + Bu + \Gamma w, \quad (4.28)$$

$$\dot{u} = u_r, \quad (4.29)$$

which can be compactly described as

$$\dot{x}_a = A_a x_a + B_a u_r + \Gamma_a w, \quad (4.30)$$

where the u_r is the new control input and the augmented state vector and matrices are defined as $x_a = [x^T \ u^T]^T$ and

$$A_a = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \quad B_a = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad \Gamma_a = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}. \quad (4.31)$$

When $(A(\alpha), B(\alpha), \Gamma(\alpha))$ are polytopic, the vertices of the corresponding polytopic augmented system can be directly computed. Moreover, the augmented polytopic system remains quadratically stabilizable when the original system is so. Hence, the various norm-based gain synthesis approaches can be applied without restrictions.

4.8 Example: CMAPSS-40k

In this section, we consider the fan speed control problem assuming that a fixed-gain feedback law of the form $u = -Kx$ is to be used across the flight envelope. The model contained in CMAPSS-40k is used for linearized plant extraction and

for realistic nonlinear simulations. We begin by choosing a six-vertex polytopic description of the linearized engine to match representative flight regimes. We then conduct a quadratic stabilizability evaluation considering input integration and then carry out several designs, observing the following:

1. Setpoint changes near different flight conditions: the \mathcal{H}_2 and \mathcal{H}_∞ designs with input integration are compared in their ability to produce consistently good responses near different operating regimes.
2. Effects of engine health status and faults: the disturbance rejection capabilities of the various designs are compared.

4.8.1 A Polytopic Description for the 40k Engine

Let the vector of incremental control inputs be defined as

$$u = [\Delta W_F \quad \Delta V S V \quad \Delta V B V]^T.$$

A six-vertex polytopic description $(A(\alpha), B(\alpha), \Gamma(\alpha))$ can be constructed by linearizing the engine at the six representative conditions shown in Table 2.4. The corresponding linearized system matrices are listed in Appendix C. Two outputs have been selected for the purposes of this example: the temperature at HPT outlet (T48) and the HPC stall margin (SM-HPC). The corresponding C , D , and Λ matrices used as output vertices are also listed in Appendix C.

4.8.2 Stabilizability of the Scaled Augmented Plant

The augmented vertices A_{ai}, B_{ai} necessary to design an integral controller are obtained directly from (4.31). Direct use augmented plant matrices in the LMI feasibility problem of (4.13) may pose numerical difficulties due to disparate scaling of A_a and B_a . Indeed, B_a is made up of only zeroes and ones, while A_a contains entries ranging from zero to thousands. To avoid numerical problems, it is advisable to obtain a *balanced realization* through similarity transformation. Defining $z_a = T x_a$, T is selected so that the transformed pair of matrices $A_{bal} = T A_a T^{-1}$ and $B_{bal} = T B_a$ generate a matrix $[A_{bal} \quad B_{bal}]$ with approximately the same row and column norms. This process is implemented in Matlab's `ssbal` command. Alternatively, the `balreal` command can be used. A transformation matrix T is obtained using an arbitrary augmented vertex and then used to transform the remaining vertices. This is a reasonable approach since all vertices share a similar scaling. Assuming that the augmented vertices are available in the workspace as

Aa1, Aa2, etc. and Ba (constant for all vertices), the following code can be used to evaluate quadratic stabilizability:

```
%Obtain balanced realizations
Cdummy=[1 0 0 0 0];Ddummy=zeros(1,3);
sys=ss(Aa1,Ba,Cdummy,Ddummy);
[sysbal,G,T,Ti]=balreal(sys);
[Abal1,Bbal]=ssdata(sysbal);
%Use the same transform for all augmented systems
Abal2=T*Aa2*Ti;Abal3=T*Aa3*Ti;
Abal4=T*Aa4*Ti;Abal5=T*Aa5*Ti;Abal6=T*Aa6*Ti;
%Prepare LMIs
setlmis([]);Q=lmivar(1,[5 1]);Y=lmivar(2,[3 5]);
lmiterm([-1 1 1 Q],1,1);
lmiterm([2 1 1 Q],1,Abal1','s');
lmiterm([2 1 1 Y],-Bbal,1,'s');
lmiterm([3 1 1 Q],1,Abal2','s');
lmiterm([3 1 1 Y],-Bbal,1,'s');
lmiterm([4 1 1 Q],1,Abal3','s');
lmiterm([4 1 1 Y],-Bbal,1,'s');
lmiterm([5 1 1 Q],1,Abal4','s');
lmiterm([5 1 1 Y],-Bbal,1,'s');
lmiterm([6 1 1 Q],1,Abal5','s');
lmiterm([6 1 1 Y],-Bbal,1,'s');
lmiterm([7 1 1 Q],1,Abal6','s');
lmiterm([7 1 1 Y],-Bbal,1,'s');
quad0=getlmis;
%Now run feasibility problem
[tmin3,xfeas]=feasp(quad0);
%Extract feasible sols
Qfeas=dec2mat(quad0,xfeas,Q);Yfeas=dec2mat(quad0,xfeas,Y);
%Calculate a feedback gain
KT=Yfeas*inv(Qfeas);
%Restore KT to original coordinates
K=KT*T;
```

The reader can verify that a feasible Q is found and that K yields stable closed-loop vertices ($A_{ai} - B_{ai}K$). Of course, the same K yields closed-loop stability for arbitrary system matrix variations within the polytope.

Note: Many books, research articles, and software assume a control input of the form $u = Kx$ rather than $u = -Kx$. Matlab's robust state feedback synthesis commands are not an exception. Using $-B$ and $-D$ in place of B and D resolves the discrepancy. This modification has been already incorporated in the code examples presented in this chapter.

4.8.3 Fixed-Gain LQR Design

An LQR gain can be found using a fixed vertex, followed by a closed-loop quadratic stability verification. As an example, the scaled vertex Abal1, Bbal is chosen

arbitrarily, along with unit LQR weights: $Q = I_5$ and $R = I_3$. In the following code, the resulting gain is used to calculate the closed-loop vertices and evaluate quadratic stability using `quadstab`:

```
%Set Q and R
Q=eye(5);R=eye(3);
[KT,S,E]=lqr(Aball,Bbal,Q,R);

%Restore KT to original coordinates
K=KT*T;
%Form polytopic closed-loop system in original coordinates
sys1=ltisys(Aa1-B*K);sys2=ltisys(Aa2-B*K);
sys3=ltisys(Aa3-B*K);sys4=ltisys(Aa4-B*K);
sys5=ltisys(Aa5-B*K);sys6=ltisys(Aa6-B*K);

polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);
[tmin,P]=quadstab(polysys)
```

The closed-loop polytopic system is verified to be quadratically stable. The fixed gain

$$K = \begin{bmatrix} -0.0108 & -0.0021 & 40.2654 & 0.3927 & -0.2583 \\ 0.0037 & -0.0056 & 0.3927 & 4.2430 & -0.0294 \\ 0.0010 & 0.0003 & -0.2583 & -0.0294 & 4.0187 \end{bmatrix} \quad (4.32)$$

produces the closed-responses shown in Fig. 4.3 when applied to the individual vertices. Although stability is guaranteed, a significant spread of response times is observed.

4.8.4 Fixed-Gain LQR in CMAPSS-40k

The gain K in (4.32) is now tested against the nonlinear engine in CMAPSS-40k. The observations made in Sect. 4.7.1 regarding controller deployment apply. The overall control law takes the form

$$u_r = -Kx_a, \quad W_f = \int_{W_f(0)} u_r(1), \quad (4.33)$$

$$VSV = \int_{VSV(0)} u_r(2), \quad (4.34)$$

$$VBV = \int_{VBV(0)} u_r(3), \quad (4.35)$$

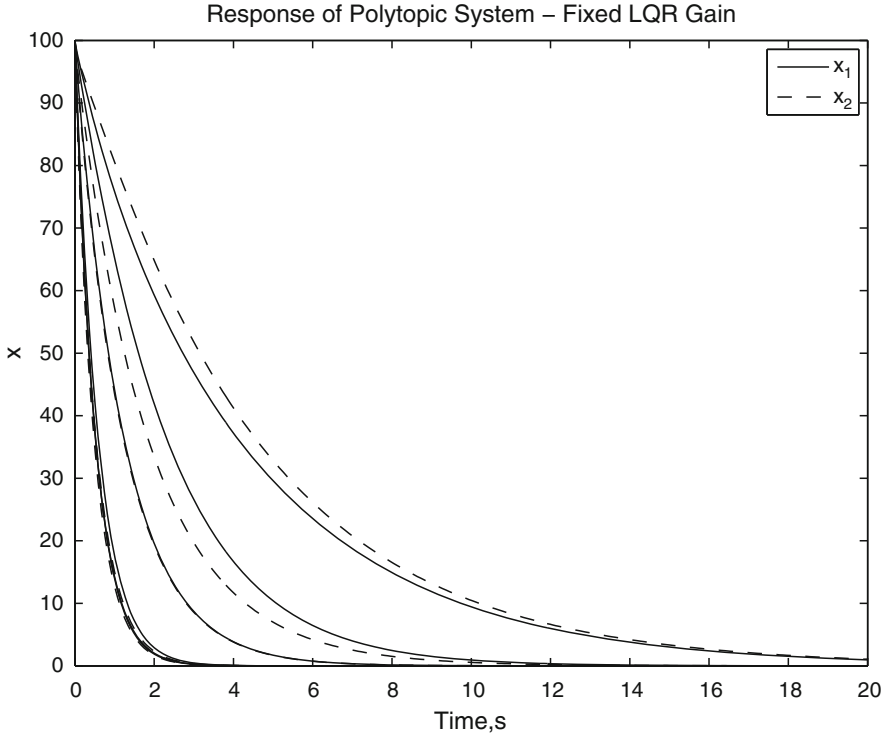


Fig. 4.3 Response of vertices with fixed LQR gain

where $x_a^T = [\bar{N}_f - N_f \quad \bar{N}_c - N_c \quad \bar{W}_F - W_F \quad \overline{VSV} - VSV \quad \overline{VBV} - VBV]$. Since the control input vector is part of the augmented state, the set of target steady values \bar{W}_F , \overline{VSV} and \overline{VBV} must be supplied to the controller. This implies that a database of such values must exist, as generated by the nonlinear engine steady-state solver. Each steady-state input triple $\bar{U} = [\bar{W}_F \quad \overline{VSV} \quad \overline{VBV}]$ yields a corresponding pair of steady states $\bar{X} = [\bar{N}_f \quad \bar{N}_c]$ through some mapping $\bar{X} = \Phi(\bar{U})$, which is affected by inlet conditions and health parameter values. This mapping is accessible only through high-fidelity nonlinear simulation, and cannot be fully captured through functions or tables due to the uncertain nature of health parameters. To conduct the simulations of this section, the nonlinear engine was driven to a steady-state using a “native” controller supplied with CMAPSS-40k. The native controller only uses W_F as a feedback-controlled input, while VSV and VBV are injected in a feedforward fashion. The steady values of the three actuators were adopted as reference inputs for this example.

Figure 4.4 shows the response of the nonlinear engine to positive step demands in fan and core speeds. The starting operating regime is Ground Idle, labeled as “A” in Table 2.4. The settling time for the state variables is about 3 s, with no overshoot. Figure 4.5 shows the response to negative step demands in fan and core speeds. The

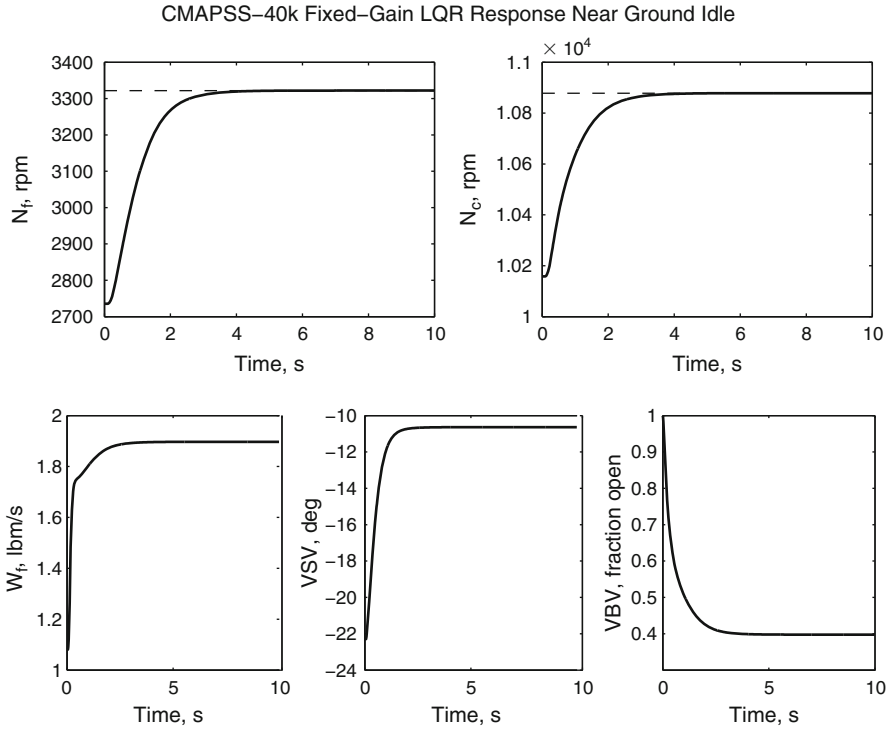


Fig. 4.4 CMAPSS-40k response with fixed LQR gain near Ground Idle

starting operating regime is Max Cruise, labeled as “D” in Table 2.4. The settling time for the state variables is now about 8 s, with no overshoot. In summary, this fixed LQR gain provides stability but is unable to maintain response speed across operating regimes. This, coupled with the need for precalculated steady references limits the applicability of integrator-based fixed-gain approaches.

4.8.5 $\mathcal{H}_2/\mathcal{H}_\infty$ Fixed Gain Synthesis: Polytopic Plant Model

We now apply the \mathcal{H}_2 , \mathcal{H}_∞ and mixed synthesis methods to the six-vertex polytopic plant and test the corresponding gains in CMAPSS-40k, comparing the results with those obtained with the “native” fan-speed controller distributed with the package. The native controller is a PI compensator around fan speed, with scheduled gains. The other two control inputs, namely VSV and VBV, are scheduled in an open-loop fashion, that is, without feedback from the state variables. Chapter 5 discusses gain scheduling in detail. We define two performance outputs: the HPT outlet temperature T_{48} and the HPC stall margin Sm_{HPC} . Allowable limits are

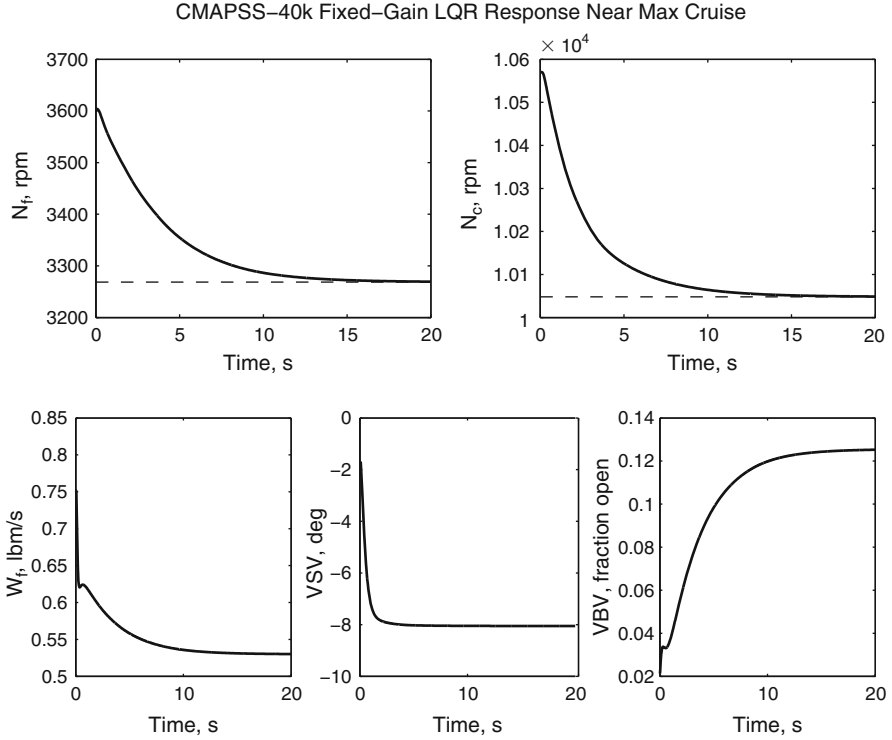


Fig. 4.5 CMAPSS-40k response with fixed LQR gain near Max Cruise

usually specified for these outputs. At the same time, these outputs tend to peak during transients. Moreover, certain health parameter inputs result in stall margin reductions. Continuing with the integral control approach, we define

$$z = [\Delta T_{48} \ \Delta SmHPC]^T = C_a(\alpha)x_a + D_{zw}(\alpha)w + D_{zu}u_r.$$

Augmented vertex matrices C_{ai} and D_{izw} are constructed from the data in Appendix C according to:

$$C_{ai} = [C_i \ | \ D_i],$$

$$D_{izw} = \Lambda_i.$$

Matrices C_i , D_i and Λ_i are formed for each vertex by stacking the corresponding C , D and Λ matrices for T_{48} and $SmHPC$. Note also that $D_{zu} = 0$, since the performance outputs are independent of the new control vector u_r , and that the vertices of the augmented disturbance input matrix are given by $\Gamma_{ai} = [\Gamma_i^T \ | \ 0]^T$.

According to the synthesis method used in the example, we will regard both components of z as \mathcal{H}_2 performance outputs, both as \mathcal{H}_∞ performance outputs, or a mix. Assuming that the augmented vertices already exist in the workspace as $Aa1, Aa2$, etc., Ba (constant for all vertices), $Ca1, Ca2$, etc., Da (zero for all vertices), $\text{Gamma}a1, \text{Gamma}a2$, etc. and $\text{Lambda}1, \text{Lambda}2$, etc., the following Matlab code creates the polytopic description with 2 performance outputs, using a balanced realization:

```

sys=ss(Aa1,Ba,Ca1,Da);
[sysbal,G,T,Ti]=balreal(sys);
[Abal1,Bbal,Cbal1,Dbal]=ssdata(sysbal);
Gammabal1=T*Gammaa1;
%Apply same balancing to all vertices
Abal2=T*Aa2*Ti;Cbal2=Ca2*Ti;Gammabal2=T*Gammaa2;
Abal3=T*Aa3*Ti;Cbal3=Ca3*Ti;Gammabal3=T*Gammaa3;
Abal4=T*Aa4*Ti;Cbal4=Ca4*Ti;Gammabal4=T*Gammaa4;
Abal5=T*Aa5*Ti;Cbal5=Ca5*Ti;Gammabal5=T*Gammaa5;
Abal6=T*Aa6*Ti;Cbal6=Ca6*Ti;Gammabal6=T*Gammaa6;
%Form polytopic system compatibly with msfsyn command:
%2 performance outs: [T48 SM-HPC]
sys1=ltisys(Abal1,[Gammabal1 -Bbal],Cbal1,[zeros(2,13) -Dbal]);
sys2=ltisys(Abal2,[Gammabal2 -Bbal],Cbal2,[zeros(2,13) -Dbal]);
sys3=ltisys(Abal3,[Gammabal3 -Bbal],Cbal3,[zeros(2,13) -Dbal]);
sys4=ltisys(Abal4,[Gammabal4 -Bbal],Cbal4,[zeros(2,13) -Dbal]);
sys5=ltisys(Abal5,[Gammabal5 -Bbal],Cbal5,[zeros(2,13) -Dbal]);
sys6=ltisys(Abal6,[Gammabal6 -Bbal],Cbal6,[zeros(2,13) -Dbal]);
polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);

```

The reader should refer to Matlab documentation for details on the plant structure assumed by `msfsyn`.

4.8.5.1 \mathcal{H}_2 Synthesis with Regional Pole Placement

When specific penalization of control effort is not included, the \mathcal{H}_2 -norm minimization may lead to large K and unreasonably high closed-loop bandwidths. It is possible to add a regional eigenvalue constraint to the minimization objective. In this example, both performance outputs are regarded as z_2 . Note that the vertex systems have $D_{zu} = 0$ (the `zeros(2,13)` entries) so that the 2-norm is finite. The following code executes the minimization:

```

r=[2 3]; %# of z2 outputs and # of controls
obj=[0 0 0 1];
region=lmireg %choose disk with center -8, radius 6
[gopt,hopt,KT,Pc1]=msfsyn(polysys,r,obj,region);
K=KT*T;

```

Closed-loop eigenvalues are required to lie in a disk centered at $-8+0i$ with a radius of 6. Attempts to reduce the radius and shift the center to obtain faster eigenvalues eventually leads to unfeasibility. The optimizing gain is

$$K = \begin{bmatrix} 0.0058 & 0.0055 & 10.8379 & 0.2571 & -0.0827 \\ -0.2535 & 0.0475 & -9.8546 & 9.5691 & 1.2961 \\ -0.9121 & 0.3095 & -9.7661 & 8.4027 & 12.3622 \end{bmatrix}.$$

The lowest bound for $\|G_{w,z_2}(s)\|_2$ returned by the program is 1.172×10^4 . The order of magnitude of this quantity is linked to problem data and should not be evaluated in an absolute sense. The norm bound is useful to compare designs sharing the same performance outputs but having different eigenvalue regions. For instance, if a disk centered at $-10+0i$ with a radius of 10 is chosen, the reader can verify that the optimum norm bound reduces to 900.27. This can be linked to a faster center for the closed-loop eigenvalues and an enlarged constraint region, as reflected in the radius specification.

Figures 4.6 and 4.7 show the responses to the same setpoint change considered in the LQR example, near Ground Idle conditions. The eigenvalue placement constraints allows to tune the controller for a faster response, matching the one obtained with the scheduled compensator. A health parameter step disturbance is injected through w at $t = 15$ s. All components of w are equal to -0.1 . Figure 4.6 shows that the ability to hold the setpoint in the presence of disturbance is roughly the same for the native controller and the \mathcal{H}_2 design, except for N_c , where the \mathcal{H}_2 design holds N_c closer to its regulation value. This is expected, since N_c is not under feedback with the native controller. Figure 4.7 shows that while the fuel flow commands are similar, the VSV and VBV inputs are not. Note that VSV saturates at the fully-open position under the \mathcal{H}_2 design.

4.8.5.2 Fixed-Gain \mathcal{H}_∞ Design

The \mathcal{H}_∞ -norm minimization with a regional eigenvalue constraint is now carried out. Both performance outputs are regarded as z_∞ . The `zeros(2,13)` entries in the code for the \mathcal{H}_2 case are changed to reflect the influence of w on the performance outputs. If Λ is used, it is difficult to find a feasible solution with the required bandwidth. Instead, the design uses 0.5Λ . Thus, the vertex definition lines must be changed to

```
sys1=ltisys(Aball, [Gammabal1 -Bbal], Cball, [0.5*Lambda1 -Dbal]);
...
sys6=ltisys(Abal6, [Gammabal6 -Bbal], Cbal6, [0.5*Lambda6 -Dbal]);
polysys=psys([sys1, sys2, sys3, sys4, sys5, sys6]);
```

States and Outputs: CMAPSS Native and H2 Design Near Ground Idle

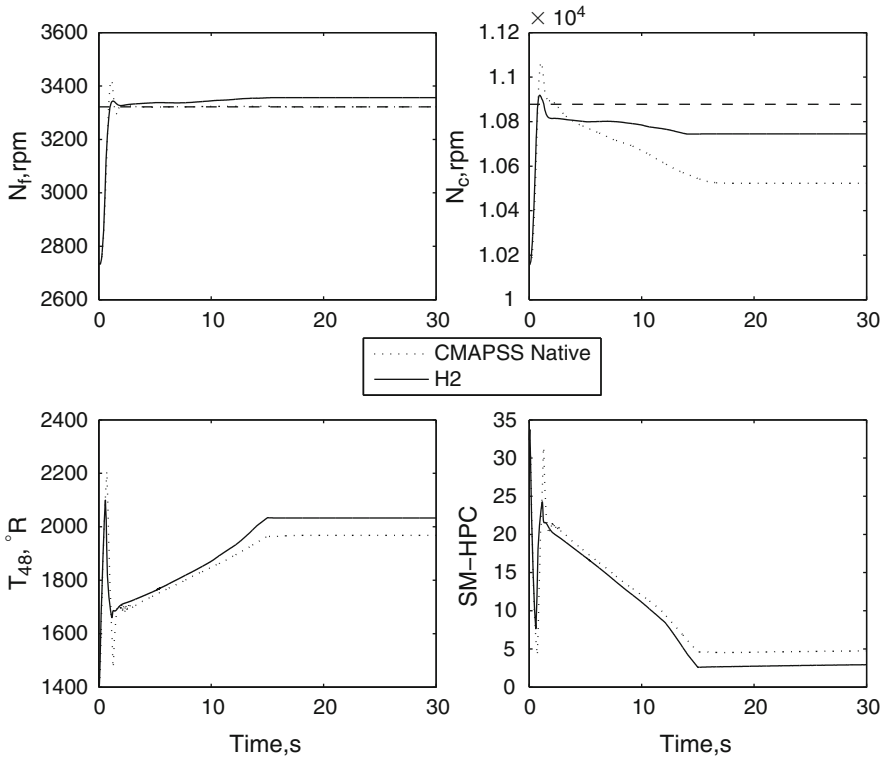


Fig. 4.6 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain \mathcal{H}_2 state feedback near Ground Idle: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

The following code is then used to regard both performance outputs as z_∞ and carry out the minimization:

```

r=[0 3]; %# of z2 outputs and # of controls
obj=[0 0 1 0];
region=lmiereg %choose disk with center -10, radius 10
[gopt,hopt,KT,Pc1]=msfsyn(polysys,r,obj,region);
K=KT*T;
    
```

Closed-loop eigenvalues are required to lie in a disk centered at $-10+0i$ with a radius of 10. The optimizing gain is

$$K = \begin{bmatrix} -0.0013 & -0.0036 & 13.9596 & -0.0877 & 0.0374 \\ -0.0179 & 0.2482 & 231.1991 & 23.6696 & -2.4600 \\ -0.1047 & 0.2329 & 227.2949 & 8.6254 & 12.9515 \end{bmatrix}.$$

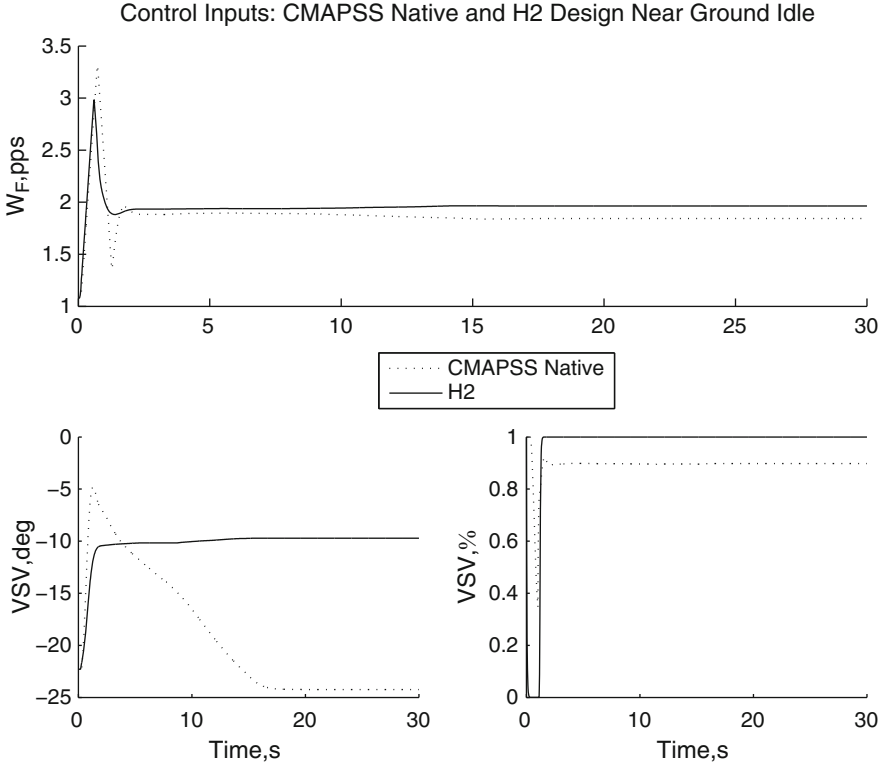


Fig. 4.7 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain \mathcal{H}_2 state feedback near Ground Idle: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

The lowest bound for γ returned by the program is 925.3. Figures 4.8 and 4.9 show the responses to the same setpoint change of the \mathcal{H}_2 example, again near Ground Idle conditions. The same health parameter step disturbance is injected at $t = 15$ s. The closed-loop eigenvalues yield a response slightly slower than the \mathcal{H}_2 design and a slightly better ability to hold the setpoint in the presence of disturbance. Figure 4.9 shows that the fuel flow control input is significantly more damped than in the \mathcal{H}_2 design. The VSV input remains saturated for most of the transient.

4.8.5.3 Mixed $\mathcal{H}_2/\mathcal{H}_\infty$

We conclude the example with a mixed minimization objective: T_{48} is regarded as an \mathcal{H}_2 performance output, while SM-HPC is regarded as an \mathcal{H}_∞ performance output. Since `msfsyn` assumes that the top rows of `Cba1`, `Lambda`, and `Dbal` correspond to the z_2 outputs, the matrices must be reordered:

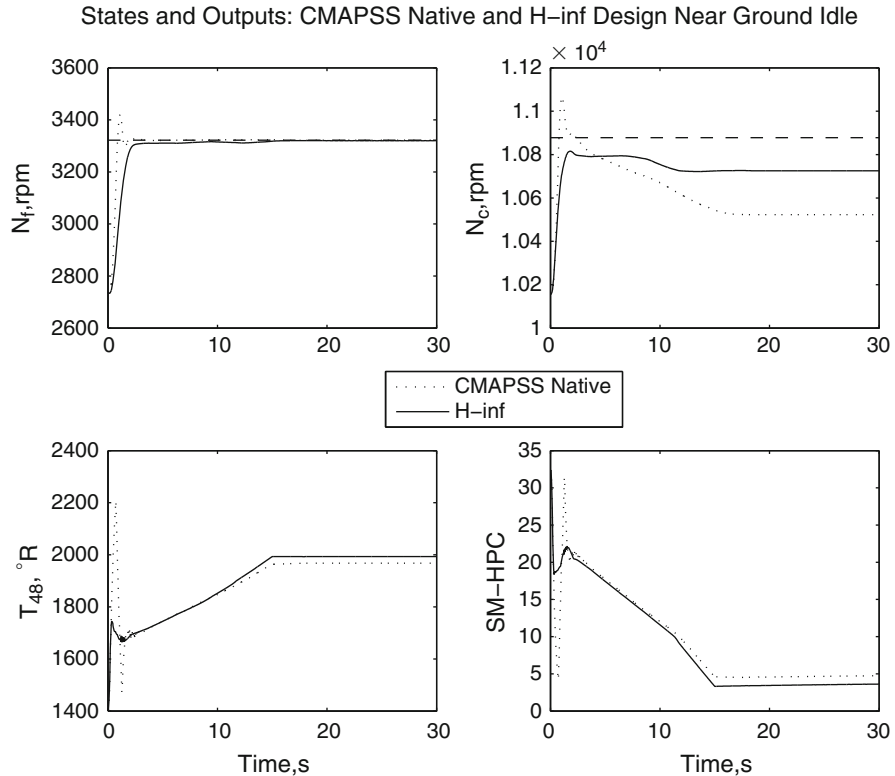


Fig. 4.8 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain \mathcal{H}_2 state feedback near Max Cruise: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

```

sys1=ltisys(Aball,[Gammabal1 -Bbal],
flipud(Cbal1),[0.75*Lambda1(2,:) -Dbal(1,:);zeros(1,16)]);
...
sys6=ltisys(Abal6,[Gammabal6 -Bbal],
flipud(Cbal6),[0.75*Lambda6(2,:) -Dbal(1,:);zeros(1,16)]);
polysys=psys([sys1,sys2,sys3,sys4,sys5,sys6]);

```

Again, Λ is scaled enough so that a feasible solution can be found. The following code is then used to and carry out the mixed-objective minimization with $a = 1$ and $b = 10$ as weights in (4.25):

```

r=[1 3];%# of z2 outputs and # of controls
obj=[0 0 1 10];
region=lmireg %choose disk with center -12, radius 10
[gopt,hopt,KT,Pcl]=msfsyn(polysys,r,obj,region);
K=KT*T;

```

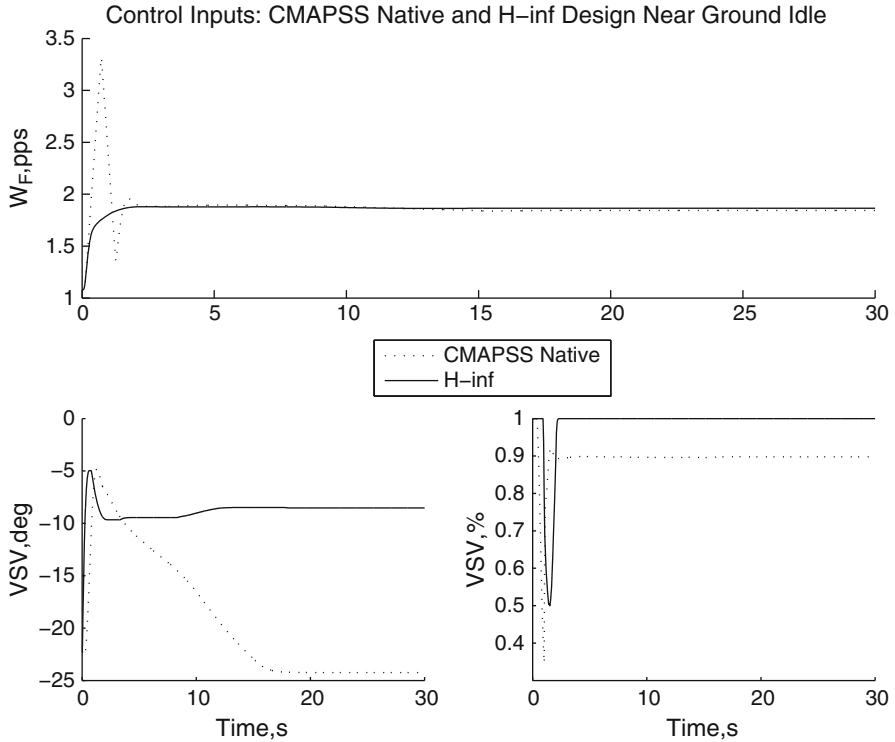


Fig. 4.9 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain \mathcal{H}_2 state feedback near Max Cruise: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

Closed-loop eigenvalues are required to lie in a disk centered at $-12+0i$ with a radius of 10. The optimizing gain is

$$K = \begin{bmatrix} 0.0163 & 0.0060 & 15.3492 & 0.0148 & -0.1114 \\ -1.0027 & 0.6748 & 161.8834 & 21.7061 & 1.6351 \\ -0.8754 & 0.4571 & 126.2015 & 5.1284 & 16.6993 \end{bmatrix}.$$

The bounds for the infinity and two-norms returned by the program are 496.3 and 6,299, respectively. These figures cannot be compared with the earlier synthesis results due to changes in the definition of performance outputs. Simulation is conducted this time near the Max Cruise condition, with a shaft deceleration command and the same health disturbance inputs as before. Figures 4.10 and 4.11 show the responses. The responses are essentially equivalent to those produced by the previous norm-based designs and the native gain-scheduled controller. In summary, four gains were designed for use in an integral control approach: LQR, \mathcal{H}_2 for both performance outputs, \mathcal{H}_∞ for both performance outputs and mixed $\mathcal{H}_2/\mathcal{H}_\infty$. The designs can be compared by measuring the closed-loop sensitivity

States and Outputs: CMAPSS Native and Mixed H₂/H-∞ Design Near Max Cruise

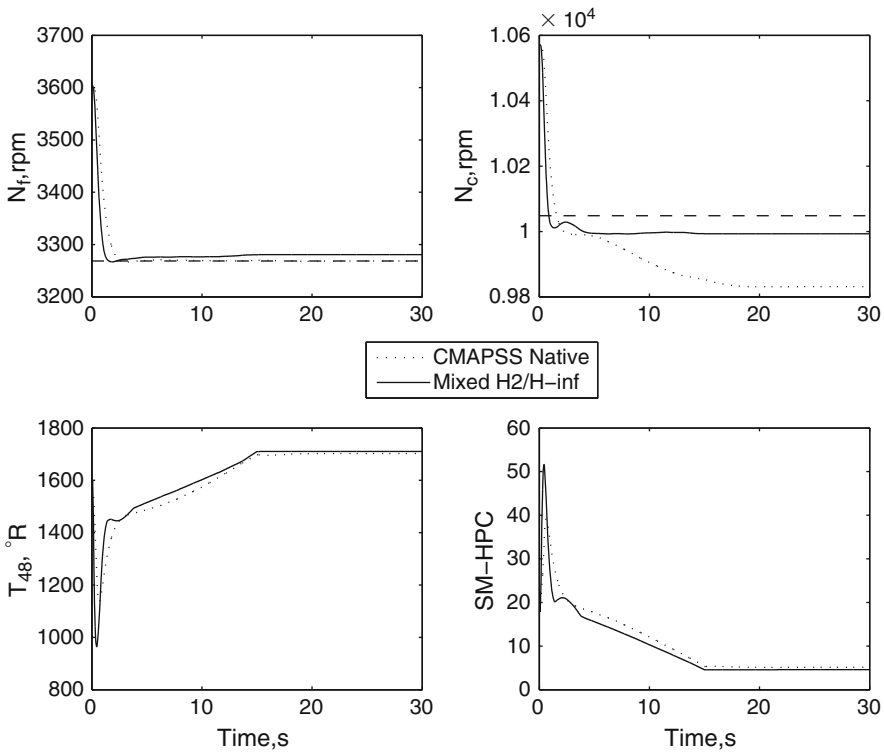


Fig. 4.10 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain mixed $\mathcal{H}_2/\mathcal{H}_\infty$ state feedback near Max Cruise: states and outputs. A step disturbance of -0.1 is applied to all health inputs at $t = 15$ s

of the performance outputs to exogenous inputs. This sensitivity can be evaluated with a plot of the maximum singular values as a function of frequency for all four designs. The worst-case peak singular value among closed-loop vertices is used. The results of this comparison are shown in Fig. 4.12.

4.9 Simplified \mathcal{H}_∞ Fan Speed Control

This chapter focused on state feedback techniques that take advantage of the availability of state measurements and multiple control inputs. As discussed in Sect. 4.7.1, information about the steady map of the engine must be incorporated in state feedback laws to account for wide parametric variations and engine nonlinearity. In contrast, a classical compensator loop using fuel flow to control

Control Inputs: CMAPSS Native and Mixed H2/H-∞ Design Near Max Cruise

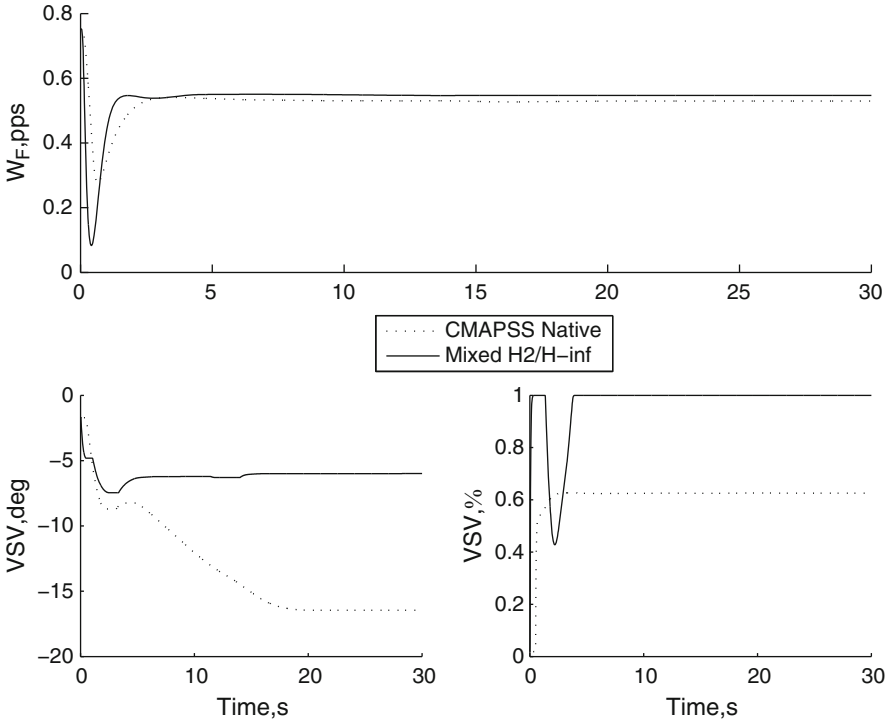


Fig. 4.11 Comparison of responses between CMAPSS-40k gain-scheduled fan speed compensator and fixed-gain $\mathcal{H}_2/\mathcal{H}_\infty$ state feedback near Max Cruise: control inputs. A step disturbance of -0.1 is applied to all health inputs at $t = 1$ s

fan speed or EPR does not require reference states (other than the target fan speed setpoint). Although a single compensator is insufficient to achieve control over the whole flight envelope or maintain critical engine variables within permissible limits, control transfer functions are the basis of traditional gain-scheduled designs and limit-protection-logic arrangements. In this section, a simple, yet systematic and effective method to design a fan speed or EPR control transfer function is presented.

4.9.1 Mixed-Sensitivity \mathcal{H}_∞ Design

Consider the SISO compensation loop shown in Fig. 4.13. Signals n and d represent sensor noise and output disturbance, respectively, while transfer function $G_d(s)$ can be used to “shape” the disturbance frequency spectrum. Elementary block-diagram

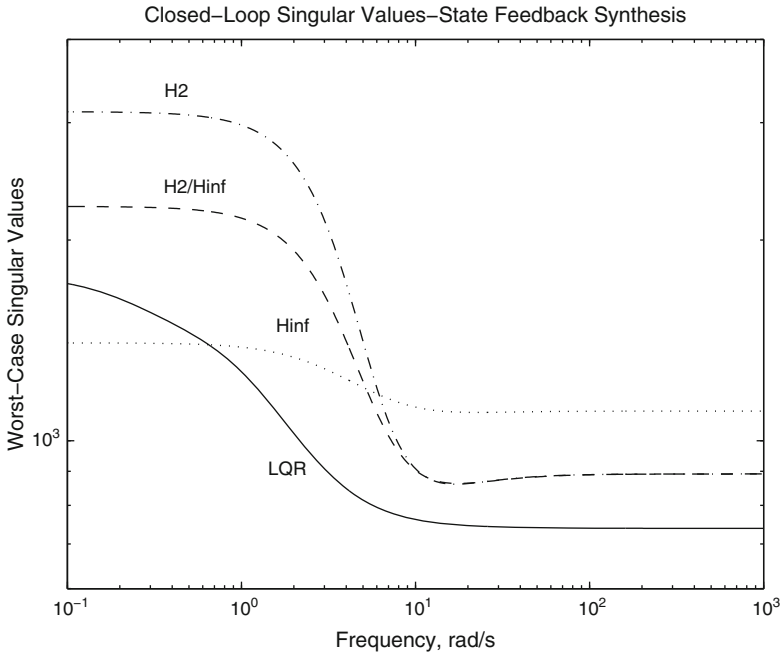
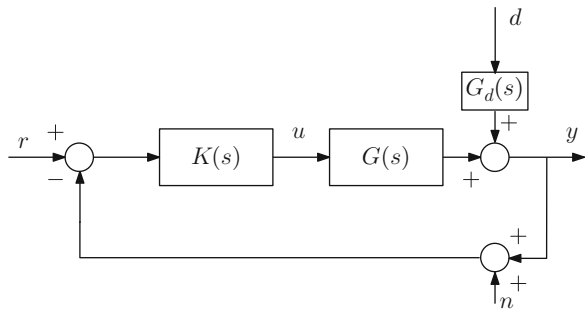


Fig. 4.12 Closed-loop comparison of worst-case maximum singular values for four synthesis methods

Fig. 4.13 Classical SISO feedback compensation loop



algebra shows that the output of the plant is related to the reference input r and to n and d as follows:

$$y = (I + GK)^{-1}GKr + (I + GK)^{-1}G_d d - (I + GK)^{-1}GKn, \quad (4.36)$$

while the following expression applies to the control input:

$$u = K(I + GK)^{-1}(r - G_d d - n). \quad (4.37)$$

Several transfer function definitions are in order. First, $T = (I + GK)^{-1}GK$ is the usual closed-loop transfer function between r and y when noise and disturbance are not considered. The transfer function $S = (I + GK)^{-1}$ is termed *sensitivity function*. This terminology is justified by the following identity:

$$\frac{dT/T}{dG/G} = S.$$

That is, the sensitivity function is a measure of the changes in T relative to changes in G . The closed-loop transfer function T is also referred to as *complementary sensitivity function*, since $S + T = I$ holds. This terminology is preferable, since several “closed-loop transfer functions” exist that relate u , r , y , n , and d .

With these definitions, (4.36) and (4.37) may be rewritten as

$$y = Tr + SG_d d - Tn, \quad (4.38)$$

$$u = KS(r - G_d d - n). \quad (4.39)$$

Various control objectives can be expressed in terms of desirable characteristics for T , S , and KS . First, the fundamental control objective is to force $y(t)$ to track $r(t)$ in a range of frequencies of interest. This implies that T should be as close to I (to 1 in the SISO case) as possible. In other words, the Bode magnitude of T should be flat, with a value of 0 dB, up to a design bandwidth. Note that n affects y in the same way as r . Care must be exercised in limiting the design bandwidth so that it does not encompass the sensor noise spectrum. Problems where the design bandwidth overlaps the noise spectrum require special filtering techniques. Next, SG_d must be kept small to minimize the influence of disturbances on y . Finally, KS must be small to maintain control input magnitudes within adequate ranges.

4.9.2 Frequency Weighting

The requirement that SG_d be “small” is made more precise by requiring that $|S(j\omega)G_d(j\omega)| < 1$ at all frequencies. Equivalently, $|S(j\omega)| < 1/|G_d(j\omega)|$ is required. Typically, no information about the disturbance spectrum is available, and therefore $G_d(s)$ is also unavailable. The designer may then model a generic disturbance as an input having strong low-frequency components, with little energy in the higher frequency range. These features distinguish disturbances from noise. Disturbances are then slowly-varying perturbations capable of producing large deviations in the plant output. According to this interpretation, $1/G_d(s)$ must have a high-pass characteristic. For this reason, it is commonplace to express the sensitivity minimization objective in the form

$$|S(j\omega)| < 1/|W_S(j\omega)|,$$

where $1/W_S(j\omega)$ is a design weight. Typical first- and second-order weights having the required characteristics are as follows:

$$W_S(s) = \frac{s/M + w_b}{s + \alpha w_b}, \quad (4.40)$$

$$W_S(s) = \frac{(s/\sqrt{M} + w_b)^2}{(s + w_b\sqrt{\alpha})^2}. \quad (4.41)$$

In both cases, w_b is the corner frequency of $1/W_S$, M is its high-frequency gain, and α is its low-frequency gain. The weights only differ in the steepness of the transition between α and M . Insensitivity to disturbances is improved by decreasing α . Note that the requirement that the magnitude of S be bounded by that of $1/W_S$ also imposes a lower-bound w_b on the bandwidth of T . Although initially conceived as a disturbance minimization objective, this form of weighting provides a mechanism to introduce a speed-of-response specification through w_b . The reader is referred to Skogestad and Postlethwaite [29] for more details.

For the purposes of controller synthesis, the sensitivity minimization objective is formulated as $\|W_S S\|_\infty < 1$. An upper-bound for the bandwidth of T and a roll-off rate can be specified as follows:

$$\|W_T T\|_\infty < 1.$$

Here, $1/W_T(s)$ is chosen to impose an upper envelope on the magnitude of $T(j\omega)$. A typical weight for T has the form

$$W_T(s) = \frac{s + w_b/M}{\alpha s + w_b}. \quad (4.42)$$

Design parameters w_b , α , and M may be chosen to coincide with those of the sensitivity weight, for simplicity.

Control magnitude weighting is likewise achieved with a specification of the form

$$\|W_u K S\|_\infty < 1.$$

A constant weight is usually sufficient and commonly chosen. All three objectives can be combined in a *mixed* or *stacked* sensitivity minimization objective of the form:

$$\min_K \|N(K)\|_\infty, \quad (4.43)$$

where $N(K)$ is given by

$$N = \begin{bmatrix} W_S S \\ W_T T \\ W_u K S \end{bmatrix}. \quad (4.44)$$

The mixed-sensitivity \mathcal{H}_∞ problem is readily solved using Matlab's `mixsyn`, part of the Robust Control Toolbox. Its use is shown in the CMAPSS example of the following section.

4.9.3 Example: Mixed \mathcal{H}_∞ Synthesis: CMAPSS-40k

Consider the problem of controlling fan speed to a setpoint using fuel flow as a single control input. The design bandwidth is chosen as $\omega_b=15$ rad/s and the remaining parameters are $M = 2$ and $\alpha = 1 \times 10^{-4}$. The first-order weight of (4.40) is used for the sensitivity, along with the complementary sensitivity weight of (4.42). Although a solution is obtainable with a large range of control weights, the weight W_u must be chosen in the order of magnitude of 1×10^5 to obtain a compensator which can work with the nonlinear CMAPSS engine model. The following code illustrates the sequence of calculations in Matlab:

```
%Plant matrices at Ground Idle: Aa and Ba assumed
%available in the workspace
C=[1 0];D=0; %Output definition for fan speed TF
G=tf(ss(A,B,C,D));
w0=15; %desired closed-loop bandwidth
alph=1/10000; %desired disturbance attenuation inside bandwidth
M=2; %desired bound on hinfnorm(S) & hinfnorm(T)
s=tf('s'); %Laplace transform variable 's'
W1=(s/M+w0)/(s+w0*A); %Sensitivity weight
W2=1e5; %Control weight
W3=(s+w0/M)/(A*s+w0); %Complementary sensitivity weight
[K,CL,GAM,INFO]=mixsyn(G,W1,W2,W3);
K=balred(K,3); %Perform balanced model reduction
%Plot results:
L=G*K; %Loop transfer function
S=inv(1+L); %Sensitivity
T=1-S; %Complementary sensitivity
sigma(GAM/W1,'k--',S,'k',GAM/W3,'k:',T,'-k');
%Manually adjust controller for explicit integration
numK=0.032341*conv([1 3.992],[1 2.439]);
denK=[conv([1 4.876],[1 14.06]) 0];
```

The control transfer function initially returned by `mixsyn` is

$$K(s) = \frac{0.032341(s + 1.5e05)(s + 3.992)(s + 2.439)}{(s + 1.5e05)(s + 14.06)(s + 4.876)(s + 0.0015)}.$$

Two important observations regarding the structure of this compensator must be made:

1. An approximate integrator has been introduced in the controller as a result of the weighting function shapes.

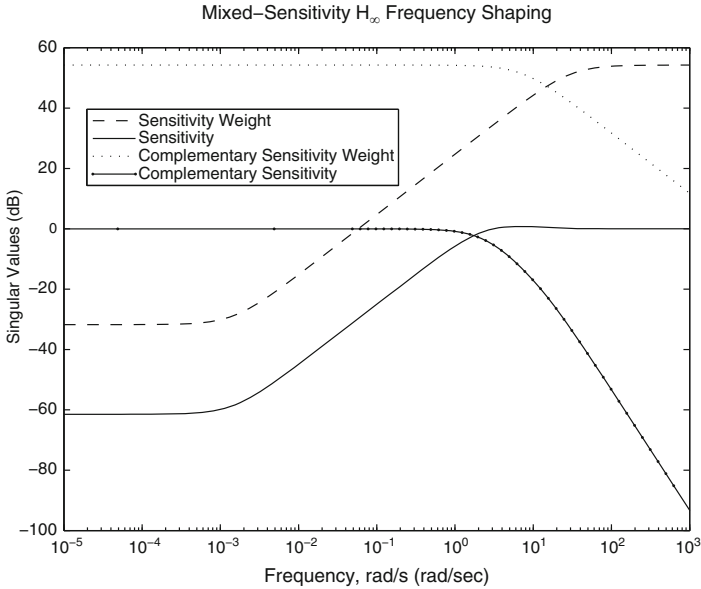


Fig. 4.14 Shaped sensitivity and complementary sensitivity functions

2. A near pole-zero cancelation is observed.
3. Some poles and zeroes have large magnitudes.

These features are not accidental, but appear rather frequently in conjunction with \mathcal{H}_∞ frequency shaping techniques. Also, compensators having a large number of poles and zeroes are typical of these designs, especially when multiple weights are used, as it is the case in MIMO systems. For this reason, controller reduction is usually required before implementation. In this example, elimination of the pole and zero at -1.5×10^5 is an obvious reduction that brings the controller to 3rd order.

The approximate integrator may be modified to force an actual pole at zero. Upon doing this, the final controller becomes:

$$K(s) = \frac{0.032341(s + 3.992)(s + 2.439)}{s(s + 14.06)(s + 4.876)}.$$

Figure 4.14 shows the achieved sensitivity and complementary sensitivity functions (after reduction) in relation to their weights. This controller was implemented in CMAPSS-40k, holding the VBV and VSV actuators constant at their Ground Idle trim values during the course of simulation. Figure 4.15 shows the fan speed, core speed, and actuator responses.

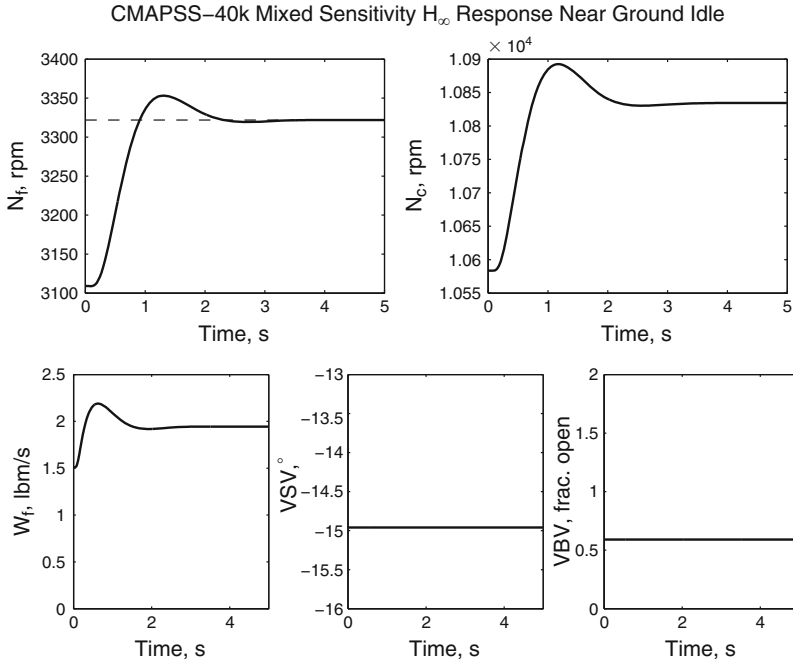


Fig. 4.15 Fan speed response with mixed sensitivity H_∞ design: CMAPSS-40k

4.9.4 Summary

The state-feedback design with input integration and fixed gain is conceptually simple. Systematic design procedures are available to synthesize the gain, according to various performance optimization requirements. The example shows that a fixed-gain design can be as effective as a classical gain-scheduled compensator in achieving robust stability against plant parameter variations arising from flight condition changes. Low sensitivity to health parameter inputs is also incorporated as an objective during the design process.

The responses achieved in the simulation examples are still unrealistic. The simulations of Sect. 4.8 show that the HPC stall margin, for instance, drops to 5% regardless of controller, a value regarded as unsafe in actual engine operations. Core acceleration, engine pressure ratio, and certain turbine temperatures are also subject to allowable operating limits. The limit protection features of CMAPSS-40k were disabled for these simulations. Robust nonlinear control approaches such as Model Predictive Control and Sliding Mode Control, considered in Chaps. 9 and 6, respectively, are capable of delivering superior performance. Limit management is a crucial consideration in GTE control design. Chapter 7 is entirely devoted to this problem.